



HAL
open science

Co-simulation of macro- and meso-scale neural circuit models: a technology ahead of its time

Lionel Kusch

► **To cite this version:**

Lionel Kusch. Co-simulation of macro- and meso-scale neural circuit models: a technology ahead of its time. Neuroscience. Aix marseille université, 2023. English. NNT : 2023AIXM0387 . tel-04527459

HAL Id: tel-04527459

<https://theses.hal.science/tel-04527459>

Submitted on 30 Mar 2024

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Distributed under a Creative Commons Attribution - NonCommercial - NoDerivatives 4.0 International License

"Science without conscience is but the
ruin of the soul"
Pantagruel, Rabelais, 1532

Affidavit

Je soussigné, Lionel Kusch, déclare par la présente que le travail présenté dans ce manuscrit est mon propre travail, réalisé sous la direction scientifique de Viktor Jirsa, dans le respect des principes d'honnêteté, d'intégrité et de responsabilité inhérents à la mission de recherche. Les travaux de recherche et la rédaction de ce manuscrit ont été réalisés dans le respect à la fois de la charte nationale de déontologie des métiers de la recherche et de la charte d'Aix-Marseille Université relative à la lutte contre le plagiat.

Ce travail n'a pas été précédemment soumis en France ou à l'étranger dans une version identique ou similaire à un organisme examinateur.

Fait à Marseille le 01/08/2023



Cette œuvre est mise à disposition selon les termes de la [Licence Creative Commons Attribution - Pas d'Utilisation Commerciale - Pas de Modification 4.0 International](https://creativecommons.org/licenses/by-nc-nd/4.0/).

Liste de publications et participation aux conférences

Liste des publications et/ou brevets réalisées dans le cadre du projet de thèse :

1. Brain simulation as a cloud service : The Virtual Brain on EBRAINS
[publish in NeuroImage]
Michael Schirner, Lia Domide, Dionysios Perdikis, Paul Triebkorn, Leon Stefanovski, Roopa Pai, Paula Prodan, Bogdan Valean, Jessica Palmer, Chloé Langford, André Blickensdörfer, Michiel van der Vlag, Sandra Diaz-Pier, Alexander Peyser, Wouter Klijn, Dirk Pleiter, Anne Nahm, Oliver Schmid, Marmaduke Woodman, Lyuba Zehl, Jan Fousek, Spase Petkoski, Lionel Kusch, Meysam Hashemi, Daniele Marinazzo, Jean-François Mangin, Agnes Flöel, Simisola Akintoye, Bernd Carsten Stahl, Michael Cepic, Emily Johnson, Gustavo Deco, Anthony R. McIntosh, Claus C. Hilgetag, Marc Morgan, Bernd Schuller, Alex Upton, Colin McMurtrie, Timo Dickscheid, Jan G. Bjaalie, Katrin Amunts, Jochen Mersmann, Viktor Jirsa, Petra Ritter
2. A comprehensive neural simulation of slow-wave sleep and highly responsive wakefulness dynamics
[publish in Frontiers in Computational Neuroscience]
Jennifer S. Goldman, Lionel Kusch, David Aquilue, Bahar Hazal Yalçinkaya, Damien Depannemaecker, Kevin Ancourt, Trang-Anh E. Nghiem, Viktor Jirsa and Alain Destexhe
3. Multiscale co-simulation design pattern for neuroscience applications
[under-review in Frontiers in Neuroinformatics]
Lionel Kusch, Sandra Diaz, Wouter Klijn, Kim Sontheimer, Christophe Bernard, Abigail Morrison and Viktor Jirsa
4. Synchronization in spiking neural networks with delays
[under-review in Chaos]
Lionel Kusch, Martin Breyton, Spase Petkoski, Viktor Jirsa

Participation aux conférences et écoles d'été au cours de la période de thèse :

1. EINT Workshop : Mean-field approaches to the dynamics of neuronal networks
2. EINT Springschool 2018
3. NEST Conference 2018
4. HBP summit 2018
5. HBP CodeJam 2018
6. HBP 3rd Student Conference
7. NeuroFrance 2019
8. NEST Conference 2019
9. The Predictive Brain Conference 2019
10. HBP CodeJam 2019
11. School of Brain Cells and Circuits : Modeling the brain and its pathologies
12. SFN Conference 2019
13. HBP summit 2020
14. HBP summit 2021 (online)
15. Virtual Brains & Digital Twins workshop in 2022
16. WP1 meeting in 2022
17. HBP summit 2023

Résumé

Le cerveau est un système complexe basé sur des phénomènes d'émergence à différentes échelles. De nombreuses méthodes existent pour intégrer les données de différentes échelles, mais l'une d'entre elles consiste à assembler des modèles de chaque échelle. Dans cette thèse, nous présenterons un cosimulateur composé de plusieurs simulateurs dédiés à la simulation de modèles d'une échelle spécifique. Nous nous limiterons aux modèles cérébraux à macro- et méso-échelle. Tout d'abord, nous décrirons un modèle de co-conception en sciences et informatiques pour coupler efficacement des simulateurs. Ce modèle de conception est utilisé pour construire un workflow entre The Virtual Brain et NEST. Pour souligner l'intérêt du couplage de ces deux simulateurs, nous proposons de simuler un modèle multiéchelle de la région CA1 de l'hippocampe cellulaire de la souris intégré dans un réseau cérébral complet de la souris impliquant des enregistrements de micro- et macro-électrodes virtuelles. Deuxièmement, nous proposons d'étudier l'interdépendance entre les modèles macro- et méso-échelle afin d'indiquer le choix d'un modèle pour passer le passage à une échelle inférieure. Nous choisissons d'analyser le rôle du délai des connexions à longue portée dans un réseau neuronal bidimensionnel. En utilisant l'exploration des paramètres, nous démontrons que le comportement principal de ce réseau comprend des ondes d'activité stationnaires ou mouvantes et dépend de l'intensité du bruit, de l'homéostasie E/I et de l'adaptation de la tension, qui est modulé par le délai de la connexion à longue portée. En outre, nos résultats illustrent l'interaction entre les propriétés d'un seul neurone, la connectivité et la composition du réseau neuronal, et le délai de la connexion à longue portée. Enfin, nous concluons que ce type de projet est principalement un projet logiciel axé sur le potentiel de réalisation sans véritable objectif scientifique. Sans l'inclusion de la recherche transdisciplinaire et la clarification de l'utilisation scientifique du champ moyen et des réseaux neuronaux, cette technologie restera inutilisée. La facilité d'utilisation de la co-simulation dépend également d'une définition plus robuste de la fonction de transformation entre les échelles afin de garantir la confiance dans la simulation multiéchelle. En outre, nous disposons également d'un résultat préliminaire sur l'interdépendance entre les modèles à méso-échelle et à macro-échelle pour indiquer le choix du modèle pour passer le passage à une échelle supérieure. Nous choisissons de caractériser l'approximation du champ moyen Ad Ex avec une entrée excitatrice statique ou oscillatoire. Nous incluons également les travaux parallèles à cette thèse, à savoir la participation à un article de modélisation de l'indice de complexité des perturbations de l'état de sommeil lent et de l'état d'éveil à réponse élevée, ainsi qu'un article de synthèse des

outils de The Virtual Brain sur EBRAINS.

Mots clés : co-simulation, multi-échelle, modèle de réseau cérébral, réseau de neurones à pointes, simulation neuronale, modèle de champ moyen, retard temporel, dynamique du réseau, synchronisation, systèmes non linéaires

Abstract

The brain is a complex system based on emergence phenomena from different scales. Multiple methods exist for integrating the data from different scales, but one of them is assembling models of each scale. In this thesis, we will present a co-simulator composed of multiple simulators dedicated to simulating models of a specific scale. We will restrain to macro- and meso- scale brain models. Firstly, we will describe a software science co-design pattern for efficiently coupling simulators. This design pattern is used for building a workflow between The Virtual Brain and NEST. To highlight the interest in the coupling of these two simulators, we propose to simulate a multiscale model of the CA1 region of the cellular-level hippocampus of the mouse embedded into a whole mouse brain network involving micro and macro electrode recordings. Secondly, we propose to investigate the interdependence between macro- and meso- scale models to indicate the choice of a model for downscaling. We choose to analyze the role of the delay of long-range connection in a two-dimensional neural network. Using parameters exploration, we demonstrate that the principal behavior of this network comprises standing or traveling waves of activity and depends on noise strength, E/I balance, and voltage adaptation, which are modulated by the delay of the long-range connection. Moreover, our results illustrate the interplay of single neuron properties, connectivity and composition of the neuronal network, and long-range connectivity for the emergent spatiotemporal activity. Finally, we conclude this type of project is mainly a capability-driven software project without a real scientific goal. Without the inclusion of transdisciplinary research and clarification on the scientific usage of mean-field and neural networks, this technology will stay unused. The usability of co-simulation also depends on a more robust definition of the transformation function between scales to ensure certainty in the multiscale simulation. Additionally, we also have a preliminary result on the interdependence between meso- and macro- scale models to indicate the choice of a model for upscaling. We choose to characterize the approximation of Mean Ad Ex with static or oscillatory excitatory input. We also include side work of this thesis, which is participation in a modeling paper of the perturbation complexity index of slow-wave sleep state and high response wakefulness state and an overview paper of the tools of The Virtual Brain on EBRAINS.

Keywords: co-simulation, multiscale, brain network model, spiking neural network, neural simulation, mean-field model, time delay, network dynamic, synchronization, nonlinear systems

Remerciements

J'aimerais commencer par remercier l'ensemble des membres de mon jury qui ont accepté de participer à ma soutenance de thèse : Dr. Fulvia Palesi, Dr. Maria-Leena Linne, Dr. Egidio D'Angelo et Dr. Laurent Perrinet. En particulier, je remercie Dr. Linne et Dr. D'Angelo pour avoir accepté d'évaluer ce manuscrit et Dr. Perrinet pour avoir accepté de présider ce jury.

Un grand merci à Viktor de m'avoir donné ma chance, il y a 5 ans et demi juste après un entretien à distance. Je me souviens d'être surpris d'avoir pu participer à un workshop avant même le début de ma thèse. Pendant ma thèse, tu m'as offert une autonomie, une stabilité financière et une liberté dans mon travail qui m'aurait été difficile avec un autre directeur. Je te remercie d'avoir eu la patience de me laisser le temps d'affiner ma réflexion jusqu'à affiner cette thèse et surtout de m'avoir supporté tous au long de ce chemin malgré une certaine différence de point de vue, en particulier sur les modèles de champs moyens.

Je souhaite remercier en particulier mes proches collaborateurs : Wouter, Sandra, Jennifer, Damien et Spase. Merci, Wouter pour tes conseils, ta prospective de la cosmulation et de l'ambition que tu portes en toi. Sandra, j'aimerais te remercier pour le support que tu m'as fourni et de ta gentillesse. Jennifer, tu m'as accordé une confiance que j'ai fortement appréciée malgré mon manque d'expérience. Damien, je te remercie pour les discussions philosophiques et presque existentielles que l'on a eues. Spase, malgré la difficulté de communication que l'on a eue, tu m'as supporté et aidé durant toutes ces années, mais surtout éviter toute la paperasse de HBP.

J'aimerais aussi remercier les collègues de mon équipe avec qui j'ai eu de nombreuses conversations scientifiques, mais aussi sur la vie. Je voudrais remercier en particulier Daniele d'avoir partagé de bon moment ensemble ainsi que Borana d'avoir accepté d'écouter mes plaintes plus ou moins raisonnables et de me montrer le bon côté de choses malgré ma vision assez pessimiste de la science. Je souhaite aussi remercier mes autres collègues de bureaux Hiba, Adam, Jyotika, Martin et Jan. En particulier, Jan pour son expertise en informatique et ces différents comics d'informaticien qui égaye mes journées. Mais aussi Martin qui m'a bien aidé dans la rédaction de ma thèse et de mes papiers ainsi que de me distraire avec ces problèmes informatiques. Dans le même registre, je souhaite remercier Mario qui a essuyé les plâtres sur mes débuts d'écriture. J'ai eu la chance de pouvoir rencontrer des personnes avec qui j'ai pu échanger nos réflexions comme Viktor, Meysam, Giovanni, Pierpaolo, Kashyap, Mathieu, Maciek, Vineet, Bahar, Gian Marco, Sora. Je remercie aussi, les précédents doctorants, Francesca et Marisa qui m'ont montré la voie, les actuels doctorants or

docteurs, Sophie, Anirudh, Mikhael, Luca et Diego, pour nous avoir entraînés et les futurs doctorants, Aitakin, Amir, Chloé, Anastasios, Manef et Paul, pour avoir pris en compte certain de mes conseils. Spécialement Aitakin que j'ai apprécié de l'avoir vu évoluer et de m'avoir apporté son affection. Je désire aussi remercier Marmaduke pour son support dans le domaine informatique et de ses précisions sur l'environnement de développement des logiciels.

Je remercie l'INS de m'avoir accueilli en son sein. En particulier, je voudrais remercier pour la fraîcheur des discussions durant le déjeuner et leur sympathie l'équipe de physionet : Pascal, Myriam, Maeva, Romanos, Christophe, Antoine, Heba, Antone, Wesley, Ana, Mattias, Prya, Galyna, Sawssan, Dominica et spécialement Thomas pour s'est exposé historique et pour partager sa vision de la science ainsi que Narim pour son chocolat et la profondeur de ces réflexions sur la vie. Je désire aussi remercier l'équipe administrative qui s'occupe de moi pour la bonne réalisation de ma thèse et des déplacements que j'ai effectués : Conchetta, Noémi, Lisa, Audrey, Irène, Véronique, Sonia, Séverine et Anuska. Je remercie aussi Mireille et Driss pour leur gentillesse et leur préoccupation pour les doctorants en particulier dans le laboratoire.

Je souhaite aussi remercier les personnes de Hippo'thèse (Alexandre, Hary, Clara, Alexendra, Victoria, Bouthaina, Michaël, Jean Yves, Fanny, Sofia, Mélanie, Coralie, Konrad, Valdimir, ...) , de DynAmu (Kim, Laure, Vladimir, Cédric, Ouassim, Tiphaine, Christine, ...) , la CJC (Quentin, Solène, Kévin, Magali, Julie, ...) dont j'ai fait partie et qui défende les doctorants et anime la vie des doctorants sur Marseille. Je remercie aussi le conseil de l'école doctorale pour m'avoir écouté et des autres représentants pour notre entraide (Victoria, Florian, Louise, ...), en particulier la direction de l'école doctorale Jean-Louis Mege, Christophe Porchet et Aurélie Aufray pour leur dévouement envers les doctorants en difficulté de mon école doctorale. Je souhaite aussi remercié, la responsable de la formation doctorale, Sandrine Alligier, pour son support lors des formations auquel j'ai participé à organiser et des formations proposées. Je désire aussi remercier le comité du PhD programme de la Neuroschool (Frédérique, Anne, Wafa, Anna, Isabelle, Laurie, Ingrid, ...) pour leur écoute et les formations qu'il propose. Je remercie les personnes qui m'ont aidé à organiser les Neurodays en 2019 malgré mon absence à l'évènement.

Je souhaite remercier les personnes qui ont géré Human Brain Project qui m'a permis de découvrir des sujets diverses, de partager de nombreuses expériences et de collaborer avec beaucoup de scientifiques dans un des plus gros projets européens. Je souhaite en particulier remercier le groupe de cosimulation (Denis, Fahad, Rolando, Kim, Alper, Michiel, Lena, ...), les personnes travaillant sur le showcase 2 (Sergio, Daniela, Jonathan, ...). Je souhaite aussi remercier l'équipe de The Virtual Brain qui m'a aidé dans mon projet (Paola, Lia, Jochen, Randi, Petra, ...) et la communauté de NEST qui sont très sympathiques (Hans, Markus, Jochen, ...).

Pour finir, je souhaite remercier ma famille qui a supporté mon manque de communication et respecter mon travail. Je souhaite remercier l'ensemble des personnes que j'ai pu rencontrer durant cette thèse à travers mon travail de recherche, mais aussi mes activités extérieures, dance, escalade et afterlab.

Table des matières

Affidavit	3
Liste de publications et participation aux conférences	4
Résumé	6
Abstract	8
Remerciements	9
Table des matières	11
Table des figures	15
Liste des tableaux	20
I Introduction	21
1. Modelling in neuroscience	24
1.1. Dynamical Systems	25
1.1.1. System and equations	25
1.1.2. Dynamical system theory	25
1.1.3. Fixed-point stability	26
1.1.4. Bifurcation theory	28
1.1.5. Software for numerical continuation	30
1.2. Neuron model	30
1.2.1. What is a Neuron?	30
1.2.2. Hodgkin and Huxley model	32
1.2.3. Characterize neurons	34
1.2.4. Point neuron models	46
1.2.5. Adaptive and exponential integrator and fire neuron (Ad Ex) . .	48
1.3. Network of point neurons	50
1.3.1. Model of synapse	51
1.3.2. Network connectivity	53
1.3.3. Characterization of a neural network	54

1.3.4. Network of adaptive exponential integrate and fire neurons . . .	55
1.4. Mean-field model	56
1.4.1. Neural Mass AD EX	57
2. Cluster programming and research infrastructure	59
2.1. Parallel and Distributed Computing	60
2.1.1. A computer	60
2.1.2. A cluster	61
2.1.3. parallelization and distributed computing	64
2.2. Human Brain Project and EBRAINS	66
2.2.1. Humain Brain Project	66
2.2.2. EBRAINS	68
II Original Work	70
3. Multiscale co-simulation design pattern for neuroscience applications	71
3.1. Chapter overview	72
3.2. Introduction	73
3.3. Results	76
3.3.1. Virtual experiment of hippocampal CA1 embedded in a full mouse brain	76
3.3.2. Output signal from the virtual experiment	77
3.3.3. Workflow between NEST and TVB	82
3.3.4. Performance	83
3.4. Materials and Methods	86
3.4.1. Models	86
3.4.2. Communication between modules	87
3.4.3. Performance tests	88
3.4.4. Implementation details	89
3.5. Discussion	90
4. Synchronization in spiking neural networks with short and long connections and time delays	94
4.1. Chapter overview	95
4.2. Introduction	96
4.3. Modeling the Network of Neurons	97
4.4. Results	100
4.4.1. Independent contributions of LRCs and noise	103
4.4.2. Synergic effect of LRCs and noise	103
4.4.3. Modulation by adaptation current	104
4.4.4. Effect of excitability of the voltage membrane	105
4.4.5. Network effects : balanced network	106
4.4.6. Network effects : network size	108
4.4.7. Topology effects : bi-directional long-range connections	109

4.5. Discussion	109
5. Discussion	113
5.1. Method for co-simulation	113
5.1.1. Putting the co-design model into perspective using a holistic framework	113
5.2. Multiscale modeling	114
5.2.1. The concept	115
5.2.2. Model for transforming neural data between scale	116
5.3. Limitation	117
5.3.1. Limitation of co-design pattern	117
5.3.2. Should we trust a co-simulation?	118
5.3.3. Modeling limitation	119
5.4. Perspective	119
5.4.1. Difficulty of formulating scientific question using co-simulation	119
5.4.2. Youth of brain network model	119
5.4.3. History of co-simulation in neuroscience, a technology-driven proposal	120
5.4.4. Missing of transdisciplinarity in neuroscience	121
5.5. Summary	122
Bibliographie	123
III Annex	168
A. Dynamics of mean-field model of adaptive exponential integrate-and-fire network	169
A.1. Chapter overview	170
A.2. Introduction	171
A.3. Models and methods	172
A.3.1. AdEx network	172
A.3.2. Mean-field of AdEx	173
A.3.3. Transfer function fitting	175
A.3.4. Bifurcation analysis	176
A.3.5. Spike train analysis	176
A.3.6. Mean-field analysis	177
A.4. Results	177
A.4.1. Steady-state and transfer function	177
A.4.2. Dynamical comparison	181
A.5. Discussion	185
B. A comprehensive neural simulation of slow-wave sleep and highly responsive wakefulness dynamics	188
B.1. Chapter overview	188

C. Brain simulation as a cloud service : The Virtual Brain on EBRAINS	206
C.1. Chapter overview	206
D. Supplementary material of chapter 1	219
D.1. Parameters for classification of neuron dynamic	220
E. Supplementary material of chapter 3	222
E.1. Table of content of Supplementary Notes, Figures and Tables	223
E.1. Supplementary Notes	225
E.1.1. Supplementary Note 1 Guidelines Input/Output (I/O) interface	225
E.1.2. Supplementary Note 2 Guideline for implementation of transfer module	226
E.1.3. Supplementary Note 3 Detail characterization of the workflow TVB-NEST	227
E.2. Supplementary Table	230
E.3. Supplementary Figures	249
F. Supplementary material of chapter 4	273
F.1. Table of content of Supplementary Notes, Figures and Tables	274
F.2. Supplementary Note 1 : Databases description	276
F.2.1. Quality of database	276
F.3. Supplementary Note 2 : supplementary results	276
F.3.1. Firing rate and associated measures	276
F.3.2. Inter-spiking interval and associated measures	277
F.3.3. Phase and associated measures	277
F.3.4. Bursts and associated measures	278
F.4. Supplementary Table	279
F.5. Supplementary Figures	283
G. Supplementary material of Annex A	317
G.1. Supplementary Notes	319
G.1.1. Supplementary Note 1 : Additional analysis	319
G.1.2. Supplementary Note 2 : Data available	320
G.1.3. Supplementary Note 3 : Supplementary results of spiking neural network	321
G.1.4. Supplementary Note 4 : Supplementary results of mean-field . .	324
G.2. Supplementary Tables	325
G.2.1. Supplementary Table 1 : Description of the network simulation simulation	325
G.2.2. Supplementary Table 2 : Description of the mean-field simulation	331
G.3. Supplementary Figures	335
H. Supplementary material of Annex B	349
I. Supplementary material of Annex C	360

Table des figures

1.1. Example of phase line and phase plane	26
1.2. Classification of Fixed-Points	27
1.3. Example of homoclinic and heterocline orbits	28
1.4. Example of limit point bifurcation	29
1.5. Example of Hopf bifurcation	29
1.6. Schematic neuron	31
1.7. Mechanism of action potential described by Hodgkin Huxley model .	33
1.8. Example of class 1 excitatory neuron	36
1.9. Example of class 2 excitatory neuron	36
1.10. Example of phasic spike neuron	37
1.11. Example of phasic burst neuron	38
1.12. Example of tonic spike neuron	38
1.13. Example of tonic burst neuron	39
1.14. Example of spike frequency adaptation neuron	39
1.15. Example of mixed neuron	40
1.16. Example of irregular neuron	40
1.17. Example of delay spike neuron	41
1.18. Example of delay burst neuron	41
1.19. Example of single spike neuron	42
1.20. Example of spike latency neuron	42
1.21. Example of depolarizing after-potential neuron	43
1.22. Example of rebound spike neuron	43
1.23. Example of rebound burst neuron	44
1.24. Example of bi-stability neuron	44
1.25. Example of integrator neuron	45
1.26. Example of resonator neuron	46
1.27. Phase portrait and physiological state diagram of FitzHugh-Nagumo model	47
1.28. adaptive and exponential integrator and fire neuron	49
1.29. Structure of a typical chemical synapse	50
1.30. Model of synaptic current after activation	52
2.1. Distributed vs. Parallel Computing	60
2.2. Computer parts and components	61

2.3.	Slurm components	62
2.4.	Architecture comparison of the Virtual Machine (a), Docker container (b), and Singularity (c)	63
2.5.	Multithreading vs. Multiprocessing	64
2.6.	Interaction of MPI and the process manager through PMI	66
2.7.	Human Brain Project Timeline	67
2.8.	TVB on EBRAINS cloud services	68
3.1.	Multiscale co-simulation design pattern and Example of an application in neuroscience	75
3.2.	The virtual mouse brain experiment	78
3.3.	Spiking neural network in 3 different states of the left CA1	80
3.4.	Three different states of CA1 in mouse brain	81
3.5.	Architecture and performance of the co-simulation	85
4.1.	Network topology	98
4.2.	Three characteristic network dynamics	101
4.3.	Competing network oscillation from noise and long-range connections	102
4.4.	Parameter space network synchronization analysis of adaptation current properties in relation to delay of long-range connections	104
4.5.	Parameter space network synchronization analysis of voltage membrane properties in relation to delay of long-range connections	105
4.6.	Parameter space network synchronization analysis of ratio excitatory/inhibitory synaptic weight versus delay of long-range connection for different values of noise	106
4.7.	Parameter space network synchronization analysis of synaptic weights in relation to delay of long-range connections for different sizes of the network	108
A.1.	Mean-field and network dynamic without adaptation	178
A.2.	Impact of spike-trigger adaptation on the network dynamic	180
A.3.	Fitting of Transfer functions	182
A.4.	Time scale of the excitatory population in the presence of a sinusoidal Poisson generator	183
A.5.	Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 7Hz	184
E.1.	Zoom on 1s for the spiking neural network with regular bursting state	249
E.2.	Mouse Brain activity for Asynchronous Irregular	250
E.3.	Mouse Brain activity for Synchronize Irregular	251
E.4.	Mouse Brain activity for Regular Bursting	252
E.5.	Detail sequence diagram of the co-simulation	253
E.6.	Sequence diagram of the communication protocol with NEST	254
E.7.	State diagram of NEST wrapper	255
E.8.	State diagram of transfer components for interaction with NEST	256

E.9.	Sequence diagram of the communication protocol	257
E.10.	State diagram of wrapper of TVB modules	258
E.11.	State diagram of transfer components	259
E.12.	File organisation of the transformer module	260
E.13.	State diagram of transfer components which transform data from one scale to another	261
E.14.	Structure diagram of the two transfer modules with the description of each component	262
E.15.	Class diagram of the transfer modules.	263
E.16.	Communication between components in the transfer module	264
E.17.	Details of the performance with the increase of neurons	265
E.18.	Details of the performance with the increase of synchronize time	266
E.19.	Details of the performance depending of the number of process and thread for NEST	267
E.20.	Performance of the co-simulation on one supercomputer	268
E.21.	Performance of the co-simulation on one supercomputer	269
E.22.	Performance of the co-simulation on one supercomputer	270
E.23.	Details of the timer of one run for the reference configuration	271
E.24.	Proof of concept of replacing NEST and TVB by other simulators	272
F.1.	View the grid as a torus	284
F.2.	Spike trains of disconnected network	285
F.3.	Parameter space of weight and delay of the long-range connections without noise	286
F.4.	Parameter space of weight and delay of the long-range connections with white noise	287
F.5.	Delay of long-range influence firing rate without white noise	288
F.6.	Parameter space of standard deviation of the connection kernel (σ_{long}) and delay of the long-range connections without noise	289
F.7.	Parameter space of standard deviation of the connection kernel (σ_{long}) and delay of the long-range connections with white noise	290
F.8.	Parameter space of position of the target cell and delay of the long-range connections without noise	291
F.9.	Parameter space of position of the target cell and delay of the long-range connections with white noise	292
F.10.	Parameter space of the effect of the noise and excitatory synaptic weight	293
F.11.	Three characteristics network dynamics without long-range connection for different excitatory synaptic weights in the presence of white noise	294
F.12.	Network and neurons dynamics characteristic function of adaptation current properties	295
F.13.	Parameter space of subthreshold adaptation and delay of the long-range connections with white noise	296

F.14.	Parameter space of spike-triggered adaptation and delay of the long-range connections with white noise	297
F.15.	Parameter space of time constant adaptation and delay of the long-range connections with white noise	298
F.16.	Network and neurons dynamics characteristic function of voltage membrane properties	299
F.17.	Parameter space of heterogeneity and delay of the long-range connections with white noise	300
F.18.	Parameter space of voltage reset and delay of the long-range connections with white noise	301
F.19.	Parameter space of refractory time and delay of the long-range connections with white noise	302
F.20.	Three characteristics network dynamics with uni-direction long-range connection for different refractory time	303
F.21.	Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 100 percent of active neurons	304
F.22.	Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 75 percent of active neurons .	305
F.23.	Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 50 percent of active neurons .	306
F.24.	Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 25 percent of active neurons .	307
F.25.	Parameter space of ratio excitatory/inhibitory synaptic weights and delay of the long-range connections without noise	308
F.26.	Parameter space of ratio excitatory/inhibitory synaptic weight and delay of the long-range connections with white noise (variance :1800pA)	309
F.27.	Parameter space of ratio excitatory/inhibitory synaptic weight and delay of the long-range connections with white noise (variance :2600pA)	310
F.28.	Two characteristics initial condition with uni-direction long-range connection for different weight of long-range connection without noise	311
F.29.	Comparison of the uni and bi-direction of long-range connection . .	312
F.30.	Four characteristic network dynamics with bi-directional long-range connections	313
F.31.	Parameter space of weight and delay of the long-range connections with white noise and bi-directional	314
F.32.	Parameter space of the size of the grid on these two dimensions . . .	315
F.33.	One simulation transition of state	316
G.1.	Variability of fixed-point from network simulation with and without adaptation	335
G.2.	Bifurcation diagram of external excitatory input for spike-trigger adaptation equals 0.0 pA	336
G.3.	Bifurcation diagram of external excitatory input for spike-trigger adaptation equals 30.0 pA	337

G.4.	Bifurcation diagram of external excitatory input for spike-trigger adaptation equals 60.0 pA	338
G.5.	Variability of the measure of the firing rate	339
G.6.	Variability of the measure of the firing rate	340
G.7.	Estimation and comparison of the high fixed-point	341
G.8.	Estimation and comparison of the low fixed-point	342
G.9.	Fitting of Transfer functions full	343
G.10.	Estimation of error of the transfer function for neurons in the presence of negative current	344
G.11.	Negative derivative of the transfer function for zeros firing rate	345
G.12.	Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 7Hz and adaptation ($b = 60.0pA$)	346
G.13.	Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 0Hz	347
G.14.	Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 0Hz and adaptation ($b = 60.0pA$)	348

Liste des tableaux

D.1. Parameter for classification of neuron dynamic	221
E.1. Parameter for co-simulation	248
F.1. Parameter for parameter exploration	282
G.1. Description of the network simulation simulation	329
G.2. Description of the mean-field simulation	334

Première partie
INTRODUCTION

Γνῶθι σεαυτόν : "Know thyself"
Greek philosophical maxim

One way to know thyself is to understand the emergence of thinking from the body, specifically from the brain, more precisely from brain cells, more accurately from molecules, or more minutely from atoms. Thinking is an emergent phenomenon from emergent phenomena at different scales. Modeling the brain from a physical point of view requires integrating knowledge of molecular, subcellular, cellular, brain regions and body [10]. This thesis does not have the ambition of bridging all these scales. However, my interest in grasping the state of the art about how emergent phenomena of neural networks can shape brain activities and contribute to the neuroscience effort led me to develop a software science co-simulation design pattern for multiscale co-simulation of the brain, introduced in the Chapter 3. The composition of this new tool is motivated by the scientific community's desire to create a bridge between macro- and meso- scales [10, 73], and by the wager that coupling a mean-field model and point neuron model offers exciting potential for modeling multiscale experiments. However, it is also essential to be aware of the limitation of this new tool, sketchily abort in Chapter 4 and in the Annex A.

Before describing the co-simulation design pattern, it is necessary to understand the dynamical systems theory, brain modeling, and distributed computation. I shall first introduce dynamical system theory to provide the basic notion to describe complex systems using differential equations. Additionally to this theory, the presentation of bifurcation theory offers one way to analyze a complex dynamical system which helps interpret system behavior and time series. Then, I shall characterize the assumption underlying neuron models and briefly present analyses and classifications of neuronal dynamics. Next, I shall describe constructing a network of point neurons and their reduced version, mean-field models. Afterward, I shall provide elementary concepts for grasping the idea of distributed and parallel computing and associated technology. To finalize the introduction, I include a presentation of the Human Brain Project [235, 234, 12, 13, 11] and EBRAINS [99, 310] to situate the present work in the scientific environment. The following part will introduce this method for coupling simulators and apply it to embedding a spiking neural network in a brain model. Then I will propose an analysis for helping to choose a model for downscaling data between macro- and meso- scales. The analysis is a synchronization analysis of neural networks due to short and long-range connections with time delay. Finally, I shall conclude with the limitation of this approach and the perspective on the usability of the multiscale co-simulation.

The annexes comprise additional work around this thesis and supplementary materials for each chapter. The extra work includes preliminary results of a study for helping to choose a model for upscaling data between macro- and meso- scales. This analysis is a dynamic analysis of the limitation of a mean-field approximation for constant and oscillating stimulation. Moreover, the extra-works include papers I participated in a modeling study of slow-wave sleep and high response wakefulness dynamics and an overview of brain simulation using The Virtual Brain on EBRAINS.

CHAPTER 1

Modelling in neuroscience

Contents of Chapter 1

1.1. Dynamical Systems	25
1.1.1. System and equations	25
1.1.2. Dynamical system theory	25
1.1.3. Fixed-point stability	26
1.1.4. Bifurcation theory	28
1.1.5. Software for numerical continuation	30
1.2. Neuron model	30
1.2.1. What is a Neuron?	30
1.2.2. Hodgkin and Huxley model	32
1.2.3. Characterize neurons	34
1.2.4. Point neuron models	46
1.2.4.1. FitzHugh Nagumo neuron	47
1.2.4.2. Leaky integrator and fire neuron	48
1.2.5. Adaptive and exponential integrator and fire neuron (Ad Ex)	48
1.3. Network of point neurons	50
1.3.1. Model of synapse	51
1.3.2. Network connectivity	53
1.3.3. Characterization of a neural network	54
1.3.4. Network of adaptive exponential integrate and fire neurons	55
1.4. Mean-field model	56
1.4.1. Neural Mass AD EX	57

1.1. Dynamical Systems

1.1.1 System and equations

One way to understand a system is to mimic it with mathematical models, i.e. equations, which describe the evolution of variables. The minimal number of independent variables for describing the model's state are called state variables; other variables are intermediate or output variables. The model is characterized by its dimension, i.e. the number of its state variables, and by its state space, i.e. the geometric space in which the variables on the axes are the state variables. The equation also contains two additional elements : the parameters and the functions of the evolution of variables. The main difference between parameters and variables is that during the model simulation, parameters are fixed and variables are changing.

For example, the simple mathematical model of a capacitor is the following equation :

$$\frac{\partial V}{\partial t} = \mathbf{f}(I(t), C) = \frac{I(t)}{C} \quad (1.1)$$

This equation approximates the time evolution of the compelling quantity of the capacitor, i.e. voltage (V) and current (I). In this case, I is defined before the simulation, i.e. it fixes, and V evolves in time. This means that V is the unique variable of the model. \mathbf{f} is the function that represents the approximation of the behavior of the capacitor and has one parameter C , the capacitance.

1.1.2 Dynamical system theory

Dynamical system theory [146] is the study of trajectories, i.e. sequences of model states. In this thesis, we will mainly focus on analyzing autonomous systems evolving in time, i.e. the evolution functions of variables are time-independent. One mathematical representation of autonomous systems is ordinary differential equations, i.e. equations with one or more unknown functions with only one independent variable and involving some of their derivatives. Due to the time independence, in all points of the state space of an autonomous system, it is possible to compute differential vectors, i.e. vectors representing the direction the system will take from a point in state space. A vector field is one representation of a dynamical system that takes a subset of points in state space and assigns a differential vector to them. Phase line and phase plane are the names for vector fields of one and two dimensions. The geometrical interpretation of these fields gives essential information about the model's dynamics, and it is one of the main tools of dynamic system theory.

The most interesting point in this vector field is when the vector is null, i.e. the variables are constant in time, and the differential of each variable equals zero. These points are called equilibrium points or fixed-points.

1.1.3 Fixed-point stability

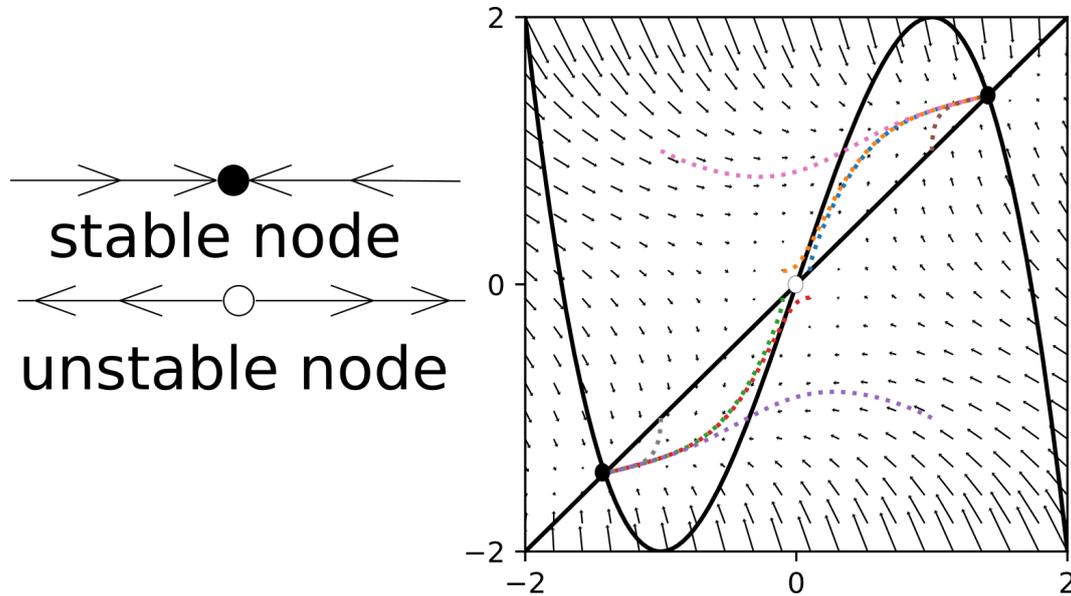


FIGURE 1.1. – Example of phase line and phase plane

The left panel is an example of a phase line for stable and unstable fixed-points. The right panel is an example of a phase plane with a vector field that contains two fixed-points and one unstable point. The two black lines are the nullcline of two system variables, and the dashed lines are trajectory examples.

These fixed-points are categorized as unstable/repulsive, stable/attractive, and saddle [278] (see left panel of figure 1.1). These categories are based on the local or Lyapunov stability and asymptotic stability of the fixed-point [229].

The Lyapunov stability : x^e is a stable fixed-point, if for every neighbourhood U of x^e there is a neighbourhood $V \subseteq U$ of x^e such that every solution $x(t)$ starting in V ($x(0) \in V$) remains in U for all $t \geq 0$, else x is an unstable fixed-point.

The asymptotically stable : x^e is a stable fixed-point, if it is Lyapunov stable and additionally V can be chosen so that $\lim_{t \rightarrow \infty} |x(t) - x^e| = 0$ for all $x(0) \in V$.

In two dimensional system, the intersection of two nullclines, i.e. lines where the differential of one state variable is zero, is a fixed-point (see right panel of figure 1.1). In the case of higher dimensions or complex functions, the stability demonstration of fixed-points is difficult. One possibility is to use the stability theory based on the system's robustness to a small perturbation. This can be done by linearizing the system close to this point and calculating the value of the Jacobian matrix [180], i.e. matrix contains all the first-order partial derivatives calculated at this point. The properties of this matrix give information about the stability of the fixed-point [362] (see figure 1.2). The sign of Jacobian determinants distinguishes between saddle (positive) and stable or unstable fixed-points (negative). A null determinant means the presence of

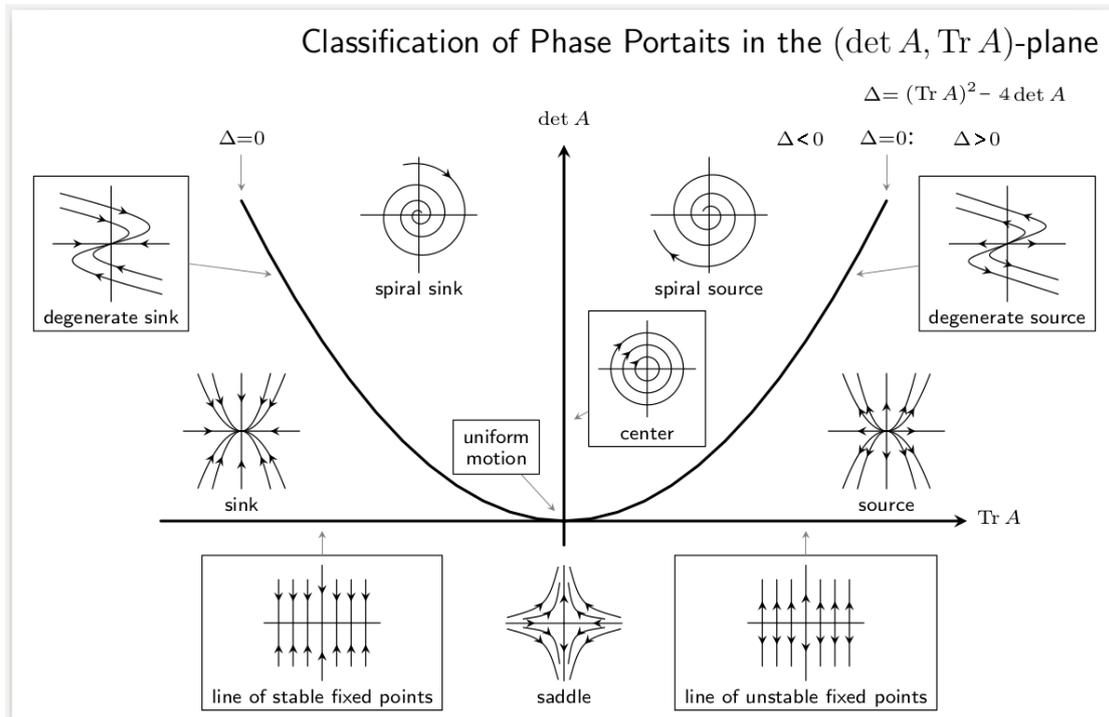


FIGURE 1.2. – Classification of Fixed-Points

Classification of linear autonomous system fixed-points $x' = Ax$, between stable or unstable. Stable fixed-points are on the left of the diagram, and unstable fixed-points more on the right (adapted from Systems of Linear Differential Equations Stability Analysis by Elmer G. Wiens [362])[331].

a line of fixed-points. The sign of Jacobian traces gives the stability of the point. If it is positive, the point is unstable. If it is negative, the point is stable. If it equals zero, the local trajectories are circles or uniform. The Jacobian discriminant defines the limits between a local circular trajectory and linear trajectories. A null discriminant is a degenerative fixed-point. A similar conclusion can be made by looking at the sign of the real part of the eigenvalue of the Jacobian and the presence of the conjugate eigenvalues.

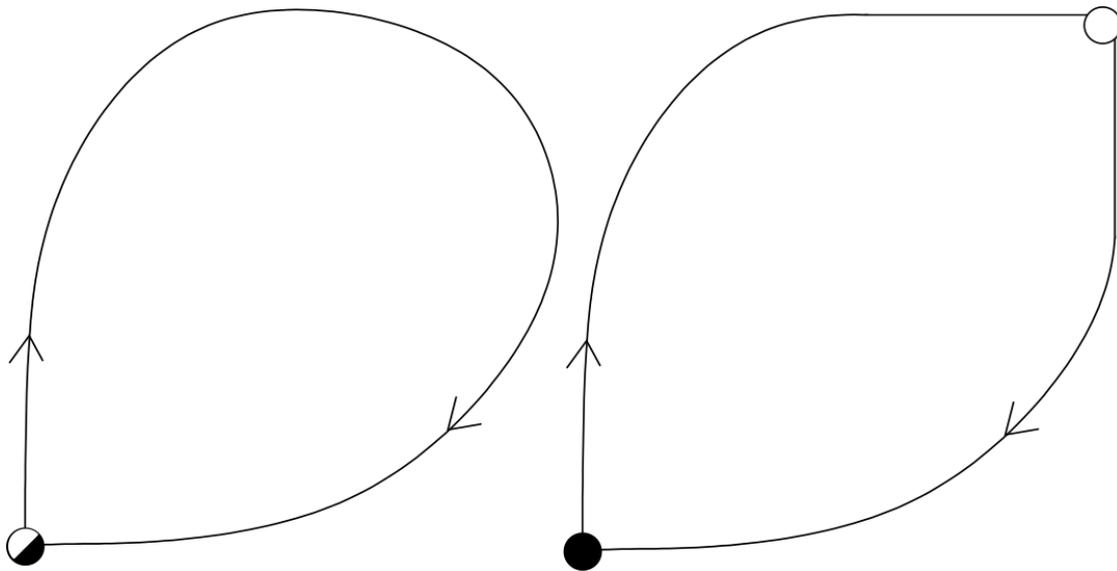


FIGURE 1.3. – Example of homoclinic and heterocline orbits

The left figure displays a homoclinic orbit with one saddle fixed-point. The right figure illustrates a heteroclinic orbit with a stable and unstable fixed-point.

Moreover, in two and higher dimensions, closed trajectory indicates a periodic orbit or cycle ($(x(0), y(0)) = (x(T), y(T))$ for some $T \geq 0$). If this trajectory is in the plane, this orbit is called limit cycle. It is also possible to calculate the stability of the limited cycle based on its first recurrence map or Poincaré map [279]. The Poincaré map is the intersection of the limit cycle with a low perpendicular dimension plan. Orbits with an infinite period are categorized into two types : homoclinic orbits, i.e. the orbit contains only one fixed-point, a saddle fixed-point, and heterogenous orbits, i.e. the orbits contain one attractor and one repulsive fixed-point (see figure 1.3).

1.1.4 Bifurcation theory

So far, the parameters have been invariant. Modifying the parameters modifies the vector field and the stability and position of the fixed-points and orbits. Bifurcation theory [15, 278, 335, 146] is the analysis of this modification for characterizing the system's dynamics depending on parameter values. One result of this analysis is the bifurcation diagram, a map that divides parameter space in dynamically equivalent systems. There exist at least four types of bifurcation of order one, i.e. based on the variation of one parameter : limit point, Hopf, pitchfork and transcritical [335, 146]. The following paragraph details the limit point and the Hopf bifurcation because the analyzed models in this thesis present these two bifurcations.

Saddle node, limit point or blue sky bifurcation [15](see figure 1.4) :

The variation of the fixed-point will push a stable and unstable point to collide and disappear. At the bifurcation, the system has only one saddle fixed-point (see figure 1.2). The simplest bifurcation of this type is $\dot{x} = \mu - x^2$ where $\mu \in \mathbb{R}$ [146]. From the point of view of stability, the bifurcation occurs when an eigenvalue equals zero.

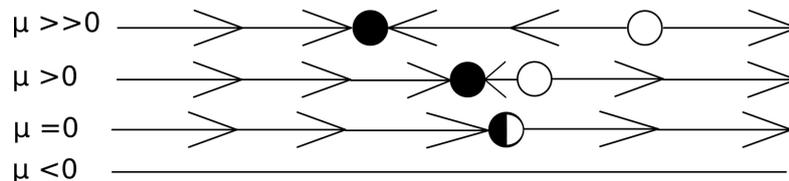


FIGURE 1.4. – Example of limit point bifurcation

The two fixed-points merge and disappear by changing the parameter μ .

Poincaré-Andronov-Hopf bifurcation [15, 165, 278](see figure 1.5) :

In this case, a limited cycle collapses to a fixed-point when the parameters vary. The simplest bifurcation of equilibria is $\dot{z} = z((\mu + i) + (\alpha + \beta i)|z|^2)$ where $z \in \mathbb{C}$ and $\mu, \alpha, \beta \in \mathbb{R}$. From the point of view of stability, the bifurcation occurs when a pair of eigenvalues become pure imaginary. Depending on the sign of α , the Hopf is supercritical; when the stable limit circles collapse with an unstable fixed-point or subcritical; when an unstable limit circle collapse with a stable fixed-point.

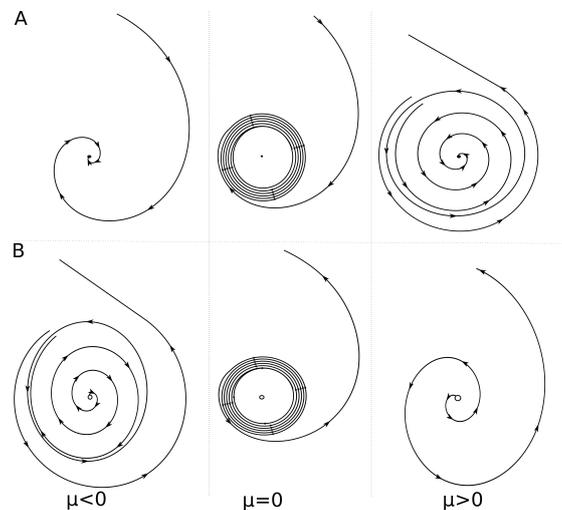


FIGURE 1.5. – Example of Hopf bifurcation

The limit circle merges with the fixed-point by changing the parameter μ . **A** supercritical case where the fixed-point is stable. **B** subcritical case where the fixed-point is unstable (adapted from KUZNETSOV [216])

1.1.5 Software for numerical continuation

The bifurcation of complex nonlinear models is very difficult to perform mathematical analysis. In this case, the mathematical analysis is coupled with computational simulation and numerical continuation [215], which only approximate the bifurcation map. In this thesis, numerical continuation was the most used methodology. The idea of numerical continuation is to trace a fixed-point or a bifurcation when one or multiple parameters change. At the same time as tracking it, the stability study identifies the occurrence of bifurcation. Multiple software realize this analysis [30], such as Matcont [94], PyDsTool [63], CoCo [77], XppAuto [109].

However, the numerical bifurcation analysis in this thesis has been done with Matcont [94] because the analyzed equations are ordinary differential equations, the detection of bifurcation is available for the fixed-point and limited cycle, and the code is open source. This last property allows minor modifications for handling errors created by the existence of a fixed-point at the limit of the state space.

1.2. Neuron model

The brain is composed of structural and functional tissues [287]. The structural tissue, stroma, is a matrix of cells that supports the functional cells' position and movement. The functional cells are categorized into two classes : excitable cells, neurons, and non excitable cells, glia. This thesis used brain models that consider only neurons due to their two properties : intercellular communication using neurotransmitter molecules in synaptic clefts or cytoplasmic continuities, gap junctions, and electric signal propagation by changing its membrane potential.

1.2.1 What is a Neuron?

As electrical signal propagation is the dominant phenomenon that characterizes neurons, this section focuses on its model and characterization. The electric signal propagation is based on the variation of the neuronal membrane potential using ion channels, membrane molecules that adjust ion concentration inside and outside the cell. The three main types of ion channels are leakage channels, voltage-gated ion channels and ion pumps. A leakage channel is an open channel that creates a concentration gradient between the inside and outside of a cell for a specific type of ion. A voltage-gated ion channel is a channel with the capability to open and close depending on the membrane potential. An ion pump is a molecule that pumps some ions from one side to the other side of the membrane. Other ion channels exist, and their behavior is more complex and possibly dependent on electrical, mechanical, or chemical signals [194]. Models present in the thesis only consider the three main channels and the two principal neurotransmitter ion channels in the synaptic cleft.

Inspired by the input-process-output pattern, neurons are modeled with three modules (see Figure 1.6) :

- dendrites : (input) branches connect to other neurons.
- soma : (process) the cell body which contains the nucleus and the functional material.
- axons : (output) roots connected to other neurons.

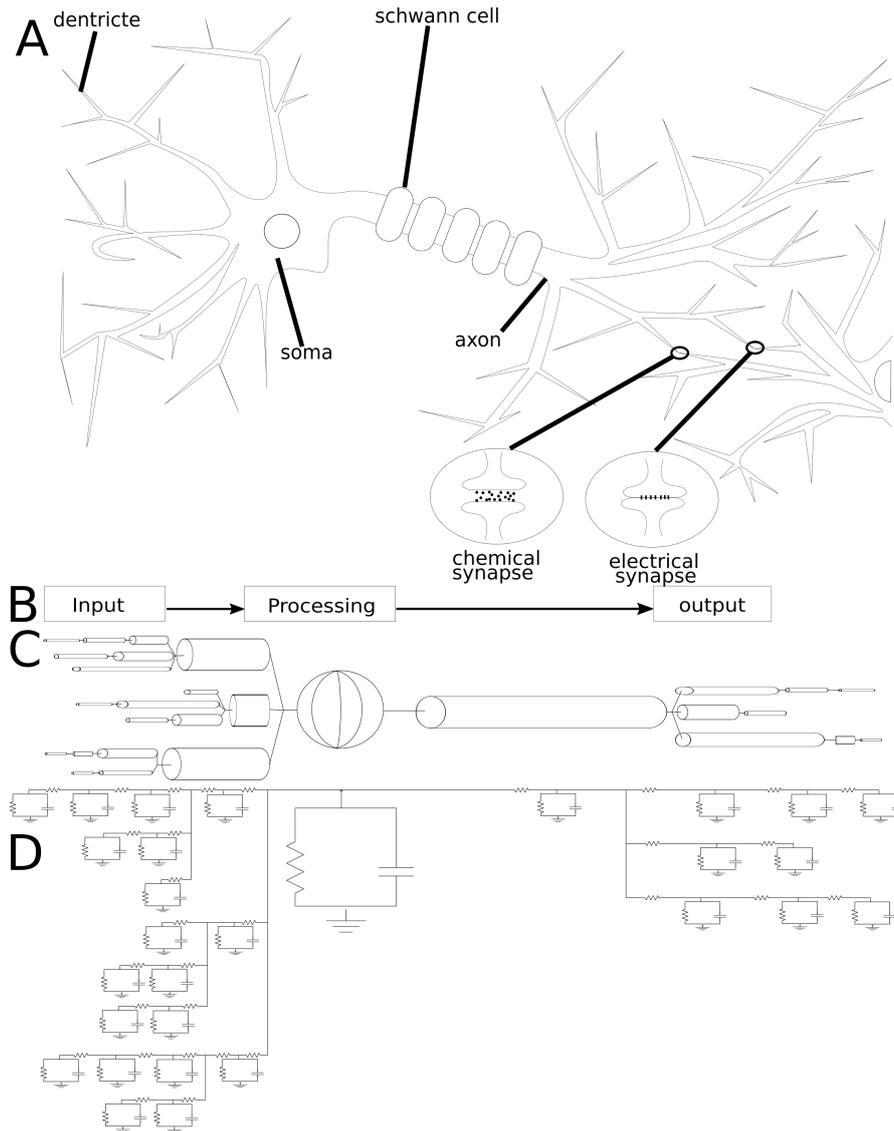


FIGURE 1.6. – Schematic neuron

Different views of a neuron and synapses. A : Cartoons of neurons with synapses. B : Conceptual model of a neuron. C : Representation of cable model composes of cylindrical isopotential compartments. D : Representation of compartmental model composes of discrete isopotential R-C compartments, the equivalent electrical circuit of the cylindrical compartments. Figured adapted from [34], from [176] and from [346]

This modeling approach leads to the simplest vision of the neuronal signaling process : dendrites collect inputs from the synapses and transfer them to the soma by depolarizing the membrane, i.e. generating an electric current; at a specific voltage (voltage threshold) the soma initiates an action potential, i.e. a self-regenerating wave of electrical activity, at the beginning of the axons; axons propagate this wave until the synapses. One common approximation of a neuron is its reduction to a point representing its soma, receiving the summation current from the connection with other neurons and transmitting an event when the voltage passes a threshold. This approach realized a significant reduction because the point neuron model does not consider the variety of interaction between incoming signals in the dendrites, i.e. the dendrite morphology, and the variation of the signal propagation along the axons, for example, the fluctuation of extra-cellular space. Other modeling approaches, such as compartmental model (neuronal model based on a tree of isopotential cylinders of different diameters and lengths) and cable model (neuronal model based on the equivalence of a neuron to an electrical circuit.) [333], integrate some of these details. However, for a better representation of ion concentration fluctuation, it is necessary to include glial cells in the modeling [287].

1.2.2 Hodgkin and Huxley model

The first historical model [111] with a comprehensive description of the action potential is the Hodgkin and Huxley model [162]. This model includes leakage channels, voltage-gated ion channels, and ion pumps. This model is based on the assumption that the dynamic of the action potential is mainly due to the difference in concentration of sodium ion, Na^+ , and potassium ion, K^+ , between intra- and extra-cellular.

$$\begin{aligned}
 c_m \frac{dV_m}{dt} &= -\bar{g}_L(V - E_L) - \bar{g}_{Na} m^3 h (V - E_{Na}) - \bar{g}_k n^4 (V - E_k) \\
 \frac{dn}{dt} &= \Phi[\alpha_n(V)(1 - n) - (\beta_n(V)n)] \\
 \frac{dm}{dt} &= \Phi[\alpha_m(V)(1 - m) - (\beta_m(V)m)] \\
 \frac{dh}{dt} &= \Phi[\alpha_h(V)(1 - h) - (\beta_h(V)h)]
 \end{aligned} \tag{1.2}$$

where c_m is the capacity of the membrane. $-\bar{g}_L(V - E_L)$ describe leak current due to the leakage ion channels, which creates a balance of concentration of Na^+ sodium, K^+ potassium and Cl^- calcium ions between intra- and extra-cellular environments. The \bar{g}_L is the leak conductance, the membrane permeability constant to ions in a passive situation. E_L is the leak reversal potential, the current at the equilibrium point of ionic concentration between intra- and extra-cellular space. The membrane capacity, leak conductance, and leak reversal potential can be quantified by measuring the resting voltage and the membrane voltage at rest and by recording membrane voltage during current injections. These measures assume that the cell is isopotential, i.e. the

membrane potential is uniform at all the points of the cell.

The $-\bar{g}_{Na}m^3h(V - E_{Na})$ is the current due to sodium voltage-gated channels. These channels are responsible for the depolarization of the cell, i.e. the increase phase of the action potential. \bar{g}_{Na} and E_{Na} are, respectively, the leak conductance and the reverse potential of the sodium channels. These values can be quantified by voltage clamp of the cell [192]. The original voltage clamp by Hodgkin and Huxley [162] is an experiment where two electrodes are implanted in the squid giant axon and a third outside. One inside electrode monitors the transmembrane voltage, and the two others inject the current to have the membrane at constant voltage (clamped). The measure of injecting current associated with voltage steps reflects the flow of ion currents. By removing the concentration of ions in extra-cellular space or by adding a channel blocker such as tetrodotoxin or tetraethylammonium, it is possible to quantify the effect of a specific ion channel on transmembrane voltage and find the value of m and h .

Similarly, the equation $-\bar{g}_kn^4(V - E_k)$ describes the current due to potassium voltage-gated channels. These channels are responsible for the repolarization of the cell, i.e. the falling phase of the action potential. \bar{g}_k and E_k are, respectively, the leak conductance and the reverse potential of the potassium channel. n describes the voltage dependence of the ion channels. Moreover, hyper-polarization after an action potential reduces the probability of the neurons creating a second action potential due to the sodium channels being closed and requiring some time for recovery.

The last constant of the equation is Φ is a temperature factor. The ion channels are exponentially dependent on the temperature ($\Phi = Q_{10}^{(T-T_{base})/10}$ where Q_{10} is the ratio ratio rates for increase in temperature of 10°C and T_{base} is the default temperature (37°C) [192]). In your case, we consider that the temperature is constant.

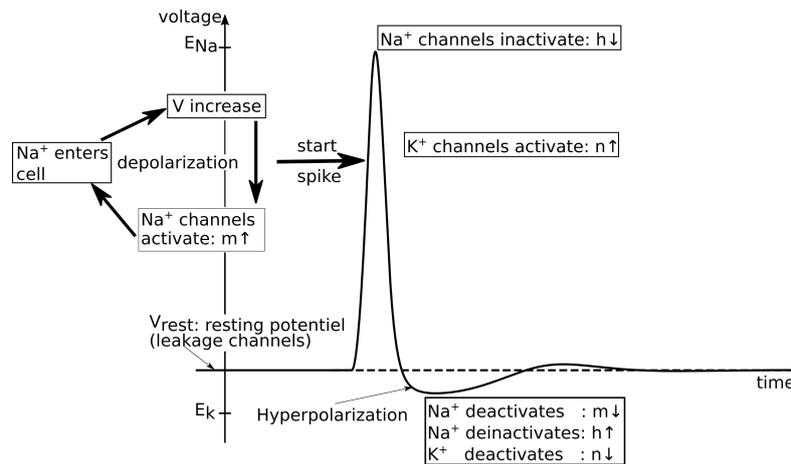


FIGURE 1.7. – Mechanism of action potential described by Hodgkin Huxley model
Figure adapted from figure 1.15 of [111] and figure 2.6 of [346]

This model's advantage is the possibility of adding other channels, such as fast sodium currents or slow potassium currents [111]. At the same time, it is a disadvantage because it increases the complexity of the model. Suppose we reduce the neurons as

the creation of action potential. In that case, analyzing a network with this model can require significant computational resources, and finding the underlying mechanisms of emergent dynamics cannot be easy. The following part describes the models used in the thesis after a characterization dynamic characterization of the neurons.

1.2.3 Characterize neurons

The model, as a simplification of reality, can be used for understanding an experience or the validation hypothesis about the mechanism of your system. The choice of the model and features for fitting and validating the model are crucial. The feature of neurons are multiple; it can be the morphology, the type of ion channels, and the electrophysiology. The essential features for point neuron models are the electrophysiology and the time of action potentials or spikes (for models, we used the terms spikes for an action potential).

Based on the Electrophys Feature Extract Library [31], one library of Ebrains, there is more than 100 features which can be extracted from an electrophysiology recording of neurons. These features are grouped into three categories : spike shape, spike event, and voltage features.

The principal feature used in this thesis is based on spike event features, especially on the firing rate, the presence of burst, regularity of the spikes, and firing rate adaptation.

The firing rate is the estimated number of spikes a neuron generates during one second. The burst is defined as a sequence of at least two spikes in a short time (In this thesis, the short time is defined as a fixed interval of 10 milliseconds, but the common definition of burst is defined as an interval between two spikes belonging to the smaller interval of bi-distribution of the interval of spikes times.)

For the regularity of neurons, there are two features : The coefficient of variation of the inter-spiking interval (ISI) is a simple method used since 1960 [274].

$$CV_{ISI} = \frac{\sigma_{ISI}}{\mu_{ISI}} \quad (1.3)$$

This measure is an evaluation of the regularity of the generation of spikes. When CV_{ISI} is close to zeros (in the thesis < 0.2), the interval between spikes should not have a significant variance, which means the neuron is regular spiking; in another case, the neuron is irregular spiking.

This measure has some deficits, especially in the case of burst, because the bi-distribution bias the estimation of the mean and the variance [320]. To include this mixed distribution, Holt, Softky, Koch, and Douglas [164] introduce CV_2 for comparing two adjacent inter-spike intervals. This new metric was improved in Shinomoto 2003 [321] and

renamed local variation of the inter-spiking interval.

$$LV_{ISI} = \frac{1}{M-1} \sum_{i=1}^{M-1} \frac{3(t_m - t_{m+1})^2}{(t_m + t_{m+1})^2} \quad (1.4)$$

where M is the total of spikes at the time of analysis.

The neurons can be classified based on these measures and the injection of currents. The firing pattern of neurons to current injection is a standard methodology in electrophysiology for characterizing them, similar to the voltage clamp experimentation for categorizing the ion channels. The following sections will present categorizations of dynamics based on Izhikevich et al. 2004 [178], which summarize the behavior of biological neurons along 20 responses to injected current, current ramps, and positive or negative pulse and Naud et al. 2008 [254] which characterize the dynamic of adaptive exponential integrate and fire neurons, the model of point neuron principal used in this thesis.

The first classification is based on the excitability of the neurons. In 1964, Hodgkin [161] proposed three classes for categorizing neurons. The class 1 neurons have a firing rate depending on the input strengths. The class 2 neuron starts to spike at a relatively large frequency when a certain level of input strength is reached. The third class is composed of the other neurons. There are multiple methods for distinguishing these dynamics. One possibility is to stimulate the neurons with a ramp current stimulation, and another option is to trace the Input Frequency curves. If the Input Frequency curve presents a jump when the neurons start to fire, it means that the neurons are class 2; in another case, the neurons are class 1.

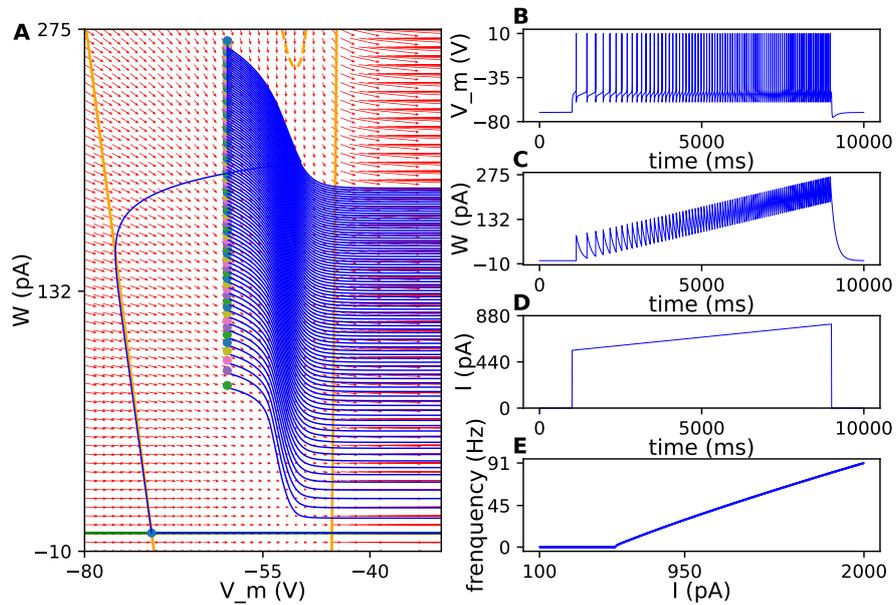


FIGURE 1.8. – Example of class 1 excitatory neuron

A Phase plane of adaptive exponential integrator and fire neurons with an example of parameters for class 1 excitatory neuron with a trajectory of the model to the response of a ramp current stimulation. The green curve presents the nullcline of the adaptation current, and the yellow is the nullcline of the voltage membrane (full : without stimulus and dashed with the maximum of input current) **B** Voltage of the neuron for a ramp current stimulation. **C** Adaptation current of the neuron for a ramp current stimulation. **D** Current of a ramp stimulation **E** Input frequency curve of the neurons evaluate over 10 seconds. (see annex D for parameters of the neuron)

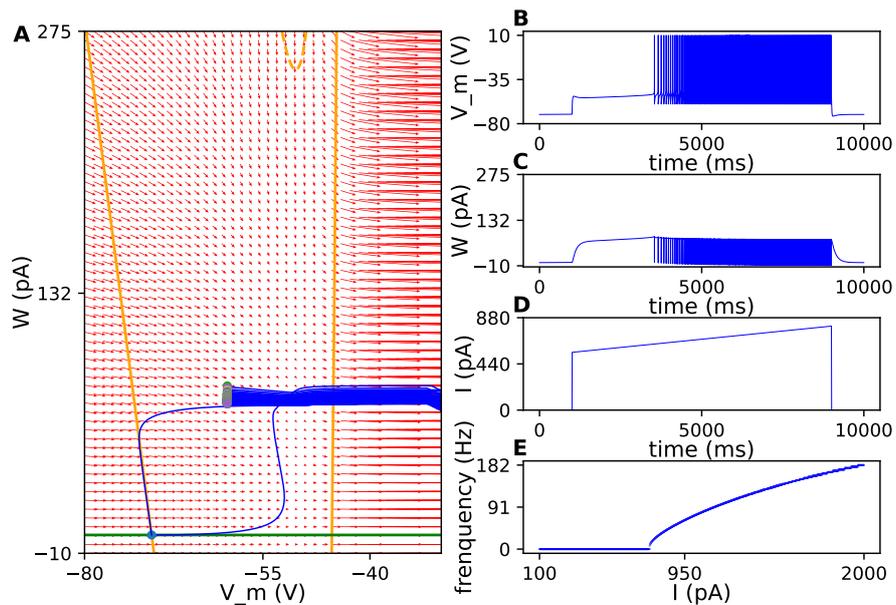


FIGURE 1.9. – Example of class 2 excitatory neuron

The second classification is based on the response of the neuron to constant stimulation : phasic spike (generate one spike), phasic burst (generate one burst), tonic spike or regular spiking (generate regular spike), tonic burst or regular bursting (generate regular burst), spike frequency adaptation (increase the frequency of the spikes), mixed model (start with a burst and continue with spikes), irregular or chaos (spike without any regularity), delay spikes (generate regular spike but the first one is delayed) and delay burst (generate regular burst but the first one is delayed).

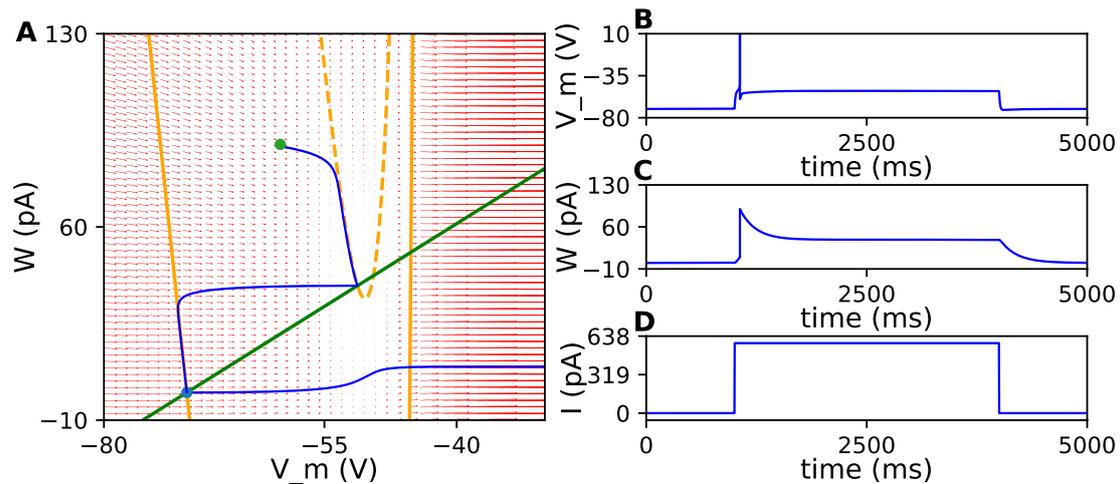


FIGURE 1.10. – Example of phasic spike neuron

A Phase plane of adaptive exponential integrator and fire neurons with an example of parameters for phasic spike neuron with a trajectory of the model to the response of a step current stimulation. The green curve presents the nullcline of the adaptation current, and the yellow is the nullcline of the voltage membrane (full : without stimulus and dashed with the step of input current) **B** Voltage of the neuron for a step current stimulation. **C** Adaptation current of the neuron for a step current stimulation. **D** Current of a step stimulation (see annex A for parameters of the neuron)

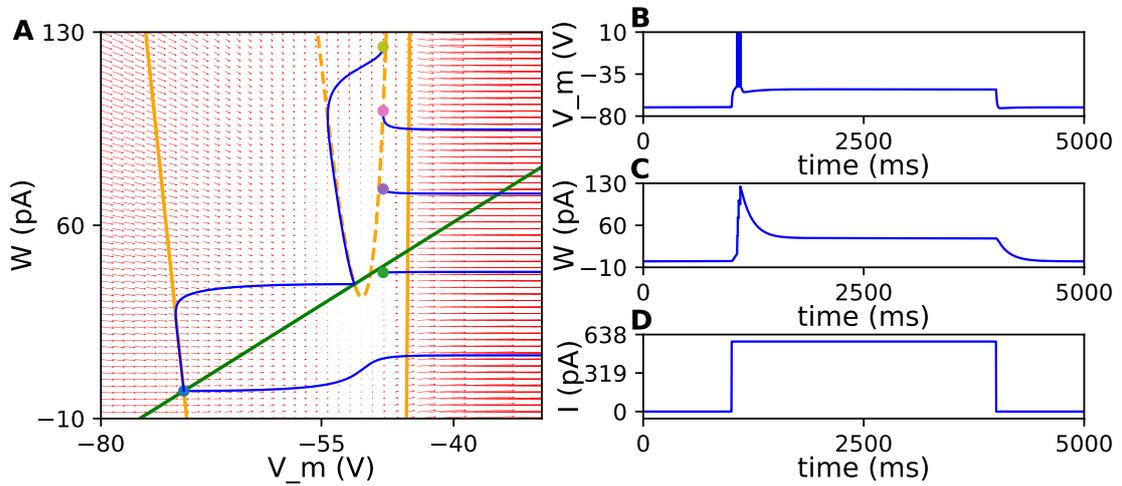


FIGURE 1.11. – Example of phasic burst neuron

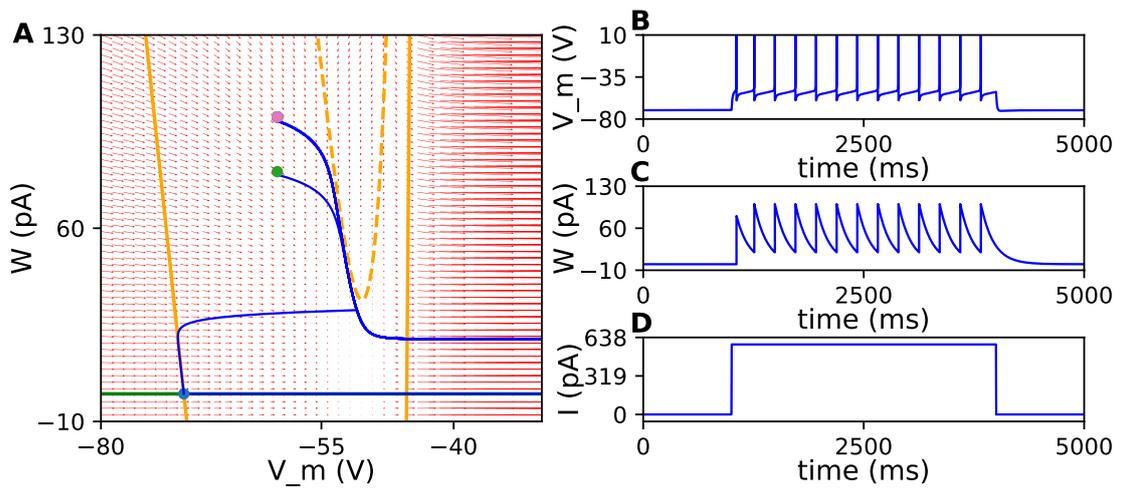


FIGURE 1.12. – Example of tonic spike neuron

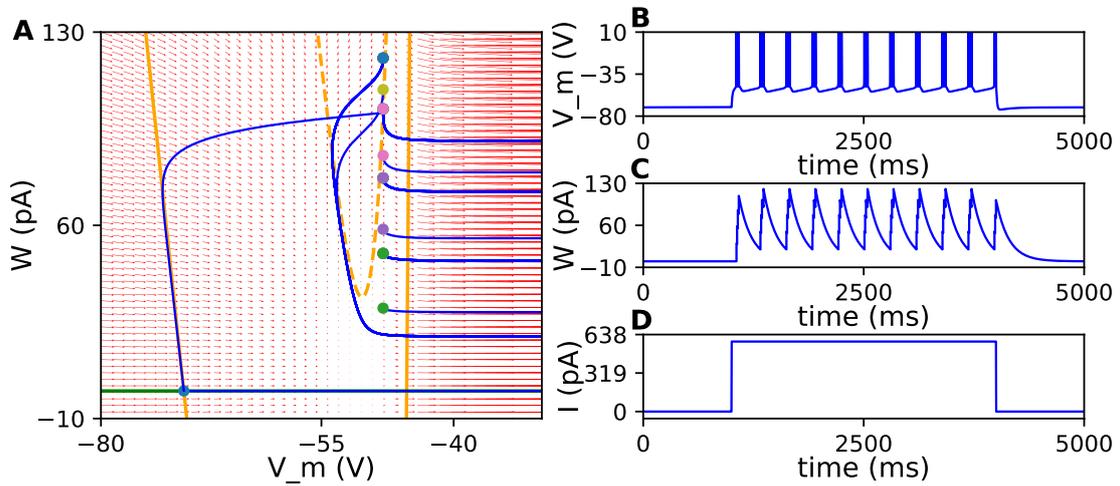


FIGURE 1.13. – Example of tonic burst neuron

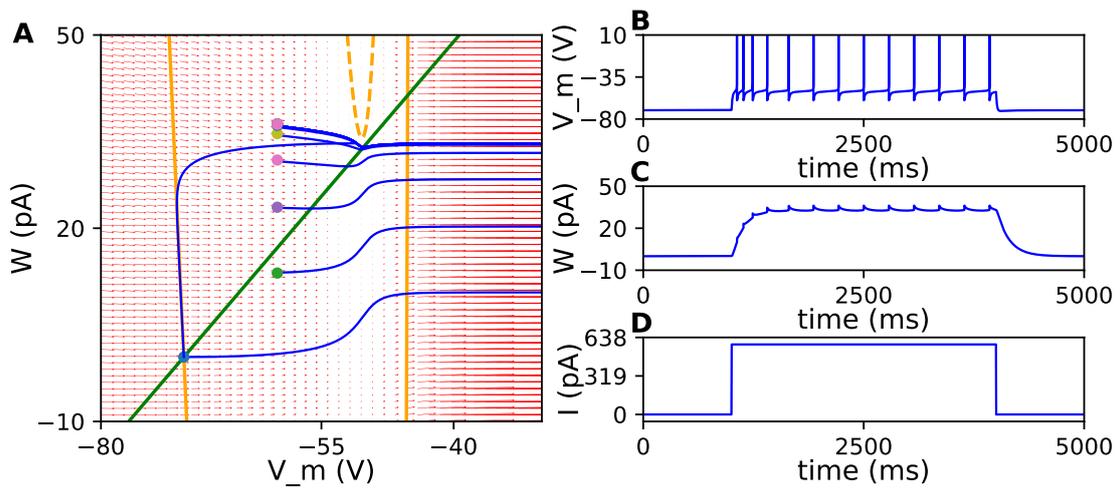


FIGURE 1.14. – Example of spike frequency adaptation neuron

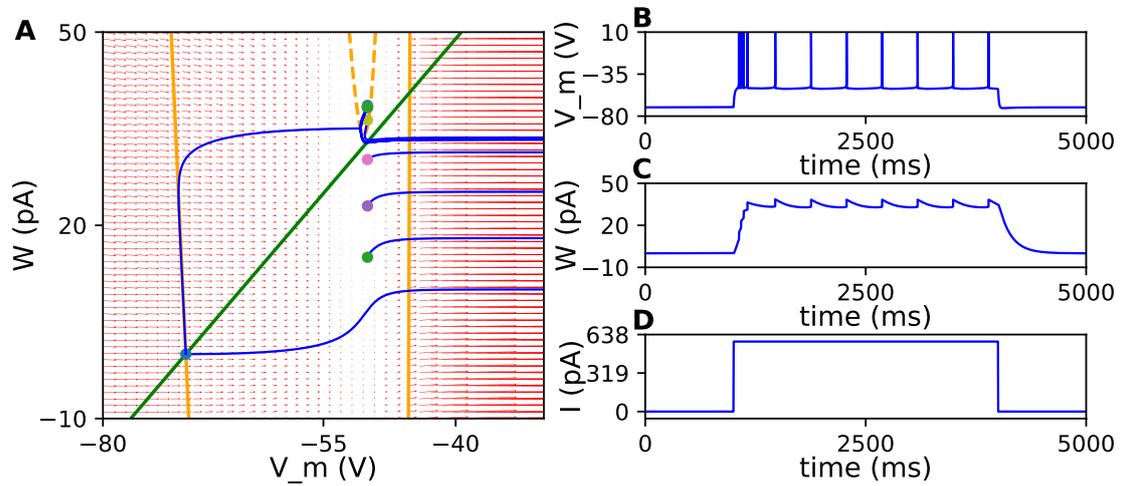


FIGURE 1.15. – Example of mixed neuron

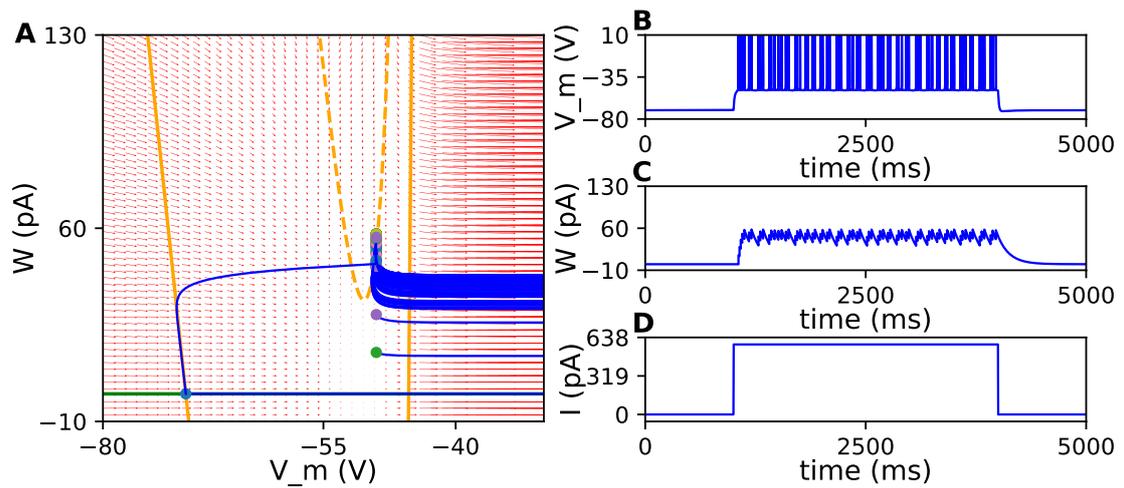


FIGURE 1.16. – Example of irregular neuron

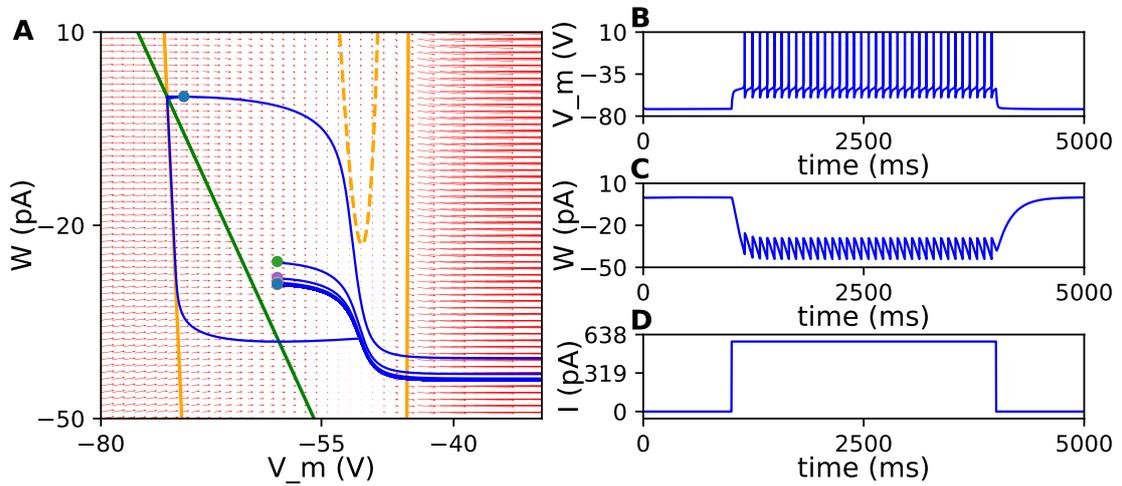


FIGURE 1.17. – Example of delay spike neuron

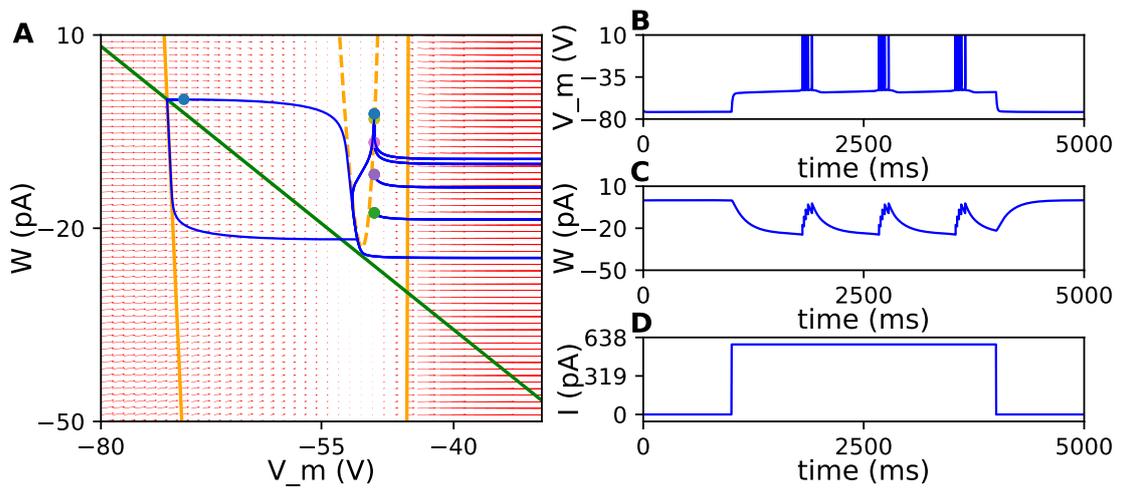


FIGURE 1.18. – Example of delay burst neuron

1. Modelling in neuroscience – 1.2. Neuron model

The third classification is based on the response of the neuron to one positive or pulse : single spike (generate a spike), spike latency (generate a spike after some delay), depolarizing after-potential (depolarization after a spike), rebound spike (spike after a negative pulse), rebound burst (burst after a negative spike) and bi-stability (a pulse push the neurons to spike or silence).

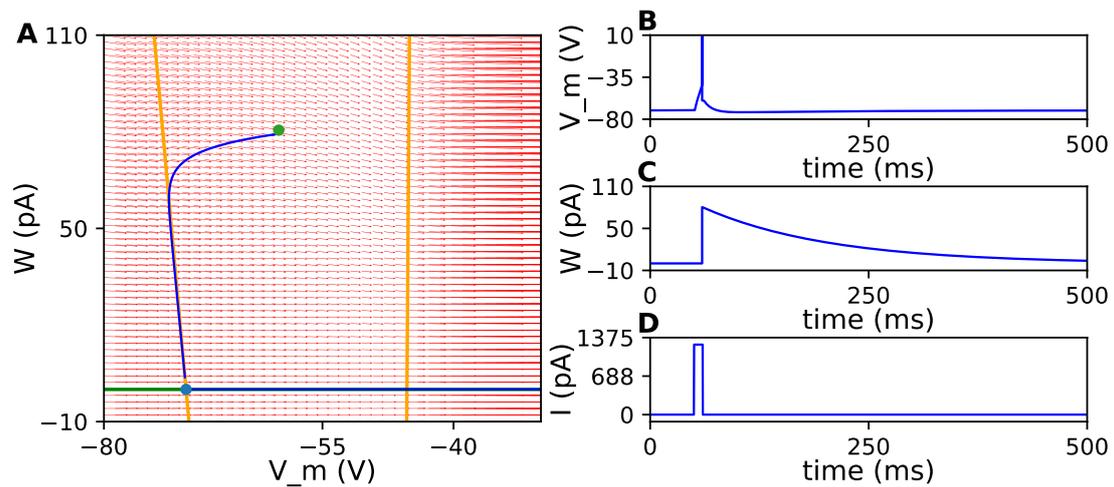


FIGURE 1.19. – Example of single spike neuron

A Phase plane of adaptive exponential integrator and fire neurons with an example of parameters for single spike neuron with a trajectory of the model to the response of a pulse. The green curve presents the nullcline of the adaptation current, and the yellow is the nullcline of the voltage membrane (full : without stimulus and dashed with the current pulse) **B** Voltage of the neuron for a pulse stimulation. **C** Adaptation current of the neuron for a pulse stimulation. **D** Current of a pulse stimulation (see annex A for parameters of the neuron)

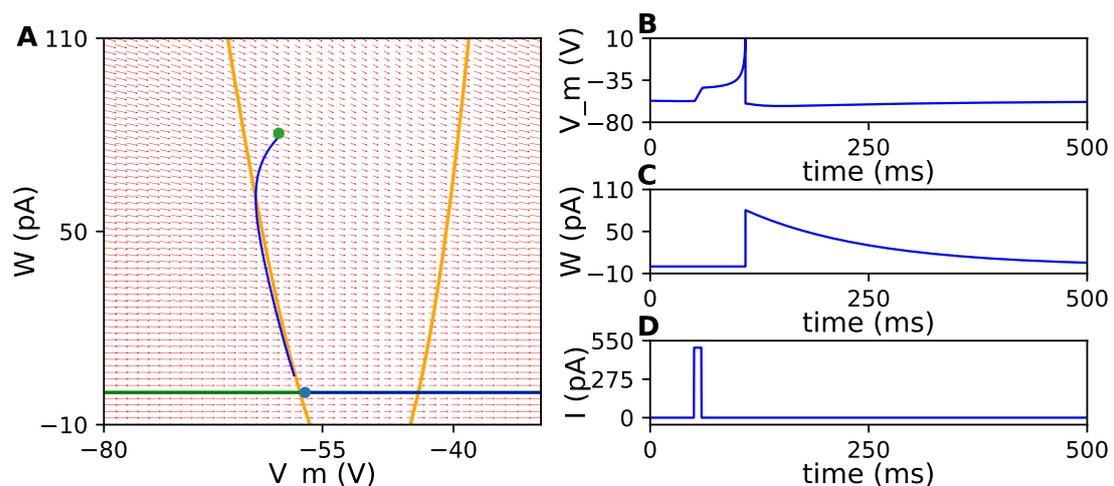


FIGURE 1.20. – Example of spike latency neuron

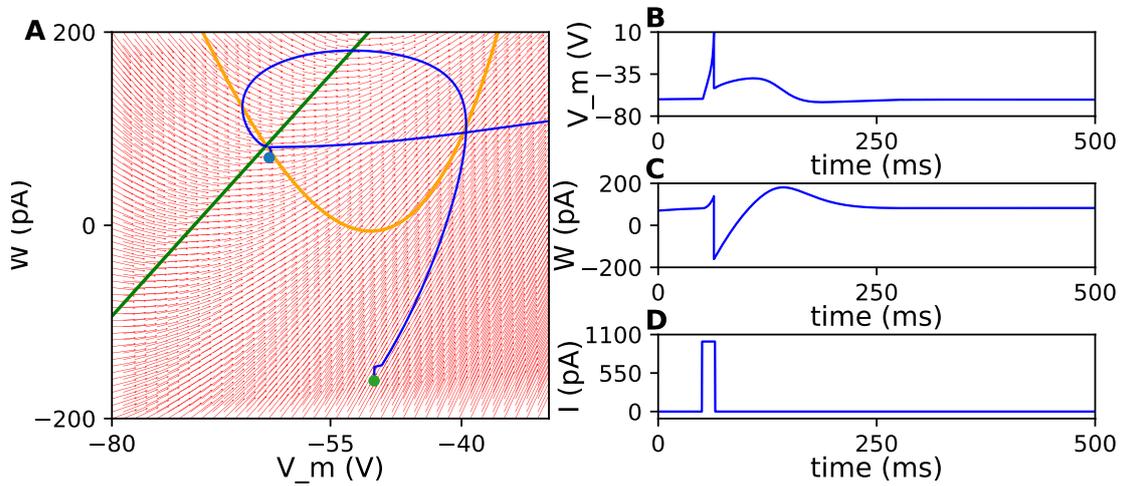


FIGURE 1.21. – Example of depolarizing after-potential neuron

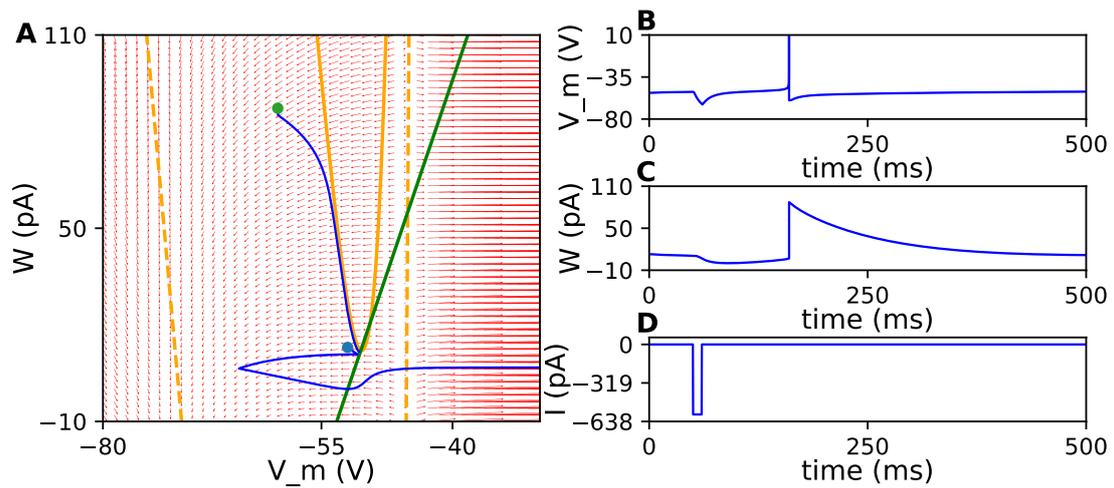


FIGURE 1.22. – Example of rebound spike neuron

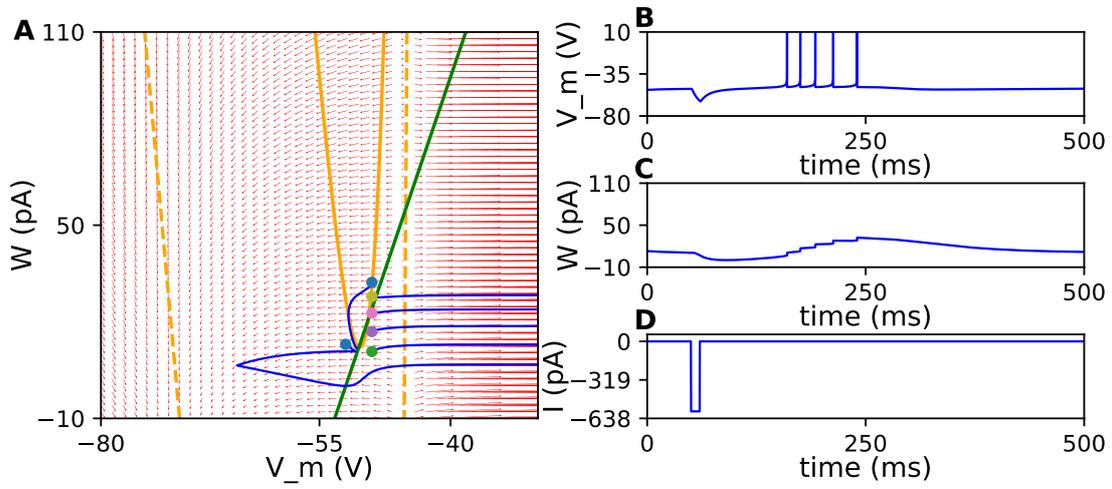


FIGURE 1.23. – Example of rebound burst neuron

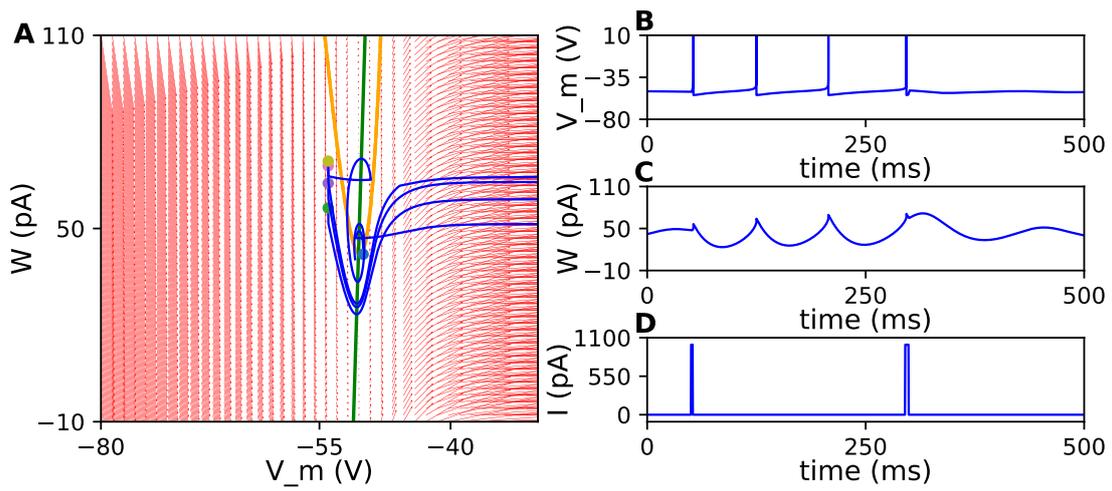


FIGURE 1.24. – Example of bi-stability neuron

The fourth classification is based on the neuron's response to one small permutation : integrator (return linearly to the same voltage) and resonator (oscillate after a perturbation). These two classes affect spike time differently in the presence of perturbation when there are oscillating. One possible quantification of this effect is the Phase Response Curve [85]. The PRC for spiking neurons measures the modification of the inter-spiking interval when a perturbation is applied at a different time of this interval. The integrator has monophasic (type I) PRC, and the resonator has biphasic (type II) PRC [154].

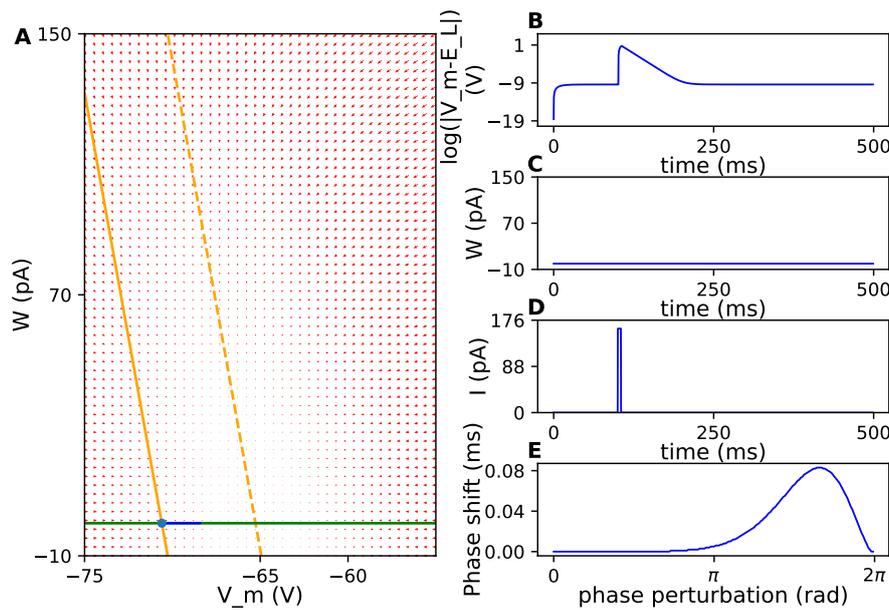


FIGURE 1.25. – Example of integrator neuron

A Phase plane of adaptive exponential integrator and fire neurons with an example of parameters for integrator neuron with a trajectory of the model to the response of a small perturbation. The green curve presents the nullcline of the adaptation current, and the yellow is the nullcline of the voltage membrane (full : without stimulus and dashed with the maximum of input current) **B** Voltage of the neuron for a small perturbation. **C** Adaptation current of the neuron for a small perturbation. **D** Current of a small perturbation **E** Phase response curve of the neuron after a transition of 1 second. (see annex A for parameters for the neuron)

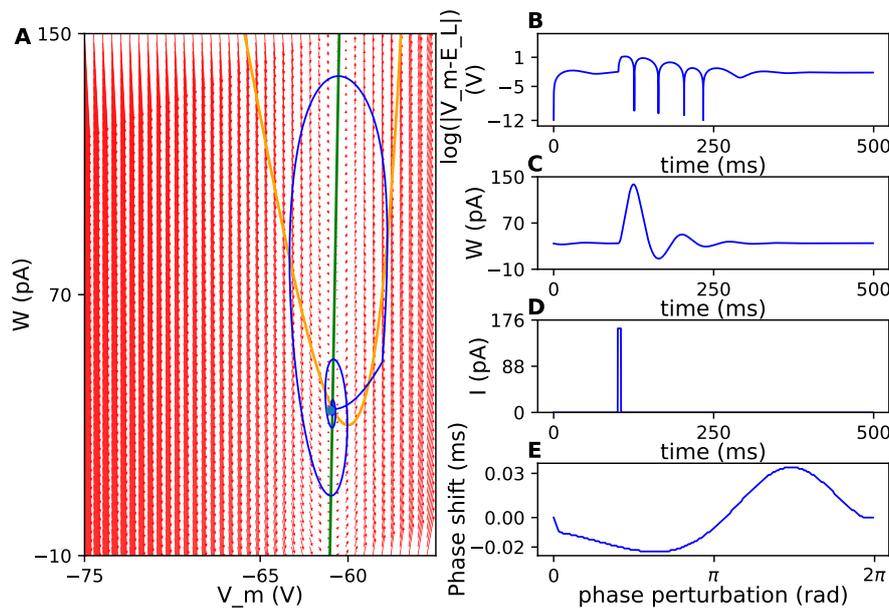


FIGURE 1.26. – Example of resonator neuron

The four classifications are based on different paradigms for classifying neuron dynamics. They overlap because, for example, the class 1 excitatory neuron is also a tonic spike, a one spike and an integrator. These classifications are based on different paradigms for identifying the neurons' behavior and fitting the neuronal model with it, which can be done with real experimentation. These different classifications are not exhaustive and restrictive to the possibility of dynamics of the adaptive exponential integrator and fire neurons [254, 345]. Moreover, the paradigms can be too simple, and other paradigms exist, such as a sequence of step current or noise [178, 336].

1.2.4 Point neuron models

Point neuron models contain a variety of models. One way to categorize them is to look at their link to biophysics principles. This categorization of point neuron models includes two categories: biophysical model, a model based on the biophysical principle, and phenomenological model, a model with parameters and/or variables which does not have direct biological meaning. Another way to categorize them is their complexity. The complexity can be quantified by different metrics such as the number of operation/flops [178], number of independent variables, number of parameters, or number of bifurcation point [54]. The difficulty of using a model is directly linked to the complexity of it. The more complex the model, the higher the probability of overfitting and the higher the degree of freedom, i.e. number of parameters unconstrained by the observations [183]. The choice of a neurons model in research is a difficult question because scientists must select a simple and realistic model simultaneously. One luck is that most often the less biophysical a model is, the less complex it is. Consequently, a scientist should choose a balanced model between

the required details and the available abstraction [157]. In this thesis, we decided to focus based on the research on the adaptive exponential integrate and fire neurons (Ad Ex) [37], which is a model of neurons focusing on the spike generation with a total abstraction of the spatial structure of the neurons and dendrites interactions. This model can reproduce the dynamic of neurons [254, 345] and reproduce the spike times of the neurons [193]. It is less abstract than rate or stochastic models, but it is less realistic than compartmental models or the Hodgkin and Huxley model. Before to present in details this model, I introduce a phenomenological model, FitzHugh-Nagumo [120, 253] neurons and the elementary model of Ad Ex, leaky integrator and fire neurons [221].

1.2.4.1 FitzHugh Nagumo neuron

From the dynamical analysis, FitzHugh and Nagumo propose a simplified dynamic of the model of the Hodgkin and Huxley model where the nullcline is similar [120, 253] with a recovery nullcline linear to the voltage.

$$\begin{aligned} \frac{dV}{dt} &= V(V - a)(1 - V) - w \\ \frac{dw}{dt} &= \epsilon(V - \gamma w) \end{aligned} \tag{1.5}$$

where $0 < a < 1$, $\epsilon > 0$ and $\gamma \geq 0$. As a model is a phenomenological model, the parameters a , ϵ , and γ do not have any biological interpretation. It is the same for the variable w . This model has been intensively analyzed [296].

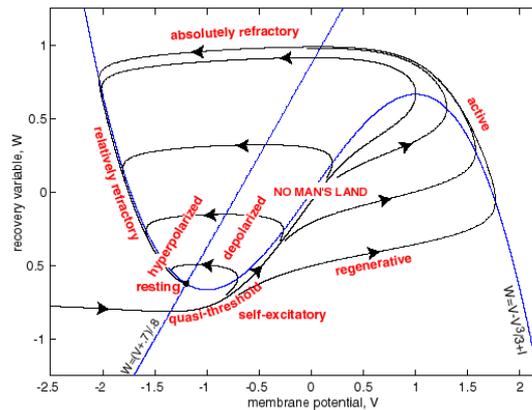


FIGURE 1.27. – Phase portrait and physiological state diagram of FitzHugh-Nagumo model

Figure from [179], modified from an image of FitzHugh 1961 [120]

1.2.4.2 Leaky integrator and fire neuron

Leaky integrator and fire [221] is based on the assumption that the never is similar to the charge of a capacitor in the presence of resistors. This model cannot be derived from Hodgkin and Huxley model [44, 349, 58]. Still, it can be considered as an approximation of it by considering only the passive properties of the membrane, leakage channel, and a reset mechanism for the action potential. The rest mechanism is that after passing a voltage threshold, the membrane voltage equals the voltage reset during a refractory time for taking account of the spike and the hyper-polarization of the membrane at the end of the action potential. This model does not consider the different phases of action potentials.

$$c_m \frac{dV_m}{dt} = -g_L(V - E_L) \quad (1.6)$$

$$\text{reset condition : } \forall t^f, V \geq V_{th}, V([t^f; t^f + t_{ref}]) = V_{reset}$$

where c_m is the capacity of the neurons, g_L is the conductance of the leakage channels, E_L is the resting potential, V_{th} is the voltage threshold, V_{reset} is the voltage reset, t_{ref} is the refractory time. This model has been intensively used in spiking neural networks because it presents the advantage of requiring low computation [178], one state variable and capturing 70 % of spike with a precision of 10 milliseconds [338]. However, this model has limited dynamic [178], and several extensions have been proposed, such as adding adaptation currents (adaptive integrate and fire neuron [357] or general integrate and fire neuron [243]), improving the shape of the spike (quadratic [110], or exponential [123]) or adapt the voltage threshold (multi-timescale adaptive threshold model [200])

1.2.5 Adaptive and exponential integrator and fire neuron (Ad Ex)

The main point neuron model of this thesis is the adaptive and exponential integrator and fire neurons (Ad Ex) [37], partially bio-physiological and partially phenomenological model. This model contains two extensions of the leaky integrator and fire neuron. The first extension is the usage of the exponential for faithfully processing fast input signals [123], improving the approximation of the subthreshold dynamic [58]. The second extension is the addition of an intrinsic slow current to capture the long-term effect of an action potential, which is essential for burst phenomena [58].

1. Modelling in neuroscience – 1.2. Neuron model

$$C_m \times \frac{dV_m}{dt} = -g_L(V_m - E_L) + g_L \Delta_T e^{\frac{V_m - V_{th}}{\Delta_T}} - w$$

$$\tau_w \times \frac{dw}{dt} = a \times (V_m - E_L) - w \quad (1.7)$$

$$\text{reset condition : } \forall t^f, V \geq V_{peak}, V_m([t^{(f)}; t^{(f)} + t_{ref}]) = V_{reset}$$

$$w(t^{(f)}) = w(t^{(f)}) + b$$

Similar to leaky integrator and fire neurons, c_m is the capacity of the neurons, g_L is the conductance of the leakage channels, E_L is the resting potential, V_{th} is the voltage threshold, V_{reset} is the voltage reset, t_{ref} is the refractory time. Additionally, the parameter of the exponential function : Δ_T is the slope factor of the voltage increase after the voltage threshold, and V_{peak} is the new threshold for the rest mechanism. Furthermore, the parameters of the slow current : a is the subthreshold adaptation, b is the spike-triggered adaptation, and τ_w is the adaptation time constant. These two extensions are phenomenology approaches because there are not derived from the biophysical principles, but their parameters can be extracted from electrophysiological experiments [37, 193].

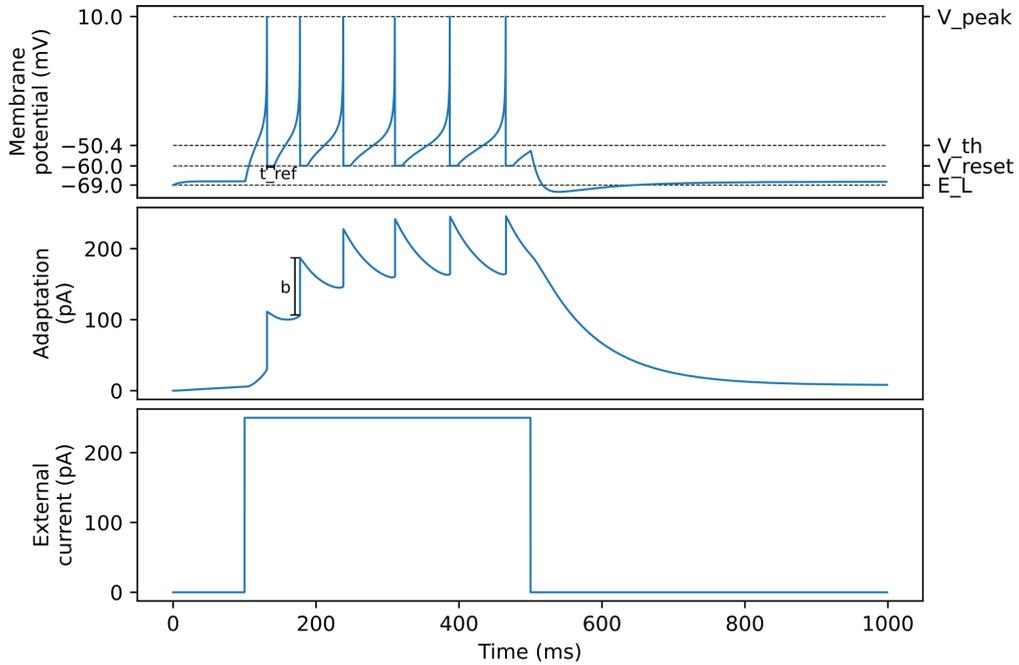


FIGURE 1.28. – adaptive and exponential integrator and fire neuron
Interpretation of some parameters from the characterization by a step currents

The Ad Ex's dynamic has been analyzed [254, 345, 220], but the model cannot reproduce all patterns of activities, especially subthreshold activities of the voltage membranes [178, 143, 348, 338].

1.3. Network of point neurons

Once the model of neurons is chosen, the next step is to define the connection between neurons. There are at least two types of intercellular communication. The first type is the cytoplasmic continuities based on an aggregate of intercellular channels, gap junctions [142], for transferring small molecules or ions between cells. The second type of communication is through chemical signals. The neurons are sensitive to multiple chemical signals produced by the brain or other organs [194]. The signals between neurons in the central neural system are classified into neurotransmitters and neuromodulators. The neuromodulators are a signal which affects locally multiple neurons mainly because there are not reabsorbed or broken down into a metabolite directly [256]. Their main function is the modulation of neuronal activities. On the contrary, neurotransmitters transfer a signal from one cell to another. In this thesis, we will consider only the neurotransmitters in synaptic cliffs of axodendritic, i.e. a space between an axon and a dendrite, and we assume that their effects are only local. In reality, the neurotransmitter interferes with the closed synapses [24].

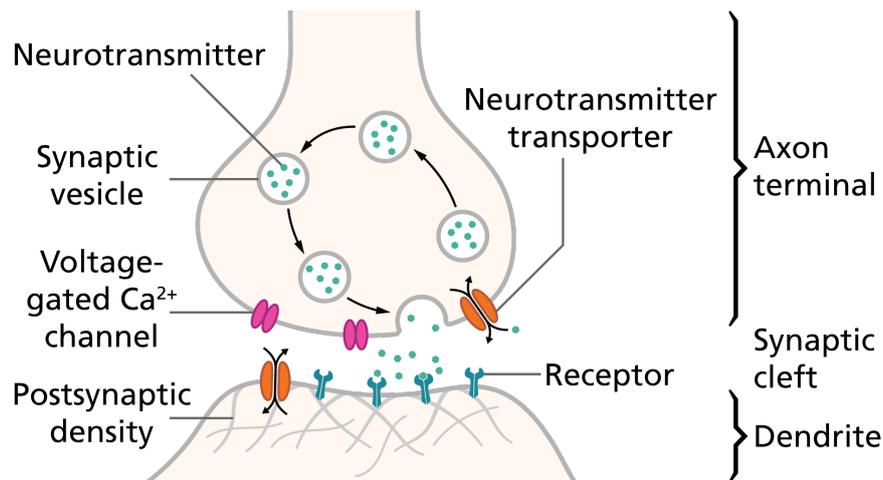


FIGURE 1.29. – Structure of a typical chemical synapse

Figure from [257].

The life cycle of neurotransmitters [70] in an axodendritic connection starts with a release of neurotransmitters by a synaptic vesicle when an action potential arrives at the pre-synaptic side, the axon of a neuron. The neurotransmitter diffuses in the synaptic cliff and activates on the post-synaptic side some receptors, which change the concentration ionic concentration, i.e. the membrane voltage of the dendrite. After the neurotransmitters are uptake by the pre-synaptic part for reform, a new vesicle and the life of the neurotransmitters can restart. The effect of the neurotransmitters depends on the type and the concentration of neurotransmitters in the vesicle, the type, the density, and the configuration of receptors at the release time, and the concentration in the cliff of other molecules such as the ions. Consequently, the signal is transmitted with different intensities and different duration or not at all. If the

glia cells are included in the model, the signal influences and is influenced by the astrocytes [231]. One phenomenological way to capture these effects is called short-term plasticity and significantly impacts network dynamic [351, 80]. In this thesis, we assume that the signal is always transmitted with the same intensity and duration, i.e. without short-term plasticity. Additionally, we consider only two types of signal, excitatory signal, associated with the glutamate neurotransmitter which activates the AMPA and NMDA receptors, and inhibitory signal, associated with α -aminobutyric acid/GABA neurotransmitter which activates GABA $_{\alpha}$ and GABA $_{\beta}$ receptors [111, 194, 80]

1.3.1 Model of synapse

Based on these assumptions, there are still multiple models of synapses. The first consideration is to take the different properties of the post-synaptic membrane. Some models assume the effect of synaptic is independent of it and always has the same effect on the membrane voltage, such as current synapse ($I_{syn}(t) = w * s_{syn}(t)$). Other models consider only the voltage dependence of the effect and change the membrane voltage by taking at count it, such as conductance synapses ($I_{syn}(t) = g_{syn} * s_{syn}(t)(V(t) - V_{syn})$). Other models include extra information depending on the chemical reaction of the receptors, such as the concentration of magnesium ions for NMDA receptors [111], which imply the consideration of short-term plasticity.

The second consideration is the modeling of the effect of the release of neurotransmitters. The most common way to model this phenomenon is by using the percentage of activated and deactivated receptors using a normalized double exponential waveform [111]. The normalization of the function is because the intensity of the signal is modeled as a multiplicative factor, synaptic weight.

$$P_s(t) = d \left(\exp^{-\frac{t}{\tau_1}} - \exp^{-\frac{t}{\tau_2}} \right) H(t)$$

$$d = \frac{1}{\exp(-t_{max}/\tau_1) - \exp(-t_{max}/\tau_2)} \tag{1.8}$$

$$\tau_{max} = \frac{\log(\tau_1/\tau_2)}{1/\tau_2 - 1/\tau_1} \text{ (time of peak value)}$$

with τ_1 is the growing rate of the receptor activation, τ_2 is the decay rate of receptor deactivation and $H(t)$ is the Heaviside step function (1 if $t > 0$ else 0). The value of τ_1 is influenced by the probability of activating receptors by a neurotransmitter after its release in the synaptic cliff. τ_2 is influenced by the diffusion out of the cliff, enzyme-mediated degradation, and uptake mechanism [80].

1. Modelling in neuroscience – 1.3. Network of point neurons

If we assume that τ_1 equals τ_2 , the function become an alpha function [111] :

$$P_s(t) = \frac{t}{\tau_2} \left(\exp^{-\frac{t}{\tau_2}} \right) H(t) \quad (1.9)$$

This formulation of the synapse was introduced by Rall et al. in 1967 [290], but often people do not use it because the interpretation of τ_2 is not clear, and it complicates the mathematical analysis of neural network [93].

If we assume that the rising time is very fast, i.e. $\tau_1 \rightarrow \infty$, the function becomes an exponential function [111] :

$$P_s(t) = \exp^{-\frac{t}{\tau_2}} H(t) \quad (1.10)$$

This formulation does not conserve the linearity of the synaptic input because it creates a discontinuity at zeros. However, it is more suitable for neural network analysis [93].

If we assume that the rising and decay time is very fast, i.e. assuming the synapse are instantaneous, the function becomes a delta function [111] :

$$P_s(t) = \delta(t)H(t) \quad (1.11)$$

This model is the simplest, and it is interesting because it reduces the computation resources of a simulation and for studies where the post-synaptic currents are negligible.

In this thesis, the synaptic models are exponential and alpha functions.

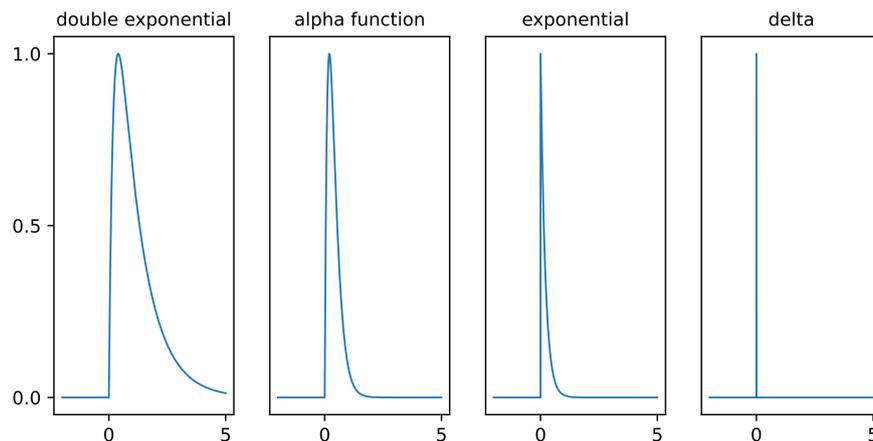


FIGURE 1.30. – Model of synaptic current after activation

1.3.2 Network connectivity

Creating a neuronal network requires a model of neurons, a model of synapses, as well as the pairs of connected neurons, the weights of these connections, and the delays in transferring signals.

Multiple methods exist for connecting neurons, but their main difference is the integration of spatial information. Mathematical studies provide three types of connectivity : fully connected [116], sparsely connected [16] or spatially connected [19].

For fully connected networks, all neurons are connected to all neurons.

For sparsely connected networks, the connectivity can be defined with a pairwise probability connection based on probability distribution such as Bernoulli distribution [181] or Poisson distribution [280, 245], or by defining the number of incoming synapses, outgoing synapses or the total numbers of a neuron or population. Additionally of the connectivity rules, some constraints [314] can be added, such as the presence of autapse synapses [173, 286], i.e. the possibility of a neuron to connect to itself, or the presence of multapse synapse, i.e. the possibility to connect multiple time to the same neurons.

Spatial connectivity requires a spatial distribution of the neurons. In mathematical studies, the spatial distribution of neurons is based on a grid along a line or plane. In these studies, the boundaries can affect the dynamic network because the neurons close to it are less connected. For toy models, the boundary condition is often assumed to be periodic, i.e. the boundaries are connected to form a continuous space. This assumption can be considered line and plan infinite, respectively, equivalent to a ring and a torus. The spatial connectivity, in these cases, is defined using a kernel. The kernel of probability is a function that defines the probability of connection depending on the spatial distance between two neurons. Another method of the spatial distribution of neurons is to group them regarding the position of their soma and use morphology distribution to define connections between populations [281, 305, 306]. For example, Potjans et al. 2014 [281] group neurons into six layers and define connectivity based on a fixed number of connections between layers derived from an anatomical map. In the case where the three dimensions position of all neurons is defined, recent methods use the morphology of neurons to define pairwise connections, such as parametric anatomical modeling [288] or spherical probability clouds [131] for more precise connectivity. In this thesis, the network connectivity is sparse connectivity based on Bernoulli distribution or spatial connectivity based on a two-dimensional Gaussian kernel probability (see Chapter 4).

Connections in neuronal networks are weighted and delayed. From the assumption that neurons are equivalent to a point, the delay represents the time of spike goes from the soma of neuron A to the soma of neuron B, i.e. the time of a spike goes along the axon until the synapses, the synapse releases the neurotransmitters, neurotransmitter activates the post-synaptic receptors, and the generated current goes along the dendrites. However, in this thesis, we assume that the delay is instantaneous between neighborhood neurons because it simplifies the analysis and the comprehension of network dynamics.

The weight represents the intensity of the post-synaptic currents, i.e. the quantity and type of neurotransmitter release, the type and density of post-synaptic receptors, and the position of the synapse on the dendrite. All these properties can influence the effect of post-synaptic current [46], but in this thesis, we consider that their values are constants and unique for inhibitory and excitatory synapses.

1.3.3 Characterization of a neural network

One way to record neural networks is the introduction of electrodes in the brain to record the variation of extracellular potential. Two elements can be extracted from these recordings : action potentials (spikes in the signal) and local field potentials (general signal fluctuation). Assuming each neuron generates identical spikes with a unique shape, it is possible to classify them, determine the number of neurons, and define the spike time of each neuron (spike train). If there are multiple electrodes, it is possible to approximate neurons' location and improve the classification [45]. However, these measures do not provide direct access to the underlying network or structural network but only include information about the network dynamics or effective network. For analyzing network dynamics or, in particular, spike trains, multiple methods exist, such as a statistic description [89], characterization of dynamical relation [62], description of the different bands of oscillation and their interaction [45, 198, 230], or embedding in a low dimension space which is linked to behavior [36, 308]. In the case of this thesis, we focus on the statistical description of the network dynamics (mean firing rate and mean of regularity, see sub-section 1.2.3), particularly population synchronization.

The level of synchronization can be evaluated by different metrics [208, 207], but in this thesis, we used only two metrics.

The first metric is the coefficient of variation of mean firing rate [41].

$$CV_{FR} = \frac{\sigma_{FR}}{\mu_{IFR}} \quad (1.12)$$

For the interpretation, CV_{FR} higher than 1.0 means that the network is synchronized.

The second metric is the order-parameter R defined by Kuramoto 1984 [212] based on the pseudo phase of the neurons. The pseudo phase of the spiking neuron n is defined by considering that the phase between two spikes is constant [32] :

$$\psi_n(t) = 2\pi m + 2\pi \frac{t - t_{n,m}}{t_{n,m+1} - t_{n,m}} \quad (1.13)$$

where m is the m -th spike of the neuron n .

From this definition, the order-parameter R of a population is defined as :

$$R(t) = \left| \frac{\sum_{n=1}^N \exp(i\psi_n(t))}{N} \right| \quad (1.14)$$

The mean over the time of the order-parameter quantifies network synchronization and is included between 1.0 (synchronous) and 0.0 (asynchronous).

1.3.4 Network of adaptive exponential integrate and fire neurons

All neural networks in this thesis are a network adaptive exponential integrate and fire neurons. Most previous studies using this type of network focused on the impact of the effect of the adaptation because anterior studies were mostly based on leaky integrator model [44] (see sub-section 1.2.4.2).

Some studies use this network model to validate biological mechanisms. For example, Destexhe et al. 2009 [92] models the thalamocortical networks to investigate the irregular asynchronous state or the up and down state of a model of cortex and thalamus and their interaction. A second example is the model of epilepsy statement and investigation of the impact of the stimulation on epilepsy [285]. A third example is the reproduction of theta(4-8Hz)-nested gamma (>30Hz) rhythms with a sparse network and with different types of synapses [319].

Some other studies analyze the impact of the connectivity, adaptation current and response of an external input. For example, Lamdenbauer et al. 2012 characterize the impact of a (subthreshold adaptation), b (spike-triggered adaptation), synaptic delays and weights on synchronization of two neurons and populations of homogeneous neurons [220]. A second example, Lamdenbauer et al. 2013 characterize the impact of the synaptic weights and external input on the mean firing rate and the mean regularity of coupled excitatory and inhibitory populations [219]. A third example, Borges et al. 2017 look at the synchronization in coupled excitatory and inhibitory populations for b , V_{reset} , synaptic weights and probability of connection [32]. A fourth example, Shiau et al. 2019 analyze the frequency and synchronization for synaptic weights, percentage of connection, τ_w (adaptation time constant), the two constants of time of a double exponential waveform of synaptic, a , b and external input [319]. A fifth example, Protachevich et al. 2020 focused on the time delay of coupled inhibitory and excitatory populations on population synchronization, mean regularity and firing rate frequency [284]. A sixth example, Protachevich et al. 2020 focused on autaptic synapses, i.e. connection of neurons to itself, on population synchronization, mean regularity and mean firing rate of inhibitory population, excitatory population and mixed population [286]. A seventh example, Augustin et al. 2013 studied synapse weights, a , b , external firing rate, τ_w on population synchronize oscillation and resonance [18]. An eighth example, Wu et al. 2021 analyze the burst stabilization for different network topologies [368].

Other studies analyze the network to define an equivalent low-dimensional system of the network. This equivalent system is used to simplify the analysis of the network or for the simulation of a larger network in the future, such as the brain [266, 18, 17, 49, 47, 33, 373, 374, 356] (see the sub-section 1.4.1 about Neural Mass of Ad EX).

1.4. Mean-field model

Before detailing reduced systems, we must define the terms and approaches for reducing a neural network to a low-dimensional system for modeling brain regions. Most reduced models are based on the assumption that the exact statement of individual neurons does not matter and can be replaced by a statistical description of populations [84]. The percentage of active synapses or the distribution of the voltage membrane or firing rate is used as the value for representing the network state at a specific time. The common formulation for describing this reduced model are Fokker Plank equations which describe the evolution of probability density over time, and master equation, which describes the probabilistic combinations of states and their transitions. The states with their transitions or probability density with its evolution can be assumed by modelers based on their intuition [364], some part of the model can be estimated from measures [182] or can be derived from a network of point neurons [41].

The derivation from a network of point neurons is too complex to be solved actually. Consequently, all available mean-fields are restrictive to specific network configurations, network dynamics or neuron models. Additionally, most of them approximate the network behavior, such as the inconsiderate high frequencies, the assuming Gaussian distribution, or linearizing some part of the model. Consequently, many assumptions and approximations are used, creating difficulty in comparing and choosing a neural field model. The rest of the paragraph presents some assumptions or approximations for reduction (for more details, see the paper of Cook et al. 2022 [66]). One common difference in assumption is the consideration of the spatial position of neurons; Neural Field models take into account the spatial propagation of neural activities, whereas the Neural Mass models do not include a spatial dimension in their model. However, the spatial dimension can be included a posteriori, approximating a Neural Field with a Neural Mass network under certain conditions [326]. Another major difference is the consideration of the type of connectivity; for simplification, some mean-fields consider only fully connected networks, while others consider some probability of connections, or others consider specific types of networks. Another difference is the approximation or not by the separation of time scale, i.e. reducing the model to slow time scales which captures the main network dynamic. Another difference is that some models assume a stable steady state of the network, like an asynchronous irregular state, or oscillating, like a synchronous regular state. A common assumption is that the parameters of the point neurons are the same in a population for simplifying the reduction. Another common assumption is to consider that the neurons are independent and that the input spikes of the neurons follow a Poisson distribution.

One common modeling of brain activities is the brain network model [185]. In this type of model, the network which constrains brain dynamics is called "connectome" [328, 150] or structural connectivity, i.e. connections between neurons or brain regions. From this point of view, brain activities are seen as an emergent phenomenon of a

network of neurons or populations of neurons. Actually, the most precise connectome is the network of neurons with detailed interactions (molecular interaction between each neuron) of the *Caenorhabditis elegans* pharynx [67]. The precision is not available for bigger animals, and the description stays at the level of brain regions. For an alive human, the standard method for getting the connectome is Diffusion Tensor Imaging (DTI) [55], which estimates the number of fibers by measuring the direction of the diffusion of water in the brain. The precision can be increased by including data from analysis of postmortem human brains, such as the direction of the fibers [369]. For the mouse brain, it is possible to extract a connectome from postmortem data of fluorescent axonal projections of each brain region [263]; this connectome is more informative than the individualized connectome computed from DTI [239]. Despite the technical limitation, the choice of scale for the connectome is still unknown. For example, the precise connectome of *Caenorhabditis elegans* pharynx does not comprehensively understand the behaviors [121]. One explanation is that the connectivity between neurons is insufficient because it lacks the plasticity of connections and the modulation of neurons' behavior. Even with these limitations, this model is used for addressing causality in brain activities. In this thesis, the brain network model is a network of Neural Mass derived from a point neuron network that describes the activities of each brain region. Neural Mass models can capture some parts of brain region dynamic, but it is challenging to evaluate their approximation. Despite the diverse Neural Mass derived from a point neuron network, few studies analyze the validity of assumptions, quantify the impact of approximations, compare the dynamic behavior, or quantify the state difference between a network of Neural Mass and the associated network of neural networks, which limits the interpretability of Neural Mass.

1.4.1 Neural Mass AD EX

As expected, multiple methods have been used for reducing a network of the Ad EX neurons. Augustin et al. 2017 [17] propose five methods based on different approximations only for fully connected networks with delayed delta synapses using Fokker Plank equations formalism. One method is the Fokker Plank equations without approximation, the most precise and complex method, but it is unsuitable for analysis and simulations. Two additional methods approximate the network activities based on the spectral decomposition of the Fokker Plank operator to the first and second order. The two last methods approximate the network activities based on the Linear-Nonlinear Cascade approximation. The Linear-Nonlinear Cascade model [266] is composed of a linear temporal filter, i.e. a function that defines the membrane voltage distribution from the estimated distribution of firing rate, and a nonlinear function, i.e. a function that defines the distribution of firing rate from the membrane voltage distribution. This study approximates the linear temporal filter by an exponential or a damped oscillator. Additionally, the section "Alternative derived models" of this paper reviews the possible extension, approximation or modification of the reduction methods for

the network of adaptive exponential integrate and fire neurons.

In this thesis, the reduction of the network used is the Mean Ad EX described by Di Volo et al. 2019 [356]. This model assumes that the network state can be reduced to its firing rate distribution and assumes that it follows a Markov process [33], i.e. the system is memoryless, only dependent on the previous state. Additionally, the model assumes that the network dynamic is an irregular asynchronous state [33, 374], i.e. the network is quasi-stationary and the neurons are not correlated. Using the Master equation formalism and these assumptions, the transition function is defined by the transfer function, i.e. a function that defines the rate of a neuron from a defined receiving excitatory and inhibitory mean firing rate.

These mean-fields were compared to data to explain some brain dynamics. The Linear-Nonlinear cascade with the exponential temporal filter was used for the model electrical stimulation of a neural population [49] and for the model whole brain with slow-wave sleep activities [47]. The Mean Ad EX was used to model the measure of consciousness based on stimulation [140] or the interaction of traveling waves in the visual cortex [374, 56].

CHAPTER 2

Cluster programming and research infrastructure

Contents of Chapter 2

2.1. Parallel and Distributed Computing	60
2.1.1. A computer	60
2.1.2. A cluster	61
2.1.3. parallelization and distributed computing	64
2.2. Human Brain Project and EBRAINS	66
2.2.1. Humain Brain Project	66
2.2.2. EBRAINS	68

2.1. Parallel and Distributed Computing

Large simulations require parallelization and/or distribution of computation to get results in a reasonable time, up to a few days or weeks. Parallel computing means performing calculations in parallel on a single computer, i.e., simultaneously realizing multiple instructions. Distributed computing is the distribution of instructions across multiple computers. However, all programs cannot be parallelized because the parallelization requires at least some independence in the instructions, i.e. some instructions do not require results of previous instructions.

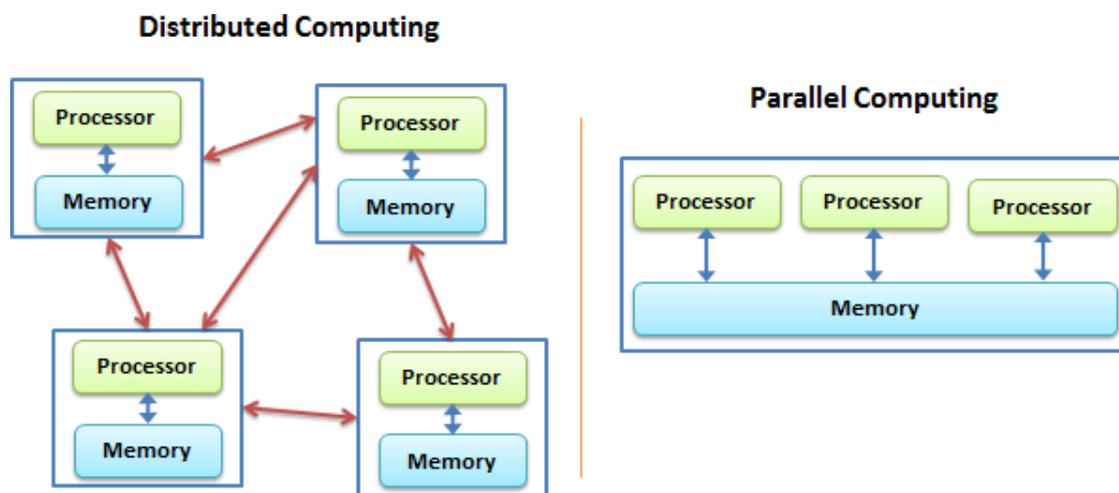


FIGURE 2.1. – Distributed vs. Parallel Computing

Figure 1 from Rehman et al. 2019 [292].

2.1.1 A computer

Understanding the efficiency and limitations of parallelization and distributed computing requires knowledge about the different components of a computer and a cluster [72]. A basic personal computer includes a screen, a tower, a mouse and a keyboard. The screen is used for showing images generated by the computer. The mouse and the keyboard are the components to interact with the computer. The tower, the center component of a computer, connects all the parts and supplies some of them with electricity. The tower is generally composed of a mother card (an electronic card that connects all components), a power supply unit, two types of memory (a stable memory, called ROM : Read-Only Memory, and a volatile memory called : RAM : Random Access Memory), two types of processors (CPU : Central Processing Unit and GPU : Graphic Processing Unit) and network connection components (wifi card, Bluetooth card, ethernet port and Universal Serial Bus (USB) ports). Computation, the core function of a computer, is carried out by CPU cores. Each core has a small memory for the computation, called cached, and a second memory to support multi-

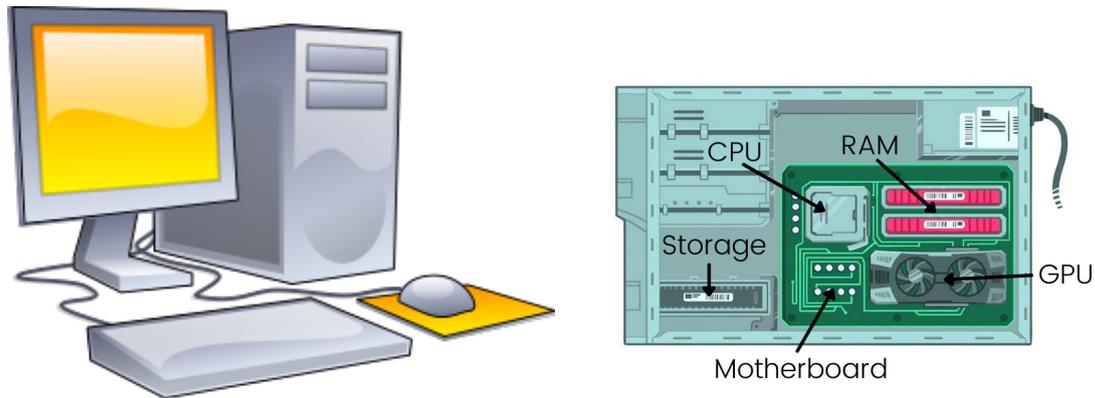


FIGURE 2.2. – Computer parts and components

The left sketch illustrates the parts of a personal computer [91], and the right sketch represents the different components of the tower [1].

threading. This second memory gives the possibility to compute other instructions when an instruction is waiting, for example, for getting data from the ROM.

The users can have two types of interface with a computer : a graphical interface or/and a text interface. The graphical interface requires a mouse to interact with the computer. The text interface is based on a command line interpreter, an engine that interprets texts to command the computer. All the commands are treated by an operative system (OS) [323], a software in charge of managing the hardware and the software resources and providing services to other programs (for example, creating a process or allocation of memory). Several implementations of OS exist, but the two main families' implementations are Windows and Unix (Linux and macOS). The thesis was carried out using Linux systems which uses an interpreter of bash language, called bash shell.

2.1.2 A cluster

A cluster is a network of nodes, i.e., towers without a screen, a keyboard and a mouse, and is designed to be used by multiple people simultaneously. Most clusters have a specific operative system, but most are based on a Linux system [27]. A cluster includes at least two types of nodes : logging and computation nodes. The logging node handles users' connections and manages the cluster's resources. Most of the connection interface is based on a text interface using ssh protocol (Secure Shell Protocol) [227], i.e. a distant shell with a secure connection. The graphic interface is possible but less practical because of the latency from the connection. A resource manager, such as SLURM(Simple Linux for Resource Management) [372], manages the other nodes to compute instructions.

The resource manager has a program for running on all the nodes to control them (slurmmd) and a program, the controller, on the logging node to interface with the users (slurmctdl). One task of the controller is to distribute and schedule the re-

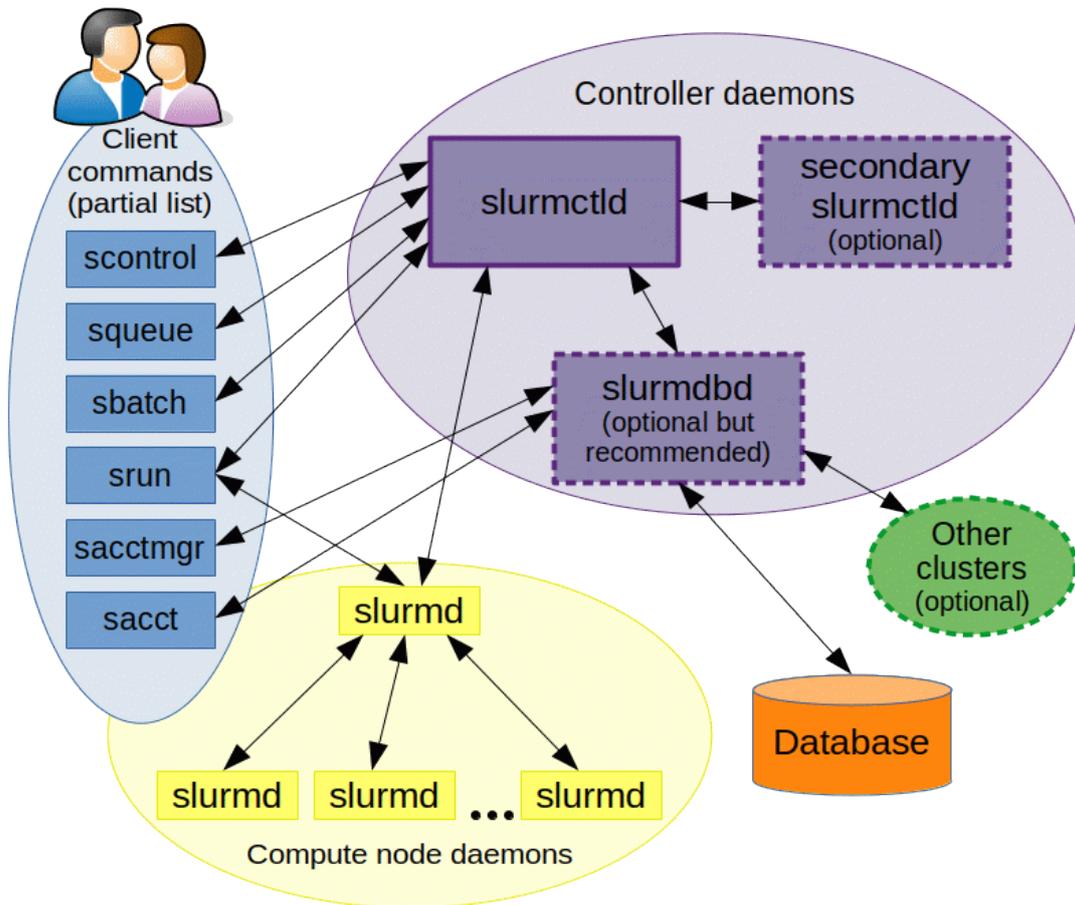


FIGURE 2.3. – Slurm components

Figure from Slurm documentation [325].

sources/nodes of the cluster depending on the users' requests and the administrator's constraints. The requests of computational resources, called jobs, contains some requirements such as constraints of memory, time of execution, and the number of nodes [325, 372] and, in the case of non-interactive jobs, a batch script (file with command lines). The resource manager handles only the computational resources. The memory resources are also shared but managed by the OS.

This common interface does not reflect the diversity of architecture and composition of clusters. Four clusters are used in this thesis and can show the cluster diversity. Piz-daint [277] is a ten years old Swiss cluster; the last update of the cluster is composed of 2 sub-networks (nodes with or without GPU) based on a Cray system. Cray system is a solution provided and supported by a private company, CRAY Incorporated, a subsidiary of Hewlett Packard Enterprise. This solution provides specific architecture and software to optimize distributed computing; for example, the scheduler is handled by the Application Level Placement Scheduler(ALPS) [195] with the same interface as SLURM.

2. Cluster programming and research infrastructure – 2.1. Parallel and Distributed Computing

DEEPEST [86] is a German prototype cluster based on three modules to evaluate the performance of the usage of heterogeneous nodes to accelerate software performances. The three modules are cluster module (node without GPU), booster module (node with GPU) and data analytics module (node with GPU and Field Programmable Gate Arrays).

JUSUF [65] is a German cluster of nodes with a large amount of RAM (256 GB) and CPUs (128 cores). One particularity of this cluster is that the computational resources are requested per CPU instead of per node to avoid the waste of resources by forcing the usage of all the CPUs.

In addition, some clusters can contain heterogeneous nodes (different numbers of CPU and memory sizes), such as the INS cluster.

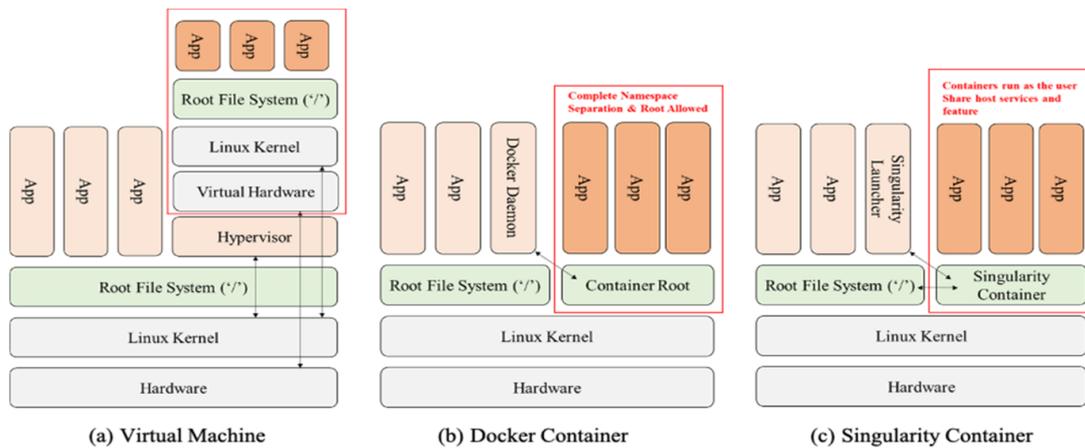


FIGURE 2.4. – Architecture comparison of the Virtual Machine (a), Docker container (b), and Singularity (c)

Figure 1 from Li et al. 2021 [224].

Before executing any program, i.e. list of instructions, on a cluster, it is necessary to port your application, i.e. engine for executing specific instructions, onto a cluster by installing or using containers. Installing an application requires identifying the existing and missing libraries for your software, identifying the compiler associated with the specific sub-network, setting up the environment, and configuring the installer of libraries. The optimal installation is a challenge because the dependence between libraries creates limitations that may even make the installation impossible and requires knowledge about the possible configurations and available optimization (for example see the different [scripts](#) of installation of co-simulation).

The second way to bring an application to a cluster is by using containers, a portable environment containing installed software and libraries. The container is instantiated from an image in a standardized virtual environment, at the difference of virtual machines that virtualize the hardware and require a second OS. Multiple standards exist for creating an image; in this thesis, the container is based on Singularity [213] and Docker [359]. The main difference between Singularity and Docker is that the

first container provides access to the host's services. For example, docker containers can not use the ssh server of the host and need their own server, which is not the case for Singularity. However, access to some services is necessary to communicate between the nodes and is essential for distributed computation. Consequently, some container runtime, i.e. the creator of the virtual environment, for docker image is specific for cluster exist such as Shifter [133] or Sarus [26]. One remark is that the usage of a container can slow the application because it requires some virtualization of the environment and does not provide all the available optimization. Still, it gives better application repeatability and facilitates software deployment on a new cluster or computer.

2.1.3 parallelization and distributed computing

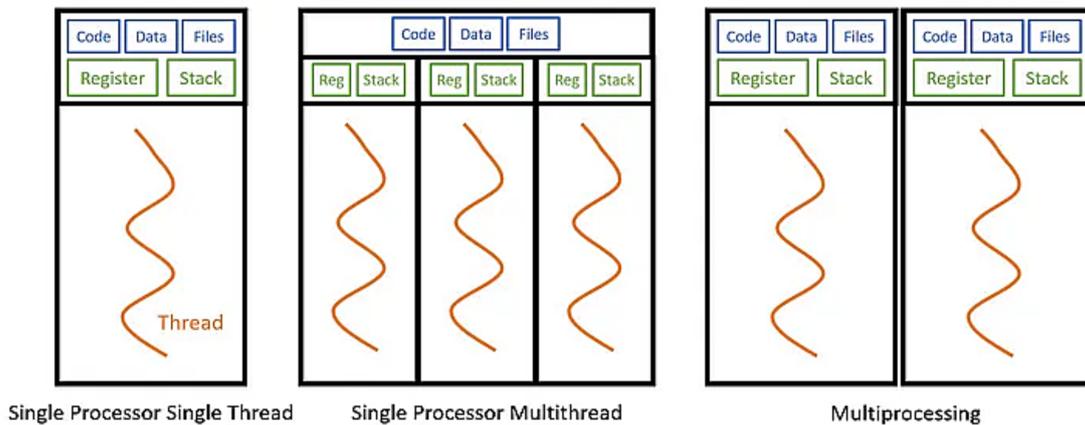


FIGURE 2.5. – Multithreading vs. Multiprocessing

Figure from [367].

This simplified overview of a computer and a cluster provide the basic concepts to understand the possibilities and the limitations of parallelization and distributed computing [72]. Parallel computing is the concept of realizing multiple instructions simultaneously to get the expected results. A program, i.e. a list of instructions, can be parallelized if some instructions are independent of the results of previous instructions. For instance, in the case of parameters exploration based on a grid, each exploration is independent and easily parallelized; it is an embarrassingly parallel workload. However, the potential for program parallelization does not mean that these programs have to be implemented in parallel due to hardware limitations. If multiple instructions are based on the same memory, at this moment, the optimal implementation use multi-threading. The multi-threading is based on a thread, i.e. a list of instructions that share the same memory, to execute a program and the capacity of the CPU to switch context. Without going into details, this tool requires coordination of reading and writing memory because the execution order of instruction is not guaranteed. Suppose the multiple parallel instructions need different parts of the memory; in

2. Cluster programming and research infrastructure – 2.1. Parallel and Distributed Computing

these cases, the optimal implementation uses multi-processing. The multi-processing is based on the usage of several simultaneous CPUs. Without going into details, this tool requires managing shared resources (memory and computation) and communications between processes, i.e. independent list of instructions, for the transfer of data and synchronization. However, this solution is slower than multi-threading because it requires data transfer between the CPUs. The parallelization will not be sufficient for considered memory (impossible to load in the RAM) because these programs will be slow due to the necessary access to ROM, the slowest accessible memory. One optimization for these programs is the distribution of computation over multiple computers. Distributed computing requires the coordination of computers and to distribute the instructions with the consideration that the connections between computers are slow. Nevertheless, multi-threading, multi-processing and distribute programming are not exclusive and can be combined for the most efficient implementation. As you can see, the optimal performance of a program is dependent on the skills of the developer to use the best strategy for paralleling the different parts of programs with the different possibilities on specific hardware with the optimized libraries.

For parameter exploration, the simpler way is to launch the same program with different parameters in parallel using a scheduler, for example, Snakemake [204], to avoid overloading a computer. The overload of a computer is when the required computational resources are higher than the available ones. In this context, the OS will run all the explorations simultaneously and distribute the calculation time between them. The consequence is a slow-down computation because each CPU must change its cached at every instruction. By default, on a cluster, the resource manager provides an interface to avoid this overload of the nodes.

For the simulation of a large homogeneous sparse network of point neurons with synapses, it recommends combining all the parallelization tools. In this case, the network cannot be handled by one computer or node and requires the simulation's distribution over multiple computers. Additionally, the neurons communicate with spikes in the sparse network, meaning that each neuron does not need to know the state of all neurons at every time. Consequently, the neurons can be computed independently in parallel with some synchronization to communicate spikes. Due to the homogeneity of the neurons, the instructions to compute the evolution of the neuron's state differ only in the values of the state itself. Consequently, the memory for the instructions differs only by the neurons' state. Due to this, the implementation can use multi-threading to exploit CPUs' capacity to execute an instruction when another is waiting, multi-processing to use multiple CPUs, and distributed programming to avoid overloading computers.

There are multiple methods for managing memory, synchronization, and communication between thread, process and computer [72]. However, some standard libraries propose elementary functions for implementing these methods. The main standard library in C++ for parallelization using shared memory, i.e. mainly for multi-threading, is Open Multi-Processing (OpenMP) [74]. The high-level standard for communication between processes on one or multiple computers, i.e. multi-processing and distributed computation, is Message Passing Interface (MPI) [342]. This latter standard has

2. Cluster programming and research infrastructure – 2.2. Human Brain Project and EBRAINS

two families of implementations : MPICH [251] (MPIVACH2 [252], Cray MPT, Intel MPI Library, Parastation MPI [250]) and OpenMPI [129, 145] (merge of FT-MPI [144], LA-MPI [112], LAM/MPI [330], and PACX-MPI [196]).

These two implementations also include a standard for managing and configuring processes, specifically, automating MPI configuration based on the SLURM requirement. This standard interface is called Process Management Interface [22, 53].

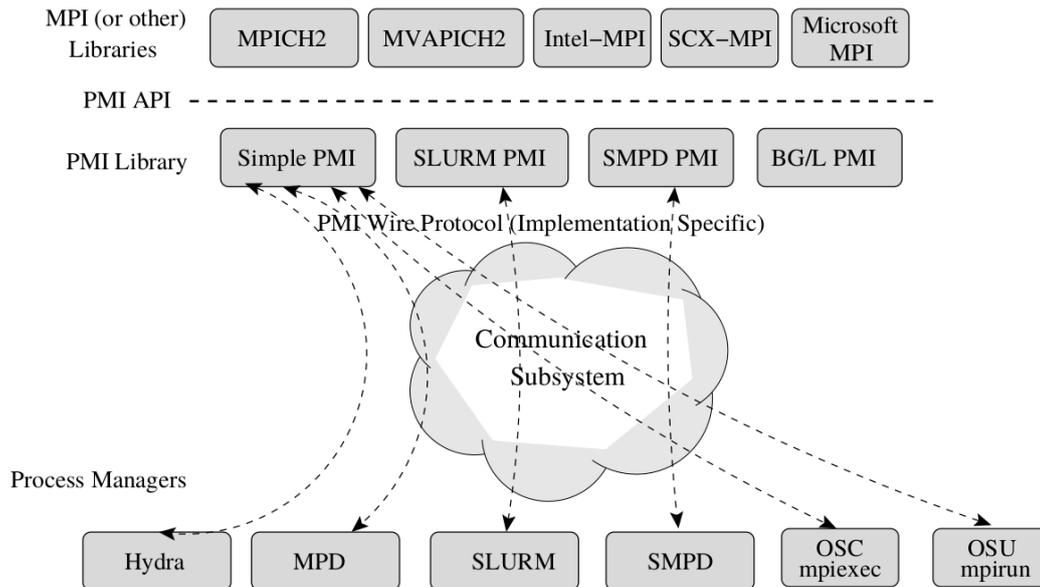


FIGURE 2.6. – Interaction of MPI and the process manager through PMI
Figure 1 from Balaji et al. 2010 [22].

2.2. Human Brain Project and EBRAINS

This thesis was done in the context of the two last stages of the Human Brain Project.

2.2.1 Humain Brain Project

The Human Brain Project (HBP) [168, 235, 234, 12, 13, 11] has granted as European Future and Emerging Technologies Flagship (FET) by the European Commission in 2011. A FET is a large-scale, long-term research initiative funded over ten years to 100 million euros annually. HBP started in 2013 and was composed of four phases : a Ramp UP phase (30 months) [341], a Specific Grant Agreement 1 (SGA1, 30 months) [169], a Specific Grant Agreement 2 (SAG2, 30 months) [170] and a Specific Grant Agreement 3 (SAG3, 30 month with an extension of 6 month due to the COVID) [171].

The project was initially scientifically led by Henry Markram [235, 234] with a vision of reverse engineering the human brain based on two main trends : developing

2. Cluster programming and research infrastructure – 2.2. Human Brain Project and EBRAINS

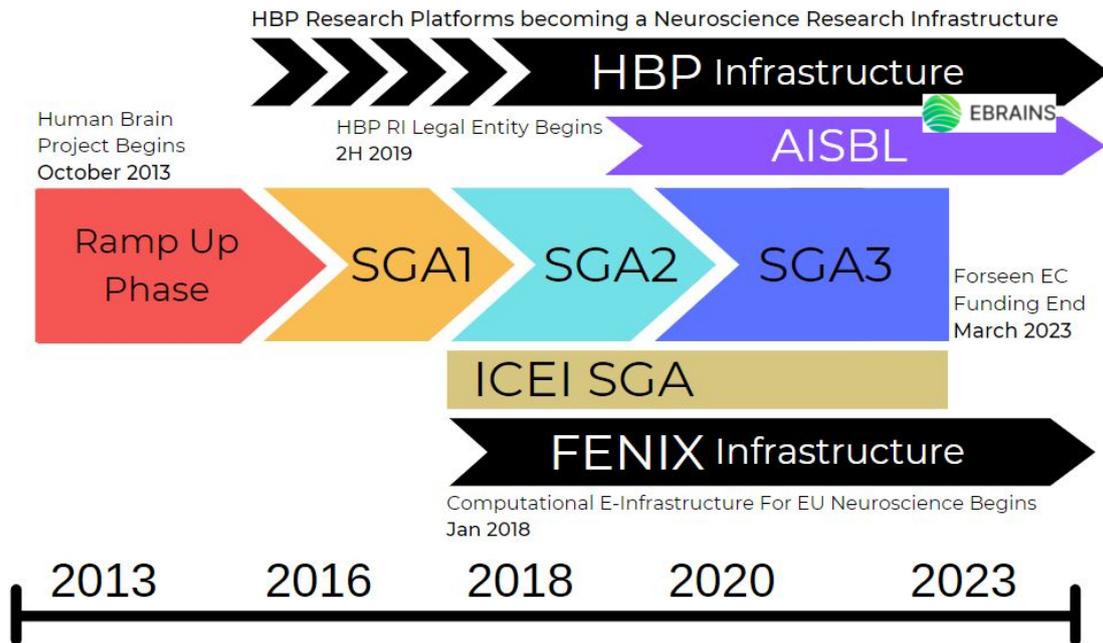


FIGURE 2.7. – Human Brain Project Timeline

Figure adapted from [168].

modern supercomputers and integrating multimodal data in models. This vision was based on the first results of the Blue Brain Project [233, 295], started in 2005 and led by Dr. Markram himself. At this time, the Blue Brain Project was developing a network of neurons modeled at the level of ion channels for a neocortical column of the rat brain using Blue Gene/L, a supercomputer design for simulating a large number of neurons. HBP is structured around six Information and Communication Technology (ICT) platforms : Neuroinformatics (management and interaction of neuroscience data based on FAIR principles (Findable, Accessible, Interoperability and Reusability) [363]), Brain Simulation (high-fidelity digital reconstruction and simulation of the brain), High-Performance Computing (development of supercomputer dedicated to brain simulation), Medical Informatics (connecting patients data from hospitals to help understand or cure illness), Neuromorphic Computing (designing hardware inspired from brain biology to increase the speed of simulations) and Neurorobotics (integration of brain model into robots). Additional to these six ICTs, HBP supported four scientific projects : Mouse brain organization (construct models of mouse brain from molecules to behavior), Human brain Organisation (characterize the functional and structural specificity of the human brain), Systems and Cognitive Neuroscience (focus on behavioral-cognitive processes and brain states), and Theoretical Neuroscience (bridge scales from cells to whole brain) [12]. Furthermore, this project dedicated efforts to reflect on ethics and societal issues relevant to the project. And it provides educational materials and training for young researchers.

During SGA1, the project was criticized for missing scientific perspectives like cognitive science or the lack of experimental neuroscience, but also for the opacity of its

governance [126]. The project took a new direction in SGA2 under the lead of Katrin Amunts in 2018. The new goal was to provide collaborative tools for decoding the brain based on multiscale reconstruction building upon scientific results from the two first phases [12], and this Ph.D. thesis started around that time. Under the pressure of the European Commission, SGA3 was more dedicated to the interoperability of the tools developed before and the demonstration of scientific contribution. HBP will end September 2023 with the finalization of EBRAINS [10], a digital research infrastructure that contains the principal datasets and tools developed during the project lifetime. Besides this platform, HBP created a European community of researchers dedicated to maintaining EBRAINS [99, 310] and to keep collaborating along a common vision [10, 310]. The support of EBRAINS is based on a network of European super-computing centers, FENIX [5, 4, 6], built during HBP.

2.2.2 EBRAINS

EBRAINS is a digital research infrastructure composed of neuroscience datasets from different animals and measured at different scales with associated tools for sharing, comparing, analyzing, modeling and visualizing this data. The following paragraph provides a presentation of the tools of EBRAINS connected to this thesis to illustrate their variety.

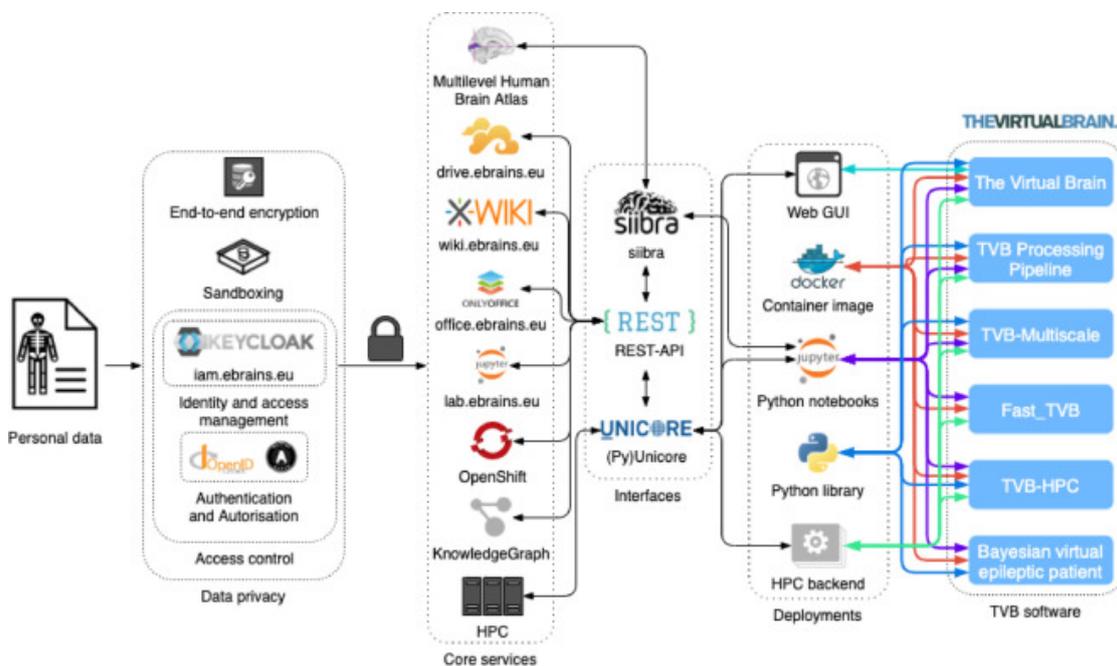


FIGURE 2.8. – TVB on EBRAINS cloud services

Figure 1 of Schirner et al. 2022 [303]

EBRAINS supports three neurons simulators : NEURON [159], Arbor [3] and NEST [98]. NEURON is a simulator, created in 1983 [159, 52], of neurons based on cable

2. Cluster programming and research infrastructure – 2.2. Human Brain Project and EBRAINS

or compartmental model written in C and C++ with a graphical user interface (GUI). The simulator has multiple extensions (for example, a python interface [160]) and optimization (for example, parallelization [242, 210]). Arbor [3] was developed by HBP to run efficient simulations of compartmental neurons on clusters and computers. NEST [152], created in 1995 [97], focuses on optimizing simulations of point neurons. This simulator is an interpreter of a simulation language interface (SLI) [97] written in C++ for improving the parallelization of simulations. This simulator has a python interface [107] and a GUI (NEST desktop) [329] dedicated to education. EBRAINS also supports a brain simulator, The Virtual Brain (TVB) [299, 300]. TVB is written in Python and has a GUI. Some derived versions propose an optimization like The Fast TVB [303] or TVB rateML [354, 303].

For the integration of different scales and the interoperability of simulators, EBRAINS offers standard interfaces to configure neuron simulators, such as PynNN [79], Brain Scaffold Builder (BSB) [81] and ConGen [156]. PyNN is a common and standard interface to create point neural networks for simulators such as NEURON [159], NEST [98] and neuromorphic platforms (SpiNNaker [267, 128, 226] and BrainScales [302, 269]). BSB [81] is a framework for managing simulations with NEST and NEURON. This framework implements functions to parameterize neurons, define connectivity, launch simulations on a specific simulator, and create visual simulation outputs. ConGen [156] offers a visual language to configure simulators and visualize networks of multiple scales.

EBRAINS also supports tools for visualization, such as ViSimpl [130] and NEST desktop [329]. ViSimpl allows multiple visualizations of neural networks, such as the morphology and placement of neurons, spiking neural network activity and connectivity, all simultaneously.

EBRAINS also supports analysis and comparison between simulation and experiments, such as LFPy [225, 149], HybridLFPy [148], Electrophysiology Analysis Toolkit (Elephant) [90]. LFPy [225, 149] is a tool for simulating extracellular potential (LFP, EEG, ECOG, MEG) generated by multicompartment neurons simulated by NEURON or Arbor. HybridLFPy [148] is an extension of LFPy for the point neuron simulation. Elephant [90] is a tool for analyzing the electrical signal and spike trains and also the statistical generation of spike trains.

EBRAINS provides a collaborative platform (<https://wiki.ebrains.eu>) and a social network (<https://community.ebrains.eu>) to facilitate access to these tools. The collaborative platform comprises a set of tools for collaboration between researchers and the training of future researchers. This platform uses xwiki [312] (for creating a collaborative environment and editing and managing web pages), OnlyOffice [264] (for collaboration on text documents), Seafile [311] (for file management) and a JupyterLab [283] environment (to run code and use EBRAINS' tools). This platform is connected to Fenix for running this collaborative environment. The overview proposed by Schirner et al. 2022 [303] provides an example of the interaction of tools and available services.

Deuxième partie
ORIGINAL WORK

CHAPTER 3

Multiscale co-simulation design pattern for neuroscience applications

Contents of Chapter 3

3.1. Chapter overview	72
3.2. Introduction	73
3.3. Results	76
3.3.1. Virtual experiment of hippocampal CA1 embedded in a full mouse brain	76
3.3.2. Output signal from the virtual experiment	77
3.3.2.1. Results at microscale	79
3.3.2.2. Results at macroscale	79
3.3.3. Workflow between NEST and TVB	82
3.3.4. Performance	83
3.4. Materials and Methods	86
3.4.1. Models	86
3.4.1.1. CA1 model	86
3.4.1.2. Mouse Brain model	86
3.4.1.3. Electrophysiological monitoring model	87
3.4.1.4. Choice of three sets of parameters	87
3.4.2. Communication between modules	87
3.4.2.1. Initialization of communication	87
3.4.2.2. Synchronization between modules	88
3.4.3. Performance tests	88
3.4.4. Implementation details	89
3.4.4.1. Deadlock due to global interpreter of python	90
3.5. Discussion	90

3.1. Chapter overview

As the core of this thesis is the multiscale co-simulation of the brain, I will first describe the software science co-simulation design pattern. This chapter describes the design pattern and illustrates its functioning along a neuroscience example. An example is the multiscale simulation of CA1 region of the cellular-level hippocampus of the mouse embedded into a full brain network involving ECOG and polytrodes recordings. This simulation is running on the co-simulator of The Virtual Brain [299] and NEST [98].

This chapter is based on an under-revision article in *Frontiers in Neuroinformatics* :
Multiscale co-simulation design pattern for neuroscience applications

Lionel Kusch¹, Sandra Diaz², Wouter Klijn², Kim Sontheimer², Christophe Bernard¹,
Abigail Morrison^{2,3,4} and Viktor Jirsa¹

¹Institut de Neurosciences des Systèmes (INS) UMR1106, Aix-Marseille Univ., Marseille, 13005, France

²Forschungszentrum Jülich GmbH, Institute for Advanced Simulation, Jülich Supercomputing Centre (JSC), Simulation and Data Laboratory Neuroscience, JARA, Jülich, 52425, Germany

³Forschungszentrum Jülich GmbH, IAS-6/INM-6, JARA, Jülich, 52425, Germany

⁴Computer Science 3 - Software Engineering, RWTH Aachen University, Aachen, 52062, Germany

Abstract

Integration of information across heterogeneous sources creates added scientific value. Interoperability of data, tools and models is, however, difficult to accomplish across spatial and temporal scales. Here we introduce the toolbox Parallel Co-Simulation, which enables the interoperation of simulators operating at different scales. We provide a software science co-design pattern and illustrate its functioning along a neuroscience example, in which individual regions of interest are simulated on the cellular level allowing us to study detailed mechanisms, while the remaining network is efficiently simulated on the population level. A workflow is illustrated for the use case of The Virtual Brain and NEST, in which the CA1 region of the cellular-level hippocampus of the mouse is embedded into a full brain network involving micro and macro electrode recordings. This new tool allows integrating knowledge across scales in the same simulation framework and validating them against multiscale experiments, thereby largely widening the explanatory power of computational models.

3.2. Introduction

The brain is a complex system that includes billions of cells that interact with each other in a nonlinear manner. As a result, even if we were able to measure what all cells are doing simultaneously, we would not gain a deep understanding of how the brain works, as emergent properties can be only understood through an integrated approach [61], ideally in a common theoretical framework to give meaning to data at all scales [125]. Such a framework using theoretical models can account for nonlinearities and subsequently explain emergent properties [186, 276]. Numerous models have been developed to study the interactions of molecules within cells, cell physiology, the activity of cell populations, full brain dynamics and human behaviour [117, 104, 172]. It is currently impossible to model the brain with all its cellular and molecular constituents due to limitations in resolution, computational resources, or available data from measurements. As a result, even if a given physio/pathological process can be modelled at the macroscopic scale, the lack of microscopic resolution at the molecular scale prevents obtaining mechanistic insight [238]. It is, therefore, important to bridge different scales, which is a challenge not unique to neuroscience. In material science, the study of composite materials requires the description of molecular interactions of individual composites and a global description for the analysis of the subsequent deformation of the composite plate [304]. In biology, to understand the effect of drugs on tumour growth, it is necessary to model the tissue of cells around the tumour, the tumour cells, and the subcellular transduction signalling pathways [293, 289]. In neuroscience, synaptic plasticity uses mechanisms of spike timing on the millisecond scale but leads to the formation of long-term memory evolving on the scale of minutes, days and weeks [103].

Our current study aims to provide a methodology to address the scientific and technical problems of multiscale co-simulation in the brain. The main difficulty of multiscale

3. Multiscale co-simulation design pattern for neuroscience applications – 3.2. Introduction

simulation is to enable the information exchange between models formulated at different scales. Such communication can be interpreted as a coupling across scales. For example, in the case of tumours, the tissue around the tumours is represented by a continuum model (first scale), which interacts with discrete tumour cells (second scale); while continuous signalling pathways are modelled in cells (third scale). At present, it is not possible to create a common coupling function amongst these three scales and each scale can use a dedicated simulator engine for optimizing the simulation. In the case of tumours, a common approach is to use COMSOL Multiphysics [64] for the tissue simulation, and Matlab [174] for the simulation at cellular and subcellular scales. Because the interaction of simulator engines is not a commonly supported feature, co-simulation of models at different scales and within a common framework is challenging. Existing solutions for co-simulation in physics [118, 141] or in biology [236, 158] cannot be easily adapted in neuroscience due to the specificity of simulators and models. There is a large number of scale-specific simulators in neuroscience, e.g. for compartmental neurons : NEURON [52], Arbor [3], Genesis [34]; for point neurons : NEST [135], Brian [334], ANNarchy [353]; for the brain network : The Virtual Brain (TVB) [299], Neurolib [48]). Most of these simulators can support multiscale simulation to a limited degree, but they remain specialised and optimised for supporting a specific model type; consequently, the usage of other model types diminishes their optimal performance. The objective of co-simulation is to remove this limitation by exploiting the advantages of each simulator within the same simulation [137, 244, 113, 100].

Schirner et al. 2022 [303] provide an overview of software tools available for TVB in the European digital neuroscience infrastructure EBRAINS. Two toolboxes for co-simulation are introduced in EBRAINS, TVB-Multiscale and Parallel Co-Simulation. The former toolbox focuses on rapid development for scientific use cases, whereas the latter focuses on optimisation of co-simulation performance and applies the co-simulation design pattern presented in this study. An illustrative example of co-simulation of multiscale models using TVB Multiscale co-simulation is virtual deep brain stimulation [237, 315].

Here we present the methodology of the Parallel Co-Simulation toolbox and illustrate its use along the example of combined microscopic Local Field Potential (LFP) and neuronal firing recordings, and macroscopic electro-CORTicoGraphy (ECOG) in mice [294]. This example aims to demonstrate computational requirements for interpreting recorded multiscale data using multiscale modelling [73]. The method is based on a software science co-design pattern[102] that dictates the separation of science and technical attributes, allowing these to be addressed in isolation where possible. This separation is based on transformer modules, which synchronise and connect simulators and include the function for transforming data between scales. A multiscale model is built from experimental data obtained in the mouse brain with ECOG cortical signals and LFP signals in the CA1 region of the hippocampus. The model is co-simulated using the simulators TVB and NEST. Three sets of model parameters related to different network dynamics are chosen to demonstrate the feasibility

3. Multiscale co-simulation design pattern for neuroscience applications – 3.2.
Introduction

and limits of this modelling approach. The following sections describe the technical details and the optimisation for co-simulation.

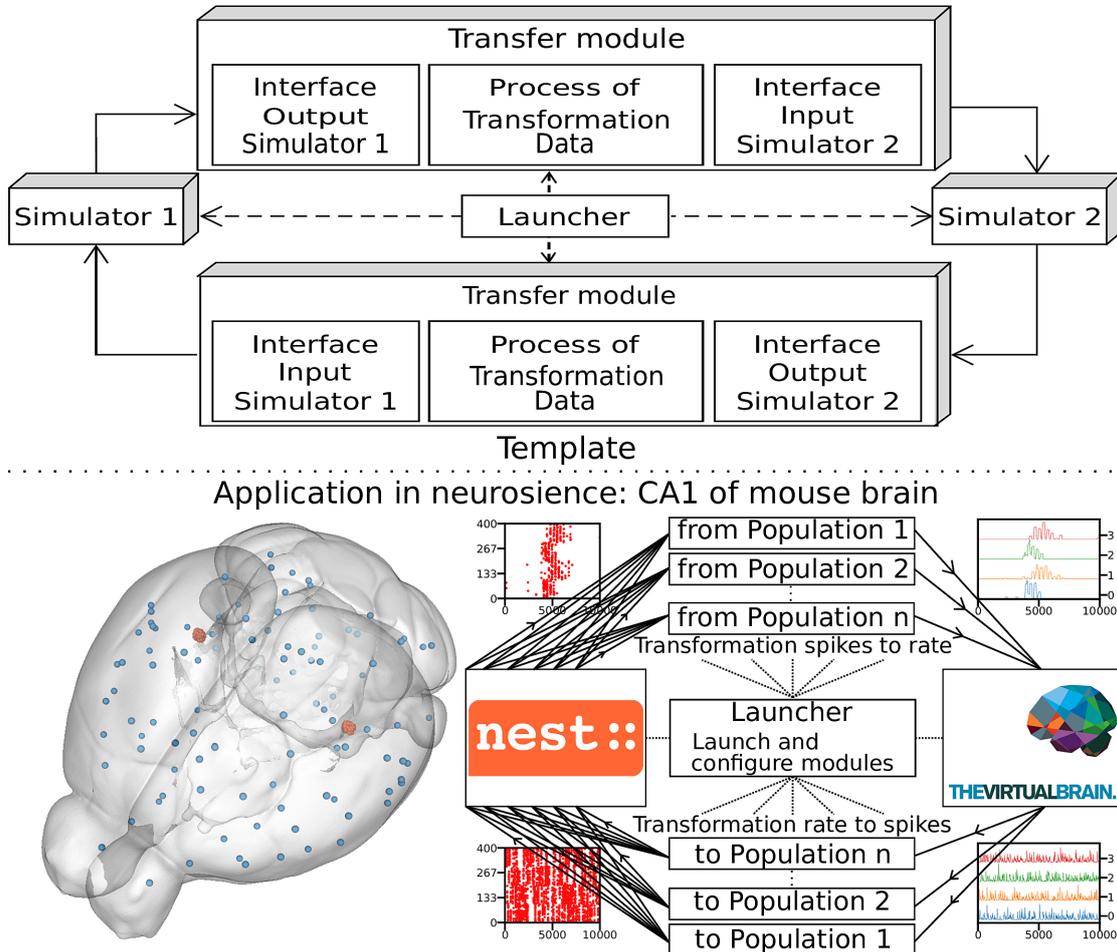


FIGURE 3.1. – Multiscale co-simulation design pattern and Example of an application in neuroscience

Top. Multiscale co-simulation design pattern between 2 simulators using transfer modules to transform and transfer data between scales. **Bottom.** Application of the co-simulation pattern for a neuroscience use case focusing on the CA1 region of a mouse brain. The **left panel** shows a rendering of the mouse brain from Allen Institute [222]. Blue spheres mark the centres of mouse brain regions, and the red spheres are a subset of neurons of the CA1. The **right panel** illustrates the co-simulation data flow between TVB [299] and NEST [152], showing the different functional modules. The four corners' plots illustrate the data type exchanged in respective information channels. The transfer modules exchange mean firing rate data with TVB (module on the right) and exchange spike times with NEST (module on the left). Each population has a specific module enabling data transfer between populations in different scales.

3.3. Results

The multiscale co-simulation software science co-design pattern formalises the interactions between parallel simulations at different scales. The data transformation among scales is performed during their transfer among simulators. This design pattern comprises 5 modules (figure 3.1a) : one launcher, two simulators, and two transfer modules. Each transfer module contains 3 components : one interface for receiving data, one interface for sending data and a transformation process. The launcher starts and handles the coordination of simulation parameters. The simulators perform scale-specific simulations. The transfer modules transfer the data from one simulator to another. During the transfer, the transformation process transforms the incoming data for the simulator on the receiver side.

This study applies the multiscale co-simulation design pattern to a virtual experiment workflow between the in-silico mouse whole-brain dynamics and the in-silico micro-scale network dynamics of the hippocampus CA1 region. The recording of the virtual CA1 and virtual mouse brain has similar positions that some multiscale experiments [294] (see figure 3.1b). The Virtual Brain (TVB) [299], an open-source platform, has been used to simulate the mouse whole-brain network activity, while NEST [152], another open-source platform, has been employed for the simulation of the CA1 neuronal network dynamics. This specific application illustrates this novel design pattern's technical limitations and demonstrates the potential for a wider range of applications.

3.3.1 Virtual experiment of hippocampal CA1 embedded in a full mouse brain

The virtual experiment of the mouse brain is composed of a brain network model, regional neuronal network models and electrophysiological sensors models.

The whole-brain animal model is a network comprised of nodes and edges, where each node contains a neural mass model to simulate the activity of each region and where edges represent the anatomical connections among the regions. The anatomical connections are defined by track lengths and an adjacency matrix representing the coupling strengths of connections between the regions of the network, the "connectome", which are extracted from tracer data from the Allen Institute [263] (figure 3.2f and 3.2g). The dynamic activity of each brain region is obtained with the Neural Mass model described by Di Volo et al. [356] (see Materials and Methods). The neuroinformatics platform The Virtual Brain (TVB) [299] performs the animal whole-brain simulation by considering both the chosen Neural Mass model and specific "connectome".

The dynamics of the two main brain regions of interest, the left and right hippocampus CA1 (figure 3.2), are modelled as a separate neural network composed of point neurons connected with static synapses. Each network comprises one inhibitory, and one excitatory homogeneous population of adaptive exponential integrate and fire

neurons [37] (see Materials and Methods). In each microcircuit, the populations of point neurons are taken to be homogeneous; that is, neurons of the same population have the same parameter values. The neuroinformatics platform NEST [152] is able to perform the regional neuronal network simulation using the aforementioned description of the microcircuit of point neurons.

To compare the simulations with empirical data, the virtual experiment contains two models of electrophysiology sensors for probing neural activity. The electrophysiological sensor models are two surface grids of 8-channel electrocortigraphy arrays and two penetrating multi-electrode arrays of 32 recording sites each. Their positions are illustrated in figure 3.2a. Figure 3.2e shows the position of the polytrodes in the mouse brain, while figure 3.2b and 3.2d depict the position of the left probes in a cross-section of the left hemisphere and the position of the point of the polytrodes in the population of neurons, respectively. Figure 3.2c displays the polytrodes with the 32 recording sites. The simulated signal from the ECOG sensor is computed using the model of a point dipole in a homogeneous space as described by Sanz-Leon et al. 2015 [300] (see Materials and Methods) and the hybridLFPy [148] software is used for computing the signal from the recording site of the implanted probes (see Materials and Methods). The latter software uses morphologies and spatial position of neurons to generate the underlying local field potential (LFP) for given spike trains of point neurons. The morphology of neurons is taken from the presented morphology in Schuman et al. 2022 [322]. The excitatory morphology is based on the pyramidal cell morphology, and inhibitory neurons are based on the basket cell morphology [322].

3.3.2 Output signal from the virtual experiment

This section describes the co-simulation results at different scales by describing the possible recordings of physiological signals from the simulation of CA1 embedded in a whole mouse brain. The Discussion section will provide an interpretation of these results to describe the advantages and the limitations of the multiscale co-simulation design pattern. As described in figure 3.2, the output modalities of one virtual experiment have some direct equivalent measure of reality experimentation such as the local field potential measure at every thirty-two sites of each polytrode electrode (figure 3.2j) and from the sixteen electrocortigraphy channels of each hemisphere (figure 3.2k). Moreover, the simulation gives access directly to the voltage membranes of the CA1 neurons (figure 3.2h), adaptive current of the CA1 neurons (figure 3.2g), spike times (figure 3.2i) and the mean firing rate of the different regions of the mouse brain (figure 3.2m). To illustrate the variability of the measures and some limitations of the coupling scale model, we choose three sets of different parameters for CA1 and Neural Masses. Each parameter represents one of three dynamic regimes of the CA1. These results are separated between micro (figure 3.3) and macro (figure 3.4) scales, but they are the output of the simulation workflow between TVB and NEST. In particular, figure 3.3 reports the mean voltage membrane, mean adaptive current, instantaneous firing rate and the signal of 12 central sites from the 32 electrode sites

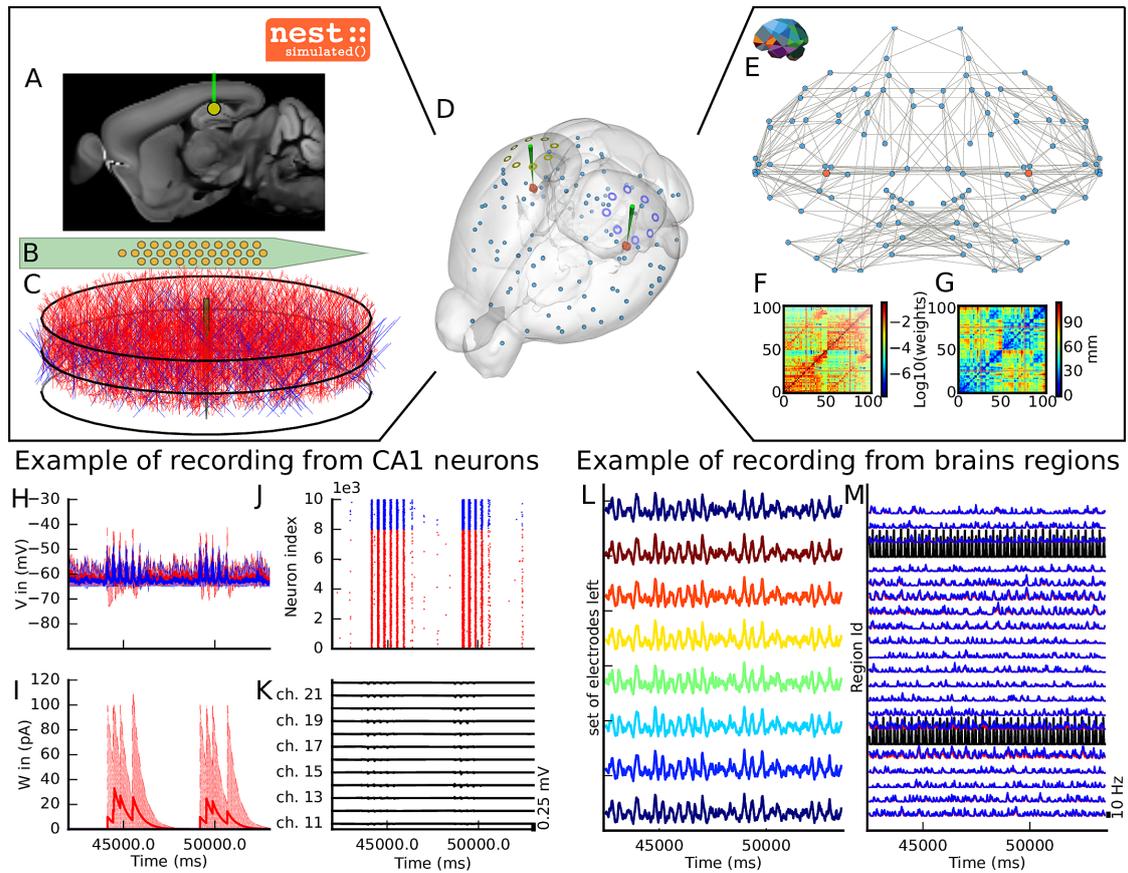


FIGURE 3.2. – The virtual mouse brain experiment

a Cross section of the mouse brain with the position of the left implanted electrode. **b** Position of the site layout of a polytrode (Neuronexus 32 models from MEAutility library). **c** The probe position inside the neural network. The red neurons are pyramidal neurons [322] and the blue neurons are basket cells [322]. **d** Mouse brain of Allen Institute [222] with the position of the 2 polytrodes and 16 ECOG electrodes. The ECOG electrodes measure the neural field from the surface of the electrode in blue for the left hemisphere and yellow for the right hemisphere. Blue spheres mark the centres of mouse brain regions, and the red spheres are a subset of neurons of the CA1. **e** Representation of the connectome of the mouse brain [263]. The blue dots are brain regions, and the red ones are CA1 regions, whose neurons are simulated with NEST. The grey links highlight the strongest anatomical connections. **f** The weights of the anatomical links in F are shown as an adjacency matrix. **g** The tract lengths associated with F are shown as an adjacency matrix. The anatomical connections are extracted from tracer data of the Allen Institute [263]. **h** example of voltage recorded from 10 excitatory and 10 inhibitory neurons. **i** Example of adaptation currents recorded from 10 inhibitory and 10 excitatory neurons. **j** Example of spike trains recording from the left CA1. **k** Example of Local Field Potential recorded from the poly-electrode generated from the spike trains and neuron morphologies. **l** Example of recording from the ECOG electrodes of the left hemisphere. **m** Example of mean firing rate of excitatory and inhibitory populations for a subset of mouse brain regions.

of the specific CA1 network. Figure 3.4 displays the results on the whole brain level : the mean firing rate of each brain region, the signal of the 16 electrocortigraphy channels and the transferring mean firing rate from the spiking neural network.

The figures 3.3 and 3.4 are separated into three different panels, which correspond to the three sets of parameters representative of the different types of dynamics exhibited by spiking neural networks (see Materials and Methods for the choice of these parameters). Panel **a** represent an asynchronous (A) state, which is characterised by a constant (flat line) the mean firing rate (see figure 3.3a top-right and figure 3.4a top-left). Panel **b** represents an irregular synchronous (IS) state, which reflects a large irregular variation of the mean firing rate (see figure 3.3b top-right and figure 3.4b top-left). Panel **c** represents regular bursting (RB) reflecting regular oscillations (see figure 3.3c top-right and figure 3.4c top-left) and a second dominant high frequency (see figure 3.3c bottom-right).

3.3.2.1 Results at microscale

The top left of panels a, b and c of figure 3.3 show the membrane voltages for ten excitatory neurons (thin red curves) and ten inhibitory neurons (thin blue curves) and mean membrane voltage of these neurons (thick curves). The middle left of panels a, b and c of figure 3.3 represent the adaptive currents from the same ensemble of neurons (thin curves) and the mean adaptive current of these neurons (thick curves). The third biological observable from the simulation is the Local Field Potential which differs among panels (see bottom left of panels a, b and c of figure 3.3). The top right of panels a, b and c of figure 3.3 display spike raster plots of the excitatory population, in red, and the inhibitory population, in blue, of the left CA1. The spiking activity is homogeneously distributed between neurons and time frames for the A state, while the other two states show co-activation of neurons with different periods. The associated instantaneous firing rate is shown in the middle right of panels a, b and c of figure 3.3. The spectral analysis of the instantaneous firing rate displays a peak around 3 Hz for the IS state (bottom left of panel b of figure 3.3), no peaks for the A state (bottom left of panel a of figure 3.3), and two peaks (around 6 Hz and 160Hz) for the RB state (bottom left of panel c of figure 3.3). For the RS state, the frequency of the first peak, 6Hz, is also present in the mean of the adaptive currents, while the second peak is associated with the burst time, as shown in further detailed in Supplementary Figure 1.

3.3.2.2 Results at macroscale

The top left of panels a, b and c of figure 3.4 display the instantaneous firing rate (light red) of the spiking neural network with the associated transferred mean firing rate of the left region of CA1 (thick red line). The neural network's different states affect the ECOG signals, as shown in the bottom left of panels a, b and c of figure 3.4. The mean firing rate of excitatory (blue) and inhibitory (red) populations of each brain region are plotted in the graph on the right part of panels a, b and c of figure 3.4 and Supplementary Figures E.2, E.3 and E.4.

3. Multiscale co-simulation design pattern for neuroscience applications – 3.3. Results

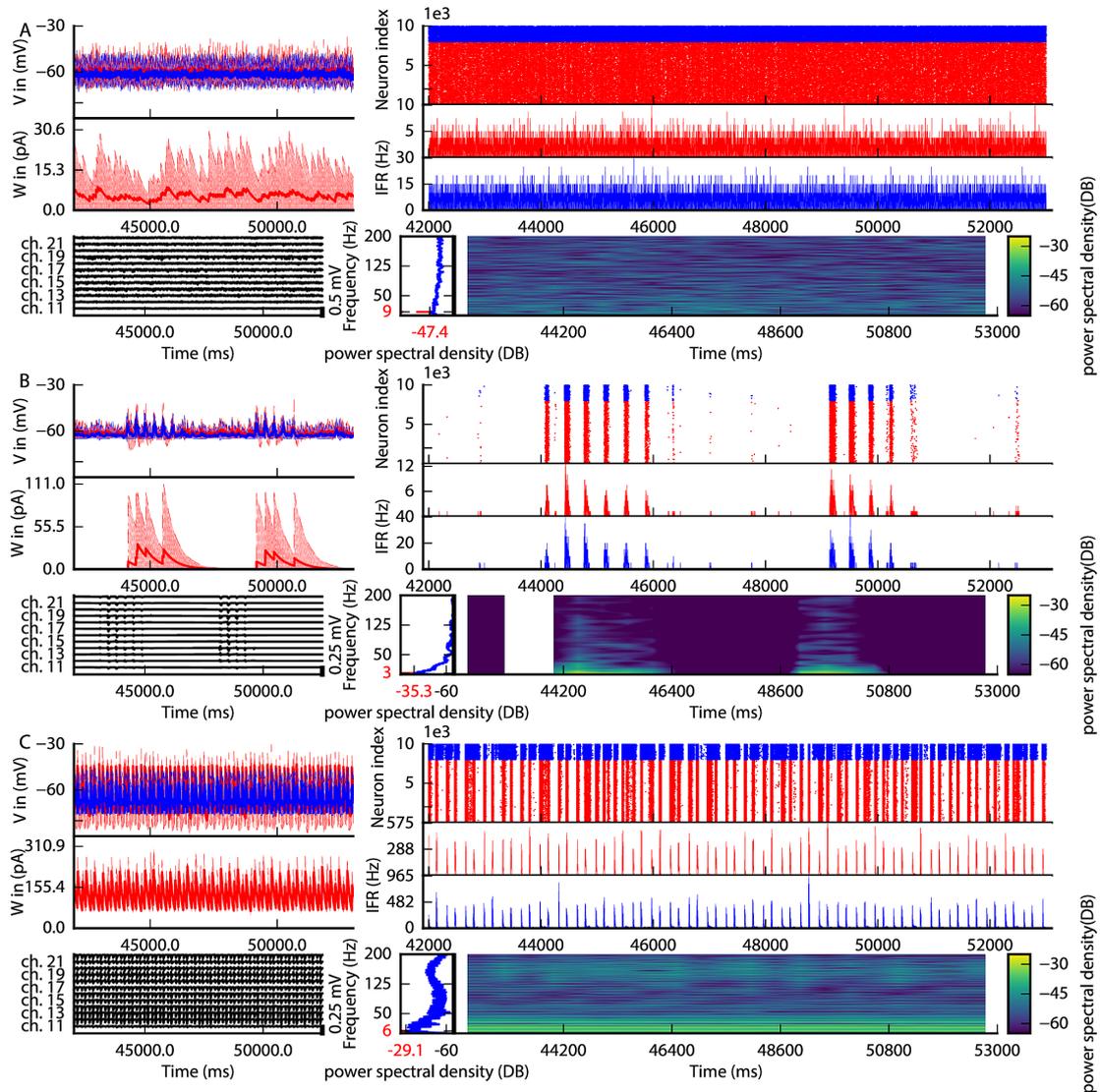


FIGURE 3.3. – Spiking neural network in 3 different states of the left CA1
 The parameterization of the spiking neural network of CA1 is chosen such that the dynamics are in an asynchronous state (a), irregular synchronization state (b) and regular bursting (c). **top-left** Voltage membrane of 20 adaptive exponential leaky and integrator neurons and their mean in a thick line. The red (blue) lines are excitatory (inhibitory) neurons. **middle-left** The adaptation currents of 10 neurons and their mean in a thick line. **bottom-left** Local field potential from the 12 sites in the middle line of the left polytrode. The local field potential is computed from the spike trains of all neurons by the software HybridLFPY[209]. **top-right** Spike trains of 10000 neurons for 11s. **middle-right** instantaneous firing rate of the excitatory (inhibitory) population above in red (blue). **bottom-right** Spectrogram and power spectrum example of the instantaneous firing rate for 10s.

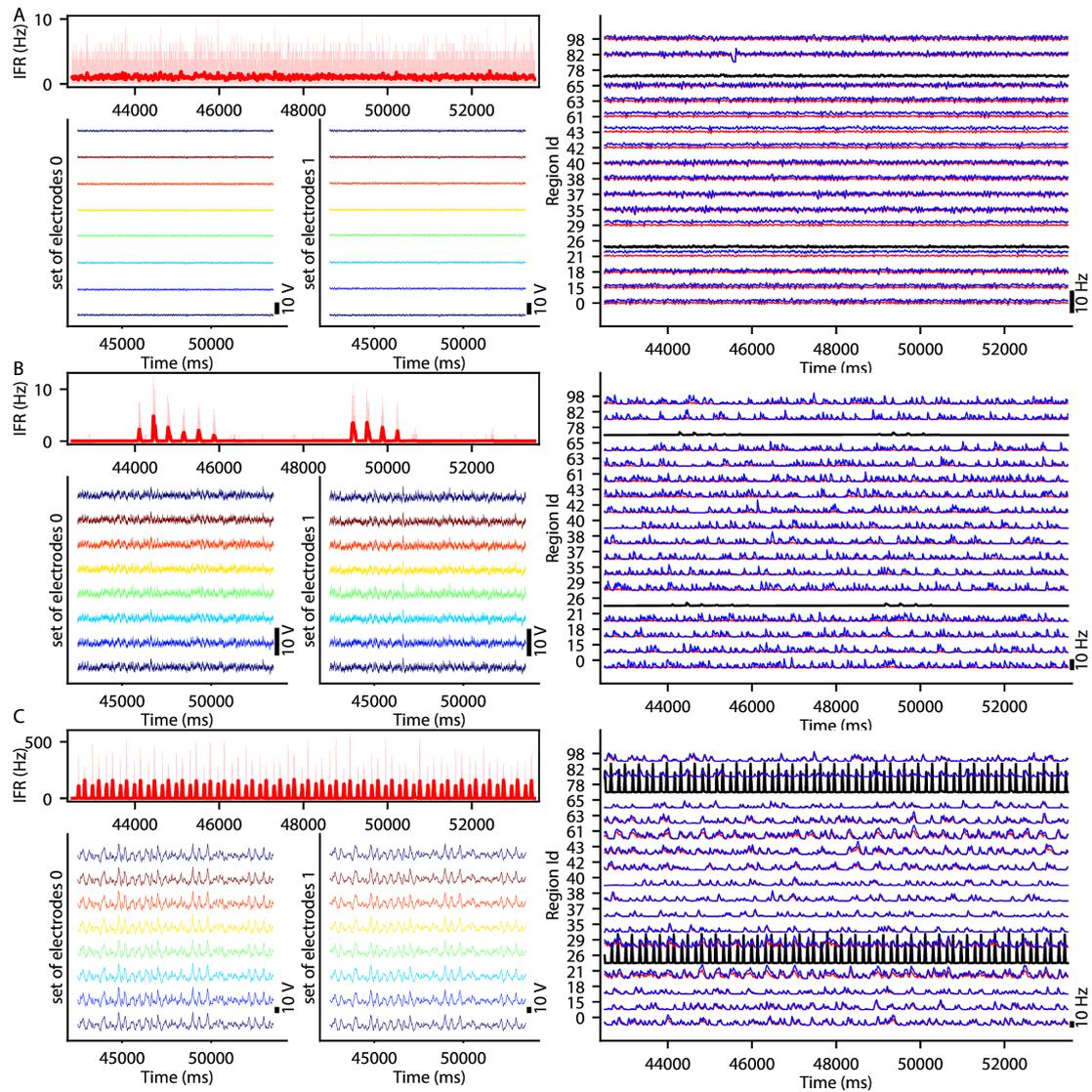


FIGURE 3.4. – Three different states of CA1 in mouse brain

The parameterization of the CA1 spiking neural network is defined to obtain an asynchronous state **(a)**, an irregular synchronization state **(b)** and a regular bursting **(c)**. **top-left** Instantaneous firing rate of spiking neural networks in light red for 11 seconds. The thick line shows the sliding window mean firing rate. **bottom-left (bottom-right)** Signal from ECOG sensors, the figure represents the recording of the 8 electrodes on the top of the left (right) hemisphere. **right part** Subset of region overview of the mean firing rates of excitatory, in red, and inhibitory, in blue, population from the model of Mean Adaptive Exponential. The two black curves are the mean firing rate of the two populations of excitatory neurons simulated with NEST [152].

3.3.3 Workflow between NEST and TVB

The previous multiscale example uses the workflow between TVB and NEST for the co-simulation. As an implementation of the design pattern, this workflow comprises five modules : two simulators (TVB and NEST), one launcher and two transfer modules. All these modules are built with the capability to be repurposed or replaced, allowing for adjustments of components of transfer modules or communication protocols (see Discussion). Two additional proofs of concept were implemented to demonstrate the possibility of the reusability of the components. The first example replaces NEST with NEURON, and the second replaces TVB with Neurolib (see Supplementary Figure E.24). Moreover, without extra development, we get a proof of concept of co-simulation between NEURON and Neurolib.

The simulators perform the actual integration of the dynamics in time and require two properties to be integrated within one optimised and coherent workflow. The first property is time delay equation management, essential for reducing data transfer overhead. The second property is the presence of a high bandwidth Input/Output (I/O) interface that facilitates the efficient exchange of data and parallel execution of the simulators. Since TVB and NEST did not have generic high bandwidth I/O interfaces by default, these had to be implemented for each simulator. Details of how these I/O interfaces were created are reported in Supplementary Note E.1.1. Briefly, the NEST interface uses the device nodes with a specific back-end, while TVB uses proxy nodes which are proxy nodes used for the interface with the external software.

The launcher prepares the environment for the simulation and initiates all the other modules, as shown in figure 3.5a (see details in the Supplementary Figure E.5). The preparation consists of creating folders for the different modules, the logger files, and the common file with all the parameters of the co-simulation. Creating the parameters file provides the functionality to enforce consistent constraints on the parameters to be shared between the modules, such as ensuring the same integration step in both simulators, which is needed for correct synchronisation between modules.

The transfer modules connect simulators by transferring data between scales and adapting the communication delay throughout the simulation. Each module is comprised of three components : two interfaces and one transformer (see Figure 3.1a and Supplementary Figure E.14). These components are implemented in different files for reusability and modularity and are tested independently to ensure robustness (see Supplementary figure E.12). The interfaces are specific to each simulator, while the transformation can be extended, modified or reused since the transformation function is implemented as an independent process (see Supplementary Note E.1.2). The components exchange data using a simple Application Programming Interface (API). The API is based on four functions and assumes that the connections are already established. The functions are "check if ready to get or send data", "transfer data", "end the transfer" and "release the connection" (see Supplementary Note E.1.2 and Supplementary Figure E.15). The API is implemented with two different technologies depending on the nature of the parallelisation of the components (multiprocessing or multithreading). In the case of multiprocessing, each component runs

in an individual process, and a Message Passing Interface (MPI) is used to transfer data. In the case of multithreading, each component runs in an individual thread in a shared process, and the data is transferred using shared memory. Multithreading uses fewer computational resources (see Performance section). However, multithreading is unstable on a supercomputer due to the global interpreter lock of python (for more details, see the section deadlock due to the global interpreter in Materials and Methods). It is recommended to use multiprocessing on a supercomputer, this solution is slower, but it is stable and allows the usage of multiple computational nodes for the transformation function.

The transformation function provides Neural Mass firing rate values by using a sliding window, shown in figure 3.5e. The panel also illustrates the inverse transformation from the mean firing rates to spike trains using a multiple interaction process [209].

The modular workflow execution is composed of three main blocks : start-up, simulation-loop and termination (see Figure 3.5a and details in the Supplementary Figure E.5).

The start-up procedure allocates a logger for each component, facilitating debugging of the co-simulation. Subsequently, the modules and their communication channels are configured according to the parameter file. At this stage, several initialisation files are generated with simulation parameters only available after instantiation of the model (e.g. id of NEST devices and MPI port description).

Once the simulation is launched, the simulator time clocks are synchronised using asynchronous message passing : At each synchronisation step, simulators receive input data, after which the next step is simulated. The transfer modules can buffer data for one synchronisation step until the receiving simulator is available for receiving. Each simulator requires an initial condition (NEST : initial voltage membrane and adaptation current and TVB : state of the node during the previous seconds) and an initial message. For TVB, this starting message is sent by the transformer processes while, for NEST, it is produced by transforming the initial condition of TVB.

Ultimately, the termination occurs at the end of the simulation by the simulators themselves (see Materials and Methods for details).

3.3.4 Performance

The evaluation of the performance is made against a fictitious workflow with optimal performance, a co-simulation with instantaneous communications between simulators. As all the modules are designed to run in parallel, the co-simulation time for each module is identical and equal to the total running time. The focus is only on the simulator timers because the time of the transformer components is dominated by the waiting time of data (see Supplementary Figure E.5). The total running time of the simulators is divided into 5 parts. The "initialisation" time is the time of configuring the simulators and creating connections. The "ending" time is the time of closing the connections, stopping the simulator engine and terminating processes. The "simulation" time is the total time of the internal computation of simulator engines. The

"wait" time is the total duration of waiting time for access to the data to transfer by the simulator interface of the transformer module. The "IO" time is the total duration of functions for exchanging data between simulators and the transfer modules minus the "wait" time.

A perfect co-simulator has the time of the slowest simulator X; thus, "wait" and "IO" times equal zero. From figure 3.5b and the Supplementary figure E.17, the actual implementation is close to ideal when the number of neurons simulated by NEST is lower than 1000. In this case, TVB is the slower simulator, and NEST spends most of the time waiting for data from TVB.

When the number of simulated neurons is between 1000 and 20000 neurons, "simulation" time of TVB is approximately the same as the sum of "simulation" time and "IO" time of NEST. In this condition, each simulator is waiting for the transformation of the data among scales.

When the number of simulated neurons is higher than 20000, NEST is the slowest simulator. In this case, the co-simulation time is determined by the "simulation" time and the "IO" of NEST. The "wait" time is set to zeros, and the "IO" time is higher than the "simulation" time (see Supplementary Figure E.17 and Supplementary Figure E.20). The two principal causes are that the communication between modules is slower than inside the modules and the increased size of the neural spike data with the increasing number of neurons (each neuron in NEST receives an individual spike train). A closer look at the performance shows that the communication spends most of the time sending individual spike trains to NEST (see Supplementary Figures E.23). However, the data size is related to the model chosen and can be reduced.

As shown by figure 3.5c and 3.5d, some optimisations can be implemented to reduce the problem of overhead time of communication. Figure 3.5c and Supplementary figure E.18 represents the time delay between brain regions when delayed data is aggregated to reduce the "IO" time and, hence, the co-simulation time. In this case, the simulators are not synchronised at each time step but at n time steps (limited by the model of connection). This aggregation can reduce co-simulation time by a factor of 6 (see Supplementary Figure E.18 and Supplementary Figure E.21). Figure 3.5d and Supplementary figure E.19 represent a reduction of co-simulation time per reduction of the "simulation" time of one simulator. The increase in NEST's resources does not modify the "IO" time until the resource is available. Since the tests are running on one computer, increasing resources for NEST increases the "simulation" time of TVB and reduces the "simulation" time of NEST. However, by deploying the workflow on high-performance computing facilities, the latter result does not replicate, and the simulation time gives similar a result with an increase in "IO" and "simulation" time because the communication between nodes is slower (see Supplementary Figures E.22).

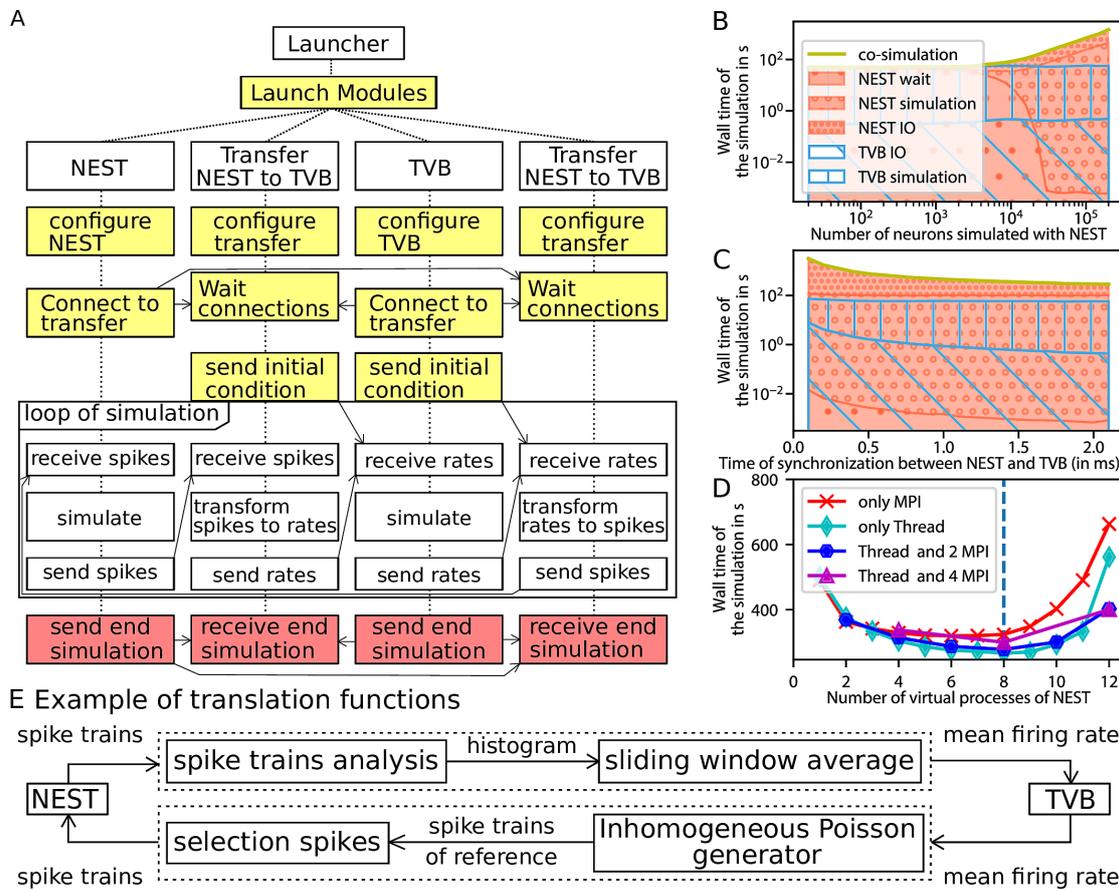


FIGURE 3.5. – Architecture and performance of the co-simulation

a The interaction among the modules and data exchanges during co-simulation execution. The boxes in yellow mark start-up : initialization and configuration, the boxes in red for the termination of the simulation and the boxes in white for the simulation phase. **b,c,d** Performance of the workflow is obtained for 1 second of simulated time (see Materials and Methods for more details). The reference implementation use 1 MPI process, 6 virtual processes/threads, a synchronization time step of 2.0 ms, and simulates 20000 neurons. **b** The wall clock time of the simulators as a function of the number of neurons. The total time of the co-simulation is represented in yellow. The "wait", "simulation" and "IO" times of NEST are represented in red surface with respectively hatches with big circles, small circles and points. The "simulation" and "IO" times of TVB [299] are represented in the blue surface with respectively hatches horizontal lines and oblique lines. **c** Simulation time depending on the synchronized time between simulator. The colour code is the same as the panel B. **d** Wall clock time depending on the number of virtual process used by NEST [152]. The green, blue, purple, red curves are associated with different parallelization strategy of NEST, respectively, only multithreading, 2 MPI processes with threads, 4 MPI processes with thread and only MPI processes. The vertical blue line represents the number of cores of the computer. **e** The "transform between spikes to rate" and "transform between rates to spikes" blocks are displayed with the different steps for transformation of data between TVB and NEST.

3.4. Materials and Methods

The simulation details and models' parametrisation are in Supplementary Table E.1. The format of this table is drawn from the proposition of Nordlie et al. 2009 [261] for spiking neural networks. This new format includes the description of brain network modelling, the description of the coupling between scales and the description of the measurements of the simulation. This format contains more details than the proposition of Nordlie et al. because it contains all the parameters for the co-simulations. The following text provides an overview of the models, communication between modules, details of the performance tests and implementation details.

3.4.1 Models

3.4.1.1 CA1 model

The spiking neural network of CA1 comprises two regions (left and right), which contain two populations, 8000 excitatory neurons and 2000 inhibitory neurons. This network is simulated by NEST [152], a neuro-informatics platform for spiking neural networks. The adaptive exponential integrate and fire neurons [37] are connected by exponential conductance-based synapses with a connection probability of 5% inside the region. The excitatory population establishes normalised weighted connections with the other regions defined by the mouse connectivity atlas. Additionally, we assume that each neuron has the same unique number of synaptic connections from other brain regions; the mouse connectome defines the repartition of these synapses. Transmission delay between regions is defined as the ratio of the distance between the regions and the transmission speed. Calculating these ratios is part of the configuration of The Virtual Brain (TVB) [299] because the data required by TVB is the track lengths between regions and the speed of the transmission. Within a region, the synaptic transmission delay is instantaneous. In addition, the neurons can receive external noise input modelled as an independent Poisson process in addition to the external stimuli received from other regions through the transfer of mean firing rates as transformed spike trains.

3.4.1.2 Mouse Brain model

The mouse brain model is simulated using The Virtual Brain [299, 240], a neuro-informatics platform for connectome-based whole-brain network modelling. The "connectome" used here is extracted from Allen Mouse Brain Connectivity Atlas [263] in 2017. The large-scale brain network is comprised of linearly coupled Neural Mass models. Specifically, the model representing each region is a second-order Mean Ad Ex model [356] with adaptation, representing the mean firing rate for an ensemble of one excitatory and one inhibitory neuronal population.

3.4.1.3 Electrophysiological monitoring model

The electrophysiological monitoring variables are computed using two models representing the cortical and implanted sensors. The electrocorticography model is a simple forward solution of a dipole at the region level. The electric field recorded by the virtual sensors at the brain level is based on two assumptions : considering the brain as a homogeneous space, and the field is generated only excitatory population. With these assumptions, the recorded field is the sum of excitatory population activities, i.e. the mean excitatory firing rate weighted by the distance between the sensors and the region's centre [300]. The implanted sensors' signals are computed from point-neuron activities using a hybrid scheme for modelling local field potentials (LFP). Specifically, each potential is simulated using hybridLFPy [148], which incorporates the recorded spike from the network and the morphology of the pyramidal and basket cells.

3.4.1.4 Choice of three sets of parameters

The parameters for Irregular Synchronous state are chosen based on Di Volo et al. paper [356]. The coupling between regions and the noise was defined after empirical exploration to get a fluctuation of the firing rate in each region.

Based on the first set of parameters, the Asynchronous state was defined by changing empirical parameters to avoid fluctuation in the brain regions. The result is a reduction of the spike-triggered adaptation of the excitatory neurons, a reduction of the number of connections between the regions, an augmentation of the inhibitory synaptic weights, a reduction of the variance of the noise and the addition of a Poisson generator for the spiking neural network.

Based on the first set of parameters, the parameters for the Regular Bursting state are chosen by changing the type of the neurons from regular spiking to regular burst neurons. This modification was realized by changing the voltage reset of the membrane and the leak of the reversal potential of the excitatory and inhibitory neurons, the spike-triggered adaptation and the time constant of the adaptation current of excitatory neurons. An empirical exploration of the models is done to get a balanced spiking neural network and the desired brain dynamic. The result of this exploration is a reduction of the connection between regions and a reduction of the connection between excitatory and inhibitory neurons, a reduction of the number of connections between brain regions and a reduction of the noise variance.

All the numerical values of the parameters are in Supplementary Table E.1.

3.4.2 Communication between modules

3.4.2.1 Initialization of communication

During the initialization of the simulation, the launcher creates a specific folder for each module and an extra folder for the logger file of all components. The launcher

creates a file with all the parameters. This file contains parameters for the simulation with dedicated sections for each module. The parameters shared between modules are duplicated by the launcher in each section to ensure there are the same.

3.4.2.2 Synchronization between modules

The transfer modules synchronize the simulation by managing the access to its internal buffer and receiving status messages from the simulators. The receiver process receives the data and aggregates them in a buffer. Rate data do not need to be buffered when using MPI communication, they are sent or received directly to the transformer process. This buffer is transferred to the transformation function when the previous data are transformed and transferred to the sender process. The sender process gets the data after sending the previous data to the simulator. It can only send the data to the simulator when it is ready. In addition, the simulator needs to await data for the next step of the simulation. Given all these constraints, the transfer module assures correct transport and keeps the components synchronized. If needed the transfer module buffers data for a simulation step. The transfer module can receive and send data concurrently and translation can be performed while waiting for the slowest simulator.

3.4.3 Performance tests

The performance tests are realised with time recorders integrated at specific places in the code. These times are aggregate duration to evaluate the running time of the co-simulation in each section. This allows evaluating the time of "simulation", "IO" and "wait" time. Each test is done for 10 trials of 1 second of simulated time for asynchronous configurations with one or two parameters which vary per trial. The results of the trials are averaged to reduce the variability of the measurements. The varied parameters of the tests are the number of spiking neurons, synchronized time between simulators and the configuration of MPI and thread of NEST.

Figure 3.5 and supplementary figure E.17, E.18 and E.19 show the result of the performance test done on DELL Precision-7540 (Intel Xeon(R) E-2286M CPU 2.40 GHz * 8 cores * 2 threads, 64 GB of Ram with Ubuntu 18.04.5). The communication between components in the transfer module was performed with the multithreading approach. Supplementary figure E.20, E.21 and E.22 are generated using the Jusuf system (<https://apps.fz-juelich.de/jsc/hps/jusuf/cluster/configuration.html>) which is composed of nodes with 2 AMD EPYC 7742 2.25 GHz * 64 cores * 2 threads, 256 (16x16) GB DDR4 with 3200 MHz, connected by InfiniBand HDR100 (Connect-X6). In this second case, the transfer module uses MPI protocol to communicate between components.

3.4.4 Implementation details

The source code of the co-simulation is open-source and contains Python script and C++ files. A singularity and a docker image are also available on singularity-hubs to replicate the figures as in the performance test. The activity diagram (see Supplementary Figure E.5) describes in detail the interaction between each module and components for this specific virtual experimentation.

The implementation of Input and Output for NEST used the existing simulator's architecture and parallelization strategy. NEST has different back-ends for the input and output data, the creation of a new backend for the communication of the data was enough for integration in the co-simulation design pattern (for more details see Supplementary Note E.1.1). For more technical details about the communication with NEST, an activity diagram (see Supplementary Figure E.6) describes the communication protocol with NEST back-end. For this specific example, the states of the wrapper of NEST and the states of transfer components which communicate with NEST are described respectively by the Supplementary Figure E.7 and E.8.

The implementation of Input and Output for TVB is different because TVB does not use MPI for its parallelization and it does not have an interface for exchanging data outside of the simulator. The creation of the interface required a dynamic modification of the simulator engine to integrate and release data from the transformer and a wrapper for communication with the transformer modules (for more details see Supplementary Note E.1.1). For more technical details about the communication with TVB, an activity diagram (see Supplementary Figure E.9) describes the communication protocol with the TVB wrapper. For this specific example, the states of the wrapper of TVB and the states of transfer components which communicate with the wrapper of TVB are described respectively by the Supplementary Figures E.10 and E.11.

The description of the transfer modules is partially described in Supplementary Note E.1.2 which focuses only on the interface with simulators. In addition to this note, the state of the different components are described in the Supplementary Figure E.8, E.11 and E.13. To better understand different instances and classes in this module, Supplementary Figure E.14 describes all the instances and their role and Supplementary Figure E.15 describes the composition of the abstract class and the simple API for communication. The communication protocol for data exchange between transfer module components differs depending on whether the parallelization strategy is multithreading or multiprocessing. In the case of multiprocessing, MPI protocol is used for data exchange. The communication protocol differs depending on the data type, as shown by panel A of Supplementary Figure E.16. The spike trains data are variable and large (it can go from a few Kilobits to more than one Megabit depending on the firing rates). The shared memory is chosen in this case. For the mean rate data, the data size is constant and small (a few Kilobits depending on the number of regions). Send and Receive function of the MPI protocol is chosen in this case. In the case of multithreading, only a shared buffer is used between threads.

3.4.4.1 Deadlock due to global interpreter of python

In the case of multithreading for internal communication in the transfer modules, the program may be in a deadlock because the interface with a simulator does not receive the information of receiving data. As it is explained in the global interpreter lock documentation, "The GIL(global interpreter lock) can cause I/O-bound threads to be scheduled ahead of CPU-bound threads, and it prevents signals from being delivered." (<https://wiki.python.org/moin/GlobalInterpreterLock>). The consequence is that some signals used by MPI are not delivered, which creates a situation where a simulator and a transformer are waiting for an MPI message from the other one, but these messages will never arrive.

3.5. Discussion

The Parallel Co-simulation toolbox presented here provides co-simulation technology linking two simulators operating at two different scales with minimal requirements and modifications. This workflow is based on the cyclic coupling topology of modules [59], i.e each module regularly receives new inputs during the simulation. The two scale-specific simulators are interchangeable due to the genericity of the transfer function, as well as the modularity and design of the transfer module (for more characterization of the workflow, see Supplementary Note E.1.3). The interfaces of the simulators and other modules serve as a software science co-design pattern and can be reused in other studies involving co-simulations.

Our approach separates the theoretical challenge of coupling models at different scales from the technical challenge of coupling the corresponding simulators. The simplicity of the design pattern allows the scientific community to advance their research project without being hindered by technical details. Best practices are advised on carrying out a task or implementing the design pattern. These challenges are not unique to using the Parallel Co-simulation toolbox, but apply to most technical implementations of multiscale modeling software. On the technical side, the design pattern does not provide guidelines for the co-simulation's robustness, management and maintenance, similar to the closely related staged deployment and support software for multiscale simulations developed in EBRAINS (<https://juser.fz-juelich.de/record/850819>). On the conceptual side, for proper use of co-simulation technology, a profound understanding of the involved models is necessary to avoid operating the models outside of their valid parameter ranges. For instance, the Neural Mass model used in this paper cannot capture the fast scale dynamics, especially the fast regimes of regular burst state [33]. In the Neural Mass model's derivation, the input firing rate of the neurons is assumed to be an adiabatic process, which is valid in some parameter regimes, but violated for the irregular synchronous state, in which rapid transitions between low and high firing rates occur. As co-simulation requires an understanding of models typically used in at least two different and non-overlapping fields, particular attention should be paid to the responsible use of multiscale models.

3. Multiscale co-simulation design pattern for neuroscience applications – 3.5. Discussion

Such caution should also be applied here when interpreting the results of the CA1 model and the brain model used in this paper. Current CA1 models are in fact typically more detailed and comprise Gandolfi et al 2022 [131], which contains the large type cells and associated connectivities, or models by Bezaire et al. 2016 [28], by Ferguson et al. 2015 [115] and by Cutsuridis et al. 2010 [71] for replication of theta rhythms produced by CA1. Numerical errors constitute another issue. As these errors cannot be estimated analytically, the alternative solution is to perform a sensitivity analysis or uncertainty quantification to determine whether or not the simulation result is reliable [68, 69].

For the validation of the co-simulation, it is essential to generate data that can be related to real-world observations, as is the case here with the model of the two types of electrodes. A critical issue is the repeatability and reproducibility of the simulations. Repeatability is ensured by managing all the random generators in each simulator and using a single parameter file for the co-simulation setup. For reproducibility, due to the complexity of the network, a table is proposed where the configuration of each simulator is reported with their version and also the description of the transformation modules (see Supplementary Table E.1). A notable property of this design pattern is the independence of its modules and components. This independence allows unit testing for each of them. Our design pattern also requires the implementation of a minimal reusable simulator interface for interaction between simulators. In a possible second stage, this interface can be adapted to a standard to increase the possibility of interaction with other simulators.

EBRAINS provides two solutions for co-simulation [303], that is TVB-Multiscale tool and the here described Parallel Co-simulation toolbox. The two tools implement conceptually and technically two different solutions. The TVB Multiscale tool focuses on user convenience, allowing for rapidly prototyping scientific use cases using a single interface to configure all modules in the co-simulation. It is based on serial approaches for the co-simulation, i.e. each module is run one after the other. The Parallel Co-simulation tool, on the other hand, focuses on optimizing performance. The detailed description, benchmarking and validation of the Parallel Co-simulation toolbox is the topic of the current manuscript. Consequently, the TVB-Multiscale tool is slower. Performance tests show that the various modules run in parallel and adapt to the slowest module (see Performance section). The waiting time of the slowest modules is quasi-null, which means there is no loss of time in the synchronization of modules. Performance in co-simulation is an important criterion, as the microscopic simulators are typically very high-dimensional and hence computationally costly. The serial approaches can be interesting when a computer does not have at least one CPU core per module because, under this condition, modules need to share resources which can slow the co-simulation. This was demonstrated by the large increase of simulation time when the number of virtual processes for NEST is higher than the number of physical CPU cores. The other important distinguishing feature of the two co-simulation toolboxes is the unique interface for multiscale simulations. Similarly to TVB-Multiscale, multiscale simulators have the advantage of having a unique interface for multiscale simulations. This unique interface simplifies the simulation

3. Multiscale co-simulation design pattern for neuroscience applications – 3.5. Discussion

configuration but also reduces the specificity of functionality for each scale, which may be disadvantageous for some situations, such as optimization. For example, in our application, spiking neuron and brain region models require different integrators to avoid numerical errors and enhance efficiency. The CA1 model is a sparsely connected network of thousands of neurons using event communications. The mouse brain is a fully connected network of hundreds of regions based on continuous communications. Consequently, the optimization strategy is different and requires specificity.

Other existing frameworks to deploy and communicate runtime data between simulators comparable to Parallel Co-simulation include the Multi-Simulation Coordinator (MUSIC) [100]. MUSIC does not include modules that facilitate translation between scales, which is needed when coupling simulators on different scales of abstraction. Our design pattern includes parallelized modules and can be extended easily to include new, potentially more computationally intensive, methods in the future. On a more technical level, a second difference is how MUSIC uses the HPC transport protocol Message Passing Interface (MPI) [241]. MUSIC takes ownership of the highest level MPI environment (MPI_COMM_WORLD); this can cause challenges when integrating MUSIC with simulators that expect exclusive ownership of this highest level. Our implementation does not touch this highest-level ownership. We use MPIs client-server functionality to connect between simulators, completely evading this challenge. This difference in MPI usage also allows better use of the HPC scheduling mechanisms as each simulator is deployed in isolation, facilitating optimal workload placement on the hardware available. MUSIC does support several features currently not implemented in our implementation of the design pattern : multi-rate integration, i.e. different frequencies of sending and receiving data from simulators, and it prevents some simulation errors by using the MPI error system. On the other hand, our implementation of the design pattern allows for easy extension with new simulators and better distribution on HPC systems.

Outside of neuroscience, standards exist for co-simulation, such as HighLevel Architecture(HLA) [301] and Functional Mock-up Interface (FMI) [14]. These standards include error management, multi-rate integration, and data management [29, 301]. The main difference between MUSIC and our design pattern with these two standards is the communication strategy between simulators. FMI provides a standard for exchanging models and for scheduled execution [14]. Features of FMI and MUSIC currently not implemented in our design pattern are : Real-time hardware interactions [14, 248]. Additionally, FMI supports signal extrapolation for error reduction [29], although this could be added in the translation modules central in our design pattern. FMI does not support concurrent execution of the different simulators, although internally, the simulations can be parallelized [14] and FMIGo proposes a parallelization implementation of FMI [218]. HLA is designed for distributed systems and provides a standard for data exchange and time management of simulators [255, 147]. HLA facilitates the re-usability and interoperability of simulators and models by describing each component's roles and interactions. It further formalizes the data exchange and coordination between simulators and follows the publish/subscribe pattern. This

standard specifies the definition of information produced or required by simulators. It provides a common data model for the reconciliation of model definitions and interoperability of simulators including during distributed runtime execution. Typically these services are implemented with a centralized communication architecture [147], sometimes described as a hub and spoke model. Both MUSIC and our design pattern use direct, peer-to-peer communication.

In summary, we have presented a new software science co-design pattern of the Parallel Co-simulation tool for coupling simulators with a transformation module. This design pattern provides the first step for developing platforms using transitional scaling models and structuring the future syntactic, semantic and conceptual issues induced by multiscale problems. The optimization for this workflow is based on the communication delay between scales. It is not generalized for all cases but recommended for models with transmission line element method [35] or waveform relaxation method [258].

Acknowledgments

This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 945539 (Human Brain Project SGA3) and Specific Grant Agreement No. 785907 (Human Brain Project SGA2). The authors would like to thank Mario Lavanga for helpful feedback, Ingles Chavez Rolando for technical support and numerous colleagues for comments about the texts and figures. Also, the authors gratefully acknowledge the support team of the supercomputer DEEP-EST and JURECA at Forschungszentrum Jülich and Piz Daint at CSCS - Swiss National Supercomputing Centre.

Code availability

The co-simulation between TVB and NEST is freely available under v2 Apache license at <https://github.com/multiscale-cosim/TVB-NEST> on the branch Paper_TV-NEST and Paper_TV-NEST_with_timer.

A docker container containing the project is freely downloaded on EBRAINS (<https://docker-registry.ebrains.eu/harbor/projects/53/repositories>). The simulated data of the three sets of parameters, the code and the singularity images are available in a Zenodo repository (<http://dx.doi.org/10.5281/zenodo.7259022>).

CHAPTER 4

Synchronization in spiking neural networks with short and long connections and time delays

Contents of Chapter 4

4.1. Chapter overview	95
4.2. Introduction	96
4.3. Modeling the Network of Neurons	97
4.4. Results	100
4.4.1. Independent contributions of LRCs and noise	103
4.4.2. Synergic effect of LRCs and noise	103
4.4.3. Modulation by adaptation current	104
4.4.4. Effect of excitability of the voltage membrane	105
4.4.5. Network effects : balanced network	106
4.4.6. Network effects : network size	108
4.4.7. Topology effects : bi-directional long-range connections	109
4.5. Discussion	109

4.1. Chapter overview

In the previous chapter, we presented the software science co-simulation design pattern. We illustrated by the simulation CA1 region of the cellular-level hippocampus of the mouse embedded into a full brain network. In this chapter, I aim to investigate data transformation from macro- to meso- scale. Synchronization is a fundamental mechanism for information processing in the brain and is affected by time delays. This investigation uses parameter exploration to investigate the reproducibility of synchronization in neural fields in the presence of long-range delayed connections with point neural networks. We demonstrate the interaction of micro-, meso- and macro-scopic parameters for the emergence of spatiotemporal patterns.

This chapter is based on an under-revision article in Chaos :

Synchronization in spiking neural networks with short and long connections and time delays

Lionel Kusch¹, Martin Breyton^{1,2}, Spase Petkoski¹, Viktor Jirsa¹

¹Institut de Neurosciences des Systèmes (INS) UMR1106, Aix-Marseille Univ., Marseille, 13005, France

²Service de Pharmacologie Clinique et Pharmacovigilance, Assistance Publique des Hôpitaux de Marseille, Marseille, 13005, France

Abstract

Synchronization in networks is fundamental for information processing in oscillatory brain networks and is strongly affected by time delays via signal propagation along long fibers. Their effect however is less evident in spiking neural networks given the discrete nature of spikes. To bridge the gap between these different modeling approaches, we study the synchronization conditions, dynamics underlying synchronization, and the role of the delay of a two-dimensional network model composed of adaptive exponential integrate-and-fire neurons. Through parameter exploration of neuronal and network properties, we map the synchronization behavior as a function of unidirectional long-range connection and the microscopic network properties and demonstrate that the principal network behaviors comprise standing or traveling waves of activity and depend on noise strength, E/I balance, and voltage adaptation, which are modulated by the delay of the long-range connection. Our results show the interplay of micro- (single neuron properties), meso- (connectivity and composition of the neuronal network), and macroscopic (long-range connectivity) parameters for the emergent spatiotemporal activity of the brain.

4.2. Introduction

Brain activity is characterized by rich spatiotemporal dynamics across scales. Among those, synchronization [350] between the rhythms of neuronal populations [45], has been shown to be relevant for brain function [127] and dysfunction [25]. Different spatial patterns of synchronization have been hallmarked both during tasks [206] and resting state [76, 232], and through various neuroimaging modalities such as fMRI (functional Magnetic Resonance Imaging), MEG/EEG (Magneto-Electro-Encephalography), and ECoG (Electrocorticography). Similarly, neurons are known to synchronize locally as seen in Local Field Potentials (LFP) through short-range connections, also in relation with brain function [21]. However, if macroscale rhythms emerge from the microscopic spiking neuronal activity [268, 260], the interplay between the two is often poorly understood.

Generative computational models have been commonly used [83, 155, 300, 324] to understand how the brain's anatomical structure, as defined by its connectome [151], impacts the coherent macroscopic patterns of brain activity. Here, we explore how emergent dynamics depend not only on the weights of the connectome but also on the time it takes to propagate through the fibers [223, 270, 136]. Together they define a space-time structure that is determinant for the network behavior, including resonance, synchronization, and emergence of brain rhythms [271].

The effects of network heterogeneity, including that of the time-delays, were studied on the macroscopic level [318, 273, 272], as well as for the combined macro- and mesoscopic representations [187, 189, 184, 188, 332] using a neural field approach, including both global and local connectivities [300, 282, 327]. However, neural fields in those cases were either modeled by mean-field representations of the neuronal masses, or

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.3. Modeling the Network of Neurons

by phenomenological neuronal models that correspond to reduced representations of Hodgkin-Huxley neurons. For the case of integrate-and-fire neuron models, there is a lack of precise and self-consistent mathematical framework to link microscopic and macroscopic dynamics within neural field theory [66]. In particular, the verification and validation of macroscopic phenomena and mechanisms emerging from spiking neuronal networks are missing.

In this study, we focus on the emergence of synchronization in neural fields coupled through long-range connections (LRCs) with finite transmission velocity [184, 188, 187, 191, 214, 376, 197]. We use a methodology similar to Jirsa and Stephanescu 2011 [191] with a more detailed and physiological model of neurons. This step is critical for synchronization of the network, as the previously used simplistic neuron models embodied strong slow attractive manifolds, which are known to be the key determinant in network synchronization, but are not physiological. As a consequence, the synchronization results reported in the literature may not always translate into real-world brain networks. To address this, we first present a detailed description of the network, then present results illustrating the link between synchronization and parameters of the LRCs as well as the effect of a noise drive. Next, we identify the robustness of this phenomenon for different categories of neurons by changing parameters of the voltage membrane (voltage reset, refractory time and heterogeneity) and parameters of the adaptation current (sub-threshold adaptation, spike-triggered adaptation and time constant adaptation current). Finally, we pinpoint network characteristics responsible for the emergence of synchronization by exploring the synaptic weight, the number of neurons and the directionality of the LRCs.

4.3. Modeling the Network of Neurons

We consider a two-dimensional grid network consisting of 200 cells (10x20), each with a size of 3x3 millimeters, and two boundaries with periodic conditions (see Figure 4.1 A) topologically equivalent to a torus (see the supplementary Figure F.1). Neurons are distributed homogeneously on the grid with the only constraint that each cell contains a fixed number of excitatory (160) and inhibitory (40) neurons (see Figure 4.1 B). All neurons were constructed identically with an adaptive exponential integrate and fire model [37] with alpha conductance synapses without delay. Only the post-synaptic weights were different between excitatory and inhibitory neurons.

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.3. Modeling the Network of Neurons

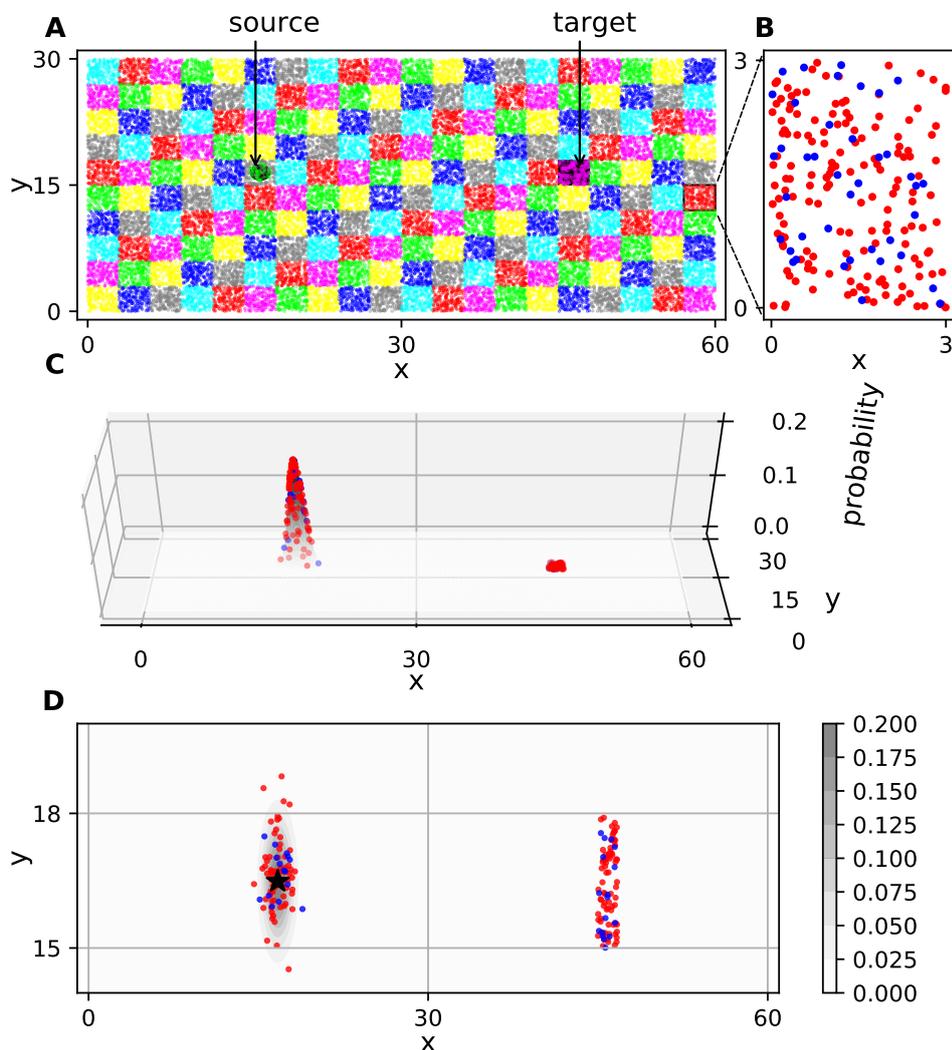


FIGURE 4.1. – Network topology

(a) The network topology is based on a two-dimensional grid network consisting of 200 cells (10x20), each with a size of 3x3 millimeters. The black surface represents the default long connections between cells, with uni-directional connections from the center of the source cell to the target cell. (b) Each cell contains 140 excitatory neurons (red) and 60 inhibitory neurons (blue), randomly positioned. (c) An example of the connected neurons of a neuron in the center of the source population is shown. The gradient of gray is a two-dimensional Gaussian probability function that defines the probability of neighborhood neurons being connected to the neuron. The red neurons are excitatory, and the blue neurons are inhibitory. The neurons on the right part of the grid represent connections due to the long-range connections. (d) is an enlargement of the center part of the grid, which shows the same connected neuron as in c. The position of the neurons is represented by a star, with different levels of gray representing the probability distribution of local connections around the neuron.

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.3. Modeling the Network of Neurons

The following set of equations describes the network :

$$\begin{aligned}
C_m \times \frac{dV_m}{dt} &= -g_L(V_m - E_L) + g_L \times \Delta_T \times e^{\frac{V_m - V_{th}}{\Delta_T}} \\
&\quad - g_e(t)(V_m - E_{ex}) - g_i(t)(V_m - E_{in}) \\
&\quad - w + I_e + \mathcal{N}(0.0, \sigma^2) \\
\tau_w \times \frac{dw}{dt} &= (a \times (V_m - E_L) - w) \\
g_e(t) &= \sum_{t_j^{(f)}} w_j \times \text{alpha}(t - t_j, \tau_{ex}) \text{ with } w_j > 0.0 \\
g_i(t) &= \sum_{t_j^{(f)}} w_j \times \text{alpha}(t - t_j, \tau_{in}) \text{ with } w_j < 0.0 \\
\text{alpha}(t, \tau) &= \frac{t}{\tau} \times e^{1 - \frac{t}{\tau}} \times \text{Heaviside}(t)
\end{aligned} \tag{4.1}$$

where V_m is the membrane voltage, w the adaptation current, g_e the excitatory synaptic conductance and g_i the inhibitory synaptic conductance of one neuron. $t_j^{(f)}$ represents spike-time from presynaptic neurons. When the membrane potential crosses a threshold V_{peak} , the membrane voltage is reset (V_{reset}), and the adaptive current is increased by a value b (for more detail, see the Supplementary Table F1). Each neuron has two external inputs : I_e , corresponding to a constant external current, and \mathcal{N} , corresponding to white noise. I_e has a default value of 0.0 pA, except for the case of heterogeneous population where the distribution over the population follows a Gaussian distribution centered in zero. The initial condition is chosen to obtain an asynchronous irregular state in disconnected networks (see the supplementary figure E2). The choice of adaptive exponential integrate and fire model is motivated by its capability of reproducing most neural firing patterns [178, 344], which led to multiple studies describing its dynamics [344, 254, 220]. Other models, such as Conductance-based Adaptive Exponential model [143], Multi-Timescale Adaptive Threshold model [200], Generalize integrate and fire neuron [43] or Hodgkin-Huxley neurons [162], also have this variety of dynamics, but they are harder to analyze due to a larger number of parameters or missing knowledge about them. Other models, such as leaky integrate and fire [221, 347] or FitzHugh-Nagumo [119, 253], are simpler and have been studied with a similar network topology [191, 355].

The neurons in the network are divided into excitatory and inhibitory categories based on the sign of their post-synaptic weights. Each neuron projects probabilistic short-range connections around it following a two-dimensional Gaussian distribution (see Equation 2). Projections of excitatory neurons decay faster ($\sigma = 0.8$) than those of inhibitory neurons ($\sigma = 1.2$) [191] (see figure 4.1 C and 4.1 D).

$$p(d_x, d_y) = e^{-\frac{(d_x - \mu_x)^2 - (d_y - \mu_y)^2}{2\sigma^2}} \tag{4.2}$$

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.4. Results

where (d_x, d_y) is the distance along the x and y axis between two neurons, and, m_x and m_y their position on the grid. To achieve an asynchronous irregular state [41] in a noise-driven balanced network of neurons, the weight of the excitatory synapses was set to 10.0 nS and the ratio (excitatory/inhibitory) to 0.11 through parameter explorations (see Supplementary Figures F25 and F26). LRCs are defined by a source cell and a target cell. In the source cell, on top of the short-range connections, 40% of excitatory neurons project to the distant target following a two-dimensional Gaussian probability density ($\sigma_{long} = 2.0$) (see figure 4.1 C and 4.1 D) centered in the target cell. The weight of the LRCs is independent of the local ones (by default : 15.0 nS), and their delay is uniform between neurons and independent of the distance.

The synchronization of the network due to LRCs can be quantified using the coefficient of variation of the firing rate (FR), as proposed by Brunel (2000) [41]. This metric is calculated by dividing the standard deviation of the firing rate (number of spikes per time bin of 3 ms) by its mean over a period of 10 s (see Equation 3).

$$CV_{FR} = \frac{\sigma_{FR}}{\mu_{FR}} \quad (4.3)$$

A CV_{FR} value higher than 1.0 between two neurons indicates that they are synchronized and asynchronous otherwise. To reduce the bias induced by initial conditions, an initial transient period of 12 s is discarded in each simulation. Each is run only once for a specific parameter set. The time-average order-parameter was also used to measure synchronization (see Supplementary Note F3 for details).

4.4. Results

We seek to recover and understand the conditions in which a network with white noise input is synchronized in the presence of long-range connections (LRCs) with transmission delays and to characterize the spatiotemporal dynamics of the network. To this end, we conducted analyses of parameter spaces of network synchronization as a function of weight and delay of unidirectional LRCs, excitatory weights and variance of white noise.

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.4. Results

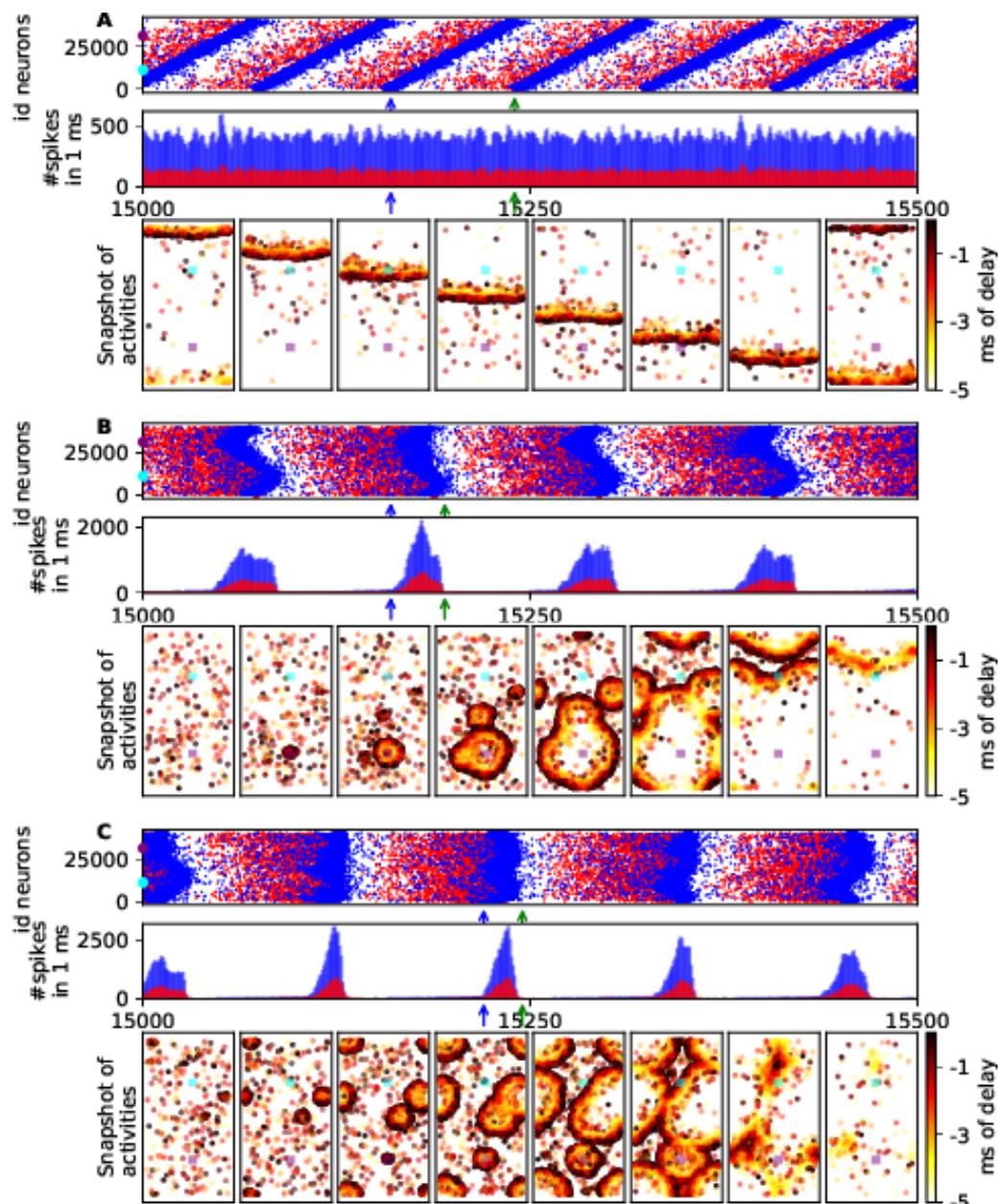


FIGURE 4.2. – Three characteristic network dynamics

A network of 40000 neurons was simulated with white noise, synaptic weights of 10 nS, and different delays of the uni-directional long-range connections (a : 40 ms, b : 80 ms, c : 100 ms). The index is ordered by cells of the grid, with excitatory neurons followed by inhibitory neurons. Each simulation is represented by spike trains and an associated histogram (bin : 1 ms) between 15.0 s and 15.5 s. The blue and green arrows indicate the eight regular snapshots' start and end. On the histogram and eight snapshots, blue and purple represent the source and the target population. (a) traveling wave; (b) regular bump from the target population with some other bumps, and (c) multiple bumps at the time.

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.4. Results

Time series analysis reveals that the synchronization spectrum is associated with corresponding spatiotemporal activity patterns. On one side of this spectrum (Figure 4.2A, corresponding to mark A in Figure 4.3B), the network is asynchronous, and the associated spatiotemporal activity is a traveling wave along the longest axis of the grid. As expected, the histogram is flat, and the spike trains forms a diagonal of spikes due to the ordering of the neuron. At the opposite side of this spectrum (Figure 4.2C corresponding to mark C in Figure 4.3B), the network is synchronous, and its activity alternates between bursts of activation and quiet periods. In this case, activity bursts are very short (less than 20 ms) and composed of multiple bumps starting from the target cell and from other random locations due to the noise. The eight snapshots indicate that when two waves collide, they collapse. Between these two extremes, spatiotemporal patterns present a gradient where synchronization is linked to shorter activity bursts and a higher number of bumps from random locations (see Figure 4.2B corresponding to mark B in Figure 4.3B).

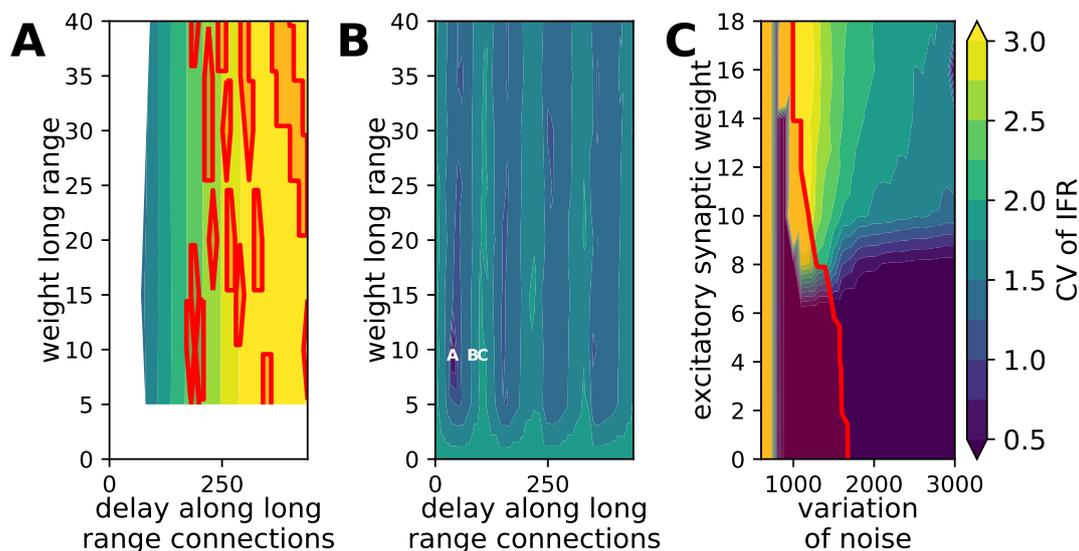


FIGURE 4.3. – Competing network oscillation from noise and long-range connections The network synchronization is quantified by the coefficient of variation of firing rate (no overlapping window of 3 ms) for 10 s after a transition period of 12 s. Figures (a) and (b) show the parameter space analysis of network synchronization without and with noise (1800pA) for different delays versus weights of long-range connections. Figure (c) displays the parameter space analysis of network synchronization for different variances of the white noise and the excitatory synaptic weights. The parameters set chosen for the simulations shown in Figure 4.2 is indicated by (a), (b), (c).

4.4.1 Independent contributions of LRCs and noise

When noise is turned off, the network is generally synchronized, as evidenced by the coefficient of variation systematically higher than 1.0 (Figure 4.3A). Time series (Figure F5) are characterized by periodic bumps starting from the target cell, forming a circular wave, and eventually colliding with itself. We find an almost proportional relationship between the level of synchronization and the delay which is also reflected in the mean firing rate and longer silent periods between bursts. This relation is preserved when varying the weight of the LRCs, the number of connections and the position of the target cell (Supplementary Figures F3, F6 and F8). However, the network is silent (white areas) when the weight of the LRCs is lower than 5 nS or the standard deviation of the Gaussian kernel is lower than 0.5, or the delay is lower than 70 ms.

When the LRCs are removed, the parameter space (Figure 4.3C and Supplementary Figure F10) shows that the network is synchronized only when noise is strong enough (variance higher than 1500 pA) and if the excitatory synaptic weight is sufficiently large (higher than 8 nS). When these conditions are met, synchronization diminishes as noise increases with little effect of the excitatory synaptic weights. A related phenomenon to this is the appearance of synchronized bumps displaying chaotic behavior [249] (see Supplementary figure F11A and F11B). When the excitatory synaptic weight is too low, below 8 nS, the system remains in an asynchronous regime (Figure F11).

4.4.2 Synergic effect of LRCs and noise

The combination of noise and LRCs results in simulations displaying activity regardless of the parameters (see Figure 4.3B and Supplementary Figures F4, F7 and F9). The level of synchronization depends on the delay along the LRCs, with a regular pattern of higher synchronization occurring at 0, 110, 220 or 330 ms. This indicates inherent oscillations with a period of 110 ms, also confirmed by a mean firing rate value around 1/0.11 Hz, shown in Supplementary Figure F12A. The observed impact of the delay on synchronization is similar to the one found in populations of oscillators [370, 273]. The same effect was also observed for continuous spiking neurons [191].

If the weight of the LRCs or the number of connections is too low, this effect is lost, and the network remains synchronized regardless of the delay. Like before, we found that the position of the target cell has no substantial impact on synchronization. In other words, if LRCs are too low, the noise is the dominant phenomenon for the emergent synchronized activity, Figure 4.3 B-C. Otherwise, LRCs perturb the synchronization, with the ratio delay and mean-firing rate determining the increased/decreased synchronization ranges.

When the synchronization is facilitated, e.g. point C in Figure 4.3B, the wave is homotopic and it propagates faster (Figure 4.2B compared to 4.2C), compared to the case when the synchronization is decreased, e.g. point B in Figure 4.3B. In certain conditions, point A in Figure 4.3B, regular traveling waves appear. In this case, the source and the target are anti-phase (close to half of the synchronized period), and

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.4. Results

the bump does not collapse but continues to travel across the network. It is worth noting that, even though the overall network synchronization is the lowest in this case, Supplementary Figure F26 G, the phase locking is the highest, Supplementary Figure F26H. This is also visible in the activity in Figure 4.2A.

In the following sections, we explore the parameters that can generate network synchronization in the presence of LRCs. We distinguish between individual neuron parameters, such as the refractory time, voltage threshold and external current, and network parameters, such as the number of neurons and the directionality of the LRCs.

4.4.3 Modulation by adaptation current

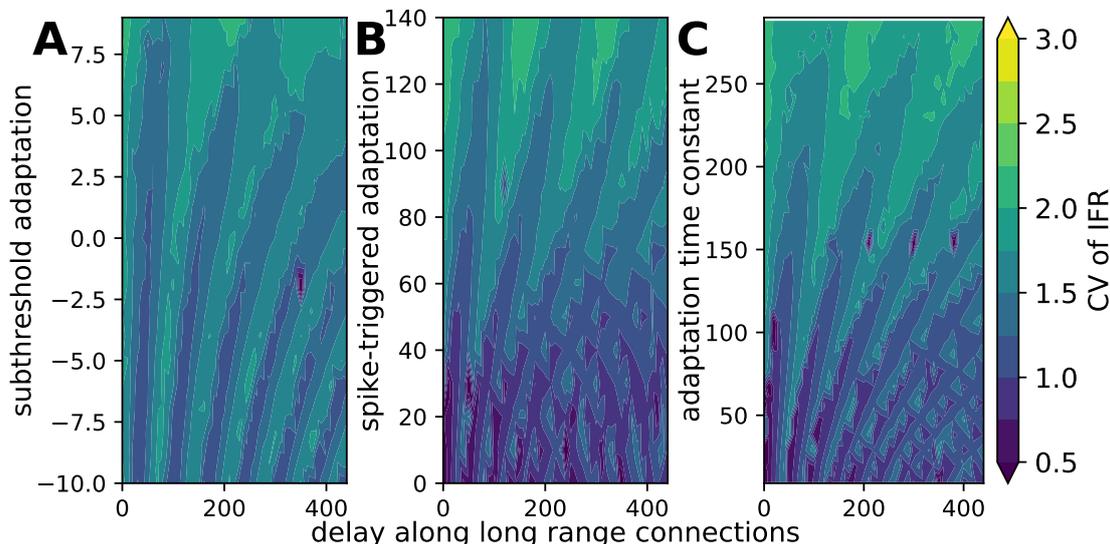


FIGURE 4.4. – Parameter space network synchronization analysis of adaptation current properties in relation to delay of long-range connections

The network synchronization is quantified by the coefficient of variation of firing rate (no overlapping window of 3 ms) for 10 s after a transition period of 12 s. Parameter space network synchronization analysis is presented, showing the relationship between the delay of long-range connections and the adaptation current properties of neurons, including subthreshold adaptation(a), spike-triggered adaptation(b), adaptation time constant(c).

The adaptive current is a mechanism that affects the frequency of the spikes. This impacts the mean-firing rate, which serves as a "natural frequency" of self-sustained network oscillations in the case of synchronization. Consequently, the impact of the time-delays is similar to that of a population of synchronized oscillators [370]. The adaptation has three main parameters : the time constant τ_w , the spike-triggered

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.4. Results

adaptation b and the sub-threshold adaptation a . As seen in Figure 4.4, synchronization can be found for all ranges of these parameters with the same periodic effect of the delay as before. This effect fades out as the adaptation parameters increase, although with a general increase of CV_{FR} and decrease of mean firing rate (see Supplementary Figures F14, F13 and F15).

4.4.4 Effect of excitability of the voltage membrane

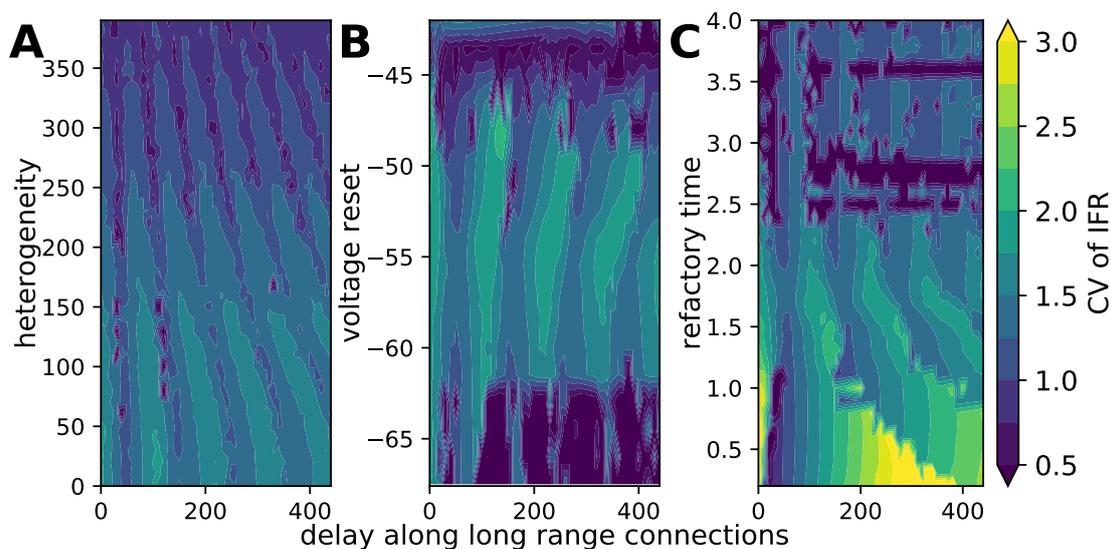


FIGURE 4.5. – Parameter space network synchronization analysis of voltage membrane properties in relation to delay of long-range connections

The network synchronization is quantified by the coefficient of variation of firing rate (no overlapping window of 3 ms) for 10 s after a transition period of 12 s. Parameter space network synchronization analysis is presented, showing the relationship between the delay of long-range connections and the voltage membrane properties of neurons, including heterogeneity (a), voltage reset (b) and refractory time (c).

Excitability of the voltage membrane has an opposite effect to adaptation and facilitates the generation of spikes and we used the external current I_e to create a heterogeneity of excitability within the population (see section Modeling the Network). As seen in Figure 4.5A, the network can be synchronized for all the heterogeneity values explored. The increase in heterogeneity reduces the coefficient of variation of the firing rate (CV_{FR}) associated with an increase in the mean firing rate (see Supplementary Figure F17). Here, the periodic effect of the delay on synchronization is spaced out as heterogeneity decreases. Although not presented here, if heterogeneity is too strong, it can also affect the balance of the network and push it into an asynchronous irregular state equivalent to the previous results on excitatory synaptic weights.

Regarding voltage reset, synchronization is mainly present between -63 mV and

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.4. Results

-45 mV and almost constant within this range for a given delay (see Figure 4.5B and Supplementary Figure F.18); otherwise the network is mostly asynchronous. This separation of regimes can be explained by a strong non-linear relationship between the mean firing rate and the voltage reset (Figure F.16D) and because the coefficient of variation is not suited to capture synchronization for extreme values of mean firing rate (highly bursty or very sparse activity). Therefore, stripes of synchronized activity still exist within the asynchronous areas.

When refractory time is varied (Figure 4.5C), the parameter space is separable in two subspaces : under and over 1.7 ms, approximately. For low refractory times, more than 95% of the neurons have irregular burst activities (Figure F.19). In this case, reducing refractory time increases the variation of the CV_{FR} proportionally to the delay. However, large values of CV_{FR} (above 3) capture the presence of large waves (see Supplementary Figure F.20A). When the refractory period exceeds 1.7 ms, neurons regularly spike with few bursts with little to no effect of the delay along the LRCs. In this scenario, very low values of CV (below 0.5), capture traveling wave activity (see Supplementary Figure F.20C).

4.4.5 Network effects : balanced network

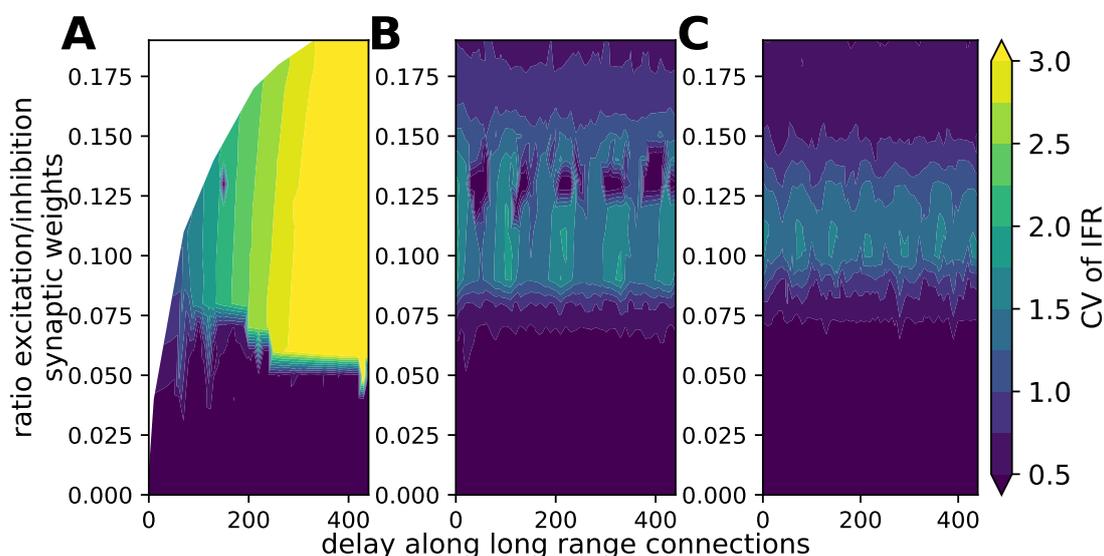


FIGURE 4.6. – Parameter space network synchronization analysis of ratio excitatory/inhibitory synaptic weight versus delay of long-range connection for different values of noise

The network synchronization is quantified by the coefficient of variation of firing rate (no overlapping window of 3 ms) for 10 s after a transition period of 12 s. This figure shows the parameter space network synchronization analysis of delay of the long-range connection versus ratio excitatory/inhibitory synaptic weight for different values of noise ((a) : 0.0pA, (b) : 1800.0pA, (c) : 2600.0pA).

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.4. Results

As seen before, for synchronization to occur the network needs to allow for wave propagation drawing mainly from a specific balance between excitation and inhibition [313]. Other factors, such as the number of neurons and the synaptic weights themselves can also affect network synchronization [191]. Plots in Figure 4.6 show the effect of the relative strength between excitatory and inhibitory synaptic weights on synchronization for different noise intensities (0 pA, 1800 pA and 2600 pA). Without noise, the results are similar to those of the parameters of the LRCs (Figure 4.3) with the same proportional relationship. The main difference with Figure 4.3 is the area of asynchronous activity for a low ratio (approximately under 0.06). This area is independent of the noise, and simulations in it have mean firing rates over 50 Hz (see Supplementary Figures E.25, E.26 and E.27). When noise is introduced, there appear three main regimes : one asynchronous regime where the network is dominated by inhibitory neurons (bottom), another asynchronous regime where excitatory neurons dominate the network (top), and in-between a synchronous "balanced" regime [153, 87]. The latter corresponds to a specific range of ratio between inhibitory and excitatory synaptic weights. The increase in noise strength narrows down this range of synchronization.

4.4.6 Network effects : network size

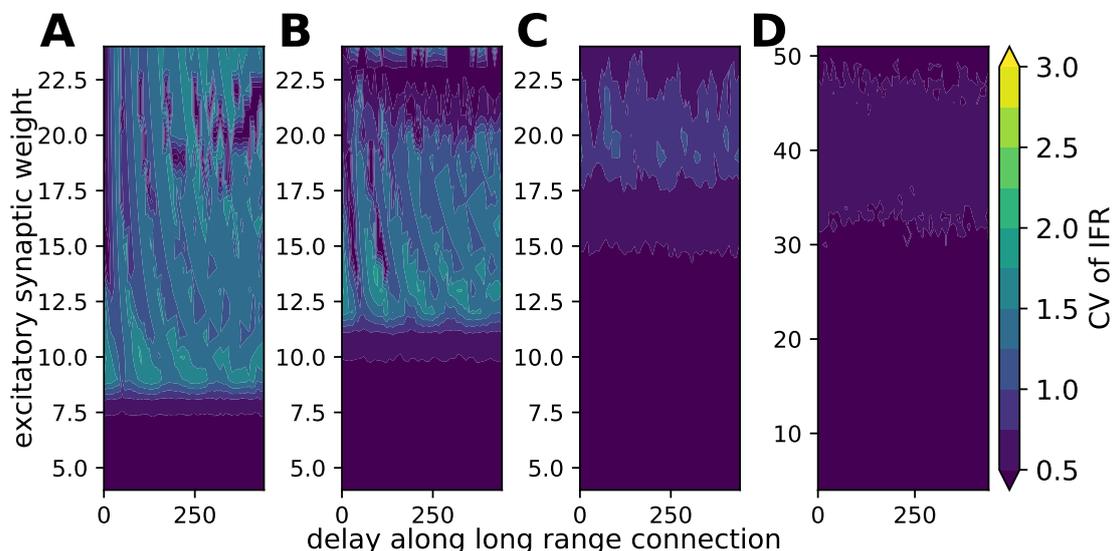


FIGURE 4.7. – Parameter space network synchronization analysis of synaptic weights in relation to delay of long-range connections for different sizes of the network

The network synchronization is quantified by the coefficient of variation of firing rate (no overlapping window of 3 ms) for 10 s after a transition period of 12 s. Parameter space network synchronization analysis is presented, showing the relationship between the delay of the long-range connection and the synaptic weight for different percentages of active neurons (**A** :100%, **B** :75%, **C** :50%, **D** :25%)

To explore the effect of the number of neurons on synchronization, we varied the percentage of active neurons in the network. The increase of synaptic weights can compensate for the reduction in the number of neurons to achieve the same mean firing rate in the network [7] (see Supplementary Figures [F21](#), [F22](#), [F23](#) and [F24](#)). As seen in Figure 4.7, the sparsening of neurons is associated with an increase in the nominal values of excitatory synaptic weights to reach synchronization. However, when the number of neurons is too low (less than 50%) increase in excitation can no longer make up for the loss of activity.

In addition, we explored different grid sizes without changing the density of the neurons and found invariant results except for small grids (less than 5 cells), see Supplementary figure [F32](#).

4.4.7 Topology effects : bi-directional long-range connections

We showed previously (see Figure F29A and 4.3B) that in the presence of unidirectional LRCs, varying the weights had little to no effect on synchronization for any fixed delay. However, when the LRCs become bidirectional, synchronization is lost an LRCs weight higher than 4 nS(F29 B).

In the absence of synchronization, activity resembles that of 4.2A with an unstable traveling wave across the network. When tuned near synchronization, we again find bumps of activity, but this time originating from both sides of the LRCs, Supplementary Figure F30B and C. If the weight of the LRCs is increased above 4 nS (from point C to D in F29 C), bumps of activation appear with origins alternating between both ends of the LRCs, with roughly one-third of them never propagating through the network. The principal cause of the disappearance of the periodic dependence on the delay in the case of bidirectional LRCs is that regardless of the noise level, the network is not synchronized. Still, a traveling wave occurs (see Supplementary Figure F28A), while the same condition for a unidirectional LRCs would have caused synchronization instead (see Supplementary Figure F28B).

4.5. Discussion

The present study analyzes the conditions for synchronization of a two-dimensional surface of adaptive exponential integrate and fire neurons [37], including long-range connections (LRCs) with transmission delay. Specifically, we identified a periodic relationship between synchronization and the delay along the LRCs, which is generally regular, depending on other conditions that we analyzed. These findings agree with previous studies which validated this phenomenon in neural fields derived from spiking neural networks [191], but also with the wider literature showing Arnold's tongues [275] for the synchronization, depending on the ratio of the natural frequency and time-delays [370, 273].

We discovered four main conditions for the modulation of synchronization by the LRCs in the network. First, the spontaneous behavior of the network with LRCs is to be synchronized at a frequency proportional to the delay along the LRCs. Adding an external input (here in the form of random noise) is essential for synchronization and to find a more complex relationship between synchronization and the delay. The second condition is that the strength of the LRCs must be strong enough (as measured by weight and width of the kernel) to have an impact on the network. However, when the LRCs become bidirectional, synchronization can no longer occur if their contribution is too strong. The third condition is that the network must be balanced between inhibition and excitation [153, 87, 313] for the LRCs to modulate synchronization. Finally, the number of neurons must be large enough [376] or sufficiently connected for this phenomenon to occur.

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.5. Discussion

By exploring neuron parameters, we have indirectly explored different types of neurons. We started with neurons coined adapting neurons in sub-threshold activities [178]. They correspond to type II neuron models (as shown in Supplementary Figure F16C) exhibiting resonator regimes [344] and type I Phase Response Curve (PRC) [220]. When modifying neuronal properties, we changed firing rate dynamics, neuron type (I/II), voltage membrane regime, and PRC (see Supplementary Figures F16 F12). For instance, when the sub-threshold is shifted to low values, the adapting neuron in sub-threshold activities becomes a type I neuron with integrator regimes. Interestingly, we found that the type of neuron is not a significant condition for the effect of the LRCs on synchronization. Furthermore, setting the spike-triggered adaptation to 0 pA makes the neuron model equivalent to an exponential integrate and fire neuron without adaptive current and we found that results also hold for this type of neuron. This provides evidence that mean-field models derived from different types of neurons would still behave according to our findings. Moreover, since for every value of the delay along the LRCs, there is at least one set of neuron parameters (i.e. a type of neuron) that will allow network synchronization to emerge, the properties of neurons could theoretically be inferred from the knowledge about the LRCs.

In terms of generative mechanisms, we identified that bumps are the main cause of synchronization captured by the CV in the network, and that it corresponds to a burst of activity where many neurons are triggered collectively in a short period of time. These bursts are generated cyclically and resemble the generative process of action potentials unfolding in a sequence of voltage accumulation, threshold crossing, spike triggering, and followed by a refractory period. In the case where the noise drive is high enough to trigger a neuronal spike at the single neuron level, local synaptic currents will increase through the synaptic weights and are mediated by the local ratio between inhibitory and excitatory neurons. After accumulation, the mean voltage becomes large enough to generate circular waves. If the network is homogeneous, the probability of these events forming anywhere on the grid is uniform, potentially generating multiple waves from different locations at the same time. When two of them meet, they collapse followed by a silent period due to the refractory time and the local increase of adaptation current, preventing neurons from generating new spikes. During that silent time the adaptation current decays until new events occur. As a result, the frequency of bursts depends on a complex interaction between the adaptive current, the synaptic current, voltage reset, refractory time, and the noise.

Lastly, the properties of the neurons and the network are changing the dynamics of the emergent phenomena in the population, such as synchronization and waves as spatiotemporal patterns. This paper identifies relations between the emergent dynamics and the properties of a long-range fiber-like connection. The comprehension and characterization of this self-organization are essential for understanding a more complex system like the brain. The delay and the weights of fibers are essential for network synchronization as well as neuronal properties. In particular, the fluctuation of adaptive current and voltage reset can impact the synchronization of a subset of neurons [78], i.e., the emergent rhythm of the neuronal population. Additionally, wave propagation is essential for synchronization as an emergent dynamic presented in this

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.5. Discussion

study. We corroborated previous findings that local modifications such as a change in the excitation/inhibition balance or a loss of neurons can disrupt higher-order phenomena potentially responsible for brain dysfunction.

The main limitation of our work is the choice of the synchronization metrics. Here, synchronization is quantified using the coefficient of variation of firing rate (CV_{FR}), which is less precise in the case of bursts (the cause of variation can be the high activities of a sub-population). Nevertheless, we were able to reproduce all our results using the time-averaged Kuramoto order-parameter [211], as shown in supplementary figures F32-F31.

Among other limitations, we can note that each network configuration was only tested once, thus bringing nuances to the conclusions that can be drawn about multiple regimes potentially coexisting for the same set of parameters. Indeed, some simulations showed transitions between regimes (see Supplementary Figure F33), making it difficult to determine if the simulation represents a complete stationary regime or only one regime in a more complex landscape. However, simulations were run long enough to reach a steady regime whenever it existed. A complete exploration of dynamical regimes would have been tedious and not the focus of this study.

Lastly, the analysis was not fully exhaustive; synaptic parameters such as Gaussian connectivity profiles [371], time constant or reversal potentials, for example, were not explored because we chose to focus on neuronal properties important for mean-field derivations.

In summary, the synchronization of a network of biological neurons with long-range connections and finite transmission speeds is modulated by the delay of these connections. The main conditions for this effect are the pre-existence of spontaneous activity driven by an external input (or noise), a significant contribution of the LRCs, a balanced network (excitation/inhibition), and a large number of neurons. Given that our objective was to provide evidence for the validity of neural field models, future work could include testing the relaxation of neural field model assumptions (e.g., plasticity, delays, and weight distributions) and using multiple heterogeneous neuron models. Additionally, further analysis with bidirectional LRCs is needed to understand spatiotemporal patterns in the presence of two LRCs. This analysis should consider larger grids or modified connectivity, as recent findings suggest that traveling waves are based on a fraction of neurons in a population [78]. Validation of this phenomenon with more recent neural field models (e.g., Zerlaut et al. 2018 [374] or Augustin et al. [18]) is also recommended.

Acknowledgments

We acknowledge the use of Fenix Infrastructure resources, which are partially funded from the European Union's Horizon 2020 research and innovation programme through the ICEI project under the grant agreement No. 800858. This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 945539 (Human Brain Project

4. Synchronization in spiking neural networks with short and long connections and time delays – 4.5. Discussion

SGA3) and Specific Grant Agreement No. 785907 (Human Brain Project SGA2) and from the Agence Nationale de la Recherche (ANR) under the grant ANR-17-CE37-0001 - CONNECTOME. The authors gratefully acknowledge the support team of the supercomputer Piz Daint at CSCS - Swiss National Supercomputing Centre.

Data and materials availability

The code for the simulation and analysis of this paper is freely available under v2 Apache license at https://bitbucket.org/lionelkusch/torus_long_range_connection/src/paper/.

Each exploration's analysis results are registered in databases available on Zenedo (<http://dx.doi.org/10.5281/zenodo.7629156>) with the associate code. Supplementary Note [E2](#) describes the contents of them.

CHAPTER 5

Discussion

In this thesis, we have introduced a method for coupling simulators and an analysis of the reproducible phenomena captured by some mean field using a neural network. In this discussion, I shall demonstrate and contextualize the relationship between these two scientific results.

5.1. Method for co-simulation

The core of this thesis is the multiscale brain simulations based on a co-simulator. Chapter 3 describes a software science co-design pattern, i.e. a set of structural blueprints for building simulations composed of multiple simulators [102]. This co-design pattern is illustrated with the coupling of NEST [98], a simulator of large-scale point neuron network, and The Virtual Brain(TVB) [299, 300], a simulator of Neural Mass network, for the simulation of a virtual mouse brain at the meso- and macro- scales. One attractive characteristic of this co-design pattern is the conservation of the simulators' scalability, i.e. the capability to parallelize and distribute (see Chapter 2) the simulation computation. Moreover, the separation of technical and scientific issues related to multiscale co-simulation is kept within the organization of the co-simulation framework by using independent specialized modules with direct communication. The core principle of this method is the creation of transfer modules dedicated to transforming the data between scales and synchronizing different simulators. I want to stress that the transformation between continuous signals (firing rate) and discrete events (spike trains) is a non-trivial and non-generic challenge[39, 141].

5.1.1 Putting the co-design model into perspective using a holistic framework

To improve the comprehension of this co-design pattern, I shall use the holistic framework of co-simulation proposed by Nguyen et al. 2017 [258], which has five conceptual layers of structuration : conceptual layer (architecture), semantic layer (formal integration), syntactic layer, dynamical layer (execution and synchronization) and technical layer.

The conceptual layer is about the meta-modeling process and the topology of the

framework. The software science co-design pattern presented here addresses the topology of the co-simulation. The meta-modeling is indirectly addressed by having designed each simulator to handle models at a specific scale and having built a transfer module using models to transform data between scales.

The semantic layer defines the role and the signification of each model, as well as their interaction. This work does not address this layer; it only encapsulates each model in a simulator and their interaction in a transformation function without a clear definition of them. In the future, additional research should be done to clarify the different notions and create an ontology to reduce the number of homonyms between the vocabulary used to describe and analyze each scale. For example, "frequency of a brain region" is ambiguous in neural networks because it can be the mean firing rate or the frequency of fluctuation of the population mean firing rate.

The syntactic layer is the harmonization of the difference of formalization between models or simulators. This is not addressed in this work. However, in neuroscience, large projects, such as Human Brain Project[11], or organizations, such as INCF[360], participate in the elaboration of multiple standards such as PYNN [79], ConGen [156], Sonata [75]. In the future, this standard can be used for parameterized simulations and standardize the interaction between modules.

The dynamic layer is the execution of co-simulation and the synchronization of the simulators during the co-simulation. This work used the transformation module to synchronize the different simulators without using centralized communications. The solution proposed in this work is flexible and adapted for two simulators. However, the synchronization and optimization between two simulators is already a difficult problem and the difficulty can increase exponentially with the number of simulators. Consequently, this solution is perhaps unsuitable for a higher number of simulators. The technical layer is related to the implementation and evaluation of co-simulation. This work addresses this layer by using Message Passing Interface [342] and simulator wrapper for communication between simulators and to keep simulators' scalability. The only technical evaluation realized here is the performance of co-simulation. In the future, the robustness and the accuracy of co-simulation, as well as the re-ability and stability of the framework, should be assessed.

To summarize, this work targets the conceptual, dynamic, and technical layers. The semantic and the syntactic layers are not directly addressed and are hidden in the transfer modules.

5.2. Multiscale modeling

To fill the lack of the semantic layer, i.e. the lack of gathering models, we provide analyses of the interdependence between models of each scale. Before presenting the result of these analyses, I shall examine the concept of a "multiscale model".

5.2.1 The concept

"Multiscale model" does not have a common and consensual definition [163]. The only consensus is that such a model is based on information from at least two scales, either temporal or spatial[59]. To understand these disagreements over the definition, it is necessary to consider the different existing multiscale modeling approaches : top-down, bottom-up, and mixed approaches [238]

The top-down approach is the study of a higher layer to infer underlying mechanisms at a lower scale. In this case, models aim to capture only the relevant information to describe a phenomenon, as seen for example in Neural Mass models (see section 1.4). A phenomenological model captures emergent dynamics without necessarily considering the underlying scale. For example, the Epileptor [190] captures the propagation of an epileptic seizure from a dynamic point of view, but it does not explicitly describe the biological mechanisms generating this dynamic. Biophysical models extract the relevant information of the lower scale for describing the emergent phenomena, i.e., the lower scale properties constraining the dynamic. They are more interpretable than phenomenological models but still suffer from a loss of information from the lower scale due to mathematical approximations and assumptions in their derivation. These limitations narrow conclusions to only potential mechanisms, and further validation is required to distinguish an artifact from the loss of information or a real emergent dynamic.

The bottom-up approach is the study of an emergent phenomenon from a lower scale. In this case, models often comprise a network of sub-models and are used to identify and characterize an emergent dynamic of the system. The limitation of this approach is the complexity of the model, the ambiguity of the results, and the extensive usage of computational resources. The model must include most details to be realistic and avoid missing important components [238], at the cost of increasing its complexity. The result of analyses can be ambiguous since including all these details can make it difficult to disentangle the central mechanism from the background noise. Additionally, the increase in details requires more computation for simulations.

The mixed approach uses bottom-up and top-down at the same time. This approach is more difficult because it requires knowledge of the two approaches and additional models for transforming data from one scale to another. However, it can take advantage of the two approaches and helps to clarify the interaction between higher and lower scales. In this approach, multiscale becomes the art of making relevant decisions, as indicated in Chopard et al. 2014 [59] "The art of multiscale modeling is then to propose a good compromise between CPU performance and accuracy by selecting the most relevant parts of the domain at an appropriate scale."

In conclusion, the disagreement over the definition of a "multiscale model" is because different approaches have been used to study emerging phenomena without specifying which or how many scales. In this thesis, we have occasionally used several definitions of this concept. Still, the definition of a "multiscale model" in this discussion is the one of the mixed approach, i.e. a model composed of sub-models at two

scales.

5.2.2 Model for transforming neural data between scale

In a mixed approach, brain multiscale models require a model for brain regions, a model for neurons, and models for connecting the scales. In this work, each model has a dedicated module in the co-design pattern. In this thesis, we restraint the research to one model for brain regions, Mean Ad Ex[356] (see section 1.4.1), and one model for neurons, adaptive exponential integrator and fire neuron [37] (see section 1.2.5). Due to the novelty of the co-simulation method, the models for the transformation of data between scales are not well defined and studied. This work proposes some elementary results for the studies of these models.

5.2.2.1 Downscaling transformation

The downscaling transformation is the transformation of the data from macro-scale (mean firing rate) to meso-scale (spike trains). This type of transformation requires additional information or some assumptions because the macro-scale description is less precise than the meso-scale description. Particularly in neuroscience, this transformation should consider the non-invertibility of the macro-scale description, also called neurodegeneracy. Following the definition, in D'Angelo et al. 2022 [73], neurodegeneracy is "a property of complex systems, such as the brain, in which multiple element combinations and system configurations can explain a specific set of ensemble signals.". As explained, this is one major limitation of the top-down approach for interpreting or downscaling macro-scale activities.

The connection between a Neural Mass and a point neuron network in a brain network model is based on white fibers, which are long-range connections in the brain. Previous results showed the effect of delayed long-range connections on the synchronization in Neural Fields [223, 270, 136, 191]. Consequently, some of these results will be reproduced to parameterize and define a downscaling model. Chapter 4 demonstrates the modulation of synchronization by the delay along long-range connections. These conditions are that the network needs to be synchronized without the long-range connections, that the long-range connections should be strong enough to impact the network, and that the network has to be balanced between excitation and inhibition activities. The properties of the neurons have also been studied and they do not significantly impact the synchronization but rather modulate the range of delays where the network is synchronized. As expected, neurodegeneracy was found, i.e. multiple sets of parameters could generate the same dynamics.

5.2.2.2 Upscaling transformation

The upscaling transformation is the transformation from meso-scale (spike trains) to macro-scale (mean firing rate). This type of transformation is a reduction of the

data from the meso-scale. The complexity and the meaning of the reduction depend on the mean-field method, i.e. the connecting variable is derived from a statistical description of the point neuron network or a phenomenological variable. In the case of phenomenological models, the reduced signal often requires only a scaling operation to the range of value of the phenomenological variables for keeping coherent dynamics, and the interpretability of the connection is non-trivial. The derived model from a point neuron network is often based on a statistical reduction of network activity, generally to the first or second order. In this case, the interdependence between the two scales is explicit and requires a statistical measure of the network activities.

In this thesis, we are in the second case, the mean field is derived from a point neuron network. The Annex A provides the result of a comparison between the Mean Ad Ex [356], and the corresponding network with a constant or oscillating external input is presented. The qualitative comparison established a close match between the mean-field fixed-point and the steady state of the network. Quantitative comparison evaluates a non-uniform difference of a few tens to a few Hz. Additionally, comparing dynamical responses to an oscillatory input between the mean-field and the network provides a difference in the phase shift with respect to the input signal and a difference in the convergence of average oscillations.

To summarize, the work introduces a multiscale co-simulation technology for the macro- and meso- scale of the brain. This co-simulation used multiscale models, which mix the top-down and bottom-up approaches. Multiscale brain modeling has the advantage of integrating knowledge from different scales. Nevertheless, it requires transformation models between scales that are currently being defined. The two main difficulties in these analyses are the neurodegeneracy and the precision of the mean field approximation.

5.3. Limitation

The limitations of these works are already discussed in detail in each chapter. I will present only some additional limitations due to the interweaving of the different previous results.

5.3.1 Limitation of co-design pattern

I want to stress that the co-simulation is not yet user-friendly, and it requires some skills in programming and coding to install and adapt to a specific usage. It is missing support for fixing bugs and helping the users. Also, as demonstrated in the previous section 5.1.1, the semantic and syntactic layers are partially reached in this thesis. To complete this analysis, I will use the framework proposed in Chopard et al. 2014 [59] to highlight some limitations. This framework considers the concepts of scale separation map, generic submodel execution loop, and coupling templates to be sufficient for bridging the gap between application design and computer implementation in the

multiscale modeling language. Regarding the previous holistic co-simulation framework, this framework addresses the concepts layer by defining basic concepts and the semantic and syntactic layers with the proposition of multiscale language.

The scale separation map is a map that places submodels in time and space to highlight the separation of scale between them. In this thesis, I consider multiscale problems with the separation of space into different brain regions with different resolutions, and the time is the same for all scales. Using the scale separation map, the multiscale problem is classified as multidomain with space separation and time overlap.

Using the proposition of coupling template, I can define that the coupling in this thesis is cyclic (the two simulators have multiple interactions during the co-simulation), fixed in synchronization (synchronization points between the two simulators are regular in time and defined before the beginning), and fixed in the number of instances (the number of submodels is not evolving during the co-simulation).

From this characterization, the co-design pattern presented here is dedicated to a specific type of multiscale model. However, the state of the art and the lack of examples in Chopard et al. 2014 for this type of multiscale show the knowledge deficit around this topic.

5.3.2 Should we trust a co-simulation?

A major limitation of the co-simulation is the certainty in the output [69]. Gathering simulators and models of different scales does not necessarily mean the simulation output is still correct. Multiscale co-simulation requires quantifying the model's uncertainty and controlling the numerical errors of the simulator before trusting the output of a co-simulation [69].

For phenomenological models, the validation should at least ensure that the range of input values to the network corresponds with the dynamic range of the phenomenological model. For mean-field derived from a point neuron network, the validations should at least quantify the maximum difference between the mean-field model and the network dynamic in the context of a network of population, as brain network model, and quantify the variability of the measure of the network in the up-scale transformation. Nevertheless, the most important consideration is the consistency between a neural network, mean-fields and transformation models while considering that their assumptions are compatible.

For example, the exposed difference between the Mean Ad Ex and the associate point neuron network (see Annex A) limits the interpretability of the co-simulation output. Additionally, the non-uniform difference between the mean field and the network makes it difficult to define a transformation function to compensate for it. This conclusion can also be generalized to other mean-field with different assumptions, such as the mean-field presented in Cakan et al. 2020 [49]. Furthermore, the analysis of the precision of the measure of the mean firing rate over windows less than 10 s presents variability higher than 1 Hz (see figure G.8) for a network in a steady state. This indi-

cates that estimating the mean firing rate for short windows used in co-simulation will present some variability that can influence the simulation's dynamic.

5.3.3 Modeling limitation

I want to raise awareness about the several assumptions for modeling the brain in this thesis, such as the focus only on neuron cells, the non-consideration of synaptic plasticity, the non-consideration of the distribution of synaptic weights or the heterogeneity of neuron properties, the absence a regional specificity between brain regions and the non-consideration of other scale (molecule interactions, variation of ion concentration, behaviors). Additionally, the work has been focused on specific models to avoid getting lost in the details, which limits its generalizability.

To summarize, the software science co-design pattern is specific to a type of multiscale model. Without quantification of the uncertainty, the output of a co-simulation should not be trusted. Additionally, the brain model used in this thesis is very specific, with many assumptions.

5.4. Perspective

Regardless of this limitation, the scientific context is the main obstruction of multiscale co-simulation usage.

5.4.1 Difficulty of formulating scientific question using co-simulation

The first reason is the difficulty of formulating a multiscale question for co-simulation. Multiscale co-simulation requires a simultaneous study of two scales that a simulation of one specific scale can not answer. This is partially in contradiction with the mean-field approach derived from a point neuron network because it claims that the relevant information of the meso-scale is already captured by it, and it is not necessary to include all other details. At the same time, as a large complex dynamical system, the point neuron network is already difficult to understand without adding a complex input to represent the interaction with the rest of the brain. From these statements, co-simulation of the brain's macro- and meso- scale requires including relevant details not captured by the mean-field model and an analysis that requires going beyond an oscillatory input.

5.4.2 Youth of brain network model

The second reason is that mean-field models are not mature enough. There is as yet no consensus on the relevant features needed to describe the state of a neural network,

and there are a variety of approximations or assumptions about the properties of neural networks without thorough comparison between these approaches [17]. Moreover, there are few studies about the impact of these assumptions and approximation in a network of populations and the quantification of the maximum difference between the neural network and the mean-field approximation. The current mean-field methods in neuroscience do not provide a precise macroscopic description for using them in interconnecting scale models. This is not surprising because brain network modeling has only seven years of application, since the non-invasive measurement of connectome [328, 150] (see section 1.4).

5.4.3 History of co-simulation in neuroscience, a technology-driven proposal

The third reason is that most of the development of co-simulation in neuroscience is mainly a technology-driven proposal without a specific scientific application [134]. We shall learn this from the history of previous developments of co-simulation. The first project of co-simulation was done during the United States' Human Brain Project (1991-2006) [167, 203, 138, 202, 316, 38, 82, 132, 298]. This technical solution is called Neo-sim [137] and was developed for coupling Genesis [34] and NEURON [159], two multi-compartment neuron simulators. As explained in Gewaltig et al. 2014 [134], "The developers departed at the end of the grant with very little to show for their efforts, and none of the software was ever reused. This was possible because NeoSim was a purely technology-driven proposal without a specific scientific application." Ten years later, INCF [360], International Neuroinformatics Coordinating Facility, funded MUSIC [100, 105], Multi-Simulation Coordinator, tools for coordinating co-simulation. MUSIC and it is still under development. Its first application [100] was to link a molecular simulator, MOOSE [291], with the meso-scale simulator NEST [98]. In parallel, the Blue Brain Project developed an interaction between simulators such as NeoCortical Simulator [101] and NEURON [159], or STEPS 4.0 [57] and NEURON. Some year later, the Human Brain Project [235, 234, 12, 13, 11] (see subsection 2.2.1) has developed different solutions in addition to this work responding to various usage : TVB-multiscale [303] (see Annex C) is a co-simulator between macro-scale and meso-scale with a vision centered around the macro-scale, i.e. the co-simulation is view has an extension of TVB; Neurorobotics platform [199, 113, 8, 114] is a platform which couples multiple simulators such as NEST [98], TVB [299], Gazebo [201] and open-Sim [88]. This platform was created to study the embodiment of brain models and the development of robots. Most of this co-simulation work is being integrated into EBRAINS [99, 310] (see subsection 2.2.2). These different co-simulation solutions use different technologies, depending on the scientific and software context. The common denominator is that they have been developed in large-scale projects, but their scientific benefits have yet to be demonstrated, except for the Neurorobotics platforms [20, 175].

5.4.4 Missing of transdisciplinarity in neuroscience

The fourth reason is the lack of pluralism in neuroscience, as explained by KraKauer et al. 2017 [205]. Pluralism means considering different points of view on the same phenomenon, which implies accepting the explanation of diversity. My personal take is that neuroscience is missing transdisciplinary neuroscientists. To develop this argument, I will use part of the argumentation in Loubser 2015 [228], which describes how to become a transdisciplinary theologian but with conclusions not restricted to theology that can apply to all disciplines. I'll assume that theology and neuroscience have in common that their different disciplines share different beliefs to develop the following.

Loubser's main argument is that Paul Cilliers' description of complexity [60], rooted in a post-foundationalist epistemology, can facilitate transdisciplinarity. Loubser introduces transdisciplinarity using the definition proposed by Basarab Nicolescu : "transdisciplinarity concerns the dynamics generated by the action of several levels of reality at once" [259], which is different from multidisciplinary and interdisciplinary approaches. For example, modeling in neurosciences is a multidisciplinary field, as this work demonstrates by situating itself at the intersection of physics, computer science and biology, but is not transdisciplinary research.

Loubser proposes that transdisciplinary epistemology and complexity epistemology are linked because both disciplines can be understood as mutually traversing networks [265] and general systems theory attempts to reconnect disciplines through a "transversal language" [247]. Using this vision, it is possible to propose a model, reduction or interpretation of this system, i.e. the network of disciplines. Disciplines are linked by boundaries that influence the conduct of transdisciplinary studies. The particularity of disciplinary boundaries is that they constantly evolve, forming part of the system while at the same time describing it.

Modeling a network of disciplines is a tool for making the complex interaction between disciplines intelligible. It requires decisions, such as which minimal information to keep, and values, such as the legitimate equality of different disciplines without hierarchy. Woerman and Cillier 2012 [366] argue that personal lives shape modeling choices and judgments, and modelers are responsible for these choices and judgments. However, choices are complicated because complex systems are not made up of the sum of their parts, but also of emergent behavior due to interactions between their components. For better judgment, Loubser recommends understanding the context and history of disciplines, as well as knowledge about knowers. For a transdisciplinary researcher, having a fruitful conversation with knowers requires knowledge of the knower's ethics, epistemology and words, as well as some skill in empathy.

I would say that the ill-defined boundary between macro- and meso-scale brain modeling, and the competition between these two communities, makes it difficult to carry out transdisciplinary studies. For example, I was surprised that modelers of different scales disagreed about the ease of use of models and their interests. A mean-field modeler may consider that simulating millions of neurons is less important than a medium-field model because it is too complex or chaotic. As a result, they only use

the smaller scale to validate their findings. Conversely, a point neuron modeler may consider that a mean-field model loses too much detail or that the approximation is too broad to be worth using and/or examining its conclusions. As a result, they only use models on a larger scale to pre-parameterize their networks. In addition, some experimentalists may consider that using a computer model in neuroscience is irrelevant because, in their view, mathematical models are irrelevant.

To summarize, co-simulation in neuroscience was developed in the context of the interoperability of simulators supported inside a scientific project or international federation. This interoperability was guided by developing technical solutions to support and sometimes demonstrate the interaction between scientific communities. Currently, scientific communities do not use them, which explains the lack of ontology common between communities. One possibility is that our understanding of the brain is not mature enough to use this technology, i.e. scientific research does not require models with this level of complexity. Either the Neural Mass models are not sufficiently accurate or do not capture the relevant information about the emerging dynamics.

5.5. Summary

The main contribution of this thesis is the development of co-simulation between macro and meso scales and the analysis of these two scales to link them. However, this tool uses models with several limitations that limit its use and interest. The main limitations are the lack of a methodology for quantifying co-simulation uncertainty, numerical error and consistency between meso-scale, macro-scale and transformation models. As Gewaltig et al. 2014 [134] mention, "In such projects [capability-driven software projects], developers need almost superhuman prescience if they want to build something useful." I hope that my collaborators and the scientific vision of HBP offered me a sense of anticipation to deliver a valuable tool despite the current lack of scientific interest.

Bibliographie

- [1] *5 Parts of a Computer | Different Main & Basic Components | Names & Info*. en-US. URL : <https://www.idtech.com/blog/parts-of-a-computer> (visité le 01/06/2023) (cf. p. 61).
- [2] J. H. AHRENS et U. DIETER. « Computer Generation of Poisson Deviates from Modified Normal Distributions ». en. In : *ACM Transactions on Mathematical Software* 8.2 (juin 1982), p. 163-179. ISSN : 0098-3500, 1557-7295. DOI : [10.1145/355993.355997](https://doi.org/10.1145/355993.355997). URL : <https://dl.acm.org/doi/10.1145/355993.355997> (visité le 23/01/2021) (cf. p. 328).
- [3] Nora Abi AKAR, Ben CUMMING, Vasileios KARAKASIS et al. « Arbor — A Morphologically-Detailed Neural Network Simulation Library for Contemporary High-Performance Computing Architectures ». In : *2019 27th Euromicro International Conference on Parallel, Distributed and Network-Based Processing (PDP)*. ISSN : 2377-5750. Fév. 2019, p. 274-282. DOI : [10.1109/EMPDP.2019.8671560](https://doi.org/10.1109/EMPDP.2019.8671560) (cf. p. 68, 69, 74, 226).
- [4] Sadaf ALAM, Javier BARTOLOME, Sanzio BASSINI et al. « Fenix : Distributed e-Infrastructure Services for EBRAINS ». en. In : *Brain-Inspired Computing*. Sous la dir. de Katrin AMUNTS, Lucio GRANDINETTI, Thomas LIPPERT et al. Lecture Notes in Computer Science. Cham : Springer International Publishing, 2021, p. 81-89. ISBN : 978-3-030-82427-3. DOI : [10.1007/978-3-030-82427-3_6](https://doi.org/10.1007/978-3-030-82427-3_6) (cf. p. 68).
- [5] Sadaf R. ALAM, Javier BARTOLOME, Sanzio BASSINI et al. « Archival Data Repository Services to Enable HPC and Cloud Workflows in a Federated Research e-Infrastructure ». In : *2020 IEEE/ACM International Workshop on Interoperability of Supercomputing and Cloud Technologies (SuperCompCloud)*. Nov. 2020, p. 39-44. DOI : [10.1109/SuperCompCloud51944.2020.00012](https://doi.org/10.1109/SuperCompCloud51944.2020.00012) (cf. p. 68).
- [6] Sadaf R. ALAM, Javier BARTOLOME, Michele CARPENE et al. « Fenix : a Pan-European federation of supercomputing and cloud e-infrastructure services ». In : *Communications of the ACM* 65.4 (mars 2022), p. 46-47. ISSN : 0001-0782. DOI : [10.1145/3511802](https://doi.org/10.1145/3511802). URL : <https://dl.acm.org/doi/10.1145/3511802> (visité le 12/06/2023) (cf. p. 68).

- [7] Sacha Jennifer van ALBADA, Moritz HELIAS et Markus DIESMANN. « Scalability of Asynchronous Networks Is Limited by One-to-One Mapping between Effective Connectivity and Correlations ». In : *PLOS Computational Biology* 11.9 (1^{er} sept. 2015), e1004490. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1004490](https://doi.org/10.1371/journal.pcbi.1004490). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1004490> (visité le 17/04/2019) (cf. p. 108).
- [8] Ugo ALBANESE, Kenny SHARMA, Benedikt FELDOTTO et al. *HBP Neurorobotics Platform*. Avr. 2020. DOI : [10.5281/zenodo.3763356](https://doi.org/10.5281/zenodo.3763356). URL : <https://zenodo.org/record/3763356> (visité le 20/07/2023) (cf. p. 120).
- [9] Christoffer G. ALEXANDERSEN, Chloe DUPRAT, Aitakin EZZATI et al. « A mean-field to capture asynchronous irregular dynamics of conductance-based networks of adaptive quadratic integrate-and-fire neuron models ». In : (juin 2023). DOI : [10.1101/2023.06.22.546071](https://doi.org/10.1101/2023.06.22.546071). URL : <https://doi.org/10.1101/2023.06.22.546071> (cf. p. 185).
- [10] Katrin AMUNTS, Markus AXER, Lise BITSCH et al. *The coming decade of digital brain research - A vision for neuroscience at the intersection of technology and computing*. eng. Rapp. tech. Zenodo, mars 2023. DOI : [10.5281/zenodo.7764003](https://doi.org/10.5281/zenodo.7764003). URL : <https://zenodo.org/record/7764003> (visité le 12/06/2023) (cf. p. 23, 68).
- [11] Katrin AMUNTS, Javier DEFELIPE, Cyriel PENNARTZ et al. « Linking Brain Structure, Activity, and Cognitive Function through Computation ». en. In : *eNeuro* 9.2 (mars 2022). Publisher : Society for Neuroscience Section : Theory/New Concepts. ISSN : 2373-2822. DOI : [10.1523/ENEURO.0316-21.2022](https://doi.org/10.1523/ENEURO.0316-21.2022). URL : <https://www.eneuro.org/content/9/2/ENEURO.0316-21.2022> (visité le 12/06/2023) (cf. p. 23, 66, 114, 120).
- [12] Katrin AMUNTS, Christoph EBELL, Jeff MULLER et al. « The Human Brain Project : Creating a European Research Infrastructure to Decode the Human Brain ». en. In : *Neuron* 92.3 (nov. 2016), p. 574-581. ISSN : 0896-6273. DOI : [10.1016/j.neuron.2016.10.046](https://doi.org/10.1016/j.neuron.2016.10.046). URL : <https://www.sciencedirect.com/science/article/pii/S0896627316307966> (visité le 09/08/2022) (cf. p. 23, 66-68, 120).
- [13] Katrin AMUNTS, Alois C. KNOLL, Thomas LIPPERT et al. « The Human Brain Project—Synergy between neuroscience, computing, informatics, and brain-inspired technologies ». en. In : *PLOS Biology* 17.7 (juill. 2019). Publisher : Public Library of Science, e3000344. ISSN : 1545-7885. DOI : [10.1371/journal.pbio.3000344](https://doi.org/10.1371/journal.pbio.3000344). URL : <https://journals.plos.org/plosbiology/article?id=10.1371/journal.pbio.3000344> (visité le 12/06/2023) (cf. p. 23, 66, 120).
- [14] ANDREAS JUNGHANNS, CLÁUDIO GOMES, CHRISTIAN SCHULZE et al. « The Functional Mock-up Interface 3.0 - New Features Enabling New Applications ». In : *Proceedings of 14th Modelica Conference 2021, Linköping, Sweden, September 20-24, 2021*. 14th Modelica Conference 2021 (20-24 sept. 2021). Sous la dir. de Martin SJÖLUND, Lena BUFFONI, Adrian POP et al. Linköping, 27 sept. 2021,

- p. 17-26. DOI : [10.3384/ecp2118117](https://doi.org/10.3384/ecp2118117). URL : <https://ecp.ep.liu.se/index.php/modelica/article/view/178> (visité le 08/02/2022) (cf. p. 92).
- [15] A. A. ANDRONOV et A. A. VITT. *Theory of Oscillators*. Pergamon, 1966. ISBN : 978-0-08-009981-1. URL : <https://www.biblio.com/book/theory-oscillators-andronov-aa-aa-vitt/d/1513746515> (visité le 14/03/2023) (cf. p. 28, 29).
- [16] Mojtaba Madadi ASL, Alireza VALIZADEH et Peter A. TASS. « Decoupling of interacting neuronal populations by time-shifted stimulation through spike-timing-dependent plasticity ». en. In : *PLOS Computational Biology* 19.2 (fév. 2023). Publisher : Public Library of Science, e1010853. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1010853](https://doi.org/10.1371/journal.pcbi.1010853). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1010853> (visité le 26/04/2023) (cf. p. 53).
- [17] Moritz AUGUSTIN, Josef LADENBAUER, Fabian BAUMANN et al. « Low-dimensional spike rate models derived from networks of adaptive integrate-and-fire neurons : Comparison and implementation ». en. In : *PLOS Computational Biology* 13.6 (juin 2017), e1005545. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1005545](https://doi.org/10.1371/journal.pcbi.1005545). URL : <http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1005545> (visité le 20/02/2018) (cf. p. 55, 57, 120).
- [18] Moritz AUGUSTIN, Josef LADENBAUER et Klaus OBERMAYER. « How adaptation shapes spike rate oscillations in recurrent neuronal networks ». English. In : *Frontiers in Computational Neuroscience* 7 (2013). ISSN : 1662-5188. DOI : [10.3389/fncom.2013.00009](https://doi.org/10.3389/fncom.2013.00009). URL : <https://www.frontiersin.org/articles/10.3389/fncom.2013.00009/full> (visité le 23/02/2018) (cf. p. 55, 111).
- [19] Daniele AVITABILE, Joshua L. DAVIS et Kyle WEDGWOOD. « Bump Attractors and Waves in Networks of Leaky Integrate-and-Fire Neurons ». en. In : *SIAM Review* 65.1 (fév. 2023), p. 147-182. ISSN : 0036-1445, 1095-7200. DOI : [10.1137/20M1367246](https://doi.org/10.1137/20M1367246). URL : <https://epubs.siam.org/doi/10.1137/20M1367246> (visité le 24/04/2023) (cf. p. 53).
- [20] Jyotika BAHUGUNA, Philipp WEIDEL et Abigail MORRISON. « Exploring the role of striatal D1 and D2 medium spiny neurons in action selection using a virtual robotic framework ». en. In : *European Journal of Neuroscience* 49.6 (mars 2019), p. 737-753. ISSN : 0953-816X, 1460-9568. DOI : [10.1111/ejn.14021](https://doi.org/10.1111/ejn.14021). URL : <https://onlinelibrary.wiley.com/doi/10.1111/ejn.14021> (visité le 22/08/2023) (cf. p. 120).
- [21] SN BAKER, JM KILNER, EM PINCHES et al. « The role of synchrony and oscillations in the motor output ». In : *Experimental brain research* 128 (1999), p. 109-117 (cf. p. 96).

- [22] Pavan BALAJI, Darius BUNTINAS, David GOODELL et al. « PMI : A Scalable Parallel Process-Management Interface for Extreme-Scale Systems ». en. In : *Recent Advances in the Message Passing Interface*. Sous la dir. de David HUTCHISON, Takeo KANADE, Josef KITTLER et al. T. 6305. Series Title : Lecture Notes in Computer Science. Berlin, Heidelberg : Springer Berlin Heidelberg, 2010, p. 31-41. DOI : [10.1007/978-3-642-15646-5_4](https://doi.org/10.1007/978-3-642-15646-5_4). URL : http://link.springer.com/10.1007/978-3-642-15646-5_4 (visité le 06/07/2021) (cf. p. 66).
- [23] Abhirup BANDYOPADHYAY, Giovanni RABUFFO, Carmela CALABRESE et al. « Mean-field approximation of network of biophysical neurons driven by conductance-based ion exchange ». In : (nov. 2021). DOI : [10.1101/2021.10.29.466427](https://doi.org/10.1101/2021.10.29.466427). URL : <https://doi.org/10.1101/2021.10.29.466427> (cf. p. 185).
- [24] B. BARBOUR et M. HÄUSSER. « Intersynaptic diffusion of neurotransmitter ». eng. In : *Trends in Neurosciences* 20.9 (sept. 1997), p. 377-384. ISSN : 0166-2236. DOI : [10.1016/s0166-2236\(96\)20050-5](https://doi.org/10.1016/s0166-2236(96)20050-5) (cf. p. 50).
- [25] Fabrice BARTOLOMEI, Patrick CHAUVEL et Fabrice WENDLING. « Epileptogenicity of brain structures in human temporal lobe epilepsy : A quantified study from intracerebral EEG ». In : *Brain* 131.7 (2008), p. 1818-1830. ISSN : 14602156. DOI : [10.1093/brain/awn111](https://doi.org/10.1093/brain/awn111) (cf. p. 96).
- [26] Lucas BENEDICIC, Felipe A. CRUZ, Alberto MADONNA et al. « Sarus : Highly Scalable Docker Containers for HPC Systems ». en. In : *High Performance Computing*. Sous la dir. de Michèle WEILAND, Guido JUCKELAND, Sadaf ALAM et al. Lecture Notes in Computer Science. Cham : Springer International Publishing, 2019, p. 46-60. ISBN : 978-3-030-34356-9. DOI : [10.1007/978-3-030-34356-9_5](https://doi.org/10.1007/978-3-030-34356-9_5) (cf. p. 64).
- [27] Emiliano BETTI, Marco CESATI, Roberto GIOIOSA et al. « A global operating system for HPC clusters ». en. In : *2009 IEEE International Conference on Cluster Computing and Workshops*. New Orleans, LA, USA : IEEE, août 2009, p. 1-10. ISBN : 978-1-4244-5011-4. DOI : [10.1109/CLUSTR.2009.5289191](https://doi.org/10.1109/CLUSTR.2009.5289191). URL : <https://ieeexplore.ieee.org/document/5289191/> (visité le 01/06/2023) (cf. p. 61).
- [28] Marianne J BEZAIRE, Ivan RAIKOV, Kelly BURK et al. « Interneuronal mechanisms of hippocampal theta oscillations in a full-scale model of the rodent CA1 circuit ». In : *eLife* 5 (23 déc. 2016). Sous la dir. de Frances K SKINNER. Publisher : eLife Sciences Publications, Ltd, e18566. ISSN : 2050-084X. DOI : [10.7554/eLife.18566](https://doi.org/10.7554/eLife.18566). URL : <https://doi.org/10.7554/eLife.18566> (visité le 09/04/2023) (cf. p. 91).
- [29] Torsten BLOCKWITZ, Martin OTTER, Johan AKESSON et al. « Functional Mockup Interface 2.0 : The Standard for Tool independent Exchange of Simulation Models ». In : *9th International MODELICA Conference, Munich, Germany*. Nov. 2012, p. 173-184. DOI : [10.3384/ecp12076173](https://doi.org/10.3384/ecp12076173). URL : https://ep.liu.se/en/conference-article.aspx?series=eep&issue=76&Article_No=17 (visité le 10/05/2023) (cf. p. 92).

- [30] Mark BLYTH, Ludovic RENSON et Lucia MARUCCI. *Tutorial of numerical continuation and bifurcation theory for systems and synthetic biology*. 12 août 2020. arXiv : 2008.05226 [q-bio]. URL : <http://arxiv.org/abs/2008.05226> (visité le 20/02/2023) (cf. p. 30).
- [31] Luca L. BOLOGNA, Roberto SMIRIGLIA, Dario CURRERI et al. « The EBRAINS NeuroFeatureExtract : An Online Resource for the Extraction of Neural Activity Features From Electrophysiological Data ». In : *Frontiers in Neuroinformatics* 15 (2021). ISSN : 1662-5196. URL : <https://www.frontiersin.org/articles/10.3389/fninf.2021.713899> (visité le 11/04/2023) (cf. p. 34).
- [32] F. S. BORGES, P. R. PROTACHEVICZ, E. L. LAMEU et al. « Synchronised firing patterns in a random network of adaptive exponential integrate-and-fire neuron model ». In : *Neural Networks* 90 (juin 2017), p. 1-7. ISSN : 0893-6080. DOI : 10.1016/j.neunet.2017.03.005. URL : <http://www.sciencedirect.com/science/article/pii/S0893608017300588> (visité le 22/02/2018) (cf. p. 54, 55, 277, 321).
- [33] Sami El BOUSTANI et Alain DESTEXHE. « A Master Equation Formalism for Macroscopic Modeling of Asynchronous Irregular Activity States ». en. In : (2009), p. 55 (cf. p. 55, 58, 90).
- [34] James M. BOWER et David BEEMAN. *The Book of GENESIS*. New York, NY : Springer, 1998. DOI : 10.1007/978-1-4612-1634-6. URL : <http://link.springer.com/10.1007/978-1-4612-1634-6> (visité le 21/04/2023) (cf. p. 31, 74, 120).
- [35] R BRAUN et P KRUS. « Multi-threaded distributed system simulations using the transmission line element method ». In : *SIMULATION* 92.10 (1^{er} oct. 2016). Publisher : SAGE Publications Ltd STM, p. 921-930. ISSN : 0037-5497. DOI : 10.1177/0037549716667243. URL : <https://doi.org/10.1177/0037549716667243> (visité le 28/10/2021) (cf. p. 93).
- [36] Connor BRENNAN et Alexander PROEKT. « A quantitative model of conserved macroscopic dynamics predicts future motor commands ». en. In : *eLife* 8 (juill. 2019), e46814. ISSN : 2050-084X. DOI : 10.7554/eLife.46814. URL : <https://elifesciences.org/articles/46814> (visité le 09/08/2019) (cf. p. 54).
- [37] Romain BRETTE et Wulfram GERSTNER. « Adaptive Exponential Integrate-and-Fire Model as an Effective Description of Neuronal Activity ». In : *Journal of Neurophysiology* 94.5 (1^{er} nov. 2005), p. 3637-3642. ISSN : 0022-3077. DOI : 10.1152/jn.00686.2005. URL : <http://www.physiology.org/doi/10.1152/jn.00686.2005> (visité le 15/02/2018) (cf. p. 47-49, 77, 86, 97, 109, 116, 172, 230, 231, 279, 326).

- [38] J. F. BRINKLEY et C. ROSSE. « Imaging and the Human Brain Project : A Review ». en. In : *Methods of Information in Medicine* 41.04 (2002), p. 245-260. ISSN : 0026-1270, 2511-705X. DOI : [10.1055/s-0038-1634485](https://doi.org/10.1055/s-0038-1634485). URL : <http://www.thieme-connect.de/DOI/DOI?10.1055/s-0038-1634485> (visité le 09/08/2022) (cf. p. 120).
- [39] David BROMAN, Lev GREENBERG, Edward A. LEE et al. « Requirements for hybrid cosimulation standards ». en. In : *Proceedings of the 18th International Conference on Hybrid Systems : Computation and Control*. Seattle Washington : ACM, avr. 2015, p. 179-188. ISBN : 978-1-4503-3433-4. DOI : [10.1145/2728606.2728629](https://doi.org/10.1145/2728606.2728629). URL : <https://dl.acm.org/doi/10.1145/2728606.2728629> (visité le 11/07/2023) (cf. p. 113).
- [40] C. G. BROYDEN. « The Convergence of a Class of Double-rank Minimization Algorithms 1. General Considerations ». In : *IMA Journal of Applied Mathematics* 6.1 (mars 1970), p. 76-90. ISSN : 0272-4960. DOI : [10.1093/imamat/6.1.76](https://doi.org/10.1093/imamat/6.1.76). URL : <https://doi.org/10.1093/imamat/6.1.76> (visité le 20/06/2023) (cf. p. 176).
- [41] Nicolas BRUNEL. « Dynamics of Sparsely Connected Networks of Excitatory and Inhibitory Spiking Neurons ». en. In : *Journal of Computational Neuroscience* 8.3 (mai 2000), p. 183-208. ISSN : 0929-5313, 1573-6873. DOI : [10.1023/A:1008925309027](https://doi.org/10.1008925309027). URL : <https://link-springer-com.lama.univ-amu.fr/article/10.1023/A:1008925309027> (visité le 09/02/2018) (cf. p. 54, 56, 100, 276, 277, 321).
- [42] Nicolas BRUNEL et Vincent HAKIM. « Population Density Model ». In : *Encyclopedia of Computational Neuroscience*. Springer New York, 2015, p. 2447-2465. DOI : [10.1007/978-1-4614-6675-8_74](https://doi.org/10.1007/978-1-4614-6675-8_74). URL : https://doi.org/10.1007/978-1-4614-6675-8_74 (cf. p. 185).
- [43] Nicolas BRUNEL, Vincent HAKIM et Magnus J. E. RICHARDSON. « Firing-rate resonance in a generalized integrate-and-fire neuron with subthreshold resonance ». In : *Physical Review E* 67.5 (19 mai 2003), p. 051916. ISSN : 1063-651X, 1095-3787. DOI : [10.1103/PhysRevE.67.051916](https://doi.org/10.1103/PhysRevE.67.051916). URL : <https://link.aps.org/doi/10.1103/PhysRevE.67.051916> (visité le 06/12/2022) (cf. p. 99).
- [44] A. N. BURKITT. « A Review of the Integrate-and-fire Neuron Model : I. Homogeneous Synaptic Input ». In : *Biological Cybernetics* 95.1 (juill. 2006), p. 1-19. ISSN : 0340-1200, 1432-0770. DOI : [10.1007/s00422-006-0068-6](https://doi.org/10.1007/s00422-006-0068-6). URL : <http://link.springer.com/10.1007/s00422-006-0068-6> (visité le 18/07/2022) (cf. p. 48, 55).
- [45] György BUZSÁKI. *Rhythms of the Brain*. en. Oxford University Press, oct. 2006. ISBN : 978-0-19-530106-9. DOI : [10.1093/acprof:oso/9780195301069.001.0001](https://doi.org/10.1093/acprof:oso/9780195301069.001.0001). URL : <https://academic.oup.com/book/11166> (visité le 03/05/2023) (cf. p. 54, 96).

- [46] György BUZSÁKI et Kenji MIZUSEKI. « The log-dynamic brain : how skewed distributions affect network operations ». In : *Nature reviews. Neuroscience* 15.4 (avr. 2014), p. 264-278. ISSN : 1471-003X. DOI : [10.1038/nrn3687](https://doi.org/10.1038/nrn3687). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4051294/> (visité le 26/04/2023) (cf. p. 54).
- [47] Caglar CAKAN, Cristiana DIMULESCU, Liliia KHAKIMOVA et al. « Spatiotemporal Patterns of Adaptation-Induced Slow Oscillations in a Whole-Brain Model of Slow-Wave Sleep ». In : *Frontiers in Computational Neuroscience* 15 (2022). ISSN : 1662-5188. URL : <https://www.frontiersin.org/articles/10.3389/fncom.2021.800101> (visité le 09/05/2023) (cf. p. 55, 58).
- [48] Caglar CAKAN, Nikola JAJCAY et Klaus OBERMAYER. « neurolib : A Simulation Framework for Whole-Brain Neural Mass Modeling ». In : *Cognitive Computation Online* (oct. 2021). ISSN : 1866-9964. DOI : [10.1007/s12559-021-09931-9](https://doi.org/10.1007/s12559-021-09931-9) SMASH. URL : <https://doi.org/10.1007/s12559-021-09931-9%20SMASH> (cf. p. 74, 226).
- [49] Caglar CAKAN et Klaus OBERMAYER. « Biophysically grounded mean-field models of neural populations under electrical stimulation ». en. In : *PLOS Computational Biology* 16.4 (avr. 2020). Publisher : Public Library of Science, e1007822. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1007822](https://doi.org/10.1371/journal.pcbi.1007822). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007822> (visité le 31/07/2023) (cf. p. 55, 58, 118).
- [50] Cristiano CAPONE, Matteo di VOLO, Alberto ROMAGNONI et al. « State-dependent mean-field formalism to model different activity states in conductance-based networks of spiking neurons ». In : *Physical Review E* 100.6 (déc. 2019). Publisher : American Physical Society, p. 062413. DOI : [10.1103/PhysRevE.100.062413](https://doi.org/10.1103/PhysRevE.100.062413). URL : <https://link.aps.org/doi/10.1103/PhysRevE.100.062413> (visité le 19/05/2020) (cf. p. 171).
- [51] M. CARLU, O. CHEHAB, L. DALLA PORTA et al. « A mean-field approach to the dynamics of networks of complex neurons, from nonlinear Integrate-and-Fire to Hodgkin–Huxley models ». In : *Journal of Neurophysiology* 123.3 (mars 2020). Publisher : American Physiological Society, p. 1042-1051. ISSN : 0022-3077. DOI : [10.1152/jn.00399.2019](https://doi.org/10.1152/jn.00399.2019). URL : <http://journals.physiology.org/doi/full/10.1152/jn.00399.2019> (visité le 14/10/2022) (cf. p. 171, 172, 185).
- [52] Nicholas T. CARNEVALE et Michael L. HINES. *The NEURON Book*. Cambridge : Cambridge University Press, 2006. ISBN : 978-0-521-84321-8. DOI : [10.1017/CB09780511541612](https://doi.org/10.1017/CB09780511541612). URL : <https://www.cambridge.org/core/books/neuron-book/7C8D9BD861D288E658BEB652F593F273> (visité le 29/11/2021) (cf. p. 68, 74, 226).
- [53] Ralph H. CASTAIN, David SOLT, Joshua HURSEY et al. « PMix : process management for exascale environments ». en. In : *Proceedings of the 24th European MPI Users' Group Meeting*. Chicago Illinois : ACM, sept. 2017, p. 1-10. ISBN : 978-1-4503-4849-2. DOI : [10.1145/3127024.3127027](https://doi.org/10.1145/3127024.3127027). URL : <https://doi.org/10.1145/3127024.3127027>

- [//dl.acm.org/doi/10.1145/3127024.3127027](https://doi.org/10.1145/3127024.3127027) (visité le 08/06/2023) (cf. p. 66).
- [54] John L. CASTI. « On System Complexity : Identification, Measurement, and Management ». In : *Complexity, Language, and Life : Mathematical Approaches*. Sous la dir. de John L. CASTI et Anders KARLQVIST. Biomathematics. Berlin, Heidelberg : Springer, 1986, p. 146-173. ISBN : 978-3-642-70953-1. DOI : [10.1007/978-3-642-70953-1_6](https://doi.org/10.1007/978-3-642-70953-1_6). URL : https://doi.org/10.1007/978-3-642-70953-1_6 (visité le 17/04/2023) (cf. p. 46).
- [55] Marco CATANI, Robert J. HOWARD, Sinisa PAJEVIC et al. « Virtual in Vivo Interactive Dissection of White Matter Fasciculi in the Human Brain ». en. In : *NeuroImage* 17.1 (sept. 2002), p. 77-94. ISSN : 1053-8119. DOI : [10.1006/nimg.2002.1136](https://doi.org/10.1006/nimg.2002.1136). URL : <https://www.sciencedirect.com/science/article/pii/S1053811902911365> (visité le 23/05/2023) (cf. p. 57).
- [56] Sandrine CHEMLA, Alexandre REYNAUD, Matteo di VOLO et al. « Suppressive Traveling Waves Shape Representations of Illusory Motion in Primary Visual Cortex of Awake Primate ». en. In : *Journal of Neuroscience* 39.22 (mai 2019), p. 4282-4298. ISSN : 0270-6474, 1529-2401. DOI : [10.1523/JNEUROSCI.2792-18.2019](https://doi.org/10.1523/JNEUROSCI.2792-18.2019). URL : <https://www.jneurosci.org/content/39/22/4282> (visité le 02/12/2019) (cf. p. 58, 171).
- [57] Weiliang CHEN, Tristan CAREL, Omar AWILE et al. « STEPS 4.0 : Fast and memory-efficient molecular simulations of neurons at the nanoscale ». In : *Frontiers in Neuroinformatics* 16 (2022). ISSN : 1662-5196. URL : <https://www.frontiersin.org/articles/10.3389/fninf.2022.883742> (visité le 17/07/2023) (cf. p. 120).
- [58] Ulises CHIALVA, Vicente GONZÁLEZ BOSCA et Horacio G. ROTSTEIN. « Low-dimensional models of single neurons : a review ». In : *Biological Cybernetics* (15 avr. 2023). ISSN : 1432-0770. DOI : [10.1007/s00422-023-00960-1](https://doi.org/10.1007/s00422-023-00960-1). URL : <https://doi.org/10.1007/s00422-023-00960-1> (visité le 24/04/2023) (cf. p. 48).
- [59] B. CHOPARD, Joris BORGDORFF et A. G. HOEKSTRA. « A framework for multi-scale modelling ». In : *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences* 372.2021 (6 août 2014). Publisher : Royal Society, p. 20130378. DOI : [10.1098/rsta.2013.0378](https://doi.org/10.1098/rsta.2013.0378). URL : <https://royalsocietypublishing.org/doi/full/10.1098/rsta.2013.0378> (visité le 29/10/2021) (cf. p. 90, 115, 117, 229).
- [60] Paul CILLIERS. « Why We Cannot Know Complex Things Completely ». In : *Emergence* 4.1/2 (juin 2002). Publisher : Taylor & Francis Ltd, p. 77-84. ISSN : 15213250. DOI : [10.1207/S15327000EM041&2-07](https://doi.org/10.1207/S15327000EM041&2-07). URL : <https://search.ebscohost.com/login.aspx?direct=true&db=bth&AN=7717276&lang=fr&site=ehost-live> (visité le 25/07/2023) (cf. p. 121).

- [61] Paul CILLIERS. « 3.1 Knowing Complex Systems : The Limits of Understanding ». In : *A vision of transdisciplinarity : Laying foundations for a world knowledge dialogue* 43 (2008) (cf. p. 73).
- [62] Wesley CLAWSON, Ana F VICENTE, Maëva FERRARIS et al. « Computing hubs in the hippocampus and cortex ». en. In : *SCIENCE ADVANCES* (2019) (cf. p. 54).
- [63] Robert CLEWLEY. « Hybrid Models and Biological Model Reduction with PyD-STool ». In : *PLOS Computational Biology* 8.8 (9 août 2012). Publisher : Public Library of Science, e1002628. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1002628](https://doi.org/10.1371/journal.pcbi.1002628). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002628> (visité le 21/02/2023) (cf. p. 30).
- [64] COMSOL. *COMSOL Multiphysics Reference Manual*. The COMSOL Inc., 2019 (cf. p. 74).
- [65] *Configuration — JUSUF user documentation documentation*. URL : <https://apps.fz-juelich.de/jsc/hps/jusuf/cluster/configuration.html#hardware-configuration> (visité le 08/06/2023) (cf. p. 63).
- [66] Blake J. COOK, Andre D. H. PETERSON, Wessel WOLDMAN et al. « Neural Field Models : A mathematical overview and unifying framework ». en. In : *Mathematical Neuroscience and Applications* Volume 2 (mars 2022), p. 7284. ISSN : 2801-0159. DOI : [10.46298/mna.7284](https://doi.org/10.46298/mna.7284). URL : <https://mna.episciences.org/7284> (visité le 31/05/2022) (cf. p. 56, 97).
- [67] Steven J. COOK, Charles M. CROUSE, Eviatar YEMINI et al. « The connectome of the Caenorhabditis elegans pharynx ». In : *The Journal of comparative neurology* 528.16 (nov. 2020), p. 2767-2784. ISSN : 0021-9967. DOI : [10.1002/cne.24932](https://doi.org/10.1002/cne.24932). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7601127/> (visité le 23/05/2023) (cf. p. 57).
- [68] Peter V. COVENEY, Derek GROEN et Alfons G. HOEKSTRA. « Reliability and reproducibility in computational science : implementing validation, verification and uncertainty quantification in silico ». In : *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences* 379.2197 (17 mai 2021). Publisher : Royal Society, p. 20200409. DOI : [10.1098/rsta.2020.0409](https://doi.org/10.1098/rsta.2020.0409). URL : <https://royalsocietypublishing.org/doi/10.1098/rsta.2020.0409> (visité le 01/11/2021) (cf. p. 91).
- [69] Peter V. COVENEY et Roger R. HIGHFIELD. « When we can trust computers (and when we can't) ». In : *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences* 379.2197 (17 mai 2021). Publisher : Royal Society, p. 20200067. DOI : [10.1098/rsta.2020.0067](https://doi.org/10.1098/rsta.2020.0067). URL : <https://royalsocietypublishing.org/doi/10.1098/rsta.2020.0067> (visité le 01/11/2021) (cf. p. 91, 118).

- [70] Javier CUEVAS. « Neurotransmitters and their Life Cycle ». en. In : *xPharm : The Comprehensive Pharmacology Reference*. Sous la dir. de S. J. ENNA et David B. BYLUND. New York : Elsevier, jan. 2007, p. 1-7. ISBN : 978-0-08-055232-3. DOI : [10.1016/B978-008055232-3.60016-9](https://doi.org/10.1016/B978-008055232-3.60016-9). URL : <https://www.sciencedirect.com/science/article/pii/B9780080552323600169> (visité le 23/04/2023) (cf. p. 50).
- [71] Vassilis CUTSURIDIS, Stuart COBB et Bruce P. GRAHAM. « Encoding and retrieval in a model of the hippocampal CA1 microcircuit ». In : *Hippocampus* 20.3 (2010). _eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/hipo.20661>, p. 423-446. ISSN : 1098-1063. DOI : [10.1002/hipo.20661](https://doi.org/10.1002/hipo.20661). URL : <https://onlinelibrary.wiley.com/doi/abs/10.1002/hipo.20661> (visité le 09/04/2023) (cf. p. 91).
- [72] Zbigniew J. CZECH. *Introduction to parallel computing*. en. Cambridge, United Kingdom : Cambridge University Press, 2016. ISBN : 978-1-107-17439-9 (cf. p. 60, 64, 65).
- [73] Egidio D'ANGELO et Viktor JIRSA. « The quest for multiscale brain modeling ». In : *Trends in Neurosciences* (27 juill. 2022). ISSN : 0166-2236. DOI : [10.1016/j.tins.2022.06.007](https://doi.org/10.1016/j.tins.2022.06.007). URL : <https://www.sciencedirect.com/science/article/pii/S0166223622001254> (visité le 09/08/2022) (cf. p. 23, 74, 116).
- [74] L. DAGUM et R. MENON. « OpenMP : an industry standard API for shared-memory programming ». en. In : *IEEE Computational Science and Engineering* 5.1 (mars 1998), p. 46-55. ISSN : 10709924. DOI : [10.1109/99.660313](https://doi.org/10.1109/99.660313). URL : <http://ieeexplore.ieee.org/document/660313/> (visité le 08/06/2023) (cf. p. 65).
- [75] Kael DAI, Juan HERNANDO, Yazan N. BILLEH et al. « The SONATA data format for efficient description of large-scale network models ». en. In : *PLOS Computational Biology* 16.2 (fév. 2020). Publisher : Public Library of Science, e1007696. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1007696](https://doi.org/10.1371/journal.pcbi.1007696). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1007696> (visité le 23/03/2020) (cf. p. 114).
- [76] J. S. DAMOISEAUX, S. A.R.B. ROMBOUTS, F. BARKHOF et al. « Consistent resting-state networks across healthy subjects ». In : *Proceedings of the National Academy of Sciences of the United States of America* 103.37 (2006), p. 13848-13853. ISSN : 00278424. DOI : [10.1073/pnas.0601417103](https://doi.org/10.1073/pnas.0601417103) (cf. p. 96).
- [77] Harry DANKOWICZ et Frank SCHILDER. *Recipes for Continuation*. Philadelphia, PA : Society for Industrial et Applied Mathematics, 5 oct. 2013. DOI : [10.1137/1.9781611972573](https://doi.org/10.1137/1.9781611972573). URL : <http://epubs.siam.org/doi/book/10.1137/1.9781611972573> (visité le 21/02/2023) (cf. p. 30).

- [78] Zachary W. DAVIS, Gabriel B. BENIGNO, Charlee FLETTERMAN et al. « Spontaneous traveling waves naturally emerge from horizontal fiber time delays and travel through locally asynchronous-irregular states ». In : *Nature Communications* 12.1 (18 oct. 2021). Number : 1 Publisher : Nature Publishing Group, p. 6057. ISSN : 2041-1723. DOI : [10.1038/s41467-021-26175-1](https://doi.org/10.1038/s41467-021-26175-1). URL : <http://www.nature.com/articles/s41467-021-26175-1> (visité le 13/12/2022) (cf. p. 110, 111).
- [79] Andrew DAVISON, Daniel BRÜDERLE, Jochen EPPLER et al. « PyNN : a common interface for neuronal network simulators ». In : *Frontiers in Neuroinformatics* 2 (2009). ISSN : 1662-5196. URL : <https://www.frontiersin.org/articles/10.3389/neuro.11.011.2008> (visité le 12/06/2023) (cf. p. 69, 114).
- [80] Peter DAYAN et L. F. ABBOTT. *Theoretical neuroscience : computational and mathematical modeling of neural systems*. en. Computational neuroscience. Cambridge, Mass : Massachusetts Institute of Technology Press, 2001. ISBN : 978-0-262-04199-7 (cf. p. 51).
- [81] Robin DE SCHEPPER, Alice GEMINIANI, Stefano MASOLI et al. « Model simulations unveil the structure-function-dynamics relationship of the cerebellar cortical microcircuit ». en. In : *Communications Biology* 5.1 (nov. 2022). Number : 1 Publisher : Nature Publishing Group, p. 1-19. ISSN : 2399-3642. DOI : [10.1038/s42003-022-04213-y](https://doi.org/10.1038/s42003-022-04213-y). URL : <https://www.nature.com/articles/s42003-022-04213-y> (visité le 13/06/2023) (cf. p. 69).
- [82] Erik DE SCHUTTER, Giorgio A. ASCOLI et David N. KENNEDY. « On the Future of the Human Brain Project ». en. In : *Neuroinformatics* 4.2 (2006), p. 129-130. ISSN : 1539-2791. DOI : [10.1385/NI:4:2:129](https://doi.org/10.1385/NI:4:2:129). URL : <http://link.springer.com/10.1385/NI:4:2:129> (visité le 09/08/2022) (cf. p. 120).
- [83] Gustavo DECO, Viktor K. JIRSA et Anthony R. MCINTOSH. « Emerging concepts for the dynamical organization of resting-state activity in the brain. » In : *Nat. Rev. Neurosci.* 12.1 (2011), p. 43-56. ISSN : 1471-0048. DOI : [10.1038/nrn2961](https://doi.org/10.1038/nrn2961). URL : <http://dx.doi.org/10.1038/nrn2961%20http://www.ncbi.nlm.nih.gov/pubmed/21170073> (cf. p. 96).
- [84] Gustavo DECO, Viktor K. JIRSA, Peter A. ROBINSON et al. « The Dynamic Brain : From Spiking Neurons to Neural Masses and Cortical Fields ». en. In : *PLOS Computational Biology* 4.8 (2008), e1000092. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1000092](https://doi.org/10.1371/journal.pcbi.1000092). URL : <http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000092> (visité le 27/02/2018) (cf. p. 56).
- [85] P. J. DECOURSEY. « Phase Control of Activity in a Rodent ». In : *Cold Spring Harbor Symposia on Quantitative Biology* 25.0 (1^{er} jan. 1960), p. 49-55. ISSN : 0091-7451, 1943-4456. DOI : [10.1101/SQB.1960.025.01.006](https://doi.org/10.1101/SQB.1960.025.01.006). URL : <http://symposium.cshlp.org/cgi/doi/10.1101/SQB.1960.025.01.006> (visité le 22/04/2023) (cf. p. 45).

- [86] *DEEP system*. URL : https://www.fz-juelich.de/en/ias/jsc/systems/prototype-systems/deep_system (visité le 08/06/2023) (cf. p. 63).
- [87] Nima DEGHANI, Adrien PEYRACHE, Bartosz TELENCZUK et al. « Dynamic Balance of Excitation and Inhibition in Human and Monkey Neocortex ». In : *Scientific Reports* 6 (16 mars 2016), p. 23176. ISSN : 2045-2322. DOI : [10.1038/srep23176](https://doi.org/10.1038/srep23176). URL : <https://www.nature.com/articles/srep23176> (visité le 11/02/2019) (cf. p. 107, 109).
- [88] Scott L. DELP, Frank C. ANDERSON, Allison S. ARNOLD et al. « OpenSim : Open-Source Software to Create and Analyze Dynamic Simulations of Movement ». en. In : *IEEE Transactions on Biomedical Engineering* 54.11 (nov. 2007), p. 1940-1950. ISSN : 0018-9294. DOI : [10.1109/TBME.2007.901024](https://doi.org/10.1109/TBME.2007.901024). URL : <http://ieeexplore.ieee.org/document/4352056/> (visité le 20/07/2023) (cf. p. 120).
- [89] M. DENKER, A. YEGENOGLU et S. GRÜN. « Collaborative HPC-enabled workflows on the HBP Collaboratory using the Elephant framework ». In : *Neuroinformatics 2018*. 2018, P19. DOI : [10.12751/incf.ni2018.0019](https://doi.org/10.12751/incf.ni2018.0019). URL : <https://abstracts.g-node.org/conference/NI2018/abstracts#/uuid/023bec4e-0c35-4563-81ce-2c6fac282abd> (cf. p. 54, 276, 277, 321-323).
- [90] Michael DENKER, Alper YEGENOGLU et Sonja GRÜN. « Collaborative HPC-enabled workflows on the HBP Collaboratory using the Elephant framework ». en. In : (août 2018). Publisher : G-Node. DOI : [10.12751/INCF.NI2018.0019](https://doi.org/10.12751/INCF.NI2018.0019). URL : <http://abstracts.g-node.org/abstracts/023bec4e-0c35-4563-81ce-2c6fac282abd> (visité le 13/06/2023) (cf. p. 69).
- [91] *Desktop computer*. en. Page Version ID : 1157695691. Mai 2023. URL : https://en.wikipedia.org/w/index.php?title=Desktop_computer&oldid=1157695691 (visité le 08/06/2023) (cf. p. 61).
- [92] Alain DESTEXHE. « Self-sustained asynchronous irregular states and Up-Down states in thalamic, cortical and thalamocortical networks of nonlinear integrate-and-fire neurons ». en. In : *Journal of Computational Neuroscience* 27.3 (déc. 2009), p. 493. ISSN : 0929-5313, 1573-6873. DOI : [10.1007/s10827-009-0164-4](https://doi.org/10.1007/s10827-009-0164-4). URL : <https://link-springer-com.lama.univ-amu.fr/article/10.1007/s10827-009-0164-4> (visité le 16/02/2018) (cf. p. 55).
- [93] Alain DESTEXHE, Zachary F. MAINEN et Terrence J. SEJNOWSKI. « Synthesis of models for excitable membranes, synaptic transmission and neuromodulation using a common kinetic formalism ». en. In : *Journal of Computational Neuroscience* 1.3 (août 1994), p. 195-230. ISSN : 0929-5313, 1573-6873. DOI : [10.1007/BF00961734](https://doi.org/10.1007/BF00961734). URL : <http://link.springer.com/10.1007/BF00961734> (visité le 23/04/2023) (cf. p. 52).

- [94] A. DHOOGHE, W. GOVAERTS, Yu A. KUZNETSOV et al. « New features of the software MatCont for bifurcation analysis of dynamical systems ». In : *Mathematical and Computer Modelling of Dynamical Systems* (25 juin 2008). Publisher : Taylor & Francis. DOI : [10.1080/13873950701742754](https://doi.org/10.1080/13873950701742754). URL : <https://www.tandfonline.com/doi/full/10.1080/13873950701742754> (visité le 21/02/2023) (cf. p. 30).
- [95] A. DHOOGHE, W. GOVAERTS et Yu. A. KUZNETSOV. « MATCONT ». In : *ACM Transactions on Mathematical Software* 29.2 (juin 2003), p. 141-164. DOI : [10.1145/779359.779362](https://doi.org/10.1145/779359.779362). URL : <https://doi.org/10.1145%5C%2F779359.779362> (cf. p. 176).
- [96] M. DI VOLO et I. FÉRÉZOU. « Nonlinear collision between propagating waves in mouse somatosensory cortex ». en. In : *Scientific Reports* 11.1 (oct. 2021). Number : 1 Publisher : Nature Publishing Group, p. 19630. ISSN : 2045-2322. DOI : [10.1038/s41598-021-99057-7](https://doi.org/10.1038/s41598-021-99057-7). URL : <https://www.nature.com/articles/s41598-021-99057-7> (visité le 28/05/2023) (cf. p. 171).
- [97] Markus DIESMANN, Marc-Oliver GEWALTIG et Ad AERTSEN. « SYNOD : An Environment for Neural Systems Simulations Language Interface and Tutorial ». In : (1995) (cf. p. 69).
- [98] Markus DIESMANN, Max-planck STRMUNGSFORSCHUNG et Marc-Oliver GEWALTIG. « NEST : An Environment for Neural Systems ». In : *Forschung und Wissenschaftliches Rechnen Beiträge zum Heinz-Billing-Preis* 58 (août 2002) (cf. p. 68, 69, 72, 113, 120, 186).
- [99] *Discover EBRAINS*. en. URL : <https://www.ebrains.eu/page/discover-ebrains> (visité le 14/06/2023) (cf. p. 23, 68, 120).
- [100] Mikael DJURFELDT, Johannes HJORTH, Jochen M. EPPLER et al. « Run-Time Interoperability Between Neuronal Network Simulators Based on the MUSIC Framework ». In : *Neuroinformatics* 8.1 (1^{er} mars 2010), p. 43-60. ISSN : 1559-0089. DOI : [10.1007/s12021-010-9064-z](https://doi.org/10.1007/s12021-010-9064-z). URL : <https://doi.org/10.1007/s12021-010-9064-z> (visité le 26/11/2021) (cf. p. 74, 92, 120).
- [101] Rich DREWES. « Modeling the Brain with NCS and Brainlab ». In : *Linux Journal* 2005.134 (avr. 2005), p. 2. URL : <https://www.linuxjournal.com/article/8038> (visité le 19/07/2023) (cf. p. 120).
- [102] Dominique DUDKOWSKI. « Co-design patterns for embedded network management ». en. In : *Proceedings of the 2009 workshop on Re-architecting the internet*. Rome Italy : ACM, déc. 2009, p. 61-66. ISBN : 978-1-60558-749-3. DOI : [10.1145/1658978.1658994](https://doi.org/10.1145/1658978.1658994). URL : <https://dl.acm.org/doi/10.1145/1658978.1658994> (visité le 10/07/2023) (cf. p. 74, 113).

- [103] Daniel DURSTEWITZ, Jeremy K. SEAMANS et Terrence J. SEJNOWSKI. « Neuro-computational models of working memory ». In : *Nature Neuroscience* 3.11 (nov. 2000). Number : 11 Publisher : Nature Publishing Group, p. 1184-1191. ISSN : 1546-1726. DOI : [10.1038/81460](https://doi.org/10.1038/81460). URL : http://www.nature.com/articles/nn1100_1184 (visité le 03/03/2022) (cf. p. 73).
- [104] Gaute T. EINEVOLL, Alain DESTEXHE, Sonja GRÜN et al. « The Scientific Case for Brain Simulations ». In : *Neuron* 102.4 (mai 2019). Publisher : Elsevier, p. 735-744. DOI : [10.1016/j.neuron.2019.03.027](https://doi.org/10.1016/j.neuron.2019.03.027). URL : <https://hal.archives-ouvertes.fr/hal-02192934> (visité le 28/11/2021) (cf. p. 73).
- [105] Örjan EKEBERG et Mikael DJURFELDT. « MUSIC - Multisimulation Coordinator : Request For Comments ». en. In : *Nature Precedings* (avr. 2008). Bandiera_ab-test : a Cg_type : Nature Research Journals Primary_atype : Research Publisher : Nature Publishing Group, p. 1-1. ISSN : 1756-0357. DOI : [10.1038/npre.2008.1830.1](https://doi.org/10.1038/npre.2008.1830.1). URL : <http://www.nature.com/articles/npre.2008.1830.1> (visité le 26/11/2021) (cf. p. 120).
- [106] Sami EL BOUSTANI et Alain DESTEXHE. « A Master Equation Formalism for Macroscopic Modeling of Asynchronous Irregular Activity States ». In : *Neural Computation* 21.1 (jan. 2009), p. 46-100. ISSN : 0899-7667. DOI : [10.1162/neco.2009.02-08-710](https://doi.org/10.1162/neco.2009.02-08-710). eprint : <https://direct.mit.edu/neco/article-pdf/21/1/46/818085/neco.2009.02-08-710.pdf>. URL : <https://doi.org/10.1162/neco.2009.02-08-710> (cf. p. 171, 173, 174, 182, 185, 186, 331, 332).
- [107] Jochen EPPLER, Moritz HELIAS, Eilif MULLER et al. « PyNEST : a convenient interface to the NEST simulator ». In : *Frontiers in Neuroinformatics* 2 (2009). ISSN : 1662-5196. URL : <https://www.frontiersin.org/article/10.3389/neuro.11.012.2008> (visité le 27/06/2022) (cf. p. 69).
- [108] Paul ERDOS et Alfréd RÉNYI. « On random graphs ». In : *Mathematicae* 6 (1959), p. 290-297 (cf. p. 325).
- [109] Bard ERMENTROUT. *Simulating, Analyzing, and Animating Dynamical Systems : A Guide to XPPAUT for Researchers and Students*. Google-Books-ID : xbTkXIN1GYkC. SIAM, 1^{er} jan. 2002. 298 p. ISBN : 978-0-89871-506-4 (cf. p. 30).
- [110] G. B. ERMENTROUT et N. KOPELL. « Parabolic Bursting in an Excitable System Coupled with a Slow Oscillation ». In : *SIAM Journal on Applied Mathematics* 46.2 (avr. 1986), p. 233-253. ISSN : 0036-1399, 1095-712X. DOI : [10.1137/0146017](https://doi.org/10.1137/0146017). URL : <http://epubs.siam.org/doi/10.1137/0146017> (visité le 21/04/2023) (cf. p. 48).
- [111] G. Bard ERMENTROUT et David H. TERMAN. *Mathematical Foundations of Neuroscience*. T. 35. Interdisciplinary Applied Mathematics. New York, NY : Springer, 2010. DOI : [10.1007/978-0-387-87708-2](https://doi.org/10.1007/978-0-387-87708-2). URL : <http://link.springer.com/10.1007/978-0-387-87708-2> (visité le 27/02/2023) (cf. p. 32, 33, 51, 52).

- [112] Graham E FAGG, Edgar GABRIEL, Zizhong CHEN et al. « Fault Tolerant Communication Library and Applications for High Performance Computing ». en. In : Santa Fee, oct. 2003 (cf. p. 66).
- [113] Egidio FALOTICO, Lorenzo VANNUCCI, Alessandro AMBROSANO et al. « Connecting Artificial Brains to Robots in a Comprehensive Simulation Framework : The Neurorobotics Platform ». In : *Frontiers in Neurorobotics* 11 (2017), p. 2. ISSN : 1662-5218. URL : <https://www.frontiersin.org/article/10.3389/fnbot.2017.00002> (visité le 10/05/2022) (cf. p. 74, 120).
- [114] Benedikt FELDOTTO, Fabrice O. MORIN et Alois KNOLL. « The Neurorobotics Platform Robot Designer : Modeling Morphologies for Embodied Learning Experiments ». In : *Frontiers in Neurorobotics* 16 (2022). ISSN : 1662-5218. URL : <https://www.frontiersin.org/articles/10.3389/fnbot.2022.856727> (visité le 20/07/2023) (cf. p. 120).
- [115] Katie FERGUSON, Carey HUH, Bénédicte AMILHON et al. « Network models provide insights into how oriens–lacunosum-moleculare and bistratified cell interactions influence the power of local hippocampal CA1 theta oscillations ». In : *Frontiers in Systems Neuroscience* 9 (2015). ISSN : 1662-5137. URL : <https://www.frontiersin.org/articles/10.3389/fnsys.2015.00110> (visité le 09/04/2023) (cf. p. 91).
- [116] Alberto FERRARA, David ANGULO-GARCIA, Alessandro TORCINI et al. « Population spiking and bursting in next-generation neural masses with spike-frequency adaptation ». In : *Physical Review E* 107.2 (fév. 2023). Publisher : American Physical Society, p. 024311. DOI : [10.1103/PhysRevE.107.024311](https://doi.org/10.1103/PhysRevE.107.024311). URL : <https://link.aps.org/doi/10.1103/PhysRevE.107.024311> (visité le 24/04/2023) (cf. p. 53).
- [117] A. FINKELSTEIN, J. HETHERINGTON, LINZHONG LI et al. « Computational challenges of systems biology ». In : *Computer* 37.5 (mai 2004), p. 26-33. ISSN : 0018-9162. DOI : [10.1109/MC.2004.1297236](https://doi.org/10.1109/MC.2004.1297236). URL : <http://ieeexplore.ieee.org/document/1297236/> (visité le 04/11/2021) (cf. p. 73).
- [118] Jacob FISH, Gregory J. WAGNER et Sinan KETEN. « Mesoscopic and multiscale modelling in materials ». In : *Nature Materials* 20.6 (juin 2021). Number : 6 Primary_atype : Reviews Publisher : Nature Publishing Group Subject_term : Engineering;Materials science;Mathematics and computing Subject_term_id : engineering;materials-science;mathematics-and-computing, p. 774-786. ISSN : 1476-4660. DOI : [10.1038/s41563-020-00913-0](https://doi.org/10.1038/s41563-020-00913-0). URL : <http://www.nature.com/articles/s41563-020-00913-0> (visité le 10/11/2021) (cf. p. 74).
- [119] Richard FITZHUGH. « Mathematical models of threshold phenomena in the nerve membrane ». In : *The Bulletin of Mathematical Biophysics* 17.4 (déc. 1955), p. 257-278. ISSN : 0007-4985, 1522-9602. DOI : [10.1007/BF02477753](https://doi.org/10.1007/BF02477753). URL : <http://link.springer.com/10.1007/BF02477753> (visité le 06/12/2022) (cf. p. 99).

- [120] Richard FITZHUGH. « Impulses and Physiological States in Theoretical Models of Nerve Membrane ». In : *Biophysical Journal* 1.6 (juill. 1961), p. 445-466. ISSN : 00063495. DOI : [10.1016/S0006-3495\(61\)86902-6](https://doi.org/10.1016/S0006-3495(61)86902-6). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0006349561869026> (visité le 17/04/2023) (cf. p. 47).
- [121] Steven W. FLAVELL et Andrew GORDUS. « Dynamic functional connectivity in the static connectome of *Caenorhabditis elegans* ». en. In : *Current Opinion in Neurobiology* 73 (avr. 2022), p. 102515. ISSN : 0959-4388. DOI : [10.1016/j.conb.2021.12.002](https://doi.org/10.1016/j.conb.2021.12.002). URL : <https://www.sciencedirect.com/science/article/pii/S0959438821001549> (visité le 23/05/2023) (cf. p. 57).
- [122] R. FLETCHER. « A new approach to variable metric algorithms ». In : *The Computer Journal* 13.3 (jan. 1970), p. 317-322. ISSN : 0010-4620. DOI : [10.1093/comjnl/13.3.317](https://doi.org/10.1093/comjnl/13.3.317). URL : <https://doi.org/10.1093/comjnl/13.3.317> (visité le 20/06/2023) (cf. p. 176).
- [123] Nicolas FOURCAUD-TROCMÉ, David HANSEL, Carl van VREESWIJK et al. « How Spike Generation Mechanisms Determine the Neuronal Response to Fluctuating Inputs ». In : *Journal of Neuroscience* 23.37 (17 déc. 2003). Publisher : Society for Neuroscience Section : Behavioral/Systems/Cognitive, p. 11628-11640. ISSN : 0270-6474, 1529-2401. DOI : [10.1523/JNEUROSCI.23-37-11628.2003](https://doi.org/10.1523/JNEUROSCI.23-37-11628.2003). URL : <https://www.jneurosci.org/content/23/37/11628> (visité le 21/04/2023) (cf. p. 48).
- [124] Walter J. FREEMAN, Brian C. BURKE et Mark D. HOLMES. « Aperiodic phase re-setting in scalp EEG of beta-gamma oscillations by state transitions at alpha-theta rates ». In : *Human Brain Mapping* 19.4 (juin 2003), p. 248-272. ISSN : 1065-9471. DOI : [10.1002/hbm.10120](https://doi.org/10.1002/hbm.10120). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC6871824/> (visité le 20/06/2023) (cf. p. 177, 323).
- [125] Yves FRÉGNAC. « How Blue is the Sky? » In : *eNeuro* 8.2 (1^{er} mars 2021). Publisher : Society for Neuroscience Section : Commentary. ISSN : 2373-2822. DOI : [10.1523/ENEURO.0130-21.2021](https://doi.org/10.1523/ENEURO.0130-21.2021). URL : <https://www.eneuro.org/content/8/2/ENEURO.0130-21.2021> (visité le 05/04/2023) (cf. p. 73).
- [126] Yves FRÉGNAC et Gilles LAURENT. « Neuroscience : Where is the brain in the Human Brain Project? » en. In : *Nature* 513.7516 (sept. 2014). Number : 7516 Publisher : Nature Publishing Group, p. 27-29. ISSN : 1476-4687. DOI : [10.1038/513027a](https://doi.org/10.1038/513027a). URL : <http://www.nature.com/articles/513027a> (visité le 09/08/2022) (cf. p. 68).
- [127] Pascal FRIES. « Rhythms for Cognition : Communication through Coherence ». In : *Neuron* 88.1 (2015), p. 220-235. ISSN : 10974199. DOI : [10.1016/j.neuron.2015.09.034](https://doi.org/10.1016/j.neuron.2015.09.034). URL : <http://dx.doi.org/10.1016/j.neuron.2015.09.034> (cf. p. 96).

- [128] Steve B. FURBER, Francesco GALLUPPI, Steve TEMPLE et al. « The SpiNNaker Project ». In : *Proceedings of the IEEE* 102.5 (mai 2014). Conference Name : Proceedings of the IEEE, p. 652-665. ISSN : 1558-2256. DOI : [10.1109/JPROC.2014.2304638](https://doi.org/10.1109/JPROC.2014.2304638) (cf. p. 69).
- [129] Edgar GABRIEL, Graham E. FAGG, George BOSILCA et al. « Open MPI : Goals, Concept, and Design of a Next Generation MPI Implementation ». en. In : *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Sous la dir. de David HUTCHISON, Takeo KANADE, Josef KITTLER et al. T. 3241. Series Title : Lecture Notes in Computer Science. Berlin, Heidelberg : Springer Berlin Heidelberg, 2004, p. 97-104. DOI : [10.1007/978-3-540-30218-6_19](https://doi.org/10.1007/978-3-540-30218-6_19). URL : http://link.springer.com/10.1007/978-3-540-30218-6_19 (visité le 08/06/2023) (cf. p. 66).
- [130] Sergio E. GALINDO, Pablo TOHARIA, Oscar D. ROBLES et al. « ViSimpl : Multi-View Visual Analysis of Brain Simulation Data ». In : *Frontiers in Neuroinformatics* 10 (2016). ISSN : 1662-5196. URL : <https://www.frontiersin.org/articles/10.3389/fninf.2016.00044> (visité le 13/06/2023) (cf. p. 69).
- [131] Daniela GANDOLFI, Jonathan MAPELLI, Sergio SOLINAS et al. « A realistic morpho-anatomical connection strategy for modelling full-scale point-neuron micro-circuits ». In : *Scientific Reports* 12 (août 2022), p. 13864. ISSN : 2045-2322. DOI : [10.1038/s41598-022-18024-y](https://doi.org/10.1038/s41598-022-18024-y). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9381785/> (visité le 24/04/2023) (cf. p. 53, 91).
- [132] Michael S. GAZZANIGA, John D. VAN HORN, Floyd BLOOM et al. « Continuing Progress in Neuroinformatics ». en. In : *Science* 311.5758 (jan. 2006), p. 176-176. ISSN : 0036-8075, 1095-9203. DOI : [10.1126/science.311.5758.176a](https://doi.org/10.1126/science.311.5758.176a). URL : <https://www.science.org/doi/10.1126/science.311.5758.176a> (visité le 09/08/2022) (cf. p. 120).
- [133] Lisa GERHARDT, Wahid BHIMJI, Shane CANON et al. « Shifter : Containers for HPC ». en. In : *Journal of Physics : Conference Series* 898 (oct. 2017), p. 082021. ISSN : 1742-6588, 1742-6596. DOI : [10.1088/1742-6596/898/8/082021](https://doi.org/10.1088/1742-6596/898/8/082021). URL : <https://iopscience.iop.org/article/10.1088/1742-6596/898/8/082021> (visité le 08/06/2023) (cf. p. 64).
- [134] Marc-Oliver GEWALTIG et Robert CANNON. « Current Practice in Software Development for Computational Neuroscience and How to Improve It ». en. In : *PLOS Computational Biology* 10.1 (jan. 2014). Publisher : Public Library of Science, e1003376. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1003376](https://doi.org/10.1371/journal.pcbi.1003376). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003376> (visité le 17/07/2023) (cf. p. 120, 122).
- [135] Marc-Oliver GEWALTIG et Markus DIESMANN. « NEST (NEural Simulation Tool) ». In : *Scholarpedia* 2.4 (5 avr. 2007), p. 1430. ISSN : 1941-6016. DOI : [10.4249/scholarpedia.1430](https://doi.org/10.4249/scholarpedia.1430). URL : [http://www.scholarpedia.org/article/NEST_\(NEural_Simulation_Tool\)](http://www.scholarpedia.org/article/NEST_(NEural_Simulation_Tool)) (visité le 30/11/2021) (cf. p. 74).

- [136] Anandamohan GHOSH, Y. RHO, Anthony Randal MCINTOSH et al. « Noise during rest enables the exploration of the brain's dynamic repertoire ». In : *PLoS Computational Biology* 4.10 (2008), e1000196. ISSN : 1553734X. DOI : [10.1371/journal.pcbi.1000196](https://doi.org/10.1371/journal.pcbi.1000196) (cf. p. 96, 116).
- [137] N GODDARD, G HOOD, F HOWELL et al. « NEOSIM : Portable large-scale plug and play modelling ». In : *Neurocomputing*. Computational Neuroscience : Trends in Research 2001 38-40 (1^{er} juin 2001), p. 1657-1661. ISSN : 0925-2312. DOI : [10.1016/S0925-2312\(01\)00528-8](https://doi.org/10.1016/S0925-2312(01)00528-8). URL : <https://www.sciencedirect.com/science/article/pii/S0925231201005288> (visité le 10/05/2022) (cf. p. 74, 120).
- [138] Nigel GODDARD. « The Human Brain Project : Phase I ». en. In : (1998). URL : <https://grantome.com/grant/NSF/IOS-9820016> (visité le 10/08/2022) (cf. p. 120).
- [139] Donald GOLDFARB. « A family of variable-metric methods derived by variational means ». en. In : *Mathematics of Computation* 24.109 (1970), p. 23-26. ISSN : 0025-5718, 1088-6842. DOI : [10.1090/S0025-5718-1970-0258249-6](https://doi.org/10.1090/S0025-5718-1970-0258249-6). URL : <https://www.ams.org/mcom/1970-24-109/S0025-5718-1970-0258249-6/> (visité le 20/06/2023) (cf. p. 176).
- [140] Jennifer S. GOLDMAN, Lionel KUSCH, David AQUILUE et al. « A comprehensive neural simulation of slow-wave sleep and highly responsive wakefulness dynamics ». In : *Frontiers in Computational Neuroscience* 16 (2023). ISSN : 1662-5188. URL : <https://www.frontiersin.org/articles/10.3389/fncom.2022.1058957> (visité le 09/04/2023) (cf. p. 58, 171, 177, 179, 181).
- [141] Cláudio GOMES, Casper THULE, David BROMAN et al. « Co-Simulation : A Survey ». en. In : *ACM Computing Surveys* 51.3 (mai 2019), p. 1-33. ISSN : 0360-0300, 1557-7341. DOI : [10.1145/3179993](https://doi.org/10.1145/3179993). URL : <https://dl.acm.org/doi/10.1145/3179993> (visité le 11/07/2023) (cf. p. 74, 113, 227).
- [142] Daniel A. GOODENOUGH et David L. PAUL. « Gap Junctions ». In : *Cold Spring Harbor Perspectives in Biology* 1.1 (juill. 2009), a002576. ISSN : 1943-0264. DOI : [10.1101/cshperspect.a002576](https://doi.org/10.1101/cshperspect.a002576). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2742079/> (visité le 23/04/2023) (cf. p. 50).
- [143] Tomasz GÓRSKI, Damien DEPANNEMAECCKER et Alain DESTEXHE. « Conductance-Based Adaptive Exponential Integrate-and-Fire Model ». In : *Neural Computation* 33.1 (jan. 2021), p. 41-66. ISSN : 0899-7667, 1530-888X. DOI : [10.1162/neco_a_01342](https://doi.org/10.1162/neco_a_01342). URL : <https://direct.mit.edu/neco/article/33/1/41-66/95662> (visité le 17/08/2022) (cf. p. 49, 99).
- [144] Richard L. GRAHAM, Sung-Eun CHOI, David J. DANIEL et al. « A Network-Failure-Tolerant Message-Passing System for Terascale Clusters ». en. In : *International Journal of Parallel Programming* 31.4 (2003), p. 285-303. ISSN : 08857458. DOI : [10.1023/A:1024504726988](https://doi.org/10.1023/A:1024504726988). URL : <http://link.springer.com/10.1023/A:1024504726988> (visité le 08/06/2023) (cf. p. 66).

- [145] Richard L. GRAHAM, Timothy S. WOODALL et Jeffrey M. SQUYRES. « Open MPI : A Flexible High Performance MPI ». en. In : *Parallel Processing and Applied Mathematics*. Sous la dir. de David HUTCHISON, Takeo KANADE, Josef KITTLER et al. T. 3911. Series Title : Lecture Notes in Computer Science. Berlin, Heidelberg : Springer Berlin Heidelberg, 2006, p. 228-239. DOI : [10.1007/11752578_29](https://doi.org/10.1007/11752578_29). URL : http://link.springer.com/10.1007/11752578_29 (visité le 08/06/2023) (cf. p. 66).
- [146] John GUCKENHEIMER et Philip HOLMES. *Nonlinear Oscillations, Dynamical Systems, and Bifurcations of Vector Fields*. T. 42. Applied Mathematical Sciences. New York, NY : Springer, 1983. DOI : [10.1007/978-1-4612-1140-2](https://doi.org/10.1007/978-1-4612-1140-2). URL : <http://link.springer.com/10.1007/978-1-4612-1140-2> (visité le 20/02/2023) (cf. p. 25, 28, 29).
- [147] Moritz GUTLEIN, Wojciech BARON, Christopher RENNER et al. « Performance Evaluation of HLA RTI Implementations ». In : *2020 IEEE/ACM 24th International Symposium on Distributed Simulation and Real Time Applications (DS-RT)*. Prague, Czech Republic : IEEE, sept. 2020, p. 1-8. ISBN : 978-1-72817-343-6. DOI : [10.1109/DS-RT50469.2020.9213641](https://doi.org/10.1109/DS-RT50469.2020.9213641). URL : <https://ieeexplore.ieee.org/document/9213641/> (visité le 12/05/2023) (cf. p. 92, 93).
- [148] Espen HAGEN, David DAHMEN, Maria L. STAVRINOÛ et al. « Hybrid Scheme for Modeling Local Field Potentials from Point-Neuron Networks ». In : *Cerebral Cortex* 26.12 (déc. 2016), p. 4461-4496. ISSN : 1047-3211. DOI : [10.1093/cercor/bhw237](https://doi.org/10.1093/cercor/bhw237). URL : <https://doi.org/10.1093/cercor/bhw237> (visité le 13/06/2023) (cf. p. 69, 77, 87, 238, 249).
- [149] Espen HAGEN, Solveig NÆSS, Torbjørn V. NESS et al. « Multimodal Modeling of Neural Network Activity : Computing LFP, ECoG, EEG, and MEG Signals With LFPy 2.0 ». In : *Frontiers in Neuroinformatics* 12 (2018). ISSN : 1662-5196. URL : <https://www.frontiersin.org/articles/10.3389/fninf.2018.00092> (visité le 13/06/2023) (cf. p. 69).
- [150] Patric HAGMANN. « From diffusion MRI to brain connectomics ». fr. In : (2005). DOI : [10.5075/epfl-thesis-3230](https://doi.org/10.5075/epfl-thesis-3230). URL : <https://infoscience.epfl.ch/record/33696> (visité le 27/11/2018) (cf. p. 56, 120).
- [151] Patric HAGMANN, Leila CAMMOUN, Xavier GIGANDET et al. « Mapping the structural core of human cerebral cortex ». In : *PLoS Biology* 6.7 (2008), p. 1479-1493. ISSN : 15449173. DOI : [10.1371/journal.pbio.0060159](https://doi.org/10.1371/journal.pbio.0060159) (cf. p. 96).
- [152] Jan HAHNE, Sandra DIAZ, Alexander PATRONIS et al. *NEST 3.0*. 10 juin 2021. DOI : [10.5281/zenodo.4739103](https://doi.org/10.5281/zenodo.4739103). URL : <https://zenodo.org/record/4739103> (cf. p. 69, 75-77, 81, 85, 86, 173, 225, 230, 250-252, 254, 325).
- [153] Bilal HAIDER, Alvaro DUQUE, Andrea R. HASENSTAUB et al. « Neocortical Network Activity In Vivo Is Generated through a Dynamic Balance of Excitation and Inhibition ». In : *Journal of Neuroscience* 26.17 (26 avr. 2006), p. 4535-4545. ISSN : 0270-6474, 1529-2401. DOI : [10.1523/JNEUROSCI.5297-05.2006](https://doi.org/10.1523/JNEUROSCI.5297-05.2006). URL :

- <http://www.jneurosci.org/content/26/17/4535> (visité le 13/02/2019) (cf. p. 107, 109).
- [154] D. HANSEL, G. MATO et C. MEUNIER. « Synchrony in Excitatory Neural Networks ». In : *Neural Computation* 7.2 (mars 1995), p. 307-337. ISSN : 0899-7667, 1530-888X. DOI : [10.1162/neco.1995.7.2.307](https://doi.org/10.1162/neco.1995.7.2.307). URL : <https://direct.mit.edu/neco/article/7/2/307-337/5861> (visité le 22/04/2023) (cf. p. 45).
- [155] Stewart HEITMANN et Michael BREAKSPEAR. « Putting the “ dynamic ” back into dynamic functional connectivity ». In : *Network Neuroscience* (2017), p. 1-37. DOI : [10.1162/NETN](https://doi.org/10.1162/NETN). URL : http://dx.doi.org/10.1162/NETN_a_00041 (cf. p. 96).
- [156] Patrick HERBERS, Iago CALVO, Sandra DIAZ-PIER et al. « ConGen—A Simulator-Agnostic Visual Language for Definition and Generation of Connectivity in Large and Multiscale Neural Networks ». In : *Frontiers in Neuroinformatics* 15 (2022). ISSN : 1662-5196. URL : <https://www.frontiersin.org/articles/10.3389/fninf.2021.766697> (visité le 13/06/2023) (cf. p. 69, 114).
- [157] Andreas V. M. HERZ, Tim GOLLISCH, Christian K. MACHENS et al. « Modeling Single-Neuron Dynamics and Computations : A Balance of Detail and Abstraction ». In : *Science* 314.5796 (6 oct. 2006), p. 80-85. ISSN : 0036-8075, 1095-9203. DOI : [10.1126/science.1127240](https://doi.org/10.1126/science.1127240). URL : <https://www.science.org/doi/10.1126/science.1127240> (visité le 17/04/2023) (cf. p. 47).
- [158] J. HETHERINGTON, I. D. L. BOGLE, P. SAFFREY et al. « Addressing the challenges of multiscale model management in systems biology ». In : *Computers & Chemical Engineering*. 7th World Congress of Chemical Engineering 31.8 (15 août 2007), p. 962-979. ISSN : 0098-1354. DOI : [10.1016/j.compchemeng.2006.10.004](https://doi.org/10.1016/j.compchemeng.2006.10.004). URL : <https://www.sciencedirect.com/science/article/pii/S0098135406002535> (visité le 04/11/2021) (cf. p. 74).
- [159] Michael HINES. « Efficient computation of branched nerve equations ». en. In : *International Journal of Bio-Medical Computing* 15.1 (jan. 1984), p. 69-76. ISSN : 0020-7101. DOI : [10.1016/0020-7101\(84\)90008-4](https://doi.org/10.1016/0020-7101(84)90008-4). URL : <https://www.sciencedirect.com/science/article/pii/0020710184900084> (visité le 12/06/2023) (cf. p. 68, 69, 120).
- [160] Michael HINES, Andrew DAVISON et Eilif MULLER. « NEURON and Python ». In : *Frontiers in Neuroinformatics* 3 (2009). ISSN : 1662-5196. URL : <https://www.frontiersin.org/articles/10.3389/neuro.11.001.2009> (visité le 12/06/2023) (cf. p. 69).
- [161] A. L. HODGKIN. « The local electric changes associated with repetitive action in a non-medullated axon ». In : *The Journal of Physiology* 107.2 (15 mars 1948), p. 165-181. ISSN : 0022-3751. URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392160/> (visité le 12/04/2023) (cf. p. 35).

- [162] A. L. HODGKIN et A. F. HUXLEY. « A quantitative description of membrane current and its application to conduction and excitation in nerve ». In : *The Journal of Physiology* 117.4 (28 août 1952), p. 500-544. ISSN : 0022-3751. URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC1392413/> (visité le 27/02/2023) (cf. p. 32, 33, 99).
- [163] Alfons HOEKSTRA, Bastien CHOPARD et Peter COVENEY. « Multiscale modelling and simulation : a position paper ». In : *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences* 372.2021 (août 2014). Publisher : Royal Society, p. 20130377. DOI : 10.1098/rsta.2013.0377. URL : <https://royalsocietypublishing.org/doi/full/10.1098/rsta.2013.0377> (visité le 29/10/2021) (cf. p. 115).
- [164] G. R. HOLT, W. R. SOFTKY, C. KOCH et al. « Comparison of discharge variability in vitro and in vivo in cat visual cortex neurons ». In : *Journal of Neurophysiology* 75.5 (1^{er} mai 1996), p. 1806-1814. ISSN : 0022-3077, 1522-1598. DOI : 10.1152/jn.1996.75.5.1806. URL : <https://www.physiology.org/doi/10.1152/jn.1996.75.5.1806> (visité le 11/04/2023) (cf. p. 34).
- [165] Eberhard HOPF. « Abzweigung einer periodischen Lösung von einer stationären Lösung eines Differentialsystems ». In : 1942 19 (1942) (cf. p. 29).
- [166] Chih-Hsu HUANG et Chou-Ching K. LIN. « An Efficient Population Density Method for Modeling Neural Networks with Synaptic Dynamics Manifesting Finite Relaxation Time and Short-Term Plasticity ». In : *eNeuro* 5.6 (2018). DOI : 10.1523/ENEURO.0002-18.2018. eprint : <https://www.eneuro.org/content/5/6/ENEURO.0002-18.2018.full.pdf>. URL : <https://www.eneuro.org/content/5/6/ENEURO.0002-18.2018> (cf. p. 185).
- [167] Michael F. HUERTA, Stephen H. KOSLOW et Alan I. LESHNER. « The human brain project : an international resource ». en. In : *Trends in Neurosciences* 16.11 (nov. 1993), p. 436-438. ISSN : 01662236. DOI : 10.1016/0166-2236(93)90069-X. URL : <https://linkinghub.elsevier.com/retrieve/pii/S016622369390069X> (visité le 09/08/2022) (cf. p. 120).
- [168] *Human Brain Project EC Grants*. URL : <https://www.humanbrainproject.eu/en/about-hbp/project-structure/human-brain-project-ec-grants/> (visité le 09/06/2023) (cf. p. 66, 67).
- [169] *Human Brain Project Specific Grant Agreement 1 | HBP SGA1 Project | Fact Sheet | H2020*. en. URL : <https://cordis.europa.eu/project/id/720270> (visité le 09/06/2023) (cf. p. 66).
- [170] *Human Brain Project Specific Grant Agreement 2 | HBP SGA2 Project | Fact Sheet | H2020*. en. URL : <https://cordis.europa.eu/project/id/785907> (visité le 09/06/2023) (cf. p. 66).
- [171] *Human Brain Project Specific Grant Agreement 3 | HBP SGA3 Project | Fact Sheet | H2020*. en. URL : <https://cordis.europa.eu/project/id/945539> (visité le 09/06/2023) (cf. p. 66).

- [172] Raoul HUYS, Dionysios PERDIKIS et Viktor K. JIRSA. « Functional architectures and structured flows on manifolds : A dynamical framework for motor behavior. » In : *Psychological Review* 121.3 (2014), p. 302-336. ISSN : 1939-1471, 0033-295X. DOI : [10.1037/a0037014](https://doi.org/10.1037/a0037014). URL : <http://doi.apa.org/getdoi.cfm?doi=10.1037/a0037014> (visité le 18/04/2023) (cf. p. 73).
- [173] Kaori IKEDA et John M. BEKKERS. « Autapses ». en. In : *Current Biology* 16.9 (mai 2006), R308. ISSN : 09609822. DOI : [10.1016/j.cub.2006.03.085](https://doi.org/10.1016/j.cub.2006.03.085). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0960982206014187> (visité le 24/04/2023) (cf. p. 53).
- [174] The MathWorks INC. *MATLAB version : 9.13.0 (R2022b)*. Natick, Massachusetts, United States, 2022. URL : <https://www.mathworks.com> (cf. p. 74, 176).
- [175] Junya INOUE, Alberto ANTONIETTI et Alessandra PEDROCCHI. « Cerebellar control of saccadic adaptation using a spiking neural network model integrated into the NeuroRobotics platform ». en. In : *2021 10th International IEEE/EMBS Conference on Neural Engineering (NER)*. Italy : IEEE, mai 2021, p. 643-646. ISBN : 978-1-72814-337-8. DOI : [10.1109/NER49283.2021.9441402](https://doi.org/10.1109/NER49283.2021.9441402). URL : <https://ieeexplore.ieee.org/document/9441402/> (visité le 04/09/2023) (cf. p. 120).
- [176] *IPO model*. In : *Wikipedia*. Page Version ID : 1066367820. 18 jan. 2022. URL : https://en.wikipedia.org/w/index.php?title=IPO_model&oldid=1066367820 (visité le 21/04/2023) (cf. p. 31).
- [177] Matias J. ISON, Florian MORMANN, Moran CERF et al. « Selectivity of pyramidal cells and interneurons in the human medial temporal lobe ». In : *Journal of Neurophysiology* 106.4 (oct. 2011), p. 1713-1721. DOI : [10.1152/jn.00576.2010](https://doi.org/10.1152/jn.00576.2010). URL : <https://doi.org/10.1152/jn.00576.2010> (cf. p. 179).
- [178] E.M. IZHIKEVICH. « Which Model to Use for Cortical Spiking Neurons? » In : *IEEE Transactions on Neural Networks* 15.5 (sept. 2004), p. 1063-1070. ISSN : 1045-9227. DOI : [10.1109/TNN.2004.832719](https://doi.org/10.1109/TNN.2004.832719). URL : <http://ieeexplore.ieee.org/document/1333071/> (visité le 23/04/2019) (cf. p. 35, 46, 48, 49, 99, 110).
- [179] Eugene M. IZHIKEVICH et Richard FITZHUGH. « FitzHugh-Nagumo model ». In : *Scholarpedia* 1.9 (23 sept. 2006), p. 1349. ISSN : 1941-6016. DOI : [10.4249/scholarpedia.1349](https://doi.org/10.4249/scholarpedia.1349). URL : http://www.scholarpedia.org/article/FitzHugh-Nagumo_model (visité le 21/04/2023) (cf. p. 47).
- [180] C. G. J. JACOBI. « De Determinantibus functionalibus. » In : *Journal für die reine und angewandte Mathematik (Crelles Journal)* 1841.22 (1^{er} jan. 1841). Publisher : De Gruyter Section : Journal für die reine und angewandte Mathematik, p. 319-359. ISSN : 1435-5345. DOI : [10.1515/crll.1841.22.319](https://doi.org/10.1515/crll.1841.22.319). URL : <https://www.degruyter.com/document/doi/10.1515/crll.1841.22.319/html> (visité le 17/02/2023) (cf. p. 26).

- [181] JAKOB BERNOULLI. *Jacobi Bernoulli, ... Ars conjectandi, opus posthumum. Accedit Tractatus de seriebus infinitis, et epistola Gallice scripta De ludo pilae reticularis*. Latin. impensis Thurnisiorum, fratrum, 1713. URL : http://archive.org/details/bub_gb_kz9nvk99EWoC (visité le 24/04/2023) (cf. p. 53).
- [182] Ben H. JANSEN, George ZOURIDAKIS et Michael E. BRANDT. « A neurophysiologically-based mathematical model of flash visual evoked potentials ». en. In : *Biological Cybernetics* 68.3 (jan. 1993), p. 275-283. ISSN : 0340-1200, 1432-0770. DOI : [10.1007/BF00224863](https://doi.org/10.1007/BF00224863). URL : <http://link.springer.com/10.1007/BF00224863> (visité le 03/03/2020) (cf. p. 56).
- [183] Lucas JANSON, William FITHIAN et Trevor J. HASTIE. « Effective degrees of freedom : a flawed metaphor ». In : *Biometrika* 102.2 (1^{er} juin 2015), p. 479-485. ISSN : 0006-3444. DOI : [10.1093/biomet/asv019](https://doi.org/10.1093/biomet/asv019). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4787623/> (visité le 17/04/2023) (cf. p. 46).
- [184] V. K. JIRSA et J. A. S. KELSO. « Spatiotemporal pattern formation in neural systems with heterogeneous connection topologies ». In : *Physical Review E* 62.6 (1^{er} déc. 2000), p. 8462-8465. DOI : [10.1103/PhysRevE.62.8462](https://doi.org/10.1103/PhysRevE.62.8462). URL : <https://link.aps.org/doi/10.1103/PhysRevE.62.8462> (visité le 25/09/2018) (cf. p. 96, 97).
- [185] V.K. JIRSA, K.J. JANTZEN, A. FUCHS et al. « Spatiotemporal forward solution of the EEG and MEG using network modeling ». en. In : *IEEE Transactions on Medical Imaging* 21.5 (mai 2002), p. 493-504. ISSN : 0278-0062. DOI : [10.1109/TMI.2002.1009385](https://doi.org/10.1109/TMI.2002.1009385). URL : <http://ieeexplore.ieee.org/document/1009385/> (visité le 09/06/2023) (cf. p. 56).
- [186] Viktor JIRSA et Hiba SHEHEITLI. « Entropy, free energy, symmetry and dynamics in the brain ». In : *Journal of Physics : Complexity* 3.1 (mars 2022), p. 015007. ISSN : 2632-072X. DOI : [10.1088/2632-072X/ac4bec](https://doi.org/10.1088/2632-072X/ac4bec). URL : <https://iopscience.iop.org/article/10.1088/2632-072X/ac4bec> (visité le 18/04/2023) (cf. p. 73).
- [187] Viktor K JIRSA. « Neural field dynamics with local and global connectivity and time delay ». In : *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences* 367.1891 (28 mars 2009), p. 1131-1143. ISSN : 1364-503X, 1471-2962. DOI : [10.1098/rsta.2008.0260](https://doi.org/10.1098/rsta.2008.0260). URL : <https://royalsocietypublishing.org/doi/10.1098/rsta.2008.0260> (visité le 09/11/2020) (cf. p. 96, 97).
- [188] Viktor K. JIRSA. « Connectivity and dynamics of neural information processing ». In : *Neuroinformatics* 2.2 (1^{er} juin 2004), p. 183-204. ISSN : 1559-0089. DOI : [10.1385/NI:2:2:183](https://doi.org/10.1385/NI:2:2:183). URL : <https://doi.org/10.1385/NI:2:2:183> (visité le 17/09/2019) (cf. p. 96, 97).

- [189] Viktor K. JIRSA, Kelly J. JANTZEN, Armin FUCHS et al. « Spatiotemporal forward solution of the EEG and MEG using network modeling ». In : *IEEE Transactions on Medical Imaging* 21.5 (2002), p. 493-504. ISSN : 02780062. DOI : [10.1109/TMI.2002.1009385](https://doi.org/10.1109/TMI.2002.1009385) (cf. p. 96).
- [190] Viktor K. JIRSA, William C. STACEY, Pascale P. QUILICHINI et al. « On the nature of seizure dynamics ». In : *Brain* 137.8 (1^{er} août 2014), p. 2210-2230. ISSN : 0006-8950. DOI : [10.1093/brain/awu133](https://doi.org/10.1093/brain/awu133). URL : <https://academic.oup.com/brain/article/137/8/2210/2847958> (visité le 27/02/2018) (cf. p. 115).
- [191] Viktor K. JIRSA et Roxana A. STEFANESCU. « Neural Population Modes Capture Biologically Realistic Large Scale Network Dynamics ». In : *Bulletin of Mathematical Biology* 73.2 (fév. 2011), p. 325-343. ISSN : 0092-8240, 1522-9602. DOI : [10.1007/s11538-010-9573-9](https://doi.org/10.1007/s11538-010-9573-9). URL : <http://link.springer.com/10.1007/s11538-010-9573-9> (visité le 02/02/2018) (cf. p. 97, 99, 103, 107, 109, 116).
- [192] Daniel JOHNSTON et Samuel Miao-sin WU. *Foundations of cellular neurophysiology*. Cambridge, Mass : MIT Press, 1995. 676 p. ISBN : 978-0-262-10053-3 (cf. p. 33).
- [193] Renaud JOLIVET, Ryota KOBAYASHI, Alexander RAUCH et al. « A benchmark test for a quantitative assessment of simple neuron models ». In : *Journal of Neuroscience Methods* 169.2 (avr. 2008), p. 417-424. ISSN : 01650270. DOI : [10.1016/j.jneumeth.2007.11.006](https://doi.org/10.1016/j.jneumeth.2007.11.006). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0165027007005535> (visité le 17/04/2023) (cf. p. 47, 49).
- [194] Eric R. KANDEL, James H. SCHWARTZ et Thomas M. JESSELL. *Principles of Neural Science*. 4th edition. New York : McGraw-Hill Medical, 5 jan. 2000. 1414 p. ISBN : 978-0-8385-7701-1 (cf. p. 30, 50, 51).
- [195] Michael KARO, Richard LAGERSTROM, Marlys KOHNKE et al. « The Application Level Placement Scheduler ». en. In : mai 2006 (cf. p. 62).
- [196] Rainer KELLER, Edgar GABRIEL, Bettina KRAMMER et al. « Towards Efficient Execution of MPI Applications on the Grid : Porting and Optimization Issues ». en. In : *Journal of Grid Computing* 1.2 (2003), p. 133-149. ISSN : 1570-7873. DOI : [10.1023/B:GRID.0000024071.12177.91](https://doi.org/10.1023/B:GRID.0000024071.12177.91). URL : <http://link.springer.com/10.1023/B:GRID.0000024071.12177.91> (visité le 08/06/2023) (cf. p. 66).
- [197] Christopher M. KIM, Ulrich EGERT et Arvind KUMAR. « Dynamics of multiple interacting excitatory and inhibitory populations with delays ». In : *Physical Review E* 102.2 (12 août 2020). Publisher : American Physical Society, p. 022308. DOI : [10.1103/PhysRevE.102.022308](https://doi.org/10.1103/PhysRevE.102.022308). URL : <https://link.aps.org/doi/10.1103/PhysRevE.102.022308> (visité le 13/12/2022) (cf. p. 97).

- [198] Wolfgang KLIMESCH. « The frequency architecture of brain and brain body oscillations : an analysis ». eng. In : *The European Journal of Neuroscience* 48.7 (oct. 2018), p. 2431-2453. ISSN : 1460-9568. DOI : [10.1111/ejn.14192](https://doi.org/10.1111/ejn.14192) (cf. p. 54).
- [199] Alois KNOLL, Marc-Oliver GEWALTIG, J SANDERS et al. « Neurorobotics : A strategic pillar of the Human Brain Project ». In : *Science Robotics* (2016), p. 2-3 (cf. p. 120).
- [200] Ryota KOBAYASHI, Yasuhiro TSUBO et Shigeru SHINOMOTO. « Made-to-order spiking neuron model equipped with a multi-timescale adaptive threshold ». In : *Frontiers in Computational Neuroscience* 3 (2009). ISSN : 1662-5188. URL : <https://www.frontiersin.org/articles/10.3389/neuro.10.009.2009> (visité le 21/04/2023) (cf. p. 48, 99).
- [201] N. KOENIG et A. HOWARD. « Design and use paradigms for gazebo, an open-source multi-robot simulator ». en. In : *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. T. 3. Sendai, Japan : IEEE, 2004, p. 2149-2154. ISBN : 978-0-7803-8463-7. DOI : [10.1109/IROS.2004.1389727](https://doi.org/10.1109/IROS.2004.1389727). URL : <http://ieeexplore.ieee.org/document/1389727/> (visité le 20/07/2023) (cf. p. 120).
- [202] Stephen H KOSLOW et Steven E HYMAN. « Human Brain Project : A Program for the New Millennium ». en. In : *The Einstein Quarterly Journal of Biology and Medicine* (2000), p. 10 (cf. p. 120).
- [203] Stephen H. KOSLOW et Michael F. HUERTA, éd. *Neuroinformatics : an overview of the Human Brain Project*. en. Progress in neuroinformatics research. Mahwah, N.J : L. Erlbaum, 1997. ISBN : 978-0-8058-2099-7 (cf. p. 120).
- [204] Johannes KÖSTER et Sven RAHMANN. « Snakemake—a scalable bioinformatics workflow engine ». In : *Bioinformatics* 28.19 (oct. 2012), p. 2520-2522. ISSN : 1367-4803. DOI : [10.1093/bioinformatics/bts480](https://doi.org/10.1093/bioinformatics/bts480). URL : <https://doi.org/10.1093/bioinformatics/bts480> (visité le 08/06/2023) (cf. p. 65).
- [205] John W. KRAKAUER, Asif A. GHAZANFAR, Alex GOMEZ-MARIN et al. « Neuroscience Needs Behavior : Correcting a Reductionist Bias ». en. In : *Neuron* 93.3 (fév. 2017), p. 480-490. ISSN : 0896-6273. DOI : [10.1016/j.neuron.2016.12.041](https://doi.org/10.1016/j.neuron.2016.12.041). URL : <https://www.sciencedirect.com/science/article/pii/S0896627316310406> (visité le 13/07/2023) (cf. p. 121).
- [206] Andreas K KREITER et Wolf SINGER. « Stimulus-Dependent Synchronization of Neuronal Responses in the Visual Cortex of the Awake Macaque Monkey ». In : *The Journal of Neuroscience* 76.7 (1996), p. 2381-2396. ISSN : 0270-6474. DOI : [ejn_04040369\[pii\]](https://doi.org/10.1523/JNEUROSCI.0404-96.2016) (cf. p. 96).
- [207] Thomas KREUZ. « Synchronization Measures ». In : mai 2013, p. 97-120. ISBN : 978-1-4398-5330-6. DOI : [10.1201/b14756-8](https://doi.org/10.1201/b14756-8) (cf. p. 54).

- [208] Thomas KREUZ, Julie S. HAAS, Alice MORELLI et al. « Measuring spike train synchrony ». en. In : *Journal of Neuroscience Methods* 165.1 (sept. 2007), p. 151-161. ISSN : 0165-0270. DOI : [10.1016/j.jneumeth.2007.05.031](https://doi.org/10.1016/j.jneumeth.2007.05.031). URL : <https://www.sciencedirect.com/science/article/pii/S0165027007002671> (visité le 23/05/2023) (cf. p. 54).
- [209] Alexandre KUHN, Ad AERTSEN et Stefan ROTTER. « Higher-Order Statistics of Input Ensembles and the Response of Simple Model Neurons ». In : *Neural Computation* 15.1 (jan. 2003), p. 67-101. ISSN : 0899-7667, 1530-888X. DOI : [10.1162/089976603321043702](https://doi.org/10.1162/089976603321043702). URL : <http://www.mitpressjournals.org/doi/10.1162/089976603321043702> (cf. p. 80, 83, 248).
- [210] Pramod KUMBHAR, Michael HINES, Jeremy FOURIAUX et al. « CoreNEURON : An Optimized Compute Engine for the NEURON Simulator ». In : *Frontiers in Neuroinformatics* 13 (2019). ISSN : 1662-5196. URL : <https://www.frontiersin.org/articles/10.3389/fninf.2019.00063> (visité le 12/06/2023) (cf. p. 69).
- [211] Yoshiki KURAMOTO. « Method of Phase Description I ». In : *Chemical Oscillations, Waves, and Turbulence*. Springer Series in Synergetics. Springer, Berlin, Heidelberg, 1984, p. 22-34. DOI : [10.1007/978-3-642-69689-3_3](https://doi.org/10.1007/978-3-642-69689-3_3). URL : https://link-springer-com.lama.univ-amu.fr/chapter/10.1007/978-3-642-69689-3_3 (visité le 22/02/2018) (cf. p. 111, 277, 321).
- [212] Yoshiki KURAMOTO. « Reductive Perturbation Method ». en. In : *Chemical Oscillations, Waves, and Turbulence*. Springer Series in Synergetics. Springer, Berlin, Heidelberg, 1984, p. 5-21. DOI : [10.1007/978-3-642-69689-3_2](https://doi.org/10.1007/978-3-642-69689-3_2). URL : https://link-springer-com.lama.univ-amu.fr/chapter/10.1007/978-3-642-69689-3_2 (visité le 22/02/2018) (cf. p. 54).
- [213] Gregory M. KURTZER, Vanessa SOCHAT et Michael W. BAUER. « Singularity : Scientific containers for mobility of compute ». en. In : *PLOS ONE* 12.5 (mai 2017). Publisher : Public Library of Science, e0177459. ISSN : 1932-6203. DOI : [10.1371/journal.pone.0177459](https://doi.org/10.1371/journal.pone.0177459). URL : <https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0177459> (visité le 08/06/2023) (cf. p. 63).
- [214] Katrina M. KUTCHKO et Flavio FRÖHLICH. « Emergence of Metastable State Dynamics in Interconnected Cortical Networks with Propagation Delays ». In : *PLOS Computational Biology* 9.10 (24 oct. 2013). Publisher : Public Library of Science, e1003304. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1003304](https://doi.org/10.1371/journal.pcbi.1003304). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1003304> (visité le 11/12/2022) (cf. p. 97).
- [215] Yuri A KUZNETSOV. « Elements of Applied Bifurcation Theory, Second Edition ». In : (1995) (cf. p. 30).

- [216] Yuri A. KUZNETSOV. « Andronov-Hopf bifurcation ». In : *Scholarpedia* 1.10 (2 oct. 2006), p. 1858. ISSN : 1941-6016. DOI : [10.4249/scholarpedia.1858](https://doi.org/10.4249/scholarpedia.1858). URL : http://www.scholarpedia.org/article/Andronov-Hopf_bifurcation (visité le 14/03/2023) (cf. p. 29).
- [217] Jean-Philippe LACHAUX, Eugenio RODRIGUEZ, Jacques MARTINERIE et al. « Measuring phase synchrony in brain signals ». en. In : *Human Brain Mapping* 8.4 (1999), p. 194-208. ISSN : 1097-0193. DOI : [10.1002/\(SICI\)1097-0193\(1999\)8:4<194::AID-HBM4>3.0.CO;2-C](https://doi.org/10.1002/(SICI)1097-0193(1999)8:4<194::AID-HBM4>3.0.CO;2-C). URL : <http://onlinelibrary.wiley.com/doi/abs/10.1002/%5C%28SICI%5C%291097-0193%5C%281999%5C%298%5C%3A4%5C%3C194%5C%3A%5C%3AAID-HBM4%5C%3E3.0.CO%5C%3B2-C> (visité le 20/06/2019) (cf. p. 177, 323).
- [218] Claude LACOURSIÈRE et Tomas HÄRDIN. « FMI Go! A simulation runtime environment with a client server architecture over multiple protocols ». In : *the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*. Juill. 2017, p. 653-662. DOI : [10.3384/ecp17132653](https://doi.org/10.3384/ecp17132653). URL : https://ep.liu.se/en/conference-article.aspx?series=ecp&issue=132&Article_No=72 (visité le 12/05/2023) (cf. p. 92).
- [219] Josef LADENBAUER, Moritz AUGUSTIN et Klaus OBERMAYER. « How adaptation currents change threshold, gain, and variability of neuronal spiking ». In : *Journal of Neurophysiology* 111.5 (oct. 2013), p. 939-953. ISSN : 0022-3077. DOI : [10.1152/jn.00586.2013](https://doi.org/10.1152/jn.00586.2013). URL : <http://www.physiology.org/doi/full/10.1152/jn.00586.2013> (visité le 23/02/2018) (cf. p. 55).
- [220] Josef LADENBAUER, Moritz AUGUSTIN, LieJune SHIAU et al. « Impact of Adaptation Currents on Synchronization of Coupled Exponential Integrate-and-Fire Neurons ». In : *PLOS Computational Biology* 8.4 (12 avr. 2012), e1002478. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1002478](https://doi.org/10.1371/journal.pcbi.1002478). URL : <http://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1002478> (visité le 21/02/2018) (cf. p. 49, 55, 99, 110).
- [221] Louis LAPICQUE. « Recherches quantitatives sur l'excitation électrique des nerfs traitée comme une polarisation ». In : *Journal de physiologie et de pathologie générale* 9 (1907), p. 620-635 (cf. p. 47, 48, 99).
- [222] Ed S. LEIN, Michael J. HAWRYLYCZ, Nancy AO et al. « Genome-wide atlas of gene expression in the adult mouse brain ». In : *Nature* 445.7124 (jan. 2007), p. 168-176. ISSN : 0028-0836, 1476-4687. DOI : [10.1038/nature05453](https://doi.org/10.1038/nature05453). URL : <http://www.nature.com/articles/nature05453> (visité le 15/09/2021) (cf. p. 75, 78).
- [223] Jean-Didier LEMARÉCHAL, Maciej JEDYNAK, Lena TREBAUL et al. « A brain atlas of axonal and synaptic delays based on modelling of cortico-cortical evoked potentials ». In : *Brain-A Journal of Neurology* (2021) (cf. p. 96, 116).

- [224] Guohua LI, Joon WOO et Sang LIM. « HPC Cloud Architecture to Reduce HPC Workflow Complexity in Containerized Environments ». In : *Applied Sciences* 11 (jan. 2021), p. 923. DOI : [10.3390/app11030923](https://doi.org/10.3390/app11030923) (cf. p. 63).
- [225] Henrik LINDÉN, Espen HAGEN, Szymon LESKI et al. « LFPy : a tool for biophysical simulation of extracellular potentials generated by detailed model neurons ». In : *Frontiers in Neuroinformatics* 7 (2014). ISSN : 1662-5196. URL : <https://www.frontiersin.org/articles/10.3389/fninf.2013.00041> (visité le 13/06/2023) (cf. p. 69).
- [226] Chen LIU, Guillaume BELLEC, Bernhard VOGGINGER et al. « Memory-Efficient Deep Learning on a SpiNNaker 2 Prototype ». In : *Frontiers in Neuroscience* 12 (2018). ISSN : 1662-453X. URL : <https://www.frontiersin.org/articles/10.3389/fnins.2018.00840> (visité le 13/06/2023) (cf. p. 69).
- [227] Chris M. LONVICK et Tatu YLONEN. *The Secure Shell (SSH) Protocol Architecture*. Request for Comments RFC 4251. Num Pages : 30. Internet Engineering Task Force, jan. 2006. DOI : [10.17487/RFC4251](https://doi.org/10.17487/RFC4251). URL : <https://datatracker.ietf.org/doc/rfc4251> (visité le 01/06/2023) (cf. p. 61).
- [228] Gys LOUBSER. « Becoming transdisciplinary theologians : Wentzel van Huyssteen, Paul Cilliers and Constantine Stanislavski ». en. In : *HTS Theologiese Studies / Theological Studies* 71.3 (mars 2015), 9 pages. ISSN : 2072-8050, 0259-9422. DOI : [10.4102/hts.v71i3.2901](https://doi.org/10.4102/hts.v71i3.2901). URL : <http://www.hts.org.za/index.php/HTS/article/view/2901> (visité le 29/10/2021) (cf. p. 121).
- [229] M LYAPUNOV A. « the general problem of the stability of motion ». In : *Comm. Soc. Math. Kharkow* (1892). URL : <https://cir.nii.ac.jp/crid/1573105975982887424> (visité le 16/02/2023) (cf. p. 26).
- [230] Keyvan MAHJOORY, Jan-Mathijs SCHOFFELEN, Anne KEITEL et al. « The frequency gradient of human resting-state brain oscillations follows cortical hierarchies ». In : *eLife* 9 (août 2020). Sous la dir. de Laura DUGUÉ, Laura L COLGIN et Laura DUGUÉ. Publisher : eLife Sciences Publications, Ltd, e53715. ISSN : 2050-084X. DOI : [10.7554/eLife.53715](https://doi.org/10.7554/eLife.53715). URL : <https://doi.org/10.7554/eLife.53715> (visité le 23/05/2023) (cf. p. 54).
- [231] Tiina MANNINEN, Riikka HAVELA et Marja-Leena LINNE. « Computational Models for Calcium-Mediated Astrocyte Functions ». In : *Frontiers in Computational Neuroscience* 12 (avr. 2018). DOI : [10.3389/fncom.2018.00014](https://doi.org/10.3389/fncom.2018.00014) (cf. p. 51).
- [232] D MANTINI, MG G PERRUCCI, Del GRATTA et al. « Electrophysiological signatures of resting state networks in the human brain ». In : *Proc Natl Acad Sci* 104.32 (2007), p. 13170-13175. DOI : [10.1073/pnas.0700668104](https://doi.org/10.1073/pnas.0700668104). URL : <http://www.pnas.org/cgi/content/abstract/104/32/13170> (cf. p. 96).
- [233] Henry MARKRAM. « The Blue Brain Project ». en. In : *Nature Reviews Neuroscience* 7.2 (fév. 2006). Number : 2 Publisher : Nature Publishing Group, p. 153-160. ISSN : 1471-0048. DOI : [10.1038/nrn1848](https://doi.org/10.1038/nrn1848). URL : <https://www.nature.com/articles/nrn1848> (visité le 12/06/2023) (cf. p. 67).

- [234] Henry MARKRAM. « The Human Brain Project ». en. In : *Scientific American* (2012), p. 6 (cf. p. 23, 66, 120).
- [235] Henry MARKRAM, Karlheinz MEIER, Thomas LIPPERT et al. « Introducing the Human Brain Project ». en. In : *Procedia Computer Science*. Proceedings of the 2nd European Future Technologies Conference and Exhibition 2011 (FET 11) 7 (jan. 2011), p. 39-42. ISSN : 1877-0509. DOI : [10.1016/j.procs.2011.12.015](https://doi.org/10.1016/j.procs.2011.12.015). URL : <https://www.sciencedirect.com/science/article/pii/S1877050911006806> (visité le 09/08/2022) (cf. p. 23, 66, 120).
- [236] Megan L. MATTHEWS et Amy MARSHALL-COLÓN. « Multiscale plant modeling : from genome to phenome and beyond ». In : *Emerging Topics in Life Sciences* 5.2 (5 fév. 2021). Sous la dir. de Joseph M. JEZ et Christopher N. TOPP, p. 231-237. ISSN : 2397-8554. DOI : [10.1042/ETLS20200276](https://doi.org/10.1042/ETLS20200276). URL : <https://doi.org/10.1042/ETLS20200276> (visité le 10/05/2022) (cf. p. 74).
- [237] Jil M. MEIER, Dionysios PERDIKIS, André BLICKENSDÖRFER et al. « Virtual deep brain stimulation : Multiscale co-simulation of a spiking basal ganglia model and a whole-brain mean-field model with The Virtual Brain ». In : *Experimental Neurology* 354 (1^{er} août 2022), p. 114111. ISSN : 0014-4886. DOI : [10.1016/j.expneurol.2022.114111](https://doi.org/10.1016/j.expneurol.2022.114111). URL : <https://www.sciencedirect.com/science/article/pii/S0014488622001364> (visité le 24/03/2023) (cf. p. 74).
- [238] Martin MEIER-SCHELLERSHEIM, Iain D. C. FRASER et Frederick KLAUSCHEN. « Multiscale modeling for biologists ». In : *Wiley Interdisciplinary Reviews : Systems Biology and Medicine* 1.1 (2009), p. 4-14. ISSN : 1939-005X. DOI : [10.1002/wsbm.33](https://doi.org/10.1002/wsbm.33). URL : <http://onlinelibrary.wiley.com/doi/abs/10.1002/wsbm.33> (visité le 30/09/2019) (cf. p. 73, 115).
- [239] Francesca MELOZZI, Eyal BERGMANN, Julie A. HARRIS et al. « Individual structural features constrain the mouse functional connectome ». en. In : *Proceedings of the National Academy of Sciences* 116.52 (déc. 2019), p. 26961-26969. ISSN : 0027-8424, 1091-6490. DOI : [10.1073/pnas.1906694116](https://doi.org/10.1073/pnas.1906694116). URL : <https://www.pnas.org/content/116/52/26961> (visité le 24/01/2020) (cf. p. 57).
- [240] Francesca MELOZZI, Marmaduke M. WOODMAN, Viktor K. JIRSA et al. « The Virtual Mouse Brain : A Computational Neuroinformatics Platform to Study Whole Mouse Brain Dynamics ». In : *eNeuro* 4.3 (1^{er} mai 2017), ENEURO.0111-17.2017. ISSN : 2373-2822. DOI : [10.1523/ENEURO.0111-17.2017](https://doi.org/10.1523/ENEURO.0111-17.2017). URL : <http://www.eneuro.org/content/4/3/ENEURO.0111-17.2017> (cf. p. 86, 240).
- [241] MESSAGE PASSING INTERFACE FORUM. *MPI : A Message-Passing Interface Standard Version 3.1*. 2015. URL : <https://www.mpi-forum.org/docs/mpi-3.1/mpi31-report.pdf> (cf. p. 92).

- [242] M. MIGLIORE, C. CANNIA, W.W LYTTON et al. « Parallel Network Simulations with NEURON ». In : *Journal of computational neuroscience* 21.2 (oct. 2006), p. 119-129. ISSN : 0929-5313. DOI : [10.1007/s10827-006-7949-5](https://doi.org/10.1007/s10827-006-7949-5). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2655137/> (visité le 12/06/2023) (cf. p. 69).
- [243] Ştefan MIHALAŞ et Ernst NIEBUR. « A Generalized Linear Integrate-and-Fire Neural Model Produces Diverse Spiking Behaviors ». In : *Neural computation* 21.3 (mars 2009), p. 704-718. ISSN : 0899-7667. DOI : [10.1162/neco.2008.12-07-680](https://doi.org/10.1162/neco.2008.12-07-680). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC2954058/> (visité le 21/04/2023) (cf. p. 48).
- [244] Ben MITCHINSON, Tak-Shing CHAN, Jon CHAMBERS et al. « BRAHMS : Novel middleware for integrated systems computation ». In : *Advanced Engineering Informatics*. Informatics for cognitive robots 24.1 (1^{er} jan. 2010), p. 49-61. ISSN : 1474-0346. DOI : [10.1016/j.aei.2009.08.002](https://doi.org/10.1016/j.aei.2009.08.002). URL : <https://www.sciencedirect.com/science/article/pii/S1474034609000457> (visité le 02/01/2023) (cf. p. 74).
- [245] Abraham De MOIVRE. « De mensura sortis, seu, de probabilitate eventuum in ludis a casu fortuito pendentibus ». In : *Philosophical Transactions of the Royal Society of London* 27.329 (1711). Publisher : Royal Society, p. 213-264. DOI : [10.1098/rstl.1710.0018](https://doi.org/10.1098/rstl.1710.0018). URL : <https://royalsocietypublishing.org/doi/10.1098/rstl.1710.0018> (visité le 24/04/2023) (cf. p. 53).
- [246] Ernest MONTBRIÓ, Diego PAZÓ et Alex ROXIN. « Macroscopic Description for Networks of Spiking Neurons ». In : *Physical Review X* 5.2 (juin 2015). DOI : [10.1103/physrevx.5.021028](https://doi.org/10.1103/physrevx.5.021028). URL : <https://doi.org/10.1103/physrevx.5.021028> (cf. p. 185).
- [247] Alfonso MONTUORI. « Complexity and Transdisciplinarity : Reflection on Theory and Practice ». In : *World Futures* 69 (jan. 2011), p. 200-230. DOI : [10.1080/02604027.2013.803349](https://doi.org/10.1080/02604027.2013.803349) (cf. p. 121).
- [248] Jan MOREN, Norikazu SUGIMOTO et Kenji DOYA. « Real-time Utilization of System-scale Neuroscience Models ». In : *The Brain & Neural Networks* 22.3 (2015), p. 125-132. DOI : [10.3902/jnns.22.125](https://doi.org/10.3902/jnns.22.125) (cf. p. 92).
- [249] Noga MOSHEIFF, Bard ERMENTROUT et Chengcheng HUANG. « Chaotic dynamics in spatially distributed neuronal networks generate population-wide shared variability ». In : *PLOS Computational Biology* 19.1 (10 jan. 2023), e1010843. ISSN : 1553-734X. DOI : [10.1371/journal.pcbi.1010843](https://doi.org/10.1371/journal.pcbi.1010843). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC9870129/> (visité le 25/04/2023) (cf. p. 103).
- [250] *MPICH ABI Compatibility Initiative* | MPICH. en-US. URL : <https://www.mpich.org/abi/> (visité le 08/06/2023) (cf. p. 66).
- [251] *MPICH Overview* | MPICH. en-US. URL : <https://www.mpich.org/about/overview/> (visité le 08/06/2023) (cf. p. 66).

- [252] *MVAPICH* : : *Overview*. URL : <https://mvapich.cse.ohio-state.edu/overview/> (visité le 08/06/2023) (cf. p. 66).
- [253] J. NAGUMO, S. ARIMOTO et S. YOSHIZAWA. « An Active Pulse Transmission Line Simulating Nerve Axon ». In : *Proceedings of the IRE* 50.10 (oct. 1962). Conference Name : Proceedings of the IRE, p. 2061-2070. ISSN : 2162-6634. DOI : [10.1109/JRPROC.1962.288235](https://doi.org/10.1109/JRPROC.1962.288235) (cf. p. 47, 99).
- [254] Richard NAUD, Nicolas MARCILLE, Claudia CLOPATH et al. « Firing patterns in the adaptive exponential integrate-and-fire model ». In : *Biological Cybernetics* 99.4 (1^{er} nov. 2008), p. 335. ISSN : 0340-1200, 1432-0770. DOI : [10.1007/s00422-008-0264-7](https://doi.org/10.1007/s00422-008-0264-7). URL : <https://link-springer-com.lama.univ-amu.fr/article/10.1007/s00422-008-0264-7> (visité le 15/02/2018) (cf. p. 35, 46, 47, 49, 99).
- [255] Himanshu NEEMA, Jesse GOHL, Zsolt LATTMANN et al. « Model-Based Integration Platform for FMI Co-Simulation and Heterogeneous Simulations of Cyber-Physical Systems ». In : *the 10th International Modelica Conference, March 10-12, 2014, Lund, Sweden*. Mars 2014, p. 235-245. DOI : [10.3384/ecp14096235](https://doi.org/10.3384/ecp14096235). URL : https://ep.liu.se/en/conference-article.aspx?series=ecp&issue=96&Article_No=24 (visité le 16/05/2023) (cf. p. 92).
- [256] *Neuromodulation*. en. Page Version ID : 1136090946. Jan. 2023. URL : <https://en.wikipedia.org/w/index.php?title=Neuromodulation&oldid=1136090946> (visité le 23/04/2023) (cf. p. 50).
- [257] *Neurotransmitter*. en. Page Version ID : 1150258312. Avr. 2023. URL : <https://en.wikipedia.org/w/index.php?title=Neurotransmitter&oldid=1150258312> (visité le 23/04/2023) (cf. p. 50).
- [258] Van Hoa NGUYEN, Yvon BESANGER, Quoc Tuan TRAN et al. « On Conceptual Structuration and Coupling Methods of Co-Simulation Frameworks in Cyber-Physical Energy System Validation ». In : *Energies* 10.12 (déc. 2017). Number : 12 Publisher : Multidisciplinary Digital Publishing Institute, p. 1977. DOI : [10.3390/en10121977](https://doi.org/10.3390/en10121977). URL : <https://www.mdpi.com/1996-1073/10/12/1977> (visité le 13/04/2020) (cf. p. 93, 113).
- [259] Basarab NICOLESCU. *Manifesto of transdisciplinarity*. suny Press, 2002 (cf. p. 121).
- [260] Yuval NIR, Lior FISCH, Roy MUKAMEL et al. « Coupling between Neuronal Firing Rate, Gamma LFP, and BOLD fMRI Is Related to Interneuronal Correlations ». In : *Current Biology* 17.15 (2007), p. 1275-1285. ISSN : 09609822. DOI : [10.1016/j.cub.2007.06.066](https://doi.org/10.1016/j.cub.2007.06.066) (cf. p. 96).
- [261] Eilen NORDLIE, Marc-Oliver GEWALTIG et Hans Ekkehard PLESSER. « Towards Reproducible Descriptions of Neuronal Network Models ». In : *PLOS Computational Biology* 5.8 (7 août 2009). Publisher : Public Library of Science, e1000456. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1000456](https://doi.org/10.1371/journal.pcbi.1000456). URL :

- <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1000456> (visité le 15/01/2021) (cf. p. 86, 172).
- [262] Duane Q. NYKAMP et Daniel TRANCHINA. In : *Journal of Computational Neuroscience* 8.1 (2000), p. 19-50. DOI : [10.1023/a:1008912914816](https://doi.org/10.1023/a:1008912914816). URL : <https://doi.org/10.1023/a:1008912914816> (cf. p. 185).
- [263] Seung Wook OH, Julie A. HARRIS, Lydia NG et al. « A mesoscale connectome of the mouse brain ». In : *Nature* 508.7495 (avr. 2014), p. 207-214. ISSN : 1476-4687. DOI : [10.1038/nature13186](https://doi.org/10.1038/nature13186). URL : <https://www.nature.com/articles/nature13186> (cf. p. 57, 76, 78, 86, 241).
- [264] *ONLYOFFICE Docs Enterprise Edition*. URL : <https://office.ebrains.eu/welcome/> (visité le 13/06/2023) (cf. p. 69).
- [265] Richard R. OSMER. *Practical Theology : An Introduction*. en. Google-Books-ID : dOXW_ua4ZEgC. Wm. B. Eerdmans Publishing, juill. 2008. ISBN : 978-0-8028-1765-5 (cf. p. 121).
- [266] Srdjan OSTOJIC et Nicolas BRUNEL. « From Spiking Neuron Models to Linear-Nonlinear Models ». en. In : *PLOS Computational Biology* 7.1 (jan. 2011). Publisher : Public Library of Science, e1001056. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1001056](https://doi.org/10.1371/journal.pcbi.1001056). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1001056> (visité le 30/08/2022) (cf. p. 55, 57).
- [267] Eustace PAINKRAS, Luis A. PLANA, Jim GARSIDE et al. « SpiNNaker : A 1-W 18-Core System-on-Chip for Massively-Parallel Neural Network Simulation ». In : *IEEE Journal of Solid-State Circuits* 48.8 (août 2013). Conference Name : IEEE Journal of Solid-State Circuits, p. 1943-1953. ISSN : 1558-173X. DOI : [10.1109/JSSC.2013.2259038](https://doi.org/10.1109/JSSC.2013.2259038) (cf. p. 69).
- [268] Agostina PALMIGIANO, Theo GEISEL, Fred WOLF et al. « Flexible information routing by transient synchrony ». In : *Nature Neuroscience* 20.7 (2017), p. 1014-1022. ISSN : 15461726. DOI : [10.1038/nn.4569](https://doi.org/10.1038/nn.4569) (cf. p. 96).
- [269] Christian PEHLE, Sebastian BILLAUDELLE, Benjamin CRAMER et al. « The BrainScaleS-2 Accelerated Neuromorphic System With Hybrid Plasticity ». In : *Frontiers in Neuroscience* 16 (fév. 2022), p. 795876. ISSN : 1662-4548. DOI : [10.3389/fnins.2022.795876](https://doi.org/10.3389/fnins.2022.795876). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8907969/> (visité le 13/06/2023) (cf. p. 69).
- [270] Spase PETKOSKI et Viktor K. JIRSA. « Transmission time delays organize the brain network synchronization ». In : *Philosophical Transactions of the Royal Society A : Mathematical, Physical and Engineering Sciences* 377.2153 (2019), p. 20180132. ISSN : 1364503X. DOI : [10.1098/rsta.2018.0132](https://doi.org/10.1098/rsta.2018.0132). URL : <https://royalsocietypublishing.org/doi/10.1098/rsta.2018.0132> (cf. p. 96, 116).

- [271] Spase PETKOSKI et Viktor K. JIRSA. « Normalizing the brain connectome for communication through synchronization ». In : *Network Neuroscience* 6.3 (jan. 2022), p. 722-744. ISSN : 24721751. DOI : [10.1162/netn_a_00231](https://doi.org/10.1162/netn_a_00231). URL : https://doi.org/10.1162/netn_a_00231 (cf. p. 96).
- [272] Spase PETKOSKI, J. Matias PALVA et Viktor K. JIRSA. « Phase-lags in large scale brain synchronization : Methodological considerations and in-silico analysis ». In : *PLoS Computational Biology* 14.7 (2018), p. 1-30. ISSN : 15537358. DOI : [10.1371/journal.pcbi.1006160](https://doi.org/10.1371/journal.pcbi.1006160). URL : <https://doi.org/10.1371/journal.pcbi.1006160> (cf. p. 96).
- [273] Spase PETKOSKI, Andreas SPIEGLER, Timothée PROIX et al. « Heterogeneity of time delays determines synchronization of coupled oscillators ». In : *Physical Review E* 94.1 (11 juill. 2016). Publisher : American Physical Society, p. 012209. DOI : [10.1103/PhysRevE.94.012209](https://link.aps.org/doi/10.1103/PhysRevE.94.012209). URL : <https://link.aps.org/doi/10.1103/PhysRevE.94.012209> (visité le 25/04/2023) (cf. p. 96, 103, 109).
- [274] R.R. PFEIFFER et N.Y-S. KIANG. « Spike Discharge Patterns of Spontaneous and Continuously Stimulated Activity in the Cochlear Nucleus of Anesthetized Cats ». In : *Biophysical Journal* 5.3 (mai 1965), p. 301-316. ISSN : 00063495. DOI : [10.1016/S0006-3495\(65\)86718-2](https://linkinghub.elsevier.com/retrieve/pii/S0006349565867182). URL : <https://linkinghub.elsevier.com/retrieve/pii/S0006349565867182> (visité le 11/04/2023) (cf. p. 34).
- [275] Arkady PIKOVSKY, Michael ROSENBLUM et Jürgen KURTHS. *Synchronization : a universal concept in nonlinear sciences*. Cambridge University Press, 2001. ISBN : 052153352X (cf. p. 109).
- [276] Ajay S. PILLAI et Viktor K. JIRSA. « Symmetry Breaking in Space-Time Hierarchies Shapes Brain Dynamics and Behavior ». In : *Neuron* 94.5 (juin 2017), p. 1010-1026. ISSN : 0896-6273. DOI : [10.1016/j.neuron.2017.05.013](https://www.sciencedirect.com/science/article/pii/S0896627317304142). URL : <https://www.sciencedirect.com/science/article/pii/S0896627317304142> (visité le 02/06/2023) (cf. p. 73).
- [277] *Piz Daint* | CSCS. URL : <https://www.cscs.ch/computers/piz-daint> (visité le 08/06/2023) (cf. p. 62).
- [278] H. POINCARÉ. « Mémoire sur les courbes définies par une équation différentielle ». In : *Journal de Mathématiques Pures et Appliquées* 7 (1881), p. 375-422. ISSN : 0021-7824. URL : http://www.numdam.org/item/?id=JMPA_1881_3_7__375_0 (visité le 17/02/2023) (cf. p. 26, 28, 29).
- [279] H. POINCARÉ. « Sur le problème des trois corps et les équations de la dynamique ». In : *Acta Mathematica* 13.1 (1890), p. 5-7. ISSN : 0001-5962. DOI : [10.1007/BF02392506](http://projecteuclid.org/euclid.acta/1485881727). URL : <http://projecteuclid.org/euclid.acta/1485881727> (visité le 17/02/2023) (cf. p. 28).

- [280] Siméon-Denis (1781-1840) Auteur du texte POISSON. *Recherches sur la probabilité des jugements en matière criminelle et en matière civile / par S.-D. Poisson,...* EN. 1837. URL : <https://gallica.bnf.fr/ark:/12148/bpt6k110193z> (visité le 24/04/2023) (cf. p. 53).
- [281] Tobias C. POTJANS et Markus DIESMANN. « The Cell-Type Specific Cortical Microcircuit : Relating Structure and Activity in a Full-Scale Spiking Network Model ». In : *Cerebral Cortex (New York, NY)* 24.3 (mars 2014), p. 785-806. ISSN : 1047-3211. DOI : [10.1093/cercor/bhs358](https://doi.org/10.1093/cercor/bhs358). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3920768/> (visité le 02/02/2018) (cf. p. 53).
- [282] Timothée PROIX, Viktor K. JIRSA, Fabrice BARTOLOMEI et al. « Predicting the spatiotemporal diversity of seizure propagation and termination in human focal epilepsy ». In : *Nature Communications* 9.1 (14 mars 2018). Number : 1 Publisher : Nature Publishing Group, p. 1088. ISSN : 2041-1723. DOI : [10.1038/s41467-018-02973-y](https://doi.org/10.1038/s41467-018-02973-y). URL : <https://www.nature.com/articles/s41467-018-02973-y> (visité le 27/04/2023) (cf. p. 96).
- [283] *Project Jupyter*. en. URL : <https://jupyter.org> (visité le 14/06/2023) (cf. p. 69).
- [284] Paulo R. PROTACHEVICZ, Fernando S. BORGES, Kelly C. IAROSZ et al. « Influence of Delayed Conductance on Neuronal Synchronization ». English. In : *Frontiers in Physiology* 11 (2020). ISSN : 1664-042X. DOI : [10.3389/fphys.2020.01053](https://doi.org/10.3389/fphys.2020.01053). URL : <https://www.frontiersin.org/articles/10.3389/fphys.2020.01053/full> (visité le 19/02/2021) (cf. p. 55).
- [285] Paulo R. PROTACHEVICZ, Fernando S. BORGES, Ewandson L. LAMEU et al. « Bistable Firing Pattern in a Neural Network Model ». English. In : *Frontiers in Computational Neuroscience* 13 (2019). ISSN : 1662-5188. DOI : [10.3389/fncom.2019.00019](https://doi.org/10.3389/fncom.2019.00019). URL : <https://www.frontiersin.org/articles/10.3389/fncom.2019.00019/full> (visité le 19/02/2021) (cf. p. 55).
- [286] Paulo R. PROTACHEVICZ, Kelly C. IAROSZ, Iberê L. CALDAS et al. « Influence of Autapses on Synchronization in Neural Networks With Chemical Synapses ». English. In : *Frontiers in Systems Neuroscience* 14 (2020). ISSN : 1662-5137. DOI : [10.3389/fnsys.2020.604563](https://doi.org/10.3389/fnsys.2020.604563). URL : <https://www.frontiersin.org/articles/10.3389/fnsys.2020.604563/full> (visité le 19/02/2021) (cf. p. 53, 55).
- [287] Dale PURVES, George J. AUGUSTINE, David FITZPATRICK et al., éd. *Neuroscience*. New York, 12 oct. 2017. 960 p. ISBN : 978-1-60535-380-7 (cf. p. 30, 32).
- [288] Martin PYKA, Sebastian KLATT et Sen CHENG. « Parametric Anatomical Modeling : a method for modeling the anatomical layout of neurons and their projections ». In : *Frontiers in Neuroanatomy* 8 (2014). ISSN : 1662-5129. URL : <https://www.frontiersin.org/articles/10.3389/fnana.2014.00091> (visité le 24/04/2023) (cf. p. 53).

- [289] Mohammad Mamunur RAHMAN, Yusheng FENG, Thomas E. YANKEELOV et al. « A fully coupled space–time multiscale modeling framework for predicting tumor growth ». In : *Computer Methods in Applied Mechanics and Engineering* 320 (15 juin 2017), p. 261-286. ISSN : 0045-7825. DOI : [10.1016/j.cma.2017.03.021](https://doi.org/10.1016/j.cma.2017.03.021). URL : <https://www.sciencedirect.com/science/article/pii/S004578251631249X> (visité le 29/04/2022) (cf. p. 73).
- [290] W RALL. « Distinguishing theoretical synaptic potentials computed for different soma-dendritic distributions of synaptic input. » en. In : *Journal of Neurophysiology* 30.5 (sept. 1967), p. 1138-1168. ISSN : 0022-3077, 1522-1598. DOI : [10.1152/jn.1967.30.5.1138](https://doi.org/10.1152/jn.1967.30.5.1138). URL : <https://www.physiology.org/doi/10.1152/jn.1967.30.5.1138> (visité le 23/04/2023) (cf. p. 52).
- [291] Subhasis RAY, Raamesh DESHPANDE, Niraj DUDANI et al. « A general biological simulator : the multiscale object oriented simulation environment, MOOSE ». en. In : *BMC Neuroscience* 9.1 (juill. 2008), P93. ISSN : 1471-2202. DOI : [10.1186/1471-2202-9-S1-P93](https://doi.org/10.1186/1471-2202-9-S1-P93). URL : <https://doi.org/10.1186/1471-2202-9-S1-P93> (visité le 24/07/2023) (cf. p. 120).
- [292] Khawaja Ubaid REHMAN, Kh REHMAN et Muhammad ASHRAF. « A Comparative Analysis of Distributed and Parallel Computing ». In : 13 (fév. 2019), p. 60-67. DOI : [10.21015/vtse.v13i2.507](https://doi.org/10.21015/vtse.v13i2.507) (cf. p. 60).
- [293] Katarzyna A. REJNIAK et Alexander R. A. ANDERSON. « Hybrid Models of Tumor Growth ». In : *Wiley interdisciplinary reviews. Systems biology and medicine* 3.1 (2011), p. 115-125. ISSN : 1939-5094. DOI : [10.1002/wsbm.102](https://doi.org/10.1002/wsbm.102). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC3057876/> (visité le 22/11/2021) (cf. p. 73).
- [294] Aline F. RENZ, Jihyun LEE, Klas TYBRANDT et al. « Opto-E-Dura : A Soft, Stretchable ECoG Array for Multimodal, Multiscale Neuroscience ». In : *Advanced Healthcare Materials* 9.17 (2020). _eprint : <https://onlinelibrary.wiley.com/doi/pdf/10.1002/adhm.202000814>. ISSN : 2192-2659. DOI : <https://doi.org/10.1002/adhm.202000814>. URL : <http://onlinelibrary.wiley.com/doi/abs/10.1002/adhm.202000814> (cf. p. 74, 76).
- [295] *Research – Blue Brain’s data-driven approach.* en-GB. URL : <https://www.epfl.ch/research/domains/bluebrain/research-2/> (visité le 12/06/2023) (cf. p. 67).
- [296] C. ROCȘOREANU, A. GEORGESCU et N. GIURGIȚEANU. *The FitzHugh-Nagumo Model*. Réd. par R. LOWEN. T. 10. Mathematical Modelling : Theory and Applications. Dordrecht : Springer Netherlands, 2000. DOI : [10.1007/978-94-015-9548-3](https://doi.org/10.1007/978-94-015-9548-3). URL : <http://link.springer.com/10.1007/978-94-015-9548-3> (visité le 17/04/2023) (cf. p. 47).

- [297] Alex ROXIN, Nicolas BRUNEL, David HANSEL et al. « On the Distribution of Firing Rates in Networks of Cortical Neurons ». In : *The Journal of Neuroscience* 31.45 (nov. 2011), p. 16217-16226. DOI : [10.1523/jneurosci.1677-11.2011](https://doi.org/10.1523/jneurosci.1677-11.2011). URL : <https://doi.org/10.1523/jneurosci.1677-11.2011> (cf. p. 179).
- [298] Kurt SAMSON. « WILL THE DEMISE OF THE NIH HUMAN BRAIN PROJECT HURT THE GROWTH OF NEUROINFORMATICS? » en-US. In : *Neurology Today* 6.5 (mars 2006), p. 34. ISSN : 1533-7006. URL : https://journals.lww.com/neurotodayonline/Fulltext/2006/03070/WILL_THE_DEMISE_OF_THE_NIH_HUMAN_BRAIN_PROJECT.11.aspx (visité le 09/08/2022) (cf. p. 120).
- [299] Paula SANZ LEON, Stuart A. KNOCK, M. Marmaduke WOODMAN et al. « The Virtual Brain : a simulator of primate brain network dynamics ». English. In : *Frontiers in Neuroinformatics* 7 (2013). ISSN : 1662-5196. DOI : [10.3389/fninf.2013.00010](https://doi.org/10.3389/fninf.2013.00010). URL : <https://www.frontiersin.org/articles/10.3389/fninf.2013.00010/full> (visité le 28/08/2018) (cf. p. 69, 72, 74-76, 85, 86, 113, 120, 188, 206, 230, 257).
- [300] Paula SANZ-LEON, Stuart A. KNOCK, Andreas SPIEGLER et al. « Mathematical framework for large-scale brain network modeling in The Virtual Brain ». In : *NeuroImage* 111 (mai 2015), p. 385-430. ISSN : 1053-8119. DOI : [10.1016/j.neuroimage.2015.01.002](https://doi.org/10.1016/j.neuroimage.2015.01.002). URL : <http://www.sciencedirect.com/science/article/pii/S1053811915000051> (visité le 14/03/2018) (cf. p. 69, 77, 87, 96, 113, 175, 225, 331).
- [301] Randy SAUNDERS. *IEEE Standard for Modeling and Simulation (M & S) High Level Architecture (HLA)– Framework and Rules*. Conference Name : IEEE Std 1516-2010 (Revision of IEEE Std 1516-2000). Août 2010. DOI : [10.1109/IEEESTD.2010.5553440](https://doi.org/10.1109/IEEESTD.2010.5553440) (cf. p. 92).
- [302] Johannes SCHEMMEL, Daniel BRIIDERLE, Andreas GRIIBL et al. « A wafer-scale neuromorphic hardware system for large-scale neural modeling ». en. In : *Proceedings of 2010 IEEE International Symposium on Circuits and Systems*. Paris, France : IEEE, mai 2010, p. 1947-1950. ISBN : 978-1-4244-5308-5. DOI : [10.1109/ISCAS.2010.5536970](https://doi.org/10.1109/ISCAS.2010.5536970). URL : <http://ieeexplore.ieee.org/document/5536970/> (visité le 13/06/2023) (cf. p. 69).
- [303] Michael SCHIRNER, Lia DOMIDE, Dionysios PERDIKIS et al. « Brain simulation as a cloud service : The Virtual Brain on EBRAINS ». en. In : *NeuroImage* 251 (mai 2022), p. 118973. ISSN : 1053-8119. DOI : [10.1016/j.neuroimage.2022.118973](https://doi.org/10.1016/j.neuroimage.2022.118973). URL : <https://www.sciencedirect.com/science/article/pii/S1053811922001021> (visité le 30/11/2022) (cf. p. 68, 69, 74, 91, 120).
- [304] Tamar SCHLICK, Stephanie PORTILLO-LEDESMA, Mischa BLASZCZYK et al. « A MULTISCALE VISION-ILLUSTRATIVE APPLICATIONS FROM BIOLOGY TO ENGINEERING ». In : *International Journal for Multiscale Computational Engineering* 19.2 (2021), p. 39-73. ISSN : 1543-1649. DOI : [10.1615/IntJMCompEng.2021039845](https://doi.org/10.1615/IntJMCompEng.2021039845). URL : <http://www.dl.begellhouse.com/journals/61fd1b191cf7e96f,6cd7873d3aec47b1,75086009651d6035.html> (visité le 05/11/2021) (cf. p. 73).

- [305] Maximilian SCHMIDT, Rembrandt BAKKER, Claus C. HILGETAG et al. « Multi-scale account of the network structure of macaque visual cortex ». en. In : *Brain Structure and Function* 223.3 (avr. 2018), p. 1409-1435. ISSN : 1863-2661. DOI : [10.1007/s00429-017-1554-4](https://doi.org/10.1007/s00429-017-1554-4). URL : <https://doi.org/10.1007/s00429-017-1554-4> (visité le 08/03/2019) (cf. p. 53).
- [306] Maximilian SCHMIDT, Rembrandt BAKKER, Kelly SHEN et al. « A multi-scale layer-resolved spiking network model of resting-state dynamics in macaque visual cortical areas ». en. In : *PLOS Computational Biology* 14.10 (oct. 2018), e1006359. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1006359](https://doi.org/10.1371/journal.pcbi.1006359). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1006359> (visité le 17/04/2019) (cf. p. 53).
- [307] Valentin SCHMUTZ, Wulfram GERSTNER et Tilo SCHWALGER. « Mesoscopic population equations for spiking neural networks with synaptic short-term plasticity ». In : *The Journal of Mathematical Neuroscience* 10.1 (avr. 2020). DOI : [10.1186/s13408-020-00082-z](https://doi.org/10.1186/s13408-020-00082-z). URL : <https://doi.org/10.1186/s13408-020-00082-z> (cf. p. 185).
- [308] Steffen SCHNEIDER, Jin Hwa LEE et Mackenzie Weygandt MATHIS. « Learnable latent embeddings for joint behavioural and neural analysis ». en. In : *Nature* 617.7960 (mai 2023). Number : 7960 Publisher : Nature Publishing Group, p. 360-368. ISSN : 1476-4687. DOI : [10.1038/s41586-023-06031-6](https://doi.org/10.1038/s41586-023-06031-6). URL : <https://www.nature.com/articles/s41586-023-06031-6> (visité le 23/05/2023) (cf. p. 54).
- [309] Tilo SCHWALGER et Anton V CHIZHOV. « Mind the last spike — firing rate models for mesoscopic populations of spiking neurons ». In : *Current Opinion in Neurobiology* 58 (oct. 2019), p. 155-166. DOI : [10.1016/j.conb.2019.08.003](https://doi.org/10.1016/j.conb.2019.08.003). URL : <https://doi.org/10.1016/j.conb.2019.08.003> (cf. p. 185).
- [310] *Science Vision*. en. URL : <https://www.ebrains.eu/page/science-vision> (visité le 14/06/2023) (cf. p. 23, 68, 120).
- [311] *Seafile - Open Source File Sync and Share Software*. URL : <https://www.seafile.com/en/features/> (visité le 13/06/2023) (cf. p. 69).
- [312] *Second Generation Wiki (XWiki.org)*. en. URL : <https://www.xwiki.org/xwiki/bin/view/Documentation/UserGuide/Features/SecondGenerationWiki/> (visité le 13/06/2023) (cf. p. 69).
- [313] Johanna SENK, Karolína KORVASOVÁ, Jannis SCHUECKER et al. « Conditions for wave trains in spiking neural networks ». In : *Physical Review Research* 2.2 (14 mai 2020). Publisher : American Physical Society, p. 023174. DOI : [10.1103/PhysRevResearch.2.023174](https://doi.org/10.1103/PhysRevResearch.2.023174). URL : <https://link.aps.org/doi/10.1103/PhysRevResearch.2.023174> (visité le 01/03/2022) (cf. p. 107, 109).

- [314] Johanna SENK, Birgit KRIENER, Mikael DJURFELDT et al. « Connectivity concepts in neuronal network modeling ». en. In : *PLOS Computational Biology* 18.9 (sept. 2022). Publisher : Public Library of Science, e1010086. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.1010086](https://doi.org/10.1371/journal.pcbi.1010086). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.1010086> (visité le 24/04/2023) (cf. p. 53).
- [315] Hina SHAHEEN, Swadesh PAL et Roderick MELNIK. « Multiscale co-simulation of deep brain stimulation with brain networks in neurodegenerative disorders ». In : *Brain Multiphysics* 3 (1^{er} jan. 2022), p. 100058. ISSN : 2666-5220. DOI : [10.1016/j.brain.2022.100058](https://doi.org/10.1016/j.brain.2022.100058). URL : <https://www.sciencedirect.com/science/article/pii/S2666522022000156> (visité le 24/03/2023) (cf. p. 74).
- [316] Ram SHANMUGAM. « Neuroinformatics : An overview of the Human Brain Project : Stephen H. Koslow and Michael F. Huerta (Eds.) ; Lawrence Erlbaum Associates Press, Mahwah, NJ, 1997, pp. 376. ISBN 0-8058-2099-X ». en. In : *Neurocomputing* 41.1 (oct. 2001), p. 195. ISSN : 0925-2312. DOI : [10.1016/S0925-2312\(01\)00333-2](https://doi.org/10.1016/S0925-2312(01)00333-2). URL : <https://www.sciencedirect.com/science/article/pii/S0925231201003332> (visité le 09/08/2022) (cf. p. 120).
- [317] D. F. SHANNO. « Conditioning of quasi-Newton methods for function minimization ». en. In : *Mathematics of Computation* 24.111 (1970), p. 647-656. ISSN : 0025-5718, 1088-6842. DOI : [10.1090/S0025-5718-1970-0274029-X](https://doi.org/10.1090/S0025-5718-1970-0274029-X). URL : <https://www.ams.org/mcom/1970-24-111/S0025-5718-1970-0274029-X/> (visité le 20/06/2023) (cf. p. 176).
- [318] L. W. SHEPPARD, A. C. HALE, S. PETKOSKI et al. « Characterizing an ensemble of interacting oscillators : The mean-field variability index ». In : *Physical Review E - Statistical, Nonlinear, and Soft Matter Physics* 87.1 (2013), p. 1-11. ISSN : 15393755. DOI : [10.1103/PhysRevE.87.012905](https://doi.org/10.1103/PhysRevE.87.012905) (cf. p. 96).
- [319] LieJune SHIAU et Laure BUHRY. « Interneuronal gamma oscillations in hippocampus via adaptive exponential integrate-and-fire neurons ». en. In : *Neurocomputing* 331 (fév. 2019), p. 220-234. ISSN : 0925-2312. DOI : [10.1016/j.neucom.2018.11.017](https://doi.org/10.1016/j.neucom.2018.11.017). URL : <https://www.sciencedirect.com/science/article/pii/S0925231218313456> (visité le 11/05/2023) (cf. p. 55).
- [320] Shigeru SHINOMOTO, Keiji MIURA et Shinsuke KOYAMA. « A measure of local variation of inter-spike intervals ». In : *Biosystems. Selection of Papers from the 5th International Workshop on Neural Coding* 79.1 (1^{er} jan. 2005), p. 67-72. ISSN : 0303-2647. DOI : [10.1016/j.biosystems.2004.09.023](https://doi.org/10.1016/j.biosystems.2004.09.023). URL : <http://www.sciencedirect.com/science/article/pii/S0303264704001662> (visité le 04/07/2019) (cf. p. 34, 277, 321).
- [321] Shigeru SHINOMOTO, Keisetsu SHIMA et Jun TANJI. « Differences in Spiking Patterns Among Cortical Neurons ». In : *Neural Computation* 15.12 (1^{er} déc. 2003), p. 2823-2842. ISSN : 0899-7667, 1530-888X. DOI : [10.1162/089976603322518759](https://doi.org/10.1162/089976603322518759). URL : <https://direct.mit.edu/neco/article/15/12/2823-2842/6785> (visité le 11/04/2023) (cf. p. 34, 277, 321).

- [322] Tristan SHUMAN, Daniel AHARONI, Denise J. CAI et al. « Breakdown of spatial coding and interneuron synchronization in epileptic mice ». In : *Nature neuroscience* 23.2 (fév. 2020), p. 229-238. ISSN : 1097-6256. DOI : [10.1038/s41593-019-0559-0](https://doi.org/10.1038/s41593-019-0559-0). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC7259114/> (visité le 25/09/2020) (cf. p. 77, 78, 238).
- [323] Abraham SILBERSCHATZ, Greg GAGNE et Peter B. GALVIN. *Operating System Concepts, 10th Edition*. English. Wiley, mai 2018 (cf. p. 61).
- [324] Viktor SIP, Meysam HASHEMI, Timo DICKSCHEID et al. « Characterization of regional differences in resting-state fMRI with a data-driven network model of brain dynamics ». In : *Science Advances* 9.11 (2023). ISSN : 23752548. DOI : [10.1126/sciadv.abq7547](https://doi.org/10.1126/sciadv.abq7547) (cf. p. 96).
- [325] *Slurm Workload Manager - Quick Start User Guide*. URL : <https://slurm.schedmd.com/quickstart.html> (visité le 01/06/2023) (cf. p. 62).
- [326] A. SPIEGLER et V. JIRSA. « Systematic approximations of neural fields through networks of neural masses in the virtual brain ». en. In : *NeuroImage* 83 (déc. 2013), p. 704-725. ISSN : 10538119. DOI : [10.1016/j.neuroimage.2013.06.018](https://doi.org/10.1016/j.neuroimage.2013.06.018). URL : <https://linkinghub.elsevier.com/retrieve/pii/S1053811913006551> (visité le 23/05/2023) (cf. p. 56).
- [327] Andreas SPIEGLER, Javad Karimi ABADCHI, Majid MOHAJERANI et al. « In silico exploration of mouse brain dynamics by focal stimulation reflects the organization of functional networks and sensory processing ». In : *Network Neuroscience* 4.3 (1^{er} sept. 2020), p. 807-851. ISSN : 2472-1751. DOI : [10.1162/netn_a_00152](https://doi.org/10.1162/netn_a_00152). URL : https://doi.org/10.1162/netn_a_00152 (visité le 19/03/2021) (cf. p. 96).
- [328] Olaf SPORNS, Giulio TONONI et Rolf KÖTTER. « The Human Connectome : A Structural Description of the Human Brain ». en. In : *PLOS Computational Biology* 1.4 (sept. 2005). Publisher : Public Library of Science, e42. ISSN : 1553-7358. DOI : [10.1371/journal.pcbi.0010042](https://doi.org/10.1371/journal.pcbi.0010042). URL : <https://journals.plos.org/ploscompbiol/article?id=10.1371/journal.pcbi.0010042> (visité le 23/05/2023) (cf. p. 56, 120).
- [329] Sebastian SPREIZER, Johanna SENK, Stefan ROTTER et al. « NEST Desktop, an Educational Application for Neuroscience ». In : *eNeuro* 8.6 (nov. 2021), ENEURO.0274-21.2021. ISSN : 2373-2822. DOI : [10.1523/ENEURO.0274-21.2021](https://doi.org/10.1523/ENEURO.0274-21.2021). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC8638679/> (visité le 12/06/2023) (cf. p. 69).
- [330] Jeffrey M. SQUYRES et Andrew LUMSDAINE. « A Component Architecture for LAM/MPI ». en. In : *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Sous la dir. de Gerhard GOOS, Juris HARTMANIS, Jan VAN LEEUWEN et al. T. 2840. Series Title : Lecture Notes in Computer Science. Berlin, Heidelberg : Springer Berlin Heidelberg, 2003, p. 379-387. DOI : [10.1007/978-](https://doi.org/10.1007/978-)

- 3-540-39924-7_52. URL : http://link.springer.com/10.1007/978-3-540-39924-7_52 (visité le 08/06/2023) (cf. p. 66).
- [331] *Stability theory*. In : *Wikipedia*. Page Version ID : 1091270115. 3 juin 2022. URL : https://en.wikipedia.org/w/index.php?title=Stability_theory&oldid=1091270115 (visité le 14/03/2023) (cf. p. 27).
- [332] Roxana A. STEFANESCU et Viktor K. JIRSA. « A low dimensional description of globally coupled heterogeneous neural networks of excitatory and inhibitory neurons ». In : *PLoS Computational Biology* 4.11 (2008). ISSN : 1553734X. DOI : [10.1371/journal.pcbi.1000219](https://doi.org/10.1371/journal.pcbi.1000219) (cf. p. 96).
- [333] Stefanos S. STEFANOPOULOS, George KASTELLAKIS et Panayiota POIRAZI. « Creating and Constraining Compartmental Models of Neurons Using Experimental Data ». In : *Advanced Patch-Clamp Analysis for Neuroscientists*. Sous la dir. d'Alon KORNGREEN. Neuromethods. New York, NY : Springer, 2016, p. 325-343. ISBN : 978-1-4939-3411-9. DOI : [10.1007/978-1-4939-3411-9_15](https://doi.org/10.1007/978-1-4939-3411-9_15). URL : https://doi.org/10.1007/978-1-4939-3411-9_15 (visité le 24/02/2023) (cf. p. 32).
- [334] Marcel STIMBERG, Romain BRETTE et Dan FM GOODMAN. « Brian 2, an intuitive and efficient neural simulator ». In : *eLife* 8 (août 2019). Sous la dir. de Frances K SKINNER, e47314. ISSN : 2050-084X. DOI : [10.7554/eLife.47314](https://doi.org/10.7554/eLife.47314) (cf. p. 74, 186).
- [335] Steven H. STROGATZ. *Nonlinear dynamics and Chaos : with applications to physics, biology, chemistry, and engineering*. Studies in nonlinearity. Reading, Mass : Addison-Wesley Pub, 1994. 498 p. ISBN : 978-0-201-54344-5 (cf. p. 28).
- [336] Susan SUNKIN. « Electrophysiology ». In : (2017) (cf. p. 46).
- [337] Ellika TAVERES-CACHAT, Fabio FAVOINO, Roel LOONEN et al. « Ten questions concerning co-simulation for performance prediction of advanced building envelopes ». In : *Building and Environment* 191 (15 mars 2021), p. 107570. ISSN : 0360-1323. DOI : [10.1016/j.buildenv.2020.107570](https://doi.org/10.1016/j.buildenv.2020.107570). URL : <https://www.sciencedirect.com/science/article/pii/S0360132320309379> (visité le 10/11/2021) (cf. p. 229).
- [338] Corinne TEETER, Ramakrishnan IYER, Vilas MENON et al. « Generalized leaky integrate-and-fire models classify multiple neuron types ». In : *Nature Communications* 9.1 (19 fév. 2018). Number : 1 Publisher : Nature Publishing Group, p. 709. ISSN : 2041-1723. DOI : [10.1038/s41467-017-02717-4](https://doi.org/10.1038/s41467-017-02717-4). URL : <https://www.nature.com/articles/s41467-017-02717-4> (visité le 21/04/2023) (cf. p. 48, 49).
- [339] Federico TESLER, Marja-Leena LINNE et Alain DESTEXHE. « Modeling the relationship between neuronal activity and the BOLD signal : contributions from astrocyte calcium dynamics ». en. In : *Scientific Reports* 13.1 (avr. 2023). Number : 1 Publisher : Nature Publishing Group, p. 6451. ISSN : 2045-2322.

- DOI : [10.1038/s41598-023-32618-0](https://doi.org/10.1038/s41598-023-32618-0). URL : <https://www.nature.com/articles/s41598-023-32618-0> (visité le 28/05/2023) (cf. p. 171).
- [340] Federico TESLER, Núria TORT-COLET, Damien DEPANNEMAECCKER et al. « Mean-field based framework for forward modeling of LFP and MEG signals ». en. In : *Frontiers in Computational Neuroscience* 16 (oct. 2022), p. 968278. ISSN : 1662-5188. DOI : [10.3389/fncom.2022.968278](https://doi.org/10.3389/fncom.2022.968278). URL : <https://www.frontiersin.org/articles/10.3389/fncom.2022.968278/full> (visité le 28/05/2023) (cf. p. 171).
- [341] *The Human Brain Project | HBP Project | Fact Sheet | FP7*. en. URL : <https://cordis.europa.eu/project/id/604102> (visité le 12/06/2023) (cf. p. 66).
- [342] CORPORATE THE MPI FORUM. « MPI : a message passing interface ». In : *Proceedings of the 1993 ACM/IEEE conference on Supercomputing*. Supercomputing '93. New York, NY, USA : Association for Computing Machinery, déc. 1993, p. 878-883. ISBN : 978-0-8186-4340-8. DOI : [10.1145/169627.169855](https://doi.org/10.1145/169627.169855). URL : <https://dl.acm.org/doi/10.1145/169627.169855> (visité le 08/06/2023) (cf. p. 65, 114).
- [343] Núria TORT-COLET, Francesco RESTA, Elena MONTAGNI et al. « Assessing brain state and anesthesia level with two-photon calcium signals ». In : *Scientific Reports* 13.1 (fév. 2023). DOI : [10.1038/s41598-023-30224-8](https://doi.org/10.1038/s41598-023-30224-8). URL : <https://doi.org/10.1038/s41598-023-30224-8> (cf. p. 179).
- [344] Jonathan TOUBOUL. « Bifurcation Analysis of a General Class of Nonlinear Integrate-and-Fire Neurons ». In : *SIAM Journal on Applied Mathematics* 68.4 (2008), p. 1045-1079. ISSN : 0036-1399. URL : <http://www.jstor.org.lama.univ-amu.fr/stable/40233794> (visité le 27/02/2018) (cf. p. 99, 110).
- [345] Jonathan TOUBOUL et Romain BRETTE. « Dynamics and bifurcations of the adaptive exponential integrate-and-fire model ». In : *Biological Cybernetics* 99.4 (1^{er} nov. 2008), p. 319. ISSN : 0340-1200, 1432-0770. DOI : [10.1007/s00422-008-0267-4](https://doi.org/10.1007/s00422-008-0267-4). URL : <https://link-springer-com.lama.univ-amu.fr/article/10.1007/s00422-008-0267-4> (visité le 22/02/2018) (cf. p. 46, 47, 49).
- [346] Thomas P. TRAPPENBERG. *Fundamentals of computational neuroscience*. 2nd ed. OCLC : ocn444383805. Oxford; New York : Oxford University Press, 2010. 390 p. ISBN : 978-0-19-956841-3 (cf. p. 31, 33).
- [347] Henry C. TUCKWELL. *Introduction to Theoretical Neurobiology : Volume 2 : Non-linear and Stochastic Theories*. T. 2. Cambridge Studies in Mathematical Biology. Cambridge : Cambridge University Press, 1988. ISBN : 978-0-521-35217-8. DOI : [10.1017/CB09780511623202](https://doi.org/10.1017/CB09780511623202). URL : <https://www.cambridge.org/core/books/introduction-to-theoretical-neurobiology/5B00C4410746818CE64451F201CD1> (visité le 13/12/2022) (cf. p. 99).

- [348] Tomas VAN POTTELBERGH, Guillaume DRION et Rodolphe SEPULCHRE. « Robust Modulation of Integrate-and-Fire Models ». In : *Neural Computation* 30.4 (avr. 2018), p. 987-1011. ISSN : 1530-888X. DOI : [10.1162/neco_a_01065](https://doi.org/10.1162/neco_a_01065) (cf. p. 49).
- [349] Tomas VAN POTTELBERGH, Guillaume DRION et Rodolphe SEPULCHRE. « From Biophysical to Integrate-and-Fire Modeling ». In : *Neural Computation* 33.3 (1^{er} mars 2021), p. 563-589. ISSN : 0899-7667. DOI : [10.1162/neco_a_01353](https://doi.org/10.1162/neco_a_01353). URL : https://doi.org/10.1162/neco_a_01353 (visité le 21/04/2023) (cf. p. 48).
- [350] Francisco VARELA, Jean-Philippe LACHAUX, Eugenio RODRIGUEZ et al. « The brainweb : Phase synchronization and large-scale integration ». In : *Nature Reviews Neuroscience* 2.4 (avr. 2001), p. 229-239. ISSN : 1471-003X. DOI : [10.1038/35067550](https://doi.org/10.1038/35067550). URL : <http://www.nature.com/articles/35067550> (cf. p. 96).
- [351] Juan A. VARELA, Kamal SEN, Jay GIBSON et al. « A Quantitative Description of Short-Term Plasticity at Excitatory Synapses in Layer 2/3 of Rat Primary Visual Cortex ». en. In : *Journal of Neuroscience* 17.20 (oct. 1997). Publisher : Society for Neuroscience Section : Articles, p. 7926-7940. ISSN : 0270-6474, 1529-2401. DOI : [10.1523/JNEUROSCI.17-20-07926.1997](https://doi.org/10.1523/JNEUROSCI.17-20-07926.1997). URL : <https://www.jneurosci.org/content/17/20/7926> (visité le 23/04/2023) (cf. p. 51).
- [352] Pauli VIRTANEN, Ralf GOMMERS, Travis E. OLIPHANT et al. « SciPy 1.0 : Fundamental Algorithms for Scientific Computing in Python ». In : *Nature Methods* 17 (2020), p. 261-272. DOI : [10.1038/s41592-019-0686-2](https://doi.org/10.1038/s41592-019-0686-2) (cf. p. 176).
- [353] Julien VITAY, Helge DINKELBACH et Fred HAMKER. « ANNarchy : a code generation approach to neural simulations on parallel hardware ». In : *Frontiers in Neuroinformatics* 9 (2015). ISSN : 1662-5196. URL : <https://www.frontiersin.org/articles/10.3389/fninf.2015.00019> (visité le 07/04/2023) (cf. p. 74).
- [354] Michiel van der VLAG, Marmaduke WOODMAN, Jan FOUSEK et al. « RateML : A Code Generation Tool for Brain Network Models ». In : *Frontiers in Network Physiology* 2 (2022). ISSN : 2674-0109. URL : <https://www.frontiersin.org/articles/10.3389/fnetp.2022.826345> (visité le 12/06/2023) (cf. p. 69).
- [355] Nicole VOGES et Laurent U. PERRINET. « Complex dynamics in recurrent cortical networks based on spatially realistic connectivities ». In : *Frontiers in Computational Neuroscience* 6 (2012). ISSN : 1662-5188. DOI : [10.3389/fncom.2012.00041](https://doi.org/10.3389/fncom.2012.00041). URL : <https://www.frontiersin.org/articles/10.3389/fncom.2012.00041/full> (visité le 15/05/2018) (cf. p. 99).
- [356] Matteo di VOLO, Alberto ROMAGNONI, Cristiano CAPONE et al. « Biologically Realistic Mean-Field Models of Conductance-Based Networks of Spiking Neurons with Adaptation ». In : *Neural Computation* 31.4 (fév. 2019), p. 653-680. ISSN : 0899-7667. DOI : [10.1162/neco_a_01173](https://doi.org/10.1162/neco_a_01173). URL : https://www.nature.com/articles/10.1162/neco_a_01173

- mitpressjournals.org/doi/10.1162/neco_a_01173 (visité le 04/10/2019) (cf. p. 55, 58, 76, 86, 87, 116, 117, 170, 171, 173, 174, 181, 185, 186, 188, 242, 331).
- [357] Udo WEHMEIER, Dawei DONG, Christof KOCH et al. « Modeling the mammalian visual system ». In : *Methods in neuronal modeling : From synapses to networks*. Cambridge, MA, USA : MIT Press, 1^{er} août 1989, p. 335-359. ISBN : 978-0-262-11133-1. (Visité le 21/04/2023) (cf. p. 48).
- [358] P. WELCH. « The use of fast Fourier transform for the estimation of power spectra : A method based on time averaging over short, modified periodograms ». In : *IEEE Transactions on Audio and Electroacoustics* 15.2 (juin 1967), p. 70-73. ISSN : 0018-9278. DOI : [10.1109/TAU.1967.1161901](https://doi.org/10.1109/TAU.1967.1161901). URL : <http://ieeexplore.ieee.org/document/1161901/> (visité le 05/12/2022) (cf. p. 276, 277, 321, 322).
- [359] *What is a Container? | Docker*. en-US. Nov. 2021. URL : <https://www.docker.com/resources/what-container/> (visité le 08/06/2023) (cf. p. 63).
- [360] *What we do | INCF*. URL : <https://www.incf.org/about/what-we-do> (visité le 20/07/2023) (cf. p. 114, 120).
- [361] Stefan WIELAND, Davide BERNARDI, Tilo SCHWALGER et al. « Slow fluctuations in recurrent networks of spiking neurons ». en. In : *Physical Review E* 92.4 (oct. 2015), p. 040901. ISSN : 1539-3755, 1550-2376. DOI : [10.1103/PhysRevE.92.040901](https://doi.org/10.1103/PhysRevE.92.040901). URL : <https://link.aps.org/doi/10.1103/PhysRevE.92.040901> (visité le 20/06/2023) (cf. p. 176, 323).
- [362] Elmer G. WIENS. *Systems of Linear Differential Equations : Linear Stability Analysis*. URL : <http://www.egwald.ca/linearalgebra/lineardifferentialequationsstability.php> (visité le 17/02/2023) (cf. p. 26, 27).
- [363] Mark D. WILKINSON, Michel DUMONTIER, IJsbrand Jan AALBERSBERG et al. « The FAIR Guiding Principles for scientific data management and stewardship ». en. In : *Scientific Data* 3.1 (mars 2016). Number : 1 Publisher : Nature Publishing Group, p. 160018. ISSN : 2052-4463. DOI : [10.1038/sdata.2016.18](https://doi.org/10.1038/sdata.2016.18). URL : <https://www.nature.com/articles/sdata201618> (visité le 09/06/2023) (cf. p. 67).
- [364] H. R. WILSON et J. D. COWAN. « A mathematical theory of the functional dynamics of cortical and thalamic nervous tissue ». en. In : *Kybernetik* 13.2 (sept. 1973), p. 55-80. ISSN : 0023-5946, 1432-0770. DOI : [10.1007/BF00288786](https://doi.org/10.1007/BF00288786). URL : <https://link-springer-com.lama.univ-amu.fr/article/10.1007/BF00288786> (visité le 15/03/2018) (cf. p. 56).
- [365] Hugh R. WILSON et Jack D. COWAN. « Excitatory and Inhibitory Interactions in Localized Populations of Model Neurons ». In : *Biophysical Journal* 12.1 (jan. 1972), p. 1-24. DOI : [10.1016/s0006-3495\(72\)86068-5](https://doi.org/10.1016/s0006-3495(72)86068-5). URL : [https://doi.org/10.1016/s0006-3495\(72\)86068-5](https://doi.org/10.1016/s0006-3495(72)86068-5) (cf. p. 185).

- [366] Minka WOERMANN et Paul CILLIERS. « The ethics of complexity and the complexity of ethics ». In : *South African Journal of Philosophy* 31.2 (jan. 2012). Publisher : Routledge _eprint : <https://doi.org/10.1080/02580136.2012.10751787>, p. 447-463. ISSN : 0258-0136. DOI : [10 . 1080 / 02580136 . 2012 . 10751787](https://doi.org/10.1080/02580136.2012.10751787). URL : <https://doi.org/10.1080/02580136.2012.10751787> (visité le 25/07/2023) (cf. p. 121).
- [367] Kay Jan WONG. *Multithreading and Multiprocessing in 10 Minutes*. en. Mai 2023. URL : <https://towardsdatascience.com/multithreading-and-multiprocessing-in-10-minutes-20d9b3c6a867> (visité le 15/06/2023) (cf. p. 64).
- [368] Xiaoyue WU, Congping LIN et Yiwei ZHANG. *Spiking hierarchy in an adaptive exponential integrate-and-fire network synchronization*. en. arXiv :2112.05983 [math]. Déc. 2021. URL : <http://arxiv.org/abs/2112.05983> (visité le 18/07/2022) (cf. p. 55).
- [369] Anastasia YENDIKI, Manisha AGGARWAL, Markus AXER et al. « Post mortem mapping of connectional anatomy for the validation of diffusion MRI ». en. In : *NeuroImage* 256 (août 2022), p. 119146. ISSN : 1053-8119. DOI : [10.1016/j.neuroimage.2022.119146](https://doi.org/10.1016/j.neuroimage.2022.119146). URL : <https://www.sciencedirect.com/science/article/pii/S1053811922002737> (visité le 23/05/2023) (cf. p. 57).
- [370] M. K. Stephen YEUNG et Steven H. STROGATZ. « Time Delay in the Kuramoto Model of Coupled Oscillators ». In : *Physical Review Letters* 82.3 (18 jan. 1999), p. 648-651. ISSN : 0031-9007, 1079-7114. DOI : [10.1103/PhysRevLett.82.648](https://doi.org/10.1103/PhysRevLett.82.648). URL : <https://link.aps.org/doi/10.1103/PhysRevLett.82.648> (visité le 08/11/2022) (cf. p. 103, 104, 109).
- [371] Pierre YGER, Sami EL BOUSTANI, Alain DESTEXHE et al. « Topologically invariant macroscopic statistics in balanced networks of conductance-based integrate-and-fire neurons ». In : *Journal of Computational Neuroscience* 31.2 (oct. 2011), p. 229-245. ISSN : 0929-5313, 1573-6873. DOI : [10.1007/s10827-010-0310-z](https://doi.org/10.1007/s10827-010-0310-z). URL : <http://link.springer.com/10.1007/s10827-010-0310-z> (visité le 13/12/2022) (cf. p. 111).
- [372] Andy B. YOO, Morris A. JETTE et Mark GRONDONA. « SLURM : Simple Linux Utility for Resource Management ». en. In : *Job Scheduling Strategies for Parallel Processing*. Sous la dir. de Dror FEITELSON, Larry RUDOLPH et Uwe SCHWIEGELSHOHN. Lecture Notes in Computer Science. Berlin, Heidelberg : Springer, 2003, p. 44-60. ISBN : 978-3-540-39727-4. DOI : [10.1007/10968987_3](https://doi.org/10.1007/10968987_3) (cf. p. 61, 62).
- [373] Y. ZERLAUT, B. TELEŃCZUK, C. DELEUZE et al. « Heterogeneous firing rate response of mouse layer V pyramidal neurons in the fluctuation-driven regime ». In : *The Journal of Physiology* 594.13 (juill. 2016), p. 3791-3808. ISSN : 0022-3751. DOI : [10.1113/JP272317](https://doi.org/10.1113/JP272317). URL : <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC4929333/> (visité le 22/03/2018) (cf. p. 55).

- [374] Yann ZERLAUT, Sandrine CHEMLA, Frederic CHAVANE et al. « Modeling mesoscopic cortical dynamics using a mean-field model of conductance-based networks of adaptive exponential integrate-and-fire neurons ». en. In : *Journal of Computational Neuroscience* 44.1 (fév. 2018), p. 45-61. ISSN : 0929-5313, 1573-6873. DOI : [10.1007/s10827-017-0668-2](https://doi.org/10.1007/s10827-017-0668-2). URL : <http://link.springer.com/article/10.1007/s10827-017-0668-2> (visité le 24/04/2018) (cf. p. 55, 58, 111, 171, 175, 177, 186).
- [375] Yann ZERLAUT et Alain DESTEXHE. « Enhanced responsiveness and low-level awareness in stochastic network states ». In : *Neuron* 94.5 (2017), p. 1002-1009 (cf. p. 171, 174, 181, 186).
- [376] Y. G. ZHENG et Z. H. WANG. « Network-scale effect on synchronizability of fully coupled network with connection delay ». In : *Chaos : An Interdisciplinary Journal of Nonlinear Science* 26.4 (avr. 2016), p. 043103. ISSN : 1054-1500, 1089-7682. DOI : [10.1063/1.4946812](https://doi.org/10.1063/1.4946812). URL : <http://aip.scitation.org/doi/10.1063/1.4946812> (visité le 07/12/2022) (cf. p. 97, 109).

Troisième partie
ANNEX

CHAPTER A

Dynamics of mean-field model of adaptive exponential integrate-and-fire network

Contents of Chapter A

A.1. Chapter overview	170
A.2. Introduction	171
A.3. Models and methods	172
A.3.1. AdEx network	172
A.3.2. Mean-field of AdEx	173
A.3.3. Transfer function fitting	175
A.3.4. Bifurcation analysis	176
A.3.5. Spike train analysis	176
A.3.6. Mean-field analysis	177
A.4. Results	177
A.4.1. Steady-state and transfer function	177
A.4.2. Dynamical comparison	181
A.5. Discussion	185

A.1. Chapter overview

In this chapter, I aim to investigate data transformation from meso- to macro- scale. Brain study requires integrating information from different scales, in particular, to study emergent phenomena. However, there is no systematic characterization of the approximation for the Mean Ad Ex [356]. We will describe the result of the comparison between mean-field and associate spiking neural networks.

This chapter is based on preliminary results :

Dynamics of mean-field model of adaptive exponential integrate-and-fire network

Lionel Kusch¹, Damien Depannemaecker¹, Viktor Jirsa¹

¹Institut de Neurosciences des Systèmes (INS) UMR1106, Aix-Marseille Univ., Marseilles, 13005, France

Abstract

The study of brain activity spans diverse scales and levels of description, necessitating the development of computational models alongside experimental investigations to explore integrations across scales. The high dimensionality of spiking networks presents challenges for understanding their dynamics. To tackle this, a mean-field formulation offers a potential approach for dimensionality reduction while retaining essential elements. This study centres on a previously developed mean-field model, extensively utilized in various research works. However, a systematic investigation of its properties has not been published before. Therefore, this paper aims to provide a comprehensive description and characterization of the model to assist future users in interpreting their results. The methodology includes model construction, stability analysis, and numerical simulations. Finally, we offer an overview of dynamical properties and methods to characterize a mean-field model useful for further use of this model.

A.2. Introduction

The study of brain activity is a complex and multifaceted field that involves examining different scales and levels of description. In addition to experimental work in neuroscience, computational models have been developed to investigate brain activity at different scales, ranging from molecular to whole-brain levels. These models of different natures can have different functions, such as explicative or predictive. In this paper, we are interested in models reproducing dynamical aspects of neuron membrane excitability, networked through a synaptic communication model. These networks are made up of an excitatory population and an inhibitory population in similar proportions to what is observed in the cortex. These networks are useful for studying complex dynamics associated with the large dimensions of these systems. However, having a large number of dimensions can be a limiting factor for the understanding of the representation of its dynamics. But also, when going to a larger spatial scale to constitute a network model of cerebral regions [140], if keeping this level of description, the system becomes too complex to be understood and too computationally demanding to be simulated. In these cases, it may be interesting to try to reduce the dimensions of the system, trying to keep key elements to achieve the function of the model. A possible way of reduction is the formulation of a mean-field. The present study focuses on the study of a mean-field previously developed [106, 375, 356] and already used in different studies [56, 140, 356, 374, 50, 51, 339, 96, 340, 140]. Such approaches may focus on the proper dynamics of its corresponding spiking network [51, 356, 106, 50] or on experimental results [96, 374, 56] or, to clinically observable measures [140, 340, 339]. However, despite a wide range of uses, no systematic study of the mean-field model properties was published previously. Therefore, we propose a description and characterization intended for future users of this model and thus to help them interpret their results. We propose a methodology, and the results are given

for one specific mean-field derivation (i.e. on a specific fit of the transfer function, see models and methods section); each user may need to apply these approaches to characterize its specific model. In the "model and methods" section, we detail the construction of this model, and then in the "stability analysis" section, we detail important elements of the structure underlying the dynamics. Then, we propose a characterization of its implementation in numerical simulations. Finally, in a discussion, we situate this model in the broad panorama of mean-fields, explaining its advantages and limits.

A.3. Models and methods

In this section, we first describe the Adaptive Exponential Integrate-and-Fire model [37] that constituted the spiking network, and then the corresponding mean-field model. The equations' details and the parameters' values are available in Supplementary Tables G.1 and G.2 based on the proposition of Nordlie et al. [261].

A.3.1 AdEx network

A network of 10 000 spiking neurons is constructed with a probability of connection of 5%, according to a sparse and random (Erdos-Renyi type) architecture. 20% constituted an inhibitory (FS) population, and 80% an excitatory (RS) population, accordingly to the cortex and previously modelled [51]. The dynamics of a single neuron membrane excitability is described by the AdEx model [37] corresponding to the equation (A.1)

$$C_m \frac{dV_m}{dt} = -g_L(V_m - E_L) + g_L \Delta_T e^{\left(\frac{V_m - V_{th}}{\Delta_T}\right)} - w + I_{syn} \quad (\text{A.1})$$

$$\tau_w \frac{dw}{dt} = a(V_m - E_L) - w$$

When an action potential is emitted (i.e. the membrane potential crosses a threshold), the system is reset as in the equation (A.2) :

$$\text{if } V_m \geq V_{peak} \text{ then } \begin{cases} V \rightarrow V_{reset} \\ w \rightarrow w + b \end{cases} \quad (\text{A.2})$$

The interaction into the network is through synaptic current (equation (A.3)).

$$I_{syn} = -g_E(V_m - E_E) - g_I(V_m - E_I) \quad (\text{A.3})$$

where $E_E = 0$ mV and $E_I = -80$ mV are, respectively, the reversal potentials of excitatory synapses and inhibitory synapses. For each incoming spike, the excitatory and inhibitory conductances g_E and g_I are increased respectively by the value $Q_E = 1.5$ nS

and $Q_I = 5$ nS. The conductances decrease following an exponential decay with a time constant $\tau_{syn} = 5$ ms according to the equation (A.4).

$$\frac{dg_{E/I}}{dt} = -\frac{g_{E/I}}{\tau_{syn}} \quad (\text{A.4})$$

The network is simulated using the simulator NEST [152].

A.3.2 Mean-field of AdEx

Following the formalism proposed by El Boustani and Destexhe [106], di Volo and colleagues have proposed a mean-field of the previously described spiking network [356]. The differential equations describe the evolution of the firing rate v_μ of each population $\mu = e, i$ (equation (A.5)), the covariance $c_{\lambda\eta}$ between population $\lambda\eta$ (equation (A.6)), and, of the average adaptation for the excitatory population W (equation (A.7)).

$$T \frac{dv_\mu}{dt} = (F_\mu - v_\mu) + \frac{1}{2} c_{\lambda\eta} \frac{\partial^2 F_\mu}{\partial v_\lambda \partial v_\eta} \quad (\text{A.5})$$

$$\begin{aligned} T \frac{dc_{\lambda\eta}}{dt} = & \delta_{\lambda\eta} \frac{F_\lambda(1/T - F_\eta)}{N_\lambda} + (F_\lambda - v_\lambda)(F_\eta - v_\eta) \\ & + \frac{\partial F_\lambda}{\partial v_\mu} c_{\eta\mu} + \frac{\partial F_\eta}{\partial v_\mu} c_{\lambda\mu} - 2c_{\lambda\eta}, \end{aligned} \quad (\text{A.6})$$

$$\frac{\partial W}{\partial t} = -W/\tau_w + bv_e + a(\mu_V(v_e, v_i, W) - E_L) \quad (\text{A.7})$$

The parameter T , the time constant of the firing rate equations and covariance equations, comes from an essential assumption for this derivation : the network dynamics is considered as Markovian on an infinitesimal (time resolution T) [106]. It is thus considered that each neuron emits a maximum of one spike into the time step T . Therefore, the value of T strongly constrains the maximum firing rate that can be considered according to these hypotheses. This question will be addressed in the discussion.

The parameters b , a et E_L in the equation (A.7), directly correspond to the parameters named the same in the AdEx model [356] ((A.1)).

And, the function $F_{e,i}$ relates to the transfer functions of the excitatory and inhibitory population and should be understood as $F_e(v_e + v_{ext}, v_i)$, where v_{ext} correspond to a constant value of the firing rate an external drive. The first step is to measure the output firing rate of the single neuron model while receiving excitatory and inhibitory inputs. These measures are necessary because, due to the nonlinearity of the dynamics

A. Dynamics of mean-field model of adaptive exponential integrate-and-fire network –
A.3. Models and methods

of the single neuron model, it cannot be inferred analytically, as well as determining when the system will irretrievably produce a spike. Thus, it is considered an effective or phenomenological threshold, $V_{\text{thre}}^{\text{eff}}$, a function of the statistics of the subthreshold membrane voltage dynamics. This statistic is assumed to be normally distributed, with the average membrane voltage μ_V , its standard deviation σ_V and auto-correlation time τ_V ; their calculation is described below. The effective threshold function is considered as a second-order polynomial giving the equation (A.8), where $\tau_V^N = \tau_V G_l / C$. The $\{P\}$ values are computationally found through fitting methods.

$$V_{\text{thre}}^{\text{eff}}(\mu_V, \sigma_V, \tau_V^N) = P_0 + \sum_{x \in \{\mu_V, \sigma_V, \tau_V^N\}} P_x \cdot \left(\frac{x - x^0}{\delta x^0} \right) + \sum_{x, y \in \{\mu_V, \sigma_V, \tau_V^N\}^2} P_{xy} \cdot \left(\frac{x - x^0}{\delta x^0} \right) \left(\frac{y - y^0}{\delta y^0} \right), \quad (\text{A.8})$$

Then a semi-analytic approach enables us to fit the transfer function (equation (A.9)), where erfc is the Gauss error function, and thus obtain the output firing rate.

$$F_V = \frac{1}{2\tau_V} \text{erfc} \left(\frac{V_{\text{thre}}^{\text{eff}} - \mu_V}{\sqrt{2}\sigma_V} \right), \quad (\text{A.9})$$

Considering asynchronous irregular regimes, we assume that the input spike trains follow the Poissonian statistics [356, 106, 375] and we calculate the averages (μ_{G_e, G_i}) and standard deviations (σ_{G_e, G_i}) of the conductances in the equations A.10. In these equations, K_e and K_i are the average input connectivity received respectively from the excitatory or inhibitory population. As in the spiking network, $\tau_e = \tau_i = \tau_{\text{syn}}$ are synaptic time constants, and Q_e and Q_i the quantal increment of the conductances, respectively, for the excitatory or inhibitory populations.

$$\begin{aligned} \mu_{G_e}(v_e, v_i) &= v_e K_e \tau_e Q_e, \\ \sigma_{G_e}(v_e, v_i) &= \sqrt{\frac{v_e K_e \tau_e}{2}} Q_e, \\ \mu_{G_i}(v_e, v_i) &= v_i K_i \tau_i Q_i, \\ \sigma_{G_i}(v_e, v_i) &= \sqrt{\frac{v_i K_i \tau_i}{2}} Q_i, \end{aligned} \quad (\text{A.10})$$

We can then calculate the total input conductance of the neuron μ_G and its effective membrane time constant τ_m^{eff} (equations (A.11)), with C and g_L being the same as in the AdEx model.

$$\begin{aligned}\mu_G(v_e, v_i) &= \mu_{Ge} + \mu_{Gi} + g_L, \\ \tau_m^{\text{eff}}(v_e, v_i) &= \frac{C}{\mu_G}.\end{aligned}\tag{A.11}$$

Then, we can calculate, in the equation (A.12), the mean subthreshold voltage assuming that the subthreshold moments (μ_V , σ_V , τ_V) are not affected by the exponential term of the AdEx model, and thus considering only the leakage term and the synaptic inputs.

$$\mu_V(v_e, v_i) = \frac{\mu_{Ge} E_e + \mu_{Gi} E_i + g_L E_L}{\mu_G}.\tag{A.12}$$

Thanks to the calculation developed by Zerlaut and al. [374], we obtain σ_V and τ_V in the equations (A.13) and (A.14), where $U_s = \frac{Q_s}{\mu_G} (E_s - \mu_V)$ and $s = (e, i)$.

$$\sigma_V(v_e, v_i) = \sqrt{\sum_s K_s v_s \frac{(U_s \cdot \tau_s)^2}{2(\tau_m^{\text{eff}} + \tau_s)}},\tag{A.13}$$

$$\tau_V(v_e, v_i) = \left(\frac{\sum_s (K_s v_s (U_s \cdot \tau_s)^2)}{\sum_s (K_s v_s (U_s \cdot \tau_s)^2 / (\tau_m^{\text{eff}} + \tau_s))} \right),\tag{A.14}$$

The mean-field is simulated using The Virtual Brain [300].

A.3.3 Transfer function fitting

The transfer function is a function that provides the firing rate of a population for a given excitatory and inhibitory input firing rates and mean adaptation current. To fit the phenomenological threshold of this function, we create a data set of averages of the mean firing rate of 10 seconds of fifty neurons connecting only to one excitatory and one inhibitory Poisson generator with a fixed negative input. We take twenty values for inhibitory input firing rate (between 0 and 40 Hz) and twenty values of adaptation current, i.e. a constant negative current (between 0 and 500 pA). The five hundred excitatory input firing rates (between 0. and 200 Hz) are adjusted to get a precision of the output firing rate (the interval between two values is in the range of 0.1 and 1.0 Hz). The mean output firing rate is the average over fifty neurons by removing the outliers (values higher than three times the interquartile difference). The phenomenological voltage threshold of the transfer function is fitted following the paper of Zerlaut et al. 2018 [374] with two steps. The initial values of the ten polynomial coefficients of the voltage threshold are set to (effective mean voltage, 1e-3, 1e-3, 1e-3, 1e-3, 0, 0, 0, 0, 0). The first fitting step is the minimization of mean square difference of the voltage threshold estimated from the transfer function ($V_{\text{thre}}^{\text{eff}}$) and the estimated effective threshold from the data set ($\mu_V + \sqrt{2}\sigma_V \cdot \text{erfc_inv}(2\mu_{out} \cdot \tau_V)$). The second fitting step is the minimization of the mean squared difference between the output firing

rate of the data set and the one estimated by the transfer function. The coefficients fit using Broyden–Fletcher–Goldfarb–Shanno algorithm [40, 122, 139, 317] implemented in Scipy [352].

A.3.4 Bifurcation analysis

The bifurcation analysis of the mean Ad Ex has been done using MatCont (version 7.7) [95], a numerical continuation library of MATLAB [174]. From the equilibrium point of a silence network (0.0, 0.0, 0.0, 0.0, 0.0, 0.0), we compute the change of stability depending on the external excitatory firing rate. The result of the numerical continuation is a stability diagram of the mean-field with a bifurcation point. The steps of numerical continuation are variable and adjusted depending on the sensitivity of the stability of the fixed-points.

A.3.5 Spike train analysis

In this paper, we analyze the spike trains analysis to validate the mean-field hypotheses and to compare the response to oscillatory input between the network and the mean-field. The hypotheses of the mean-field are evaluated using two measures : the time scale of the spike trains (auto-correlation time) and the minimum of the spiking interval.

The auto-correlation time is calculated based on the following equation (A.15) proposed by Wieland et al. 2015 [361] :

$$\tau_{\text{corr}} = \int_{-\tau_{\text{max}}}^{\tau_{\text{max}}} \left[\frac{\hat{C}(\tau)}{\hat{C}(0)} \right]^2 d\tau \quad (\text{A.15})$$

where $\hat{C}(\tau) = \langle x(t)x(t-\tau) \rangle \langle x \rangle^2 - \langle x \rangle \delta(\tau)$ with $x(t) = \sum_{n=1}^N \delta(t-t_n)$ for t_n is the n-th spike time of the spike trains. The minimum inter-spiking interval is the minimum time between two spikes of the same neurons. The maximum firing rate is the maximum number of spikes in 5 milliseconds. The analysis of the network's response to a sinusoidal Poisson generator is based on quantifying the phase difference between the oscillation of the mean firing rate of the excitatory neuron of the spiking network and the input oscillation. The network oscillation signal corresponds to the firing rate filter at the input oscillation frequency. The firing rate is defined as a sliding average with a window of 5 milliseconds of the instantaneous firing rate (resolution 0.1ms).

The phase difference between these two oscillatory signals is defined by the following equation(A.16) :

$$\text{meanphase} = \frac{1}{N} \sum_{n=1}^N \exp(j(\theta_{\text{input}}(t, n) - \theta_{\text{network}}(t, n))) \quad (\text{A.16})$$

where θ_{input} is the analytic signal given by the Hilbert transformation of the sinusoidal input, and $\theta_{network}$ is the analytic signal provided by the Hilbert transformation of the network oscillation signal. The network oscillation signal is the histogram of instantaneous firing rate (bin :0.1) smoothed with a sliding average window of 5 milliseconds. The absolute value of the *meanphase* corresponds to the phase locking value as defined by Lachaux et al. 1999 [217]. The angle of the *meanphase* corresponds to the phase shift [124]. Additional analyses are described in the Supplementary Note G.1.3.

A.3.6 Mean-field analysis

The mean-field signal corresponds to the excitatory mean firing rate filter at the input oscillation frequency. The rest of the analysis is the same as for the spike trains. The difference of phase is calculated based on the equation (A.16) where θ_{input} is the analytic signal given by the Hilbert transformation of the sinusoidal input and θ_{mean_field} is the analytic signal given by the Hilbert transformation of the mean-field signal. The phase locking value and the phase shift are extracted from this phase and used to quantify the mean-field's response. Additionally, the maximum firing rate of the mean-field corresponds to this maximum value of the mean firing rate. Additional analyses are described in the Supplementary Note G.1.4.

A.4. Results

In the results sections of this paper, we do not present specific simulations but instead, provide a characterization of the model itself. The results presented (unless otherwise indicated) are based on a unique set of parameters that have led to a fixed transfer function fitted for each of the populations (inhibitory and excitatory) considered. These results may differ for other fits and parameters, and we invite users of the mean-field AdEx to perform their own characterizations for their specific applications.

A.4.1 Steady-state and transfer function

The mean-field (MF) model, by construction, captures the out-of-equilibrium asynchronous irregular steady-state of the spiking neural network as a stable fixed-point. The stability analysis, according to the parameter v_{ext} , the external input, provides a bifurcation diagram. This bifurcation diagram can complete the understanding of the underlying structure of attractors while building a network of MF [374, 140].

A qualitative comparison of the mean firing rate for a subset of external input characterizes the link between the bifurcation diagram of the MF and the spiking neural network state. The external input in this paper is a parameter, constant input, for the MF and the frequency of the Poissonian spike generator for the network. Figure A.1 illustrates in the central plot the stability of the MF (black line) for different levels

of external inputs and the corresponding mean firing rate of the steady-state of the network. The bifurcation diagram shows three regions of stability associated with the two variables, the mean firing rate of each population, also existing for the network. The side figures of the diagram show two simulations corresponding to two examples taken from the stable region of the MF with the associated network simulations. The qualitative comparison of these two measures displays a similar behavior and proves that the MF captures well the steady-state of the network for different inputs. Remarkably, in the region with a very high firing rate ($\approx 200\text{Hz}$), the network is no longer in an asynchronous irregular regime, are captured by the MF. However, looking more into details, significant quantitative differences appear for certain values of external inputs.

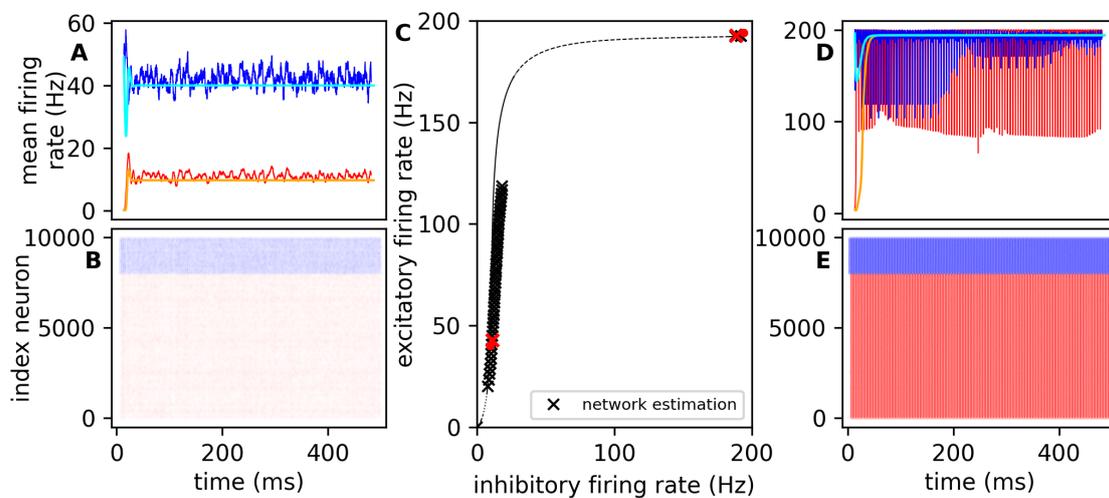


FIGURE A.1. – Mean-field and network dynamic without adaptation

The figure C represents the bifurcation of external excitatory input in the variable space of excitatory and inhibitory mean firing rate. The black line is the mean-field bifurcation with its stability (continuous line for stable fix points and dashed lines for unstable fix points). The crosses are the estimation of the fixed-point using the simulation of the spiking neural network. The red crosses and red points are the fixed-points associated with the examples of each side. The figures A, B, D and E exemplify network and mean-field simulations with an external excitatory input of 10 Hz and 80 Hz. The top figures (A, D) visualize the mean firing rate of the excitatory (red) and inhibitory (blue) neurons from the network simulation and the mean firing rate of the excitatory (orange) and inhibitory (cyan) population from the mean-field simulation. The bottom figures (B, E) are the corresponding raster plots of the spiking network simulations. The figures A and B display asynchronous irregular regime, and the figures D and E display synchronous regular regime

Before looking into detail, it is important to notice that the measure of the mean and standard deviation of the firing rate of a network's steady-state present variability in time and is dependent on the window size. The supplementary figures G.5 and G.6 show the distribution of these measures over thirty seconds for different window sizes.

This confirms that the distribution is dependent on the network state. To reduce the measure variability, the steady-state of the network is measured over four seconds after discarding one second of transient. Figure A.2 compares MF fixed-points and the corresponding mean firing rate of steady-state of the network. The results show the firing rate of each population for three different values of the parameter b , which directly affects the range of the adaptation variable w . This adaptation variable is a slower variable playing an important role in modeling phenomena such as neuromodulation [140, 343]

As above, figure A.2 confirms that the MF captures the qualitative behavior of the network. However, the MF model underestimates the mean firing rate for rates below 150 Hz and overestimates it for the mean firing rate higher than 150 Hz. The overestimation is accentuated with the increase of spike trigger, and it comes from a reduction of the mean firing rate, not present in network simulation, after the first fast increase. This may be reduced by fitting the population's transfer function with the network's steady-state in a specific range of values. However, the structure remains qualitatively the same : a stable fixed-point at the zero input (giving a firing rate = 0) that soon becomes an unstable fixed-point going through a Hopf bifurcation for negative or near-by zero values depending on the level of adaptation b . After another Hopf bifurcation, it becomes stable again, entering the main region of interest of the MF where the corresponding spiking network exhibit asynchronous irregular dynamics and where the range of firing rate observed correspond to what can be observed experimentally in the cortex [177, 297]. It remains stable for a long range of input (80Hz without adaption and >100Hz with adaptation $b = 30pA$ or $b = 60pA$). A bistability exists, with a higher fixed-point around 200Hz. The supplementary figures G.2, G.3 and G.4 provide the four additional dimensions of the bifurcation diagram. They also demonstrate that the MF does not capture quantitatively the variances and covariance. This may be linked to the size of the windows (see G.5 and G.6).

Our previous method for estimating the steady-state of the network is limited to one state, and it cannot provide evidence of the existence of bi-stability. By a reduction every 10 seconds of 1Hz of the external input of the Poisson Generator for a network in a steady-state around 200 Hz, we estimate the range of existence of the bi-stability for an external firing rate under 51 Hz. Similarly, by an increase every 10 seconds of 1Hz of the external input of a steady-state network under 150 Hz, we estimate the existence of bi-stability for an external firing rate over 48 Hz. The supplementary figures G.7 and G.8 show that the MF overestimates the bi-stability's existence. As well as previously, the comparison was also limited to a specific network and one realization of noise. To examine the generalization of the MF, the estimation of the steady-state of the network was done thirty times. The result demonstrates a low variability of the steady-state except when the external input is closed to 50Hz, around the shift of attractor (see supplementary figure G.1).

The crucial aspect for the accuracy of the MF AdEx is the fitted transfer function that, in the equations, drives the dynamics. As detailed in the method section, the final expression is based on the semi-analytical approach, building the link between the single neuron response to inhibitory and excitatory input and the MF. We aim to

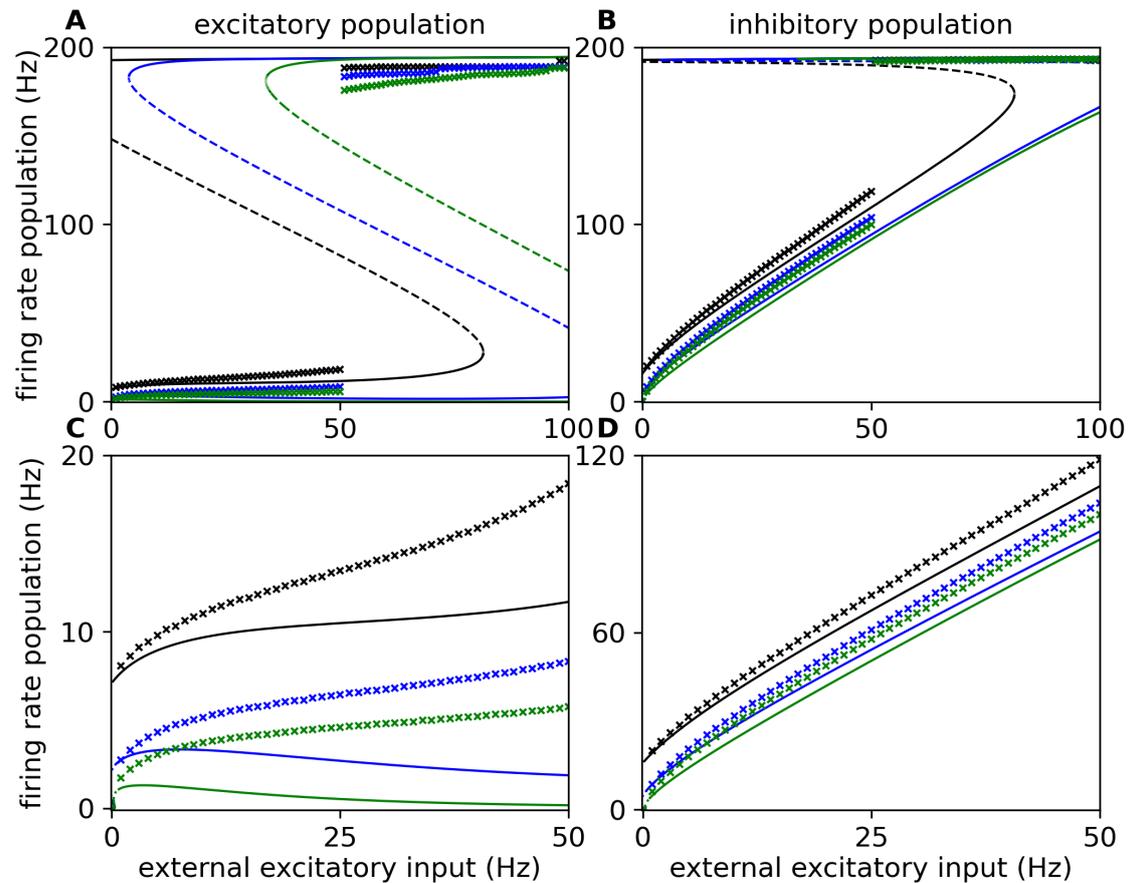


FIGURE A.2. – Impact of spike-trigger adaptation on the network dynamic
The graphic shows the bifurcation diagram, which depends on the external excitatory input for three different values of spike-trigger adaptation (black : 0 pA, blue : 30 pA, green : 60 pA). The three curves represent the stability of the fixed-point (a continuous line indicates a stable fixed-point, and a dashed line represents an unstable fixed-point). The crosses on the diagram are an approximation of the stable fixed-point obtained through simulations. The **A** and **C** diagrams are dedicated to the mean firing rate of the excitatory population, while the **B** and **D** diagrams are for the mean firing rate of the inhibitory population. The two bottom figures (C and D) enlarge the bottom part of the top graphic (A and B).

characterize it in figure A.3 provides an analysis of the MF transfer function’s ability to capture the single neuron’s firing rate in response to excitatory input. The top plot (fig. A.3A, B) shows the MF transfer function fitted. The middle plot (fig. A.3C, D) shows the single neuron firing rate as a function of excitatory input, while the bottom plot (fig. A.3E, F) shows the absolute error corresponding to the difference between the MF transfer function and the firing rate of the single neuron. The results show that the transfer function captures the firing rate well, with the maximum error occurring when the inhibitory firing rate is null. For the excitatory neurons transfer function, the

maximum error was found to be 40.089 Hz, with an error of 79.769 Hz and a relative error of 0.0056 Hz for the range between 0 Hz and 200 Hz (see supplementary figure G.9). It becomes much lower if we only consider the range between 0 Hz and 50 Hz, where the maximum error was found to be 3.407 Hz with an absolute error of 0.1743 Hz and a relative error of 0.0218 Hz. For inhibitory, between 0 Hz and 200 Hz, the maximum error was found to be 34.716, with an absolute error of 57.477 Hz and a relative error of 0.00478 Hz (see supplementary figure G.9). If we only consider the range between 0 Hz and 50 Hz, the maximum error was 1.7353 Hz with an absolute error of 0.0688 Hz and a relative error of 0.00687 Hz. In this paper, the fitting of transfer functions includes the effect of the adaptation currents by using data from simulations of neurons with constant negative currents. This was done to reduce the discrepancy between MF and network firing rate when the spike trigger increased, as shown in Di Volo et al. 2019 [356]. For this set of parameters, the effect of the adaptation does not have a significant impact (under the fitting errors without this additional data), as shown in the supplementary figure G.10.

Due to the crucial aspect of the transfer function on the MF dynamic, we analyze the sensibility of the precision of the polynomial coefficients of the excitatory transfer function. The sensibility analysis is the variation of the 10 Hz fixed-point (far from bifurcation) depending on the number of significant digits of the polynomial coefficients. For an accuracy of 10Hz, it requires at least one significant digit. For an accuracy of 1 Hz, it requires at least three significant digits. And, for an accuracy of 0.1 Hz, it requires at least four significant digits. However, the more the network dynamic is closed to a bifurcation point, the more sensitive the stability of fixed to the precision of the coefficients. Furthermore, for some specific states of the MF, the mean firing rate becomes negative (see supplementary figure G.11). Consequently, the MF becomes undefined because the transfer function is not defined for negative firing rates, as expected, negative firing rates are unrealistic cases.

However, another possible source of differences between the network and the MF observed in figure A.2 is that the network may not stay in an asynchronous irregular regime while the input increases. While investigating the MF, we did not explore the network behaviors in more detail, which would require an entirely separate study.

In conclusion, the MF captures qualitatively well the different network steady-states. Still, some quantitative differences are observed and may be improved by a better fitting of the transfer function on a specific region of interest for a particular usage of this model.

A.4.2 Dynamical comparison

This part of the analysis is the dynamical response of the MF to non-constant inputs. Indeed, this model is used to constitute a network of MF to model a cortical region [375] or whole brain modeling [140]. In such simulations, it is not anymore the steady-state of the network but the response to varying inputs that is of interest. Thus, the network may not exhibit the asynchronous irregular regime and/or the condition

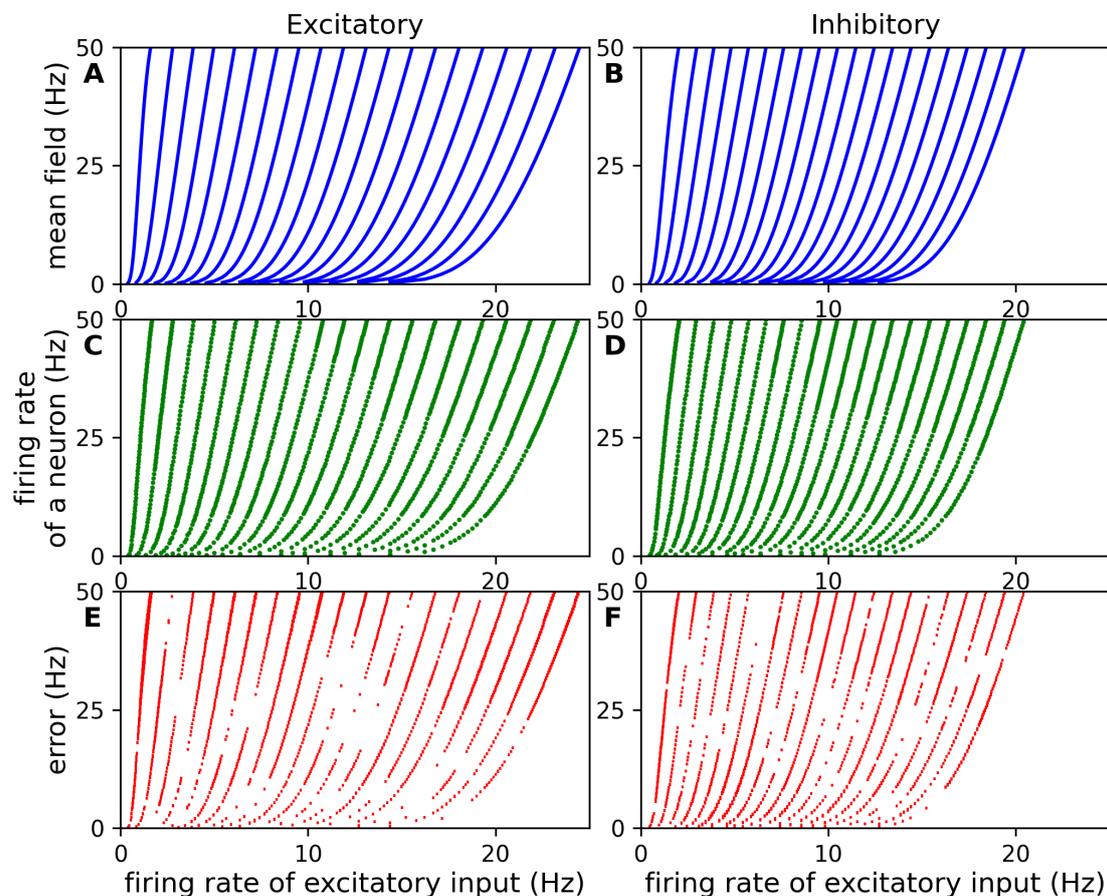


FIGURE A.3. – Fitting of Transfer functions

The left part (A, C and E) is dedicated to the excitatory neuron, while the right part (B, D and F) is dedicated to the inhibitory neuron. **A, B** : The top graph represents the transfer function of the mean-field fitted to the data. **C, D** : The middle graph shows the data used for the fitting, which is the mean firing rate of 50 independent neurons. **E, F** : The bottom graph displays the error between the transfer function and the data. Each line is associated with a different value of inhibitory firing rate (20 values evenly distributed between 0 Hz and 40 Hz).

for the Markovianity assumption under non-constant inputs. We tested for different sinusoidal variations of the external Poissonian inputs.

Figure A.4 shows the range of validity of the hypothesis of the MF. Indeed, one of the important assumptions to avoid underestimation is the presence of only a single spike of each neuron within the time windows T (see [106]). The red line corresponds to the network's auto-correlation time equals the minimum time between two spikes. The top row of the figure describes the validity range for low amplitude and/or low frequency. It shows that the MF hypothesis is valid for a wide range of low frequency inputs but breaks down for high frequency or/and high amplitude inputs.

On the other hand, the bottom row of the figure shows the effect of adaptation

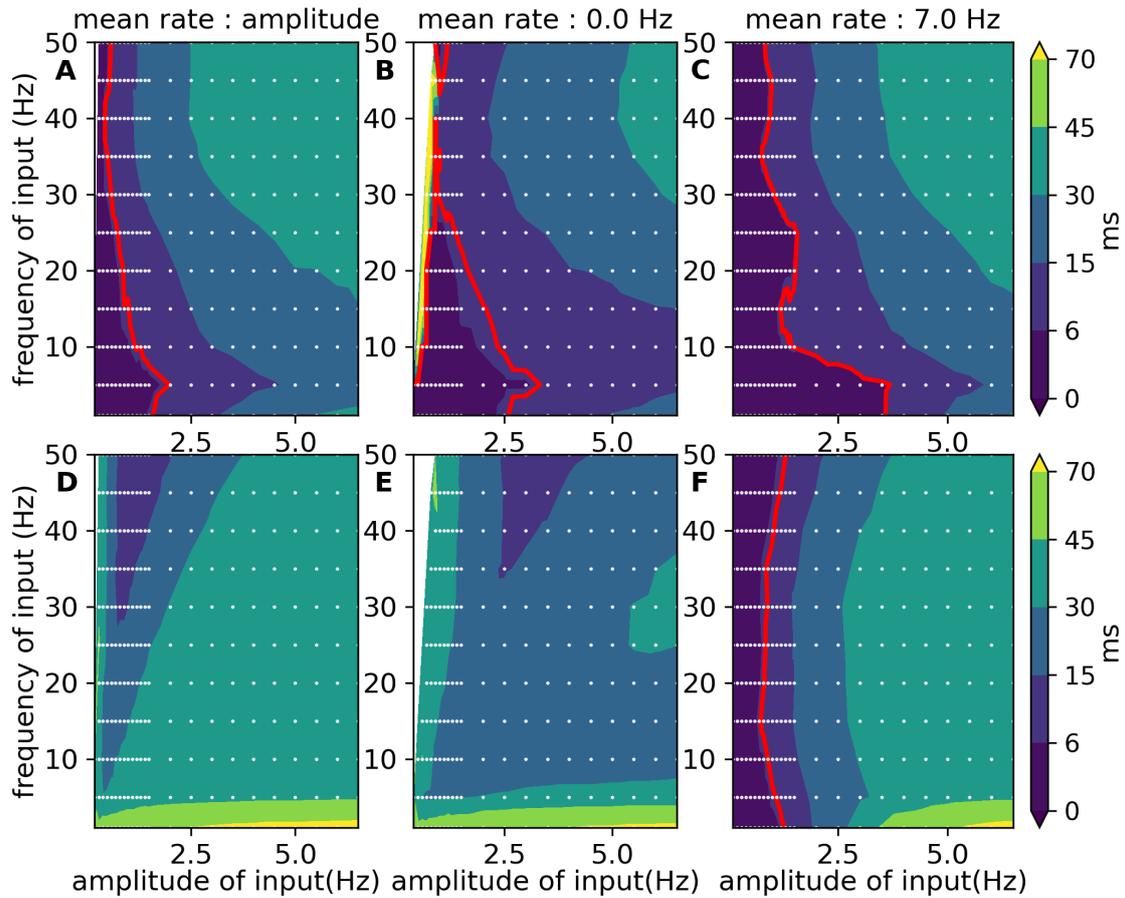


FIGURE A.4. – Time scale of the excitatory population in the presence of a sinusoidal Poisson generator

Each graphic represents the timescale of the excitatory population in the presence of stimulation by a sinusoidal Poisson generator mimicking the connection with another population. The red line represents the minimum time between two spikes equals the timescale of the network. The neurons do not have any adaptation for the top graphics (A, B and C), while for the bottom graphics (D, E, F), the value of their spike-trigger adaptation is 60.0 pA. The difference between each column is the mean rate of the sinusoidal Poisson generator. **A, D** : The mean rate equals the amplitude of the oscillation. **B, E** : The mean rate is fixed at 0.0 Hz. **C, F** : The mean rate is fixed at 7.0 Hz, respectively.

($b=60.0\text{pA}$) on the range of validity. It shows a validity range only when the average input frequency is 7.0 Hz. In other words, the MF hypothesis, while receiving non-constant input, is valid only for a narrow range of input values when the system has adaptation.

Out of the MF assumptions, we compare the dynamic of the MF and the network in response to a sinusoidal input. Figure A.5 shows that the Phase Locking Values between the input signal and the network or MF is close to 1.0, i.e. synchronized.

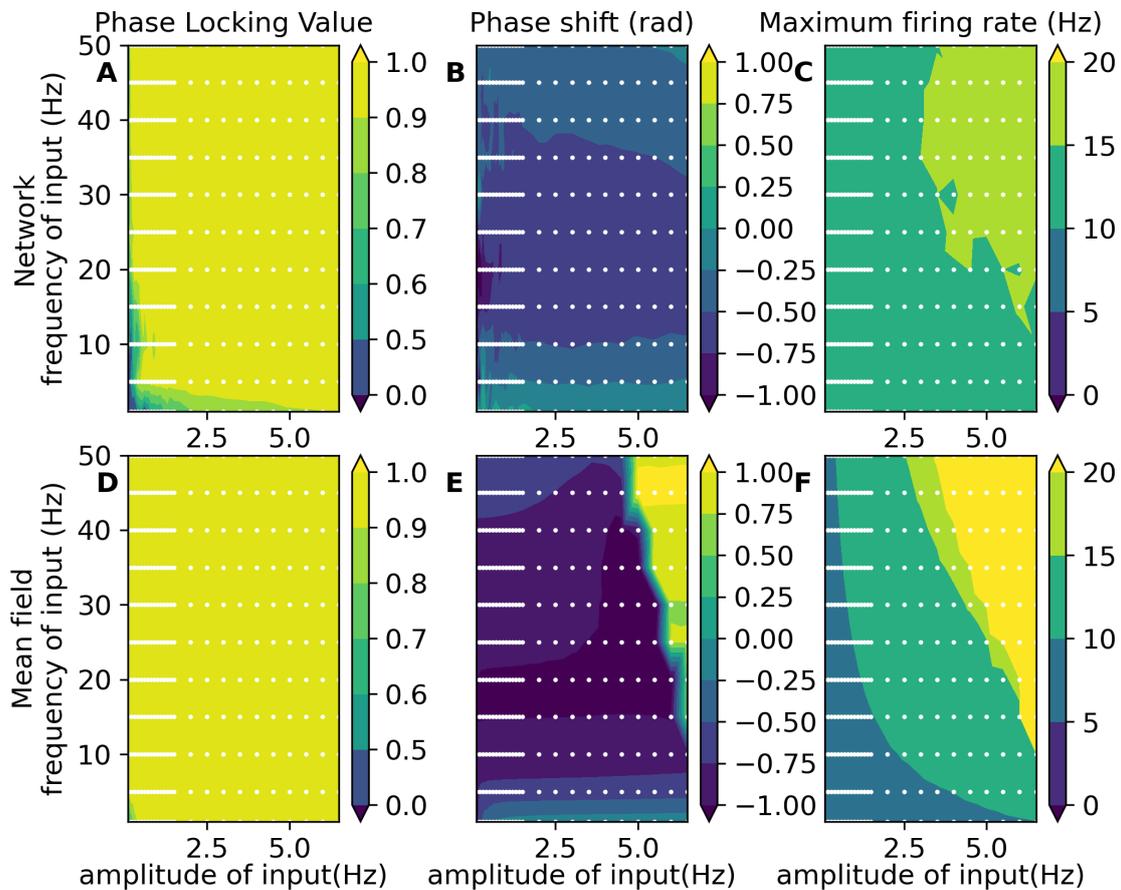


FIGURE A.5. – Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 7Hz

The graphics **A**, **B** and **C** represent the measure of the excitatory population in the spiking neural network, and the **D**, **E** and **F** graphics illustrate the measure of the excitatory mean firing rate of the mean-field. The three measures displayed here are the phase locking values in rad (**A** and **D**), the phase shift (**B** and **E**) between the oscillatory input and the mean firing rate in rad, and the excitatory population's maximum firing rate in Hz (**C** and **F**).

The only exception is when the network receives a signal with low amplitude or low frequency; this can come from the variability of the input of the inhomogeneous Poisson generator. However, the phase shift analysis gives a clear difference between the network and the MF. The MF's phase shift is more negative than the networks and is in anti-phase with the input for high amplitude and high firing rate (yellow area). This area also corresponds to a maximum firing close to 200Hz. On the contrary, the network has a maximum firing rate in the range of 10 Hz to 20 Hz. This means that the network is more attracted around the center fixed-point than the MF, i.e. has a tendency to converge toward this corresponding attractor. Including adaptation does not change these observations but increases the mismatch already noted (see

supplementary figure G.12). Nevertheless, the mismatches become more significant if the sinusoidal input is centered around 0.0 Hz. As the mean firing rate does not attract to the center fixed-point, it creates a difference in the phase locking values and phase shift (see supplementary figure G.13 and G.14). The adaptation pushes the network in anti-phase, reducing the discrepancy between the MF and the network for the phase shift. However, the other observation remains the same.

A.5. Discussion

Many different approaches are possible to capture brain dynamics at the mesoscopic scale [365, 262, 106, 42, 166, 309, 246, 307, 23], and each of these approaches is relevant to capture specific features of the dynamics of spiking networks. Some specificities of the mean-field methods studied here are the possibility of considering the sparse connectivity and the finite size of the populations. This method has been applied considering different single neuron models [106, 356, 51, 9], but the most used to build large-scale model is based on the AdEx model. Thus, we aim to offer relevant information and characterization approaches about this specific MF model.

The present characterization of the mean-field AdEx (MF) indicates that the MF provides good qualitative insights into the complex behavior of networks of the adaptive exponential integrate and fire neurons with conductance synapses. The approximation of the MF is tightly connected to its transfer function and, in particular, to the precision of the phenomenological threshold. However, this mean-field approximation does not capture the dynamical behaviors of the network for oscillatory inputs.

The bifurcation analysis of the MF reveals the co-existence of multiple fixed-point depending on the external input. In this work, we validated the existence of the co-existence of multi-steady-state in spiking neural networks. Additionally, the MF captures fixed-points around 200 Hz despite the non-validity of the MF's assumption that the network regimes are asynchronous irregular regimes. However, the quantitative comparison of the fixed-points and the steady-state present a mismatch which is accentuated by the increase of the spike trigger adaptation of the excitatory neurons. This mismatch has multiple causes, such as the size of the windows for the measures, the approximation of the transfer function or/and the degree of network synchronization. To evaluate the influence of these causes, we quantify the approximation of the transfer function. This quantification informs us about the errors which are not uniform and increase with the output firing rate of the neuron. However, the approximation can be improved if the input and output firing rate range is reduced. Additionally, sensitive analysis of the polynomial coefficients of the phenomenological threshold of the transfer function shows that reducing the imprecision at 0.1 Hz, requires at least 4 significant digits. In the presence of oscillatory input, we check the co-validity of two hypotheses : the time constant of the MF is high enough for considering that the network is memory-less and low enough for the neurons not to generate more than one spike during this interval. We found that these two hypotheses are valid simultaneously for low amplitudes of the oscillatory input and depend on

the average input. Additionally, adding the adaptation current can create cases where these two hypotheses are exclusive because the adaptation current creates a slow fluctuation of mean firing rate. Despite the validity or not of these two hypotheses, the dynamic comparison between the MF and the network presents a good match for synchronization with the input but a mismatch in mean firing rate, like previously, and a mismatch in the phase shift with the input. These mismatches are variable with the average input and are quite insensitive to the adaptation current. The possible cause of these mismatches is the adiabatic approximation of the MF methods [106].

This study is limited to a specific set of parameters and mostly to a specific network connectivity. However, the realization of thirty other networks provides a low variance except around the shift of attractor. Additionally, the qualitative dynamic is similar to that presented in Di Volo et al. 2019 [356]. The analyses do not provide exhaustive confirmation of all MF's hypotheses, such as the Gaussian distribution of the voltage membrane, the adiabatic process, and the input spike trains of each neuron are equivalent to a Poisson generator. It explains the partial conclusion of the study but provides a characterization to help the interpretation of MF simulation and clarify the link between the MF and the network. Additionally, we do not provide a study of the different versions of the phenomenological threshold [375, 374, 356] which can reduce the MF's approximation. For example, the paper of Zerlaut et al. 2018 [374] seems to better approximate the MF because the difference in phase shift is smaller. Still, it was realized with a different transfer function and simulator. Indeed, the implementation of the Poisson generator may differ between simulating environments such as between NEST [98] and Brian2 [334], which can impact the network regimes, particularly for high frequency. The good practice will be to perform a characterization of the MF for each specific application.

To summarize, the MF provides valuable insights into the network's steady-state behavior. Through the optimization of the transfer function within the desired range, the model's suitability for particular use cases can be improved. Nonetheless, to better characterize and refine the model's performance in these specific scenarios, it is imperative to undertake further investigations and analyses. This would ensure a more comprehensive understanding of the network's dynamics and improve its accuracy and effectiveness in practical applications.

Acknowledgments

This research has received funding from the European Union's Horizon 2020 Framework Programme for Research and Innovation under the Specific Grant Agreement No. 945539 (Human Brain Project SGA3) and Specific Grant Agreement No. 785907 (Human Brain Project SGA2).

Data and materials availability

The code for the simulation and analysis of this paper is freely available under v2 Apache license at https://github.com/lionelkusch/compare_zerlaut/tree/paper. Each exploration's analysis results are registered in databases available on Zenodo (<http://dx.doi.org/10.5281/zenodo.8070791>) with the associate code. Supplementary Note [G.1.1](#) and [G.1.2](#) describe the contents of them.

CHAPTER B

A comprehensive neural simulation of slow-wave sleep and highly responsive wakefulness dynamics

B.1. Chapter overview

This chapter is a side work of this thesis. It describes a model of perturbation complexity index for a state of wake or sleep. My participation in this work is the implementation of the Mean Ad Ex [356] in The Virtual Brain [299] and the help of the parametrization of the model using the bifurcation analysis presented in the previous chapter.

This chapter is based on the published article in *Frontiers in Computational Neuroscience* :

A comprehensive neural simulation of slow-wave sleep and highly responsive wakefulness dynamics

Jennifer S. Goldman¹, Lionel Kusch², David Aquilue¹, Bahar Hazal Yalçinkaya^{1,2}, Damien Depannemaecker¹, Kevin Ancourt¹, Trang-Anh E. Nghiem^{1,3}, Viktor Jirsa², and Alain Destexhe¹

¹CNRS, Institute of Neuroscience (NeuroPSI), Paris-Saclay University, Saclay, France

²Institut de Neurosciences des Systèmes, Aix-Marseille University, INSERM, Marseille, France

³Laboratoire de Physique, Ecole Normale Supérieure, Université PSL, CNRS, Sorbonne Université

Université de Paris, Paris, France



OPEN ACCESS

EDITED BY

Fernando Soler-Toscano,
Sevilla University, Spain

REVIEWED BY

Federico Stella,
Radboud University, Netherlands
Yu Zhang,
Zhejiang Lab, China

*CORRESPONDENCE

Jennifer S. Goldman
✉ jennifer.goldman@mail.mcgill.ca
Alain Destexhe
✉ alain.destexhe@cnrs.fr

RECEIVED 30 September 2022

ACCEPTED 21 December 2022

PUBLISHED 13 January 2023

CITATION

Goldman JS, Kusch L, Aquilue D,
Yalçinkaya BH, Depannemaecker D,
Ancourt K, Nghiem T-AE, Jirsa V and
Destexhe A (2023) A comprehensive
neural simulation of slow-wave sleep
and highly responsive wakefulness
dynamics.
Front. Comput. Neurosci. 16:1058957.
doi: 10.3389/fncom.2022.1058957

COPYRIGHT

© 2023 Goldman, Kusch, Aquilue,
Yalçinkaya, Depannemaecker, Ancourt,
Nghiem, Jirsa and Destexhe. This is an
open-access article distributed under
the terms of the [Creative Commons
Attribution License \(CC BY\)](https://creativecommons.org/licenses/by/4.0/). The use,
distribution or reproduction in other
forums is permitted, provided the
original author(s) and the copyright
owner(s) are credited and that the
original publication in this journal is
cited, in accordance with accepted
academic practice. No use, distribution
or reproduction is permitted which
does not comply with these terms.

A comprehensive neural simulation of slow-wave sleep and highly responsive wakefulness dynamics

Jennifer S. Goldman^{1*}, Lionel Kusch², David Aquilue¹,
Bahar Hazal Yalçinkaya^{1,2}, Damien Depannemaecker¹,
Kevin Ancourt¹, Trang-Anh E. Nghiem^{1,3}, Viktor Jirsa² and
Alain Destexhe^{1*}

¹CNRS, Institute of Neuroscience (NeuroPSI), Paris-Saclay University, Saclay, France, ²Institut de Neurosciences des Systèmes, Aix-Marseille University, INSERM, Marseille, France, ³Laboratoire de Physique, Ecole Normale Supérieure, Université PSL, CNRS, Sorbonne Université, Université de Paris, Paris, France

Hallmarks of neural dynamics during healthy human brain states span spatial scales from neuromodulators acting on microscopic ion channels to macroscopic changes in communication between brain regions. Developing a scale-integrated understanding of neural dynamics has therefore remained challenging. Here, we perform the integration across scales using mean-field modeling of Adaptive Exponential (AdEx) neurons, explicitly incorporating intrinsic properties of excitatory and inhibitory neurons. The model was run using The Virtual Brain (TVB) simulator, and is open-access in EBRAINS. We report that when AdEx mean-field neural populations are connected *via* structural tracts defined by the human connectome, macroscopic dynamics resembling human brain activity emerge. Importantly, the model can qualitatively and quantitatively account for properties of empirically observed spontaneous and stimulus-evoked dynamics in space, time, phase, and frequency domains. Large-scale properties of cortical dynamics are shown to emerge from both microscopic-scale adaptation that control transitions between wake-like to sleep-like activity, and the organization of the human structural connectome; together, they shape the spatial extent of synchrony and phase coherence across brain regions consistent with the propagation of sleep-like spontaneous traveling waves at intermediate scales. Remarkably, the model also reproduces brain-wide, enhanced responsiveness and capacity to encode information particularly during wake-like states, as quantified using the perturbational complexity index. The model was run using The Virtual Brain (TVB) simulator, and is open-access in EBRAINS. This approach not only provides a scale-integrated understanding of brain states and their underlying mechanisms, but also open access tools to investigate brain responsiveness, toward producing a more unified, formal understanding of experimental data from conscious and unconscious states, as well as their associated pathologies.

KEYWORDS

neural simulation, mean-field model, spontaneous activity, evoked responses, wake, synchronous, slow-wave sleep, human brain

1. Introduction

Brain activity is marked by complex spontaneous dynamics, particularly during conscious states when the brain is most responsive to stimuli. Though changes in spontaneous and evoked dynamics have been unambiguously empirically observed in relation to changes in brain state, their multi-scale nature has notoriously occluded a formal understanding.

Spanning from macroscopic dynamics supporting communication between brain regions to microscopic, molecular mechanisms modulating ion channels, hallmarks of consciousness have been observed across spatial scales. At the whole-brain level, conscious states are marked by complex spontaneous collective neural dynamics (Niedermeyer and Lopes da Silva, 2005; El Boustani and Destexhe, 2010) and more sustained, reliable, and complex responses to stimuli (Massimini et al., 2005; Casali et al., 2013; D'Andola et al., 2018; Dasilva et al., 2021). At the microscopic level, neuromodulation is enhanced in conscious, active states, leading to microscopic changes in cellular kinetics (McCormick, 1992). Yet, a challenging multi-scale problem still resides in comprehending how changes in the complexity of global spontaneous dynamics and whole brain responsiveness may specifically relate to microscopic neuromodulatory processes to enable neural coding during active states. Here, using mean-field models of conductance-based, Adaptive Exponential (AdEx) integrate-and-fire neurons with spike-frequency adaptation developed recently (Zerlaut et al., 2018; Capone et al., 2019; di Volo et al., 2019), constrained by human anatomy and empirically informed by local circuit parameters, we report the natural emergence of global dynamics mimicking different human brain states.

To connect microscales (neurons) to macroscales (whole brain), this work relies on previous advances at mesoscales (neural populations). The first step was modeling biologically-relevant activity states in networks of spiking neurons. Based on experimental recordings, we used the Adaptive Exponential (AdEx) integrate and fire model to simulate two main cell types identifiable in extracellular recordings of human brain (Peyrache et al., 2012): regular-spiking (RS) excitatory and fast-spiking (FS) inhibitory cells. AdEx networks were constrained by biophysical representations of synaptic conductances, which allowed the model to be compared to conductance measurements done in awake animals (Zerlaut et al., 2018) (for experiments, see Steriade et al., 2001; Rudolph et al., 2007). In such configurations, AdEx networks reproduce states observed *in vivo* (Destexhe, 2009; Jercog et al., 2017; Zerlaut and Destexhe, 2017; Zerlaut et al., 2018; Nghiem et al., 2020), notably asynchronous irregular (AI) states found experimentally in awake states, and synchronous slow waves as in deep sleep (Destexhe et al., 1999; Steriade et al., 2001; Steriade, 2003). From AdEx networks, mean-field models were derived to take into account second order statistics of AdEx networks

interacting through conductance-based synapses. We used a Master Equation formalism (El Boustani and Destexhe, 2009), modified to include adaptation (di Volo et al., 2019).

In this manuscript, we present evidence that mean-field descriptions of biophysically informed estimates of neuron networks produce macroscopic dynamics capturing essential characteristics of human wake and sleep states—due to variation in spike-frequency adaptation—when coupled by the human connectome with tract-specific delays. First, we show that simulated microscopic changes in membrane currents directly lead to the emergence of globally asynchronous vs. synchronous dynamics exhibiting distinct signatures in the frequency domain, as well as changes in inter-regional correlation structure and phase-locking, mimicking aspects of spontaneous human brain dynamics. The spatial extent of synchrony and phase relations across brain regions was observed to be an emergent property of both microscopic-scale adaptation changes and the organization of the human connectome, which allow for enhanced phase coherence at intermediate, cross-region, but not whole-brain scales in sleep-like states, consistent with the propagation of traveling waves. Further, we report enhanced brain-scale responsiveness to stimulation in simulations of asynchronous, fluctuation-driven compared to synchronous, phase-locked regimes, consistent with empirical data from conscious vs. unconscious brain states. Together, the data suggest that the TVB-AdEx model represents a scale-integrated neuroinformatics framework capable of recapitulating known features associated with human brain states as well as elucidating relationships between space-time scales in brain activity. Due to its reliance on anatomical data non-invasively available from humans, this model may further facilitate subject-specific modeling of human brain states in health and disease, including restful and active waking states, as well as sleep, anesthesia, and coma to aid future advances in personalized medicine.

2. Results

We begin by showing essential properties of the components forming the TVB-AdEx model. Next, we describe the integration of AdEx mean-fields into The Virtual Brain (TVB) simulator of EBRAINS, making the models and analyzes openly available to facilitate replication and extension of the results. The results presented here indicate that the TVB-AdEx whole human brain model captures fundamental aspects of synchronous and asynchronous brain states, both spontaneously and in response to perturbation.

2.1. Components of TVB-AdEx models

The first component of the TVB-AdEx model is at the cellular level, and consists of networks of integrate-and-fire

adaptive exponential (AdEx) neurons. As shown in previous studies (Destexhe, 2009; Zerlaut et al., 2018; di Volo et al., 2019), networks of AdEx neurons with adaptation can display asynchronous, irregular (AI) states, as well as synchronous, regular slow-wave dynamics that alternate between periods of high activity (Up) and periods of near silence (Down). The necessary mechanistic ingredients needed to obtain both dynamical regimes include leak conductance and conductance-based synaptic inputs. Each neuron's input is comprised by the firing rates of synaptically connected neurons, weighted by synaptic strengths, as well as stochastic noise (hereafter called “drive”; see Materials and Methods), related biologically to miniature postsynaptic currents. AdEx neurons have the ability to integrate synaptic inputs and fire action potentials, followed by a refractory period (Brette and Gerstner, 2005). AdEx networks with conductance-based synapses can capture features offered by more detailed and computationally expensive models, including AI states and slow-wave dynamics. Figure 1 shows an example of such AI states (Figure 1A) and Up-Down dynamics (Figure 1B) simulated by the same AdEx network, changing only the level of spike-frequency adaptation current (parameter b in the equations, see Material and Methods). In AI states, the firing of individual units remains irregular, but sustained (Figure 1A), whereas in slow-wave states the dynamics alternate between depolarized Up states with asynchronous dynamics and hyperpolarized Down states of near silence (Figure 1B). As such, changes in spike-frequency adaptation lead to differences in cellular kinetics between sleep and wake states. Biologically, spike-frequency adaptation is suppressed by enhanced concentrations of neuromodulators such as acetylcholine during active, conscious brain states that tends to close K^+ leak channels, resulting in sustained depolarization of neurons (McCormick, 1992) which promotes the emergence of asynchronous, irregular (AI) action potential firing and fluctuation-driven regimes associated with waking states. In contrast, low levels of neuromodulation during unconscious brain states leave leak K^+ channels open, leading to waves of synchronous depolarisation and hyperpolarization due to the buildup and decay of spike-frequency adaptation, accounting for the emergence of slow-wave dynamics as observed in previous modeling work (Jercog et al., 2017; Nghiem et al., 2020).

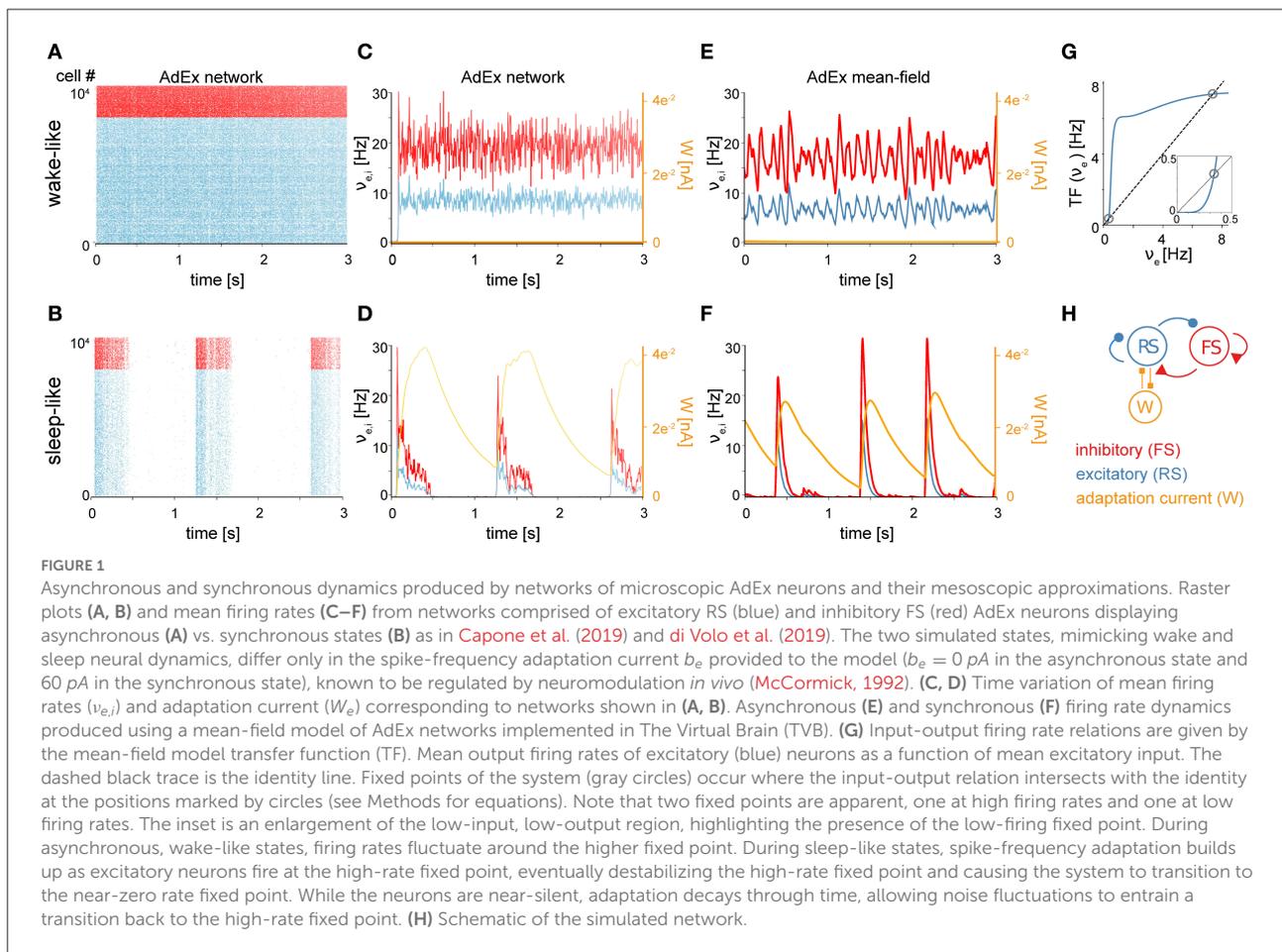
The second component of the TVB-AdEx model is a mean-field equation derived from spiking-neuron network simulations, capturing the typical dynamics of a neuron in response to inputs and hence able to describe the mean behavior of a neuronal population, Zerlaut et al. (2018) and di Volo et al. (2019) using a Master-Equation formalism (El Boustani and Destexhe, 2009). This formalism allows one to derive mean fields from conductance-based integrate-and-fire models. It has been shown that—using numerical fits of the transfer function (Zerlaut et al., 2016), an analytical expression for the relationship between a neuron's input and output rates - one can describe complex neuronal models, such as AdEx neurons,

and even Hodgkin-Huxley type biophysical models (Carlu et al., 2020). In Figures 1C, D, excitatory and inhibitory firing rates are compared between mean-field simulations using the Master Equation formalism and spiking neural network simulations (time binned population spike counts divided by time bin length $T = 0.5$ ms). The average adaptation value is also shown for this network (Figures 1C, D, orange curves). These population variables are suitably captured by the mean-field model including adaptation (Capone et al., 2019; di Volo et al., 2019). This mean-field model can exhibit both AI (Figure 1E) and Up-Down dynamics (Figure 1F). Like in the AdEx spiking network model, the transition between two states can be obtained by changing the adaptation parameter called b in both cases (di Volo et al., 2019). With no adaptation, the dynamics are fluctuation-driven around a fixed point exhibiting nonzero firing rates. With adaptation, as the neurons self-inhibit due to adaptation, the nonzero rate fixed point is progressively destabilized by adaptation buildup, driving the dynamics back to the near-zero firing rate fixed point until adaptation wears off and noise drives the system back to the vicinity of the higher-rate fixed point. Thus, with adaptation, the system displays noise-driven alternation between the two fixed-points to generate slow waves (Figure 1G). The regimes are achieved using the mean-field model, which describes excitatory (RS) and inhibitory (FS) population firing rates as well as the mean adaptation level of excitatory populations (Figure 1H).

2.2. Integration of AdEx mean-field models in TVB

We have used the simulation engines of the Human Brain Project's (HBP's) EBRAINS neuroscience research infrastructure (<https://ebrains.eu> and The Virtual Brain <https://ebrains.eu/service/the-virtual-brain>) to make access as wide as possible. Replication of the TVB-AdEx findings can be done here, with a free EBRAINS account, and users can clone the repositories to further test or extend the present capacities. The models can also be downloaded from Github at <https://gitlab.ebrains.eu/kancourt/tvb-adex-showcase3-git> to run locally.

Here, the connection of mean-field models was defined by human tractography data (<https://zenodo.org/record/4263723>, Berlin subjects/QL_20120814) from the Berlin empirical data processing pipeline (Schirner et al., 2015) (Figure 2A). A parcellation of 68 regions was used to place localized mean-field models, with long-range excitatory connections (Figure 2B) and delays (Figure 2C) defined by tract length and weight estimates in human diffusion tensor imaging (DTI) data (Sanz-Leon et al., 2015). Now it becomes possible to simulate brain-scale networks using AdEx-based mean-field models in TVB, hence the name “TVB-AdEx” model.

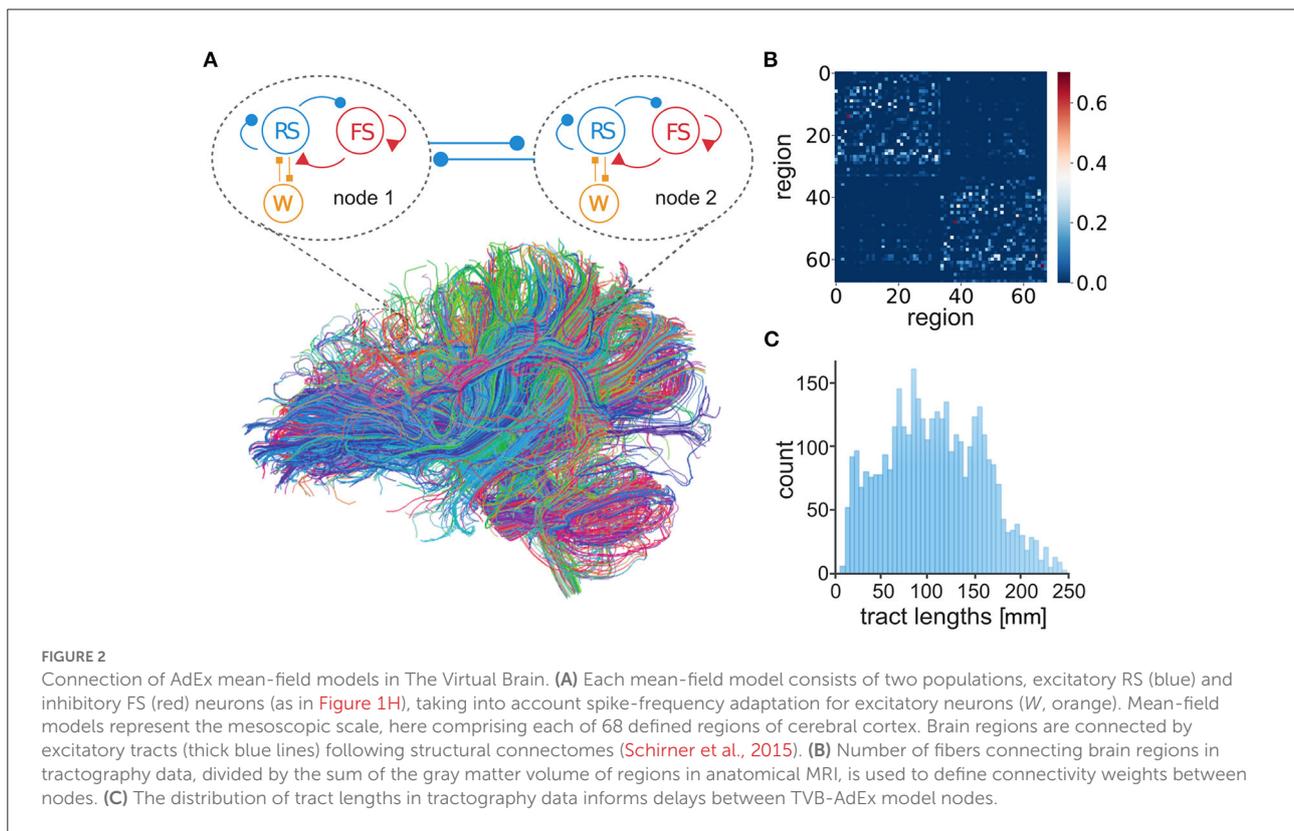


2.3. Spontaneous dynamics of large-scale networks

Having coupled AdEx mean-field models that capture average microscopic characteristics of neural activity, we sought to ascertain if hallmarks of brain-scale (macroscopic) spontaneous activity resembling human brain states were reproduced, as well as whether increases in adaptation strength account for transitions between wake-like and sleep-like macroscopic dynamics. Characterizing temporal hallmarks of simulated neural activity (Figure 3), we find that asynchronous, wake-like dynamics across nodes are recovered in the absence ($b = 0$ pA, Figure 3A), but not in the presence ($b = 60$ pA, Figure 3B) of adaptation. Power spectral analysis reveals a peak in the delta range (0.5 – 5 Hz) for the high-adaptation condition (Figure 3C) consistent with empirical data from deeply sleeping individuals. Further, the power spectrum in the low-adaptation condition shows a maximum near 10 Hz (alpha range), consistent with empirically observed dynamics during resting wakefulness (Figure 3C). Therefore, changes in a simulated microscopic process (spike-frequency

adaption) influence spectral features of macroscopic brain states, with low-adaptation regimes resembling waking states and high-adaptation regimes reminiscent of slow-wave sleep.

Increasing adaptation can also tune the spatial correlation structure of neural activity across brain states. Indeed, as shown in Figure 4, Pearson correlations across nodes are enhanced in the presence of adaptation, consistent with asynchronous dynamics seen during wakefulness vs. synchronous slow waves seen during deep sleep (Figures 4A, E). We also observe the correlation matrix is structured into two clusters corresponding to the two hemispheres in the slow-wave condition ($b = 60$ pA, Figure 4E). In addition, increased adaptation strength also causes the emergence of significantly larger correlations between inhibitory than excitatory firing rates across nodes during sleep-like dynamics (Figures 4B, F). This reveals that microscopic variation in adaptation strength alone can account for empirical reports of increased correlations between inhibitory neurons across long distances and even different cortical regions specifically for inhibitory (Peyrache et al., 2012; Le Van Quyen et al., 2016; Olcese et al., 2016). This is due to different effects of



adaptation on excitatory regular-spiking neurons and inhibitory fast-spiking neurons, key to reproducing empirical dynamics in unconscious states ([Jercog et al., 2017](#); [Nghiem et al., 2020](#)). Moreover, the Phase Lag Index (PLI) is increased during sleep-like dynamics ([Figures 4C, G](#)), suggesting systematic phase relations between nodes consistent with traveling slow waves empirically observed during spontaneous unconscious dynamics ([Destexhe et al., 1999](#); [Steriade, 2003](#)). Such phase relations, evidenced by a significantly larger PLI, are more pronounced for inhibitory than excitatory neurons in sleep-like dynamics, reminiscent of the key role of inhibitory neurons in organizing the emergence of synchronous dynamics during sleep ([Nghiem et al., 2018](#)).

Next, we investigate how the connectome's structure shapes the landscape of synchrony and phase coherence across brain regions, alongside adaptation. In particular, how do the Pearson correlation and PLI scale with spatial distance between nodes? In both $b = 0$ pA and $b = 60$ pA conditions ([Figures 5A, D](#)), the Pearson correlation between excitatory firing rates significantly decreases with Euclidean distance between regions, corresponding to tract-length-related delays in our model. A steeper negative slope is observed in the awake-like ([Figure 5A](#)) than in the slow-wave regime ([Figure 5D](#)), suggesting that the spatial extent of synchrony between regions is enhanced in the presence of high-adaptation, sleep-like dynamics.

In [Figures 5B, C, E, F](#), we show a scatter plot and box plot of the Phase-Lag Index (PLI) as a function of distance between regions. In both $b = 0$ pA and $b = 60$ pA cases, significant differences are observed in the PLI across regions (Kruskal-Wallis test, $***p < 0.001$), suggesting systematic phase relations consistent with the propagation of traveling waves are particularly predominant at intermediate scales. Specifically, in the slow-wave condition ($b = 60$ pA), we observe that the PLI between regions approximately 65 mm apart is significantly enhanced (Mann-Whitney U test, $**p < 0.01$) in comparison to the PLI at both shorter and longer distances.

Further, we test whether the predominance of slow waves at an intermediate spatial scale is an emergent property from the structure of the human connectome. To this purpose, we shuffle the connection weights successively for each region and every other region, to generate a connectome with the same distribution of connection weights between regions but retaining none of the graph structure of the empirical connectome ([Figure 5G](#)). The tract lengths are not modified. Repeating simulations with shuffled connectomes, we again compute the PLI as a function of distance. With shuffled connectomes, we find that the PLI no longer varies significantly as a function of distance in wake-like dynamics ([Figure 5H](#)). As well, the intermediate peak in PLI as a function of distance—denoting an intermediate spatial

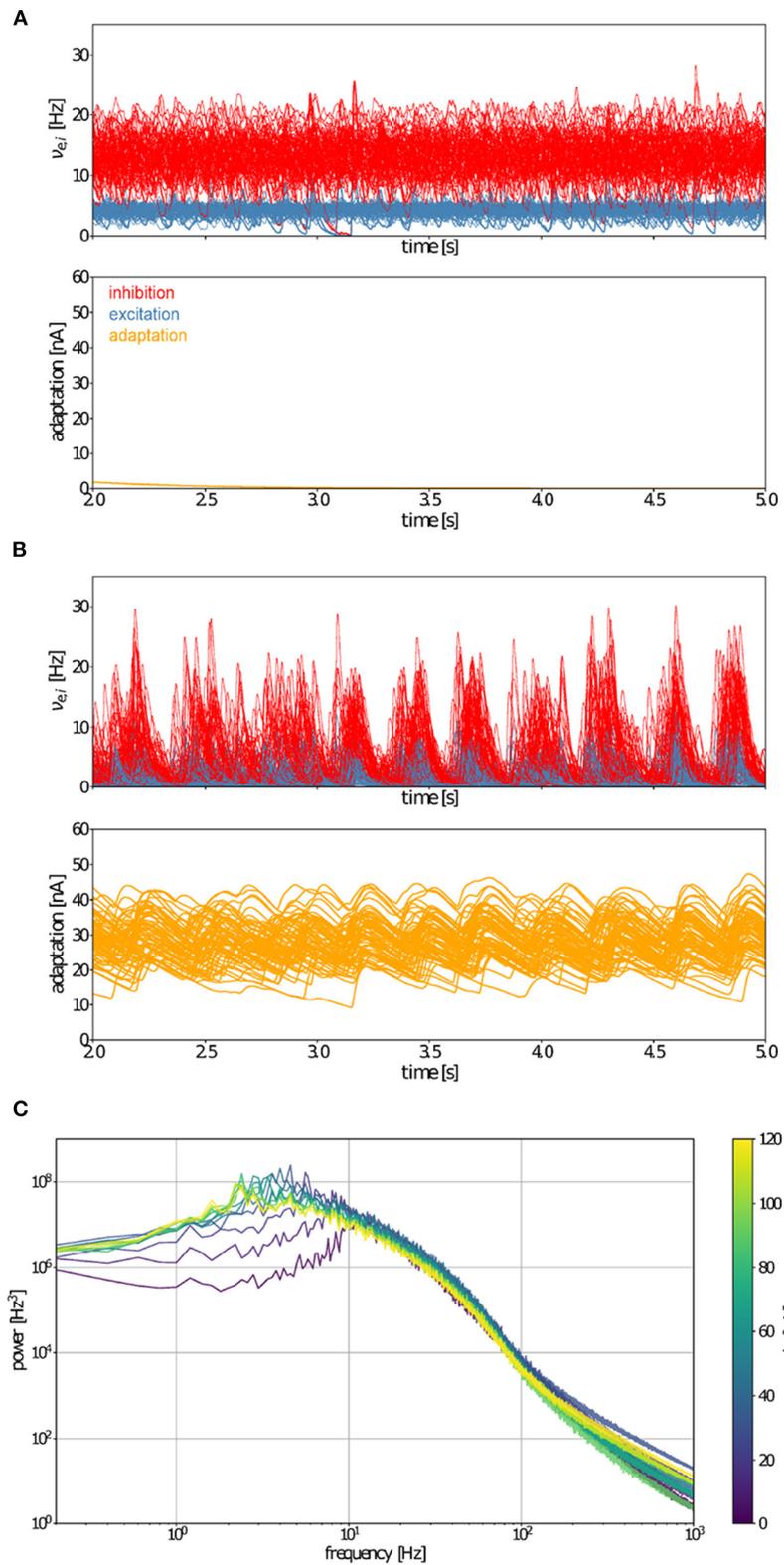
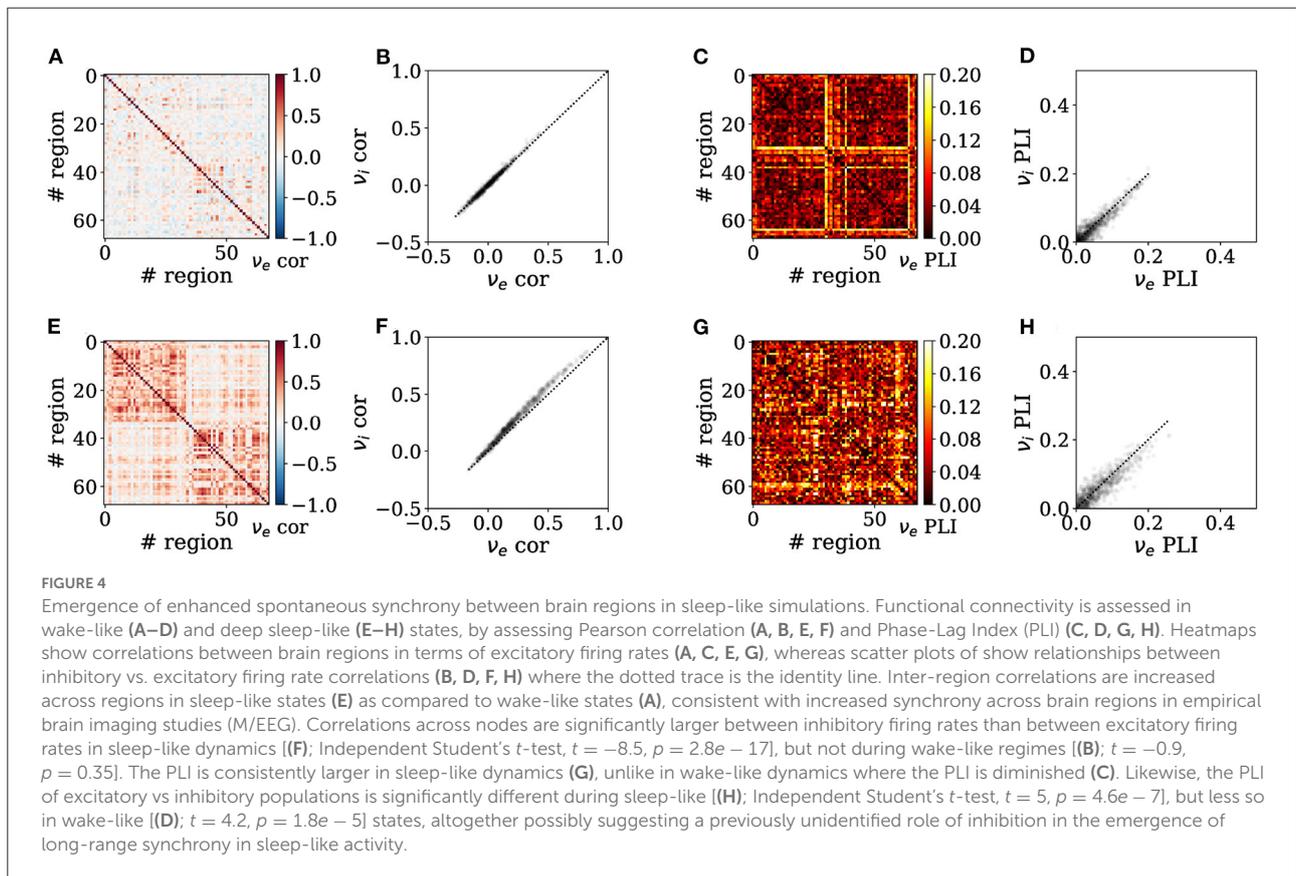


FIGURE 3 Whole-brain-scale simulations of connected AdEx mean-field models produce activity mimicking wake- and sleep-like states. Time variation of firing rates ($v_{e,i}$, top) and adaptation currents (w_e , bottom) in simulated wake- (**A**) and sleep-like (**B**) states for each of the model nodes representing 68 brain regions. When adaptation (b_e) equals 0 pA, the activity of model nodes is asynchronous (**A**), whereas the inclusion of (Continued)

FIGURE 3 (Continued)

adaptation ($b_e = 60 \text{ pA}$) leads to the emergence of synchrony between brain regions (B). (C) Fourier power spectra of signals produced by the TVB-AdEx in synchronous (sleep-like) and asynchronous (wake-like) states for different values of b_e . Note that maximal power in the sleep-like condition falls in the delta range (1–4 Hz), while it occurs near 10 Hz, in the alpha range for low adaptation, wake-like states.



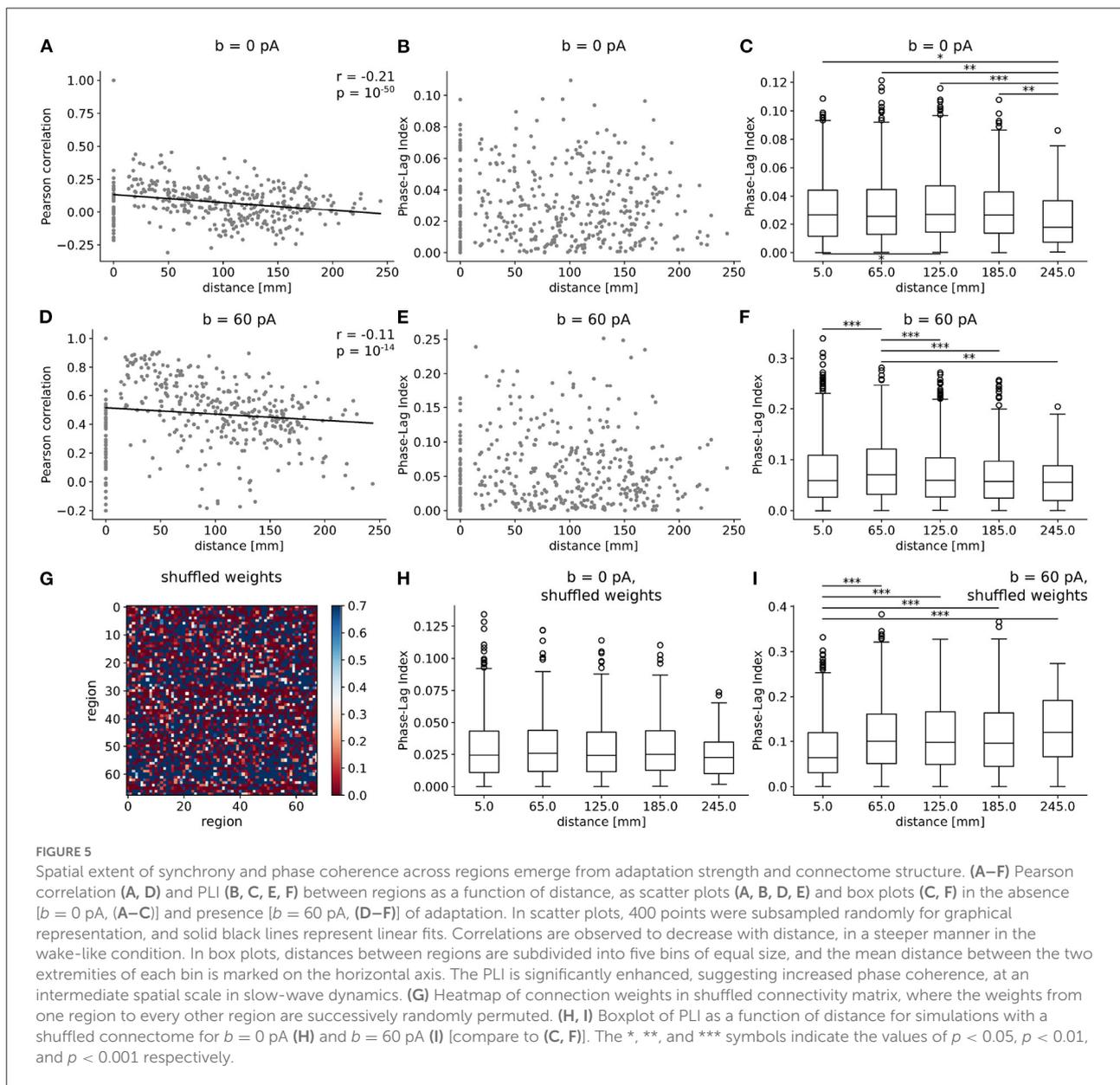
scale for traveling waves (Figure 5I)—is lost. These results suggest that the non-trivial organization of phase coherence phenomena across cortex is an emergent property of both high adaptation and the weighted graph structure of the human connectome.

Finally, the transition between wake and sleep-like dynamics when changing the level of adaptation is robust for different combinations of parameters of the model (Figure 6). With a high-density scan of the parameter space (see Methods), we find that, for lower values of spike-frequency adaptation ($b = 0 \text{ pA}$), AI states are present independently of the timescale (T) of the AdEx mean-field model network nodes and the coupling strength between nodes (S). Consequently, when increasing adaptation, there is a sharp transition from wake-like dynamics to slow-wave activity captured by a marked increase in firing rate standard deviation, again robustly across all values of T and S in the explored range.

2.4. Responsiveness to external stimulation

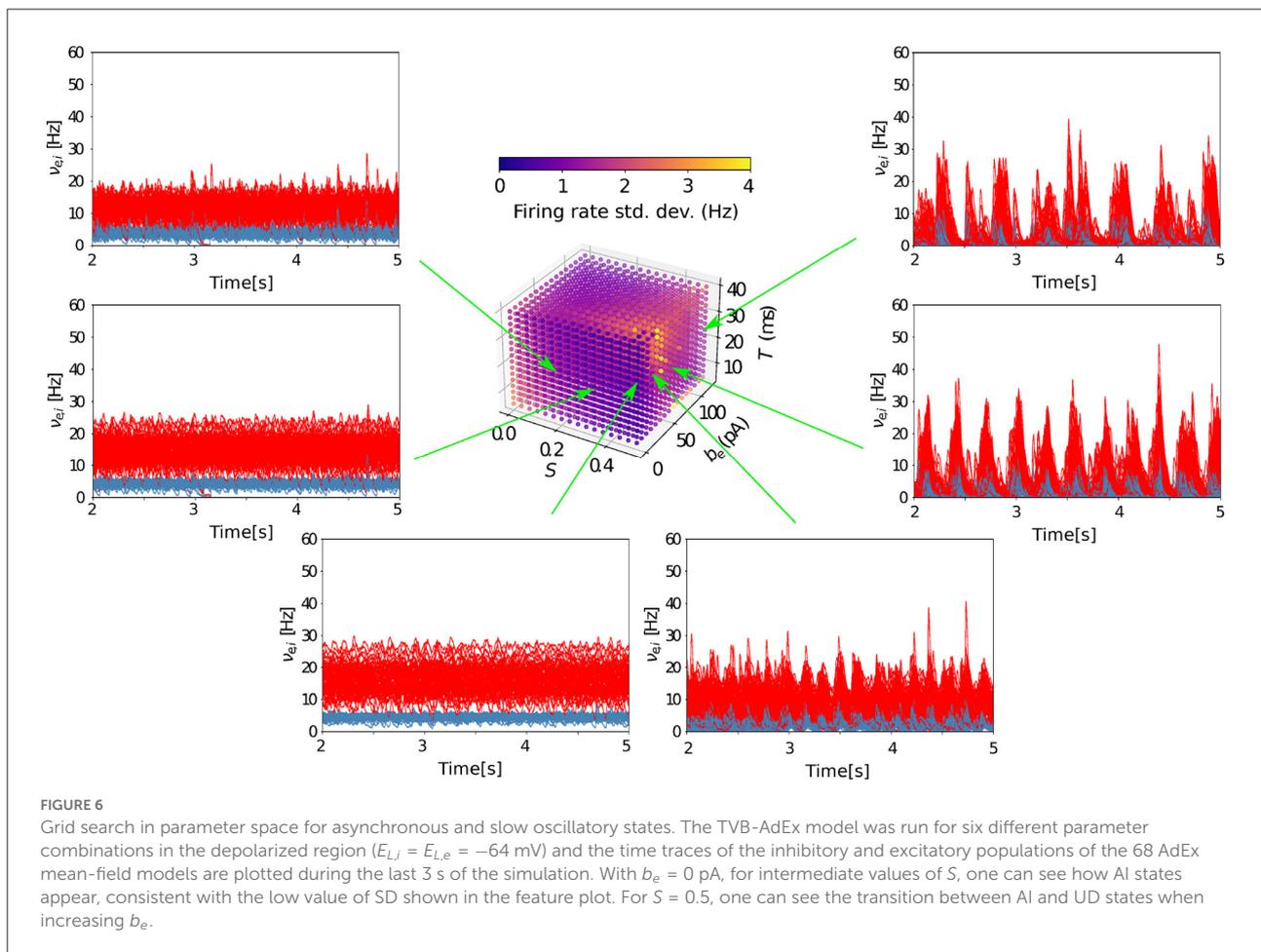
After reproducing features of spontaneous dynamics between brain states, we test the hypothesis that changing adaptation in the TVB-AdEx model can also explain differences in empirically observed stimulus-evoked brain responses, with stimuli encoded in more sustained, widespread, reliable, and complex patterns during conscious states (Massimini et al., 2005). To this end, a square wave of 0.1 Hz, matching the magnitude of stochastic drive, with 50 ms duration was input to the firing rates of the transfer function for excitatory populations in the right premotor cortex of the TVB-AdEx simulation during awake-like and slow wave sleep-like conditions, as in previously published empirical studies (Casali et al., 2013).

Figure 7 illustrates the effect of perturbing the large-scale network defined by the TVB-AdEx models. The effect of an



external stimulus is apparent for both deep sleep-like and wake-like states (Figure 7A). The average traces of the stimulated region are shown in black, and take into account the 40 realizations shown in gray. To examine the spread of activity following perturbations, the time at which the excitatory firing rate of each region becomes significantly different from the unstimulated baseline (prior to perturbation) is plotted using a color map showing earlier significant changes in brighter colors. Here, we find that responses are more widespread across time and space across brain regions in wake-like (Figure 7B) than sleep-like dynamics (Figure 7C), corresponding to experimental observations in response to Transcranial Magnetic Stimulation (TMS) (Massimini et al., 2005).

To better characterize the complexity of stimulus-evoked responses, the perturbational complexity index (PCI), used in previous experimental works involving TMS, was computed. The PCI is the ratio between Lempel-Ziv complexity, which captures the number of all possible different binary words that can be extracted from binarized responses to stimuli across time and regions, and the entropy of the binarized response that describes how often the response is above the pre-stimulus baseline (see Methods for binarization procedure). A low PCI value indicates a “simple” response to stimulus, while a high PCI value indicates more “complex” response, typically propagating more effectively to different brain areas (Casali et al., 2013). The models were perturbed with stimuli smaller (0.01 Hz) than the

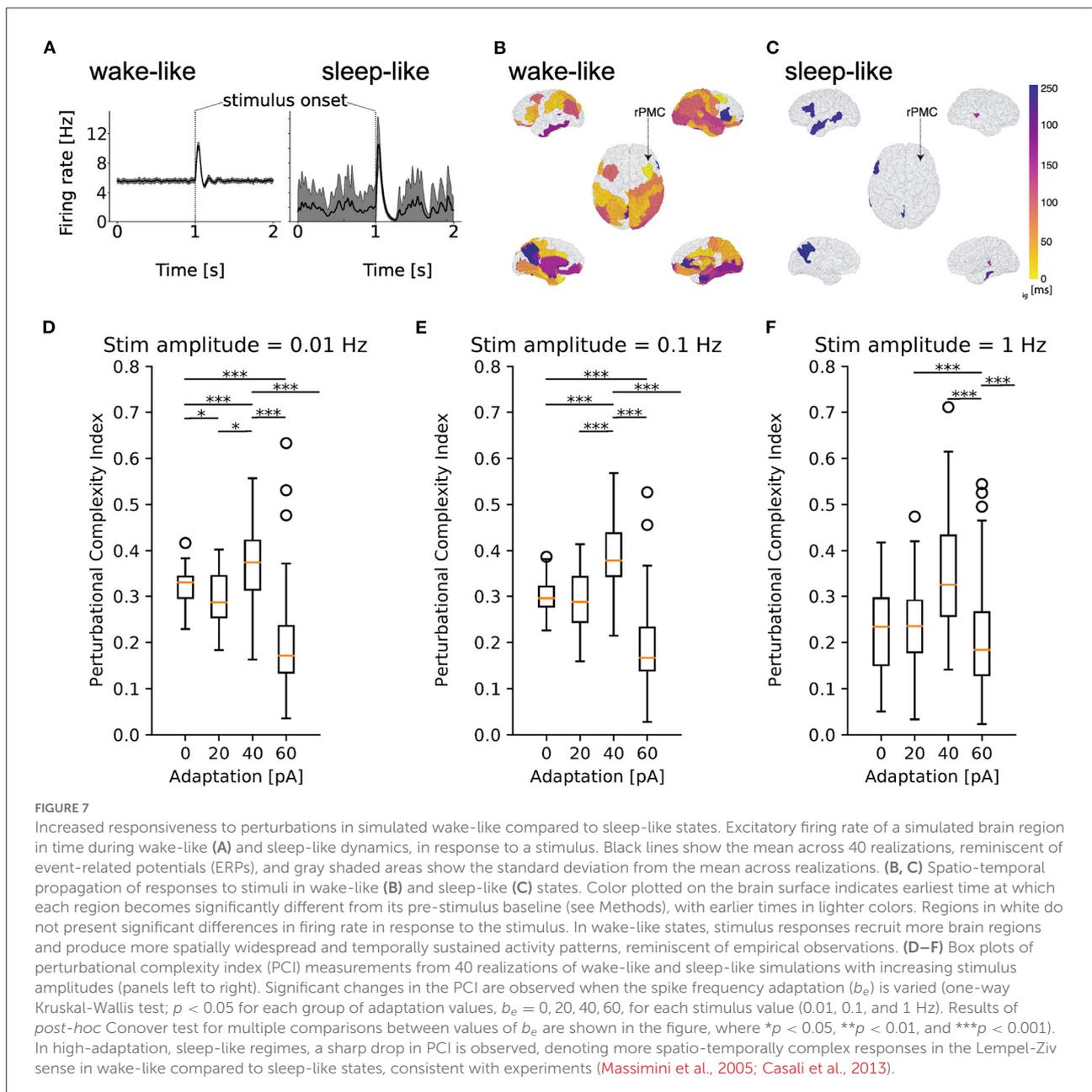


stochastic drive, comparable to the noise in amplitude (0.1 Hz), and larger than the drive (1.0 Hz) for simulations in which the value of spike frequency adaptation (b_e) was varied between 0 and 60 pA. As shown in **Figures 7D–F**, computing the PCI from the TVB-AdEx model shows that PCI values are typically higher for lower-adaptation, wake-like regimes than for higher-adaptation, slow-wave sleep like regimes. In particular, a sharp drop in PCI values is observed between $b = 40$ pA and $b = 60$ pA, suggesting an abrupt transition between highly responsive asynchronous and less responsive slow-wave dynamics as adaptation increases. For each value of noise, a one-way ANOVA revealed significant differences between PCI distributions across b -values ($p < 0.0001$), with multiple comparisons highlighting that the PCI was significantly larger for wake-like than sleep-like conditions, in particular for lower-amplitude stimuli. The same behavior was observed when comparing awake subjects with subjects in slow-wave sleep (Massimini et al., 2005; Casali et al., 2013). One may note a sharp drop of the PCI after $b = 40$ pA, for which enhanced PCI is observed, reminiscent of a transition from conscious to unconscious response dynamics. As concentrations of neuromodulators such as acetylcholine are

known to increase with attention and vigilance, $b = 40$ pA, which is at the higher-adaptation, lower-neuromodulation end of the spectrum of high-PCI states, could be reminiscent of states that lie between waking vigilance and deep sleep, such as resting wakefulness. Also note the wider distribution of PCI values in sleep-like simulations, suggesting more variable responses for each realization of the same stimulus and therefore less reliable stimulus encoding.

3. Discussion

In this paper, we demonstrated that biologically-informed scale-integrated mean-field models (di Volo et al., 2019) can be used to simulate large-scale brain networks using the TVB platform in EBRAINS. The coupled mean-field models comprising the TVB-AdEx are derived from networks of AdEx neurons and display whole-brain asynchronous and slow-wave dynamics when wired following white matter tracts from a human connectome. These results demonstrate the natural emergence of empirically observed patterns of macroscopic



brain dynamics from simulated changes at microscopic scales from both microscopic adaptation changes and the structural organization of the human connectome. The TVB-AdEx integration in EBRAINS is also of interest as EBRAINS human brain atlas services will be able to provide a large degree of cytoarchitectural detail such as region-specific neurotransmitter densities and cell types and densities and thus add to the biological realism of these virtual brain models. The vertical integration across scales is provided by TVB-AdEx-type models, taking advantage of the Big Data in EBRAINS.

TVB-AdEx mean-field models constituting each node of the connectome are designed by construction to approximate the mean and covariance of the firing rate in spiking neural networks exhibiting stable dynamics in asynchronous irregular regimes (El Boustani and Destexhe, 2009). This model was extended to two neuronal populations, excitatory neurons with adaptation and inhibitory neurons (di Volo et al., 2019), but this extension has limitations. Importantly, the model is imprecise when adaptation varies within a range larger than described here [for adaptation values higher than 100 pA (di Volo et al., 2019)] and when fast synchronous dynamics like oscillations

in the gamma range (between 40 and 80 Hz) (El Boustani and Destexhe, 2009), spindles, or ripples occur. This model is therefore likely not directly suitable for understanding the nuances associated with particular microscopic motifs comprised by transiently communicating assemblies that likely encode relevant neural information. The model is appropriate to the type of general dynamics presented here, wake-like AI and deep sleep-like slow-waves, describing large scale phenomena with relatively slow time scales. By smoothing microscopic details, we have built a computationally tractable bridge from microscopic to macroscopic scales, to elucidate how general dynamical phenomena relate to differences in neuronal interactions.

After integration in TVB, the resulting TVB-AdEx model displays a number of interesting features and several exciting perspectives for future work. A first result is the emergence of synchrony across brain regions in the presence of adaptation. In this case, the TVB-AdEx model displays synchronized slow waves with structured phase relations at a macroscopic, brain-wide level (Figure 3). This is consistent with the synchronized slow-wave dynamics observed during deep sleep in the brain (Destexhe et al., 1999; Steriade, 2003; Niedermeyer and Lopes da Silva, 2005). When the same model is set into the asynchronous-irregular regime due to the loss of spike-frequency adaptation, as is the case in the presence of acetylcholine and other neuromodulators present in higher concentrations during waking states (Jones, 2003), the large-scale network displays a lower level of synchrony (Figure 3), consistent with asynchronous dynamics typically seen in awake and aroused states (Destexhe et al., 1999; Steriade, 2003; Niedermeyer and Lopes da Silva, 2005). These different levels of synchrony are therefore emergent properties in the large-scale network, induced by changes at the microscopic level. A detailed grid scan in parameter space established the robustness of this phenomenon (Figure 6).

A second main result is that evoked dynamics are also state-dependent. When the network displays synchronized Up- and Down-states, a stimulus typically evokes a high amplitude, simple response that remains local in space and time. When the model resides in the asynchronous regime, the same stimulus evokes responses that are weaker in amplitude, but that propagate in a more elaborate way through space and time. The PCI measure applied to these two states match the experimental observations (Massimini et al., 2005; Casali et al., 2013; D'Andola et al., 2018; Dasilva et al., 2021). Again, this is an emergent property of the large-scale network.

What are possible mechanisms for such differences? A previous study (Zerlaut and Destexhe, 2017) showed that in balanced networks, not all states are equal and that asynchronous states, despite their apparently noisy character, can display higher responsiveness and support propagation of stimuli. This enhanced responsiveness of AI states can be explained by the combined effects of depolarization, membrane

potential fluctuations, and conductance state. It was proposed as a fundamental property to explain why the activity of the brain is systematically asynchronous in aroused states (Zerlaut and Destexhe, 2017). The present results are in full agreement with this mechanism, which manifests here in the asynchronous state as a propagation further in time and space, across many brain areas, associated with higher values of the PCI.

We believe that this work opens several perspectives. First, the enhanced propagation of perturbations during wake-like states could be used as a basis to explain why stimuli are perceived in asynchronous regimes, and what kind of modulation of the network activity could support phenomena such as attention and perception. Second, mean-field models can be set to also display pathological states, such as hyper-excitable or hypersynchronized states, and the TVB-AdEx model could be used to investigate seizure activity (Depannemaecker et al., 2021). Other features, such as neuronal heterogeneity, are also beginning to be included in mean-field models (di Volo and Destexhe, 2020), paving the way for enhancing biological realism in future versions of TVB-AdEx models.

In TVB, connectivity depends on the intermediate spatial resolution of coarse-graining. Here, the brain was parceled in 68 regions, with each mean-field representing a substantially large brain area. TVB allows such simple simulations using a few tens of nodes, taking into account the rough long-range connectivity according to the connectome resolved in tractography of human DTI. TVB can also simulate much finer-grained connectivity, by defining a larger number of nodes (usually on the order of hundreds to hundreds of thousands, approaching the resolution of cortical columns) (Spiegler and Jirsa, 2013). In such vertex-based simulations that shall follow the present work, local connectivity is determined by intracortical connections, whereas the white-matter connectome from DTI used here captures effects of longer-range cortico-cortical fibers.

Early stimulation studies in humans and in particular in rodents are pioneering the use of high-resolution simulations, demonstrating subtler influences of the connectome in scaffolding signal propagation through brain networks (Spiegler et al., 2016, 2020). In those studies, network nodes were equipped with generic neural mass dynamics, Andronov-Hopf-oscillators, which are theoretically appealing for their mathematical simplicity, but are limited with regard to biophysical interpretability of the results. The inclusion of high-resolution data from tracer studies in the Allen Institute was recently demonstrated in virtual mouse brain models to significantly increase the predictive power (Melozzi et al., 2019). As well, the inclusion of subject-specific, personalized connectomes in virtual brain models significantly outperforms generic simulations in predictive inter-individual variability (Melozzi et al., 2019; Hashemi et al., 2020). These studies point together to the importance of personalized brain network models in future clinical applications and affords novel methods supporting such goals (Falcon et al., 2016; Jirsa et al.,

2017). The virtual brains in [Spiegler et al. \(2016, 2020\)](#) captured the emergence of well-known resting state networks known during spontaneous activity, but also functionally specific brain responses to stimulation of regions along the processing chains of sensory systems from periphery up to primary sensory cortical areas. The latter responses heavily relied on the Default Mode Network (DMN) and were suggestive of the DMN playing a mechanistic role between functional networks. But neither brain state dependence, nor biological interpretation of the neural mass model parameters was possible, as it requires the incorporation of biological complexity and integration across scales provided here by the here by the TVB-AdEx approach. Ongoing efforts in EBRAINS aim to enrich high-resolution brain models with detailed information on regionally-variant physiological features (neurotransmitters, receptor densities, cell types, and densities) to next build the Virtual Big Brain, a high-resolution multi-scale brain network, which will be continuously updated and available to the community. The drawback of such fine grained simulations is that they typically require large computing resources as provided by EBRAINS, while coarse grained TVB simulations, as presented here, can easily be run on a standard workstation. To summarize, for the sake of the initial release of the TVB-AdEx models, we offer a relatively coarse parcellation, which will become more refined and personalized in future work.

The TVB-AdEx models presented here are constructed by connecting conductance-based, mean-field models of biologically-informed populations of neurons by a human connectome ([Sanz-Leon et al., 2015](#)). While the results presented in the main figures are made on the backbone of a single example human connectome, and many of the reported emergent phenomena could be reproduced with other topologies of connectivity, it is imperative to note that the human connectome backbone of the TVB-AdEx model is interchangeable between human subjects, representing an opportunity to construct personalized models, digital twins ([Evers and Salles, 2021](#); [Petkoski and Jirsa, 2022](#)), for any human subject for which diffusion tractography data are available. While a study of inter-individual variation is beyond the scope of this study, as a proof of principle, our data indicate differences in spectral features, particularly the power of low-frequency activity and power-law scaling, between personalized TVB-AdEx models made from two different healthy human subjects ([Supplementary Figure S1](#)). Indeed, a parameter scan using TVB-AdEx models derived from the subjects identifies different overlapping regions of parameter space in which transitions between sleep-like and wake-like activity between human subjects. Construction of next generation TVB-AdEx models with human connectomes for which biological and behavioral data are available, for example, from the Human Connectome Project cohorts, will allow researchers to use TVB-AdEx models methods introduced here to test predictions regarding inter-subject differences in

brain states and their transitions. All code is openly available in the EU digital neuroscience platform EBRAINS ([Schirner et al., 2022](#)) and on Github to facilitate progress in personalized brain modeling ([Falcon et al., 2016](#)) of neural states and their transitions in health and disease.

It is interesting to note that the global properties used here to characterize neural dynamics across brain states—synchrony, frequency spectra, responsiveness, and PCI—all reflect neural correlates of consciousness ([Skarda and Freeman, 1987](#); [Tononi and Edelman, 1998](#); [Sarasso et al., 2014](#); [Koch et al., 2016](#)). It has even been argued that asynchronous dynamics (so-called ‘activated EEG’) is so far one of the most “sensitive and reliable” neural correlates of consciousness ([Koch et al., 2016](#)). Though we have only presented results from stimulating one brain region in different brain states, in the interest of replicating experimental results ([Casali et al., 2013](#)), it is important to note that our approach offers the possibility to perturb any region within a given connectome to simulate the effects of various stimuli or tasks in a variety of states, as well as the dynamical consequences of transcranial stimulation in experimental and medical contexts. This further emphasizes the promise of the present modeling approach to understanding dynamics associated with conscious and non-conscious states, with broad potential applications in medicine and computation.

4. Materials and methods

Three types of models are used in this work: a network of spiking neurons, a mean-field model of this network, and a network of mean-field models implemented in The Virtual Brain (TVB). Here we describe these models successively.

4.1. Spiking network model

We considered networks of integrate-and-fire neuron models displaying spike-frequency adaptation, based on two previous papers ([Destexhe, 2009](#); [Zerlaut et al., 2018](#)). We used the Adaptive Exponential (AdEx) integrate-and-fire model ([Brette and Gerstner, 2005](#)). We considered a population of $N = 10^4$ neurons randomly connected with a connection probability of $p = 5\%$. We considered excitatory and inhibitory neurons, with 20% inhibitory neurons. The AdEx model permits to define two cell types, “regular-spiking” (RS) excitatory cells, displaying spike-frequency adaptation, and “fast spiking” (FS) inhibitory cells, with no adaptation. The dynamics of these neurons is given by the following equations:

$$c_m \frac{dv_k}{dt} = g_L(E_L - v_k) + g_L \Delta e^{\frac{v_k - v_{thr}}{\Delta}} - w_k + I_{syn} \quad (1)$$

$$u_w \frac{dw_k}{dt} = -w_k + b \sum_{t_{sp}(k)} \delta(t - t_{sp}(k)) + a(v_k - E_L), \quad (2)$$

where $c_m = 200$ pF is the membrane capacitance, v_k is the voltage of neuron k and, whenever $v_k > v_{peak} = -47.5$ mV for inhibitory neurons and $v_k > v_{peak} = -40.0$ mV for excitatory at time $t_{sp}(k)$, v_k is reset to the resting voltage $v_{reset} = -65$ mV and fixed to that value for a refractory time $T_{refr} = 5$ ms. The voltage threshold v_{thr} is -50 mV. The leak term g_L had a fixed conductance of $g_L = 10$ nS and the leakage reversal E_L was of -65 mV for inhibitory and -63 for excitatory. The exponential term had a different strength for RS and FS cells, i.e., $\Delta = 2$ mV ($\Delta = 0.5$ mV) for excitatory (inhibitory) cells. Inhibitory neurons were modeled as fast spiking FS neurons with no adaptation ($a = b = 0$ for all inhibitory neurons) while excitatory regular spiking RS neurons had a lower level of excitability due to the presence of adaptation (while b varied in our simulations we fixed subthreshold adaptation $a = 0$ nS and $u_w = 500$ ms).

The synaptic current I_{syn} received by neuron i is the result of the spiking activity of all neurons $j \in \text{pre}(i)$ pre-synaptic to neuron i . This current can be decomposed in the synaptic conductances evoked by excitatory E and inhibitory I pre-synaptic spikes

$$I_{syn} = G_{syn}^e(E_e - v_k) + G_{syn}^i(E_i - v_k),$$

Where $E_e = 0$ mV ($E_i = -80$ mV) is the excitatory (inhibitory) reversal potential. Excitatory synaptic conductances were modeled by a decaying exponential function that sharply increases by a fixed amount Q_E at each pre-synaptic spike, i.e.,:

$$G_{syn}^e(t) = Q_e \sum_{exc.pre} \Theta(t - t_{sp}^e(k)) e^{-(t - t_{sp}^e(k))/u_e},$$

Where Θ is the Heaviside function, $u_e = u_i = 5$ ms is the characteristic decay time of excitatory and inhibitory synaptic conductances, and $Q_e = 1.5$ nS ($Q_i = 5$ nS) the excitatory (inhibitory) quantal conductance. Inhibitory synaptic conductances are modeled using the same equation with $e \rightarrow i$. This network displays two different states according to the level of adaptation, $b = 0$ pA for asynchronous irregular states, and $b = 60$ pA for Up-Down states (see Zerlaut et al., 2018 for details).

4.2. Mean-field models

We considered a population model of a network of AdEx neurons, using a Master Equation formalism originally developed for balanced networks of integrate-and-fire neurons (El Boustani and Destexhe, 2009). This model was adapted to AdEx networks of RS and FS neurons (Zerlaut et al., 2018), and later modified to include adaptation (di Volo et al., 2019). The latter version is used here, which corresponds to the following equations using Einstein's index summation

convention where sum signs are omitted and repeated indices are summed over:

$$T \frac{\partial v_\mu}{\partial t} = (F_\mu - v_\mu) + \frac{1}{2} c_{\lambda\eta} \frac{\partial^2 F_\mu}{\partial v_\lambda \partial v_\eta} \quad (3)$$

$$T \frac{\partial c_{\lambda\eta}}{\partial t} = \delta_{\lambda\eta} \frac{F_\lambda(1/T - F_\eta)}{N_\lambda} + (F_\lambda - v_\lambda)(F_\eta - v_\eta) + \frac{\partial F_\lambda}{\partial v_\mu} c_{\eta\mu} + \frac{\partial F_\eta}{\partial v_\mu} c_{\lambda\mu} - 2c_{\lambda\eta} \quad (4)$$

$$\frac{\partial W}{\partial t} = -W/u_w + b v_e + a(\mu_V(v_e, v_i, W) - E_L), \quad (5)$$

where $\mu = \{e, i\}$ is the population index (excitatory or inhibitory), v_μ the population firing rate and $c_{\lambda\eta}$ the covariance between populations λ and η . W is a population adaptation variable (di Volo et al., 2019). The function $F_{\mu=\{e,i\}} = F_{\mu=\{e,i\}}(v_e, v_i, W)$ is the transfer function which describes the firing rate of population μ as a function of excitatory and inhibitory inputs (with rates v_e and v_i) and adaptation level W . These functions were estimated previously for RS and FS cells and in the presence of adaptation (di Volo et al., 2019).

At the first order, i.e., neglecting the dynamics of the covariance terms $c_{\lambda\eta}$, this model can be written simply as:

$$T \frac{dv_\mu}{dt} = (F_\mu - v_\mu), \quad (6)$$

Together with Equation (5). This system is equivalent to the well-known Wilson-Cowan model (Wilson and Cowan, 1972), with the specificity that the functions F need to be obtained according to the specific single neuron model under consideration. These functions were obtained previously for AdEx models of RS and FS cells (Zerlaut et al., 2018; di Volo et al., 2019) and the same are used here.

For a cortical volume modeled as two populations of excitatory and inhibitory neurons, the equations can be written as:

$$T \frac{dv_e}{dt} = \mathcal{F}_e(v_e + v_{aff} + v_{drive}, v_i) - v_e \quad (7)$$

$$T \frac{dv_i}{dt} = \mathcal{F}_i(v_e + v_{aff}, v_i) - v_i \quad (8)$$

$$\frac{dW}{dt} = -W/u_w + b v_e + a(\mu_V(v_e, v_i, W) - E_L), \quad (9)$$

where v_{aff} is the afferent thalamic input to the population of excitatory and inhibitory neurons and v_{drive} is an external noisy drive simulated by an Ornstein-Uhlenbeck process. The function μ_V is the average membrane potential of the population and is given by

$$\mu_V = \frac{\mu_{Ge}E_e + \mu_{Gi}E_i + i + g_L E_L - W}{\mu_{Ge} + \mu_{Gi} + g_L},$$

where the mean excitatory conductance is $\mu_{Ge} = v_e K_e u_e Q_e$ and similarly for inhibition.

This system describes the population dynamics of a single region, and was shown to closely match the dynamics of the spiking network (di Volo et al., 2019).

4.3. Networks of mean-field models

Extending our previous work at the mesoscale (Chemla et al., 2019; di Volo et al., 2019) to model large brain regions, we define networks of mean-field models, representing interconnected brain regions (each described by a mean-field model). We considered interactions between cortical regions as excitatory, while inhibitory connections remain local to each region. The equations of such a network, expanding the two-population mean-field (Equation 7), are given by:

$$\begin{aligned} T \frac{dv_e(k)}{dt} &= \mathcal{F}_e[v_e^{input}(k) + v_{aff}(k), v_i(k)] - v_e(k) \\ T \frac{dv_i(k)}{dt} &= \mathcal{F}_i[v_e^{input}(k) + v_{aff}(k), v_i(k)] - v_i(k) \quad (10) \\ \frac{dW(k)}{dt} &= -W(k)/u_w + bv_e(k) \\ &+ a(\mu_V(v_e(k), v_i(k), W(k)) - E_L), \quad (11) \end{aligned}$$

where $v_e(k)$ and $v_i(k)$ are the excitatory and inhibitory population firing rates at site k , respectively, $W(k)$ the level of adaptation of the population, and $v_e^{input}(k)$ is the excitatory synaptic input. The latter is given by:

$$v_e^{input}(k) = v_{drive}(k) + \sum_j C_{jk} v_e(j, t - \|j - k\|/v_c) \quad (12)$$

where the sum runs over all nodes j sending excitatory connections to node k , and C_{jk} is the strength of the connection from j to k (and is equal to 1 for $j = k$). Note that $v_e(j, t - \|j - k\|/v_c)$ is the activity of the excitatory population at node k at time $t - \|j - k\|/v_c$ to account for the delay of axonal propagation. Here, $\|j - k\|$ is the distance between nodes j and k and v_c is the axonal propagation speed.

4.4. Spontaneous activity

The Phase-Lag Index (PLI) was computed for each pair of nodes, averaged over simulation time. The Hilbert transform is employed to extract the phase $\psi(t)$ of the time series. From there, the PLI, given by

$$PLI \equiv |\langle \text{sign}(\psi_i(t) - \psi_j(t)) \rangle|, \quad (13)$$

is computed for nodes i and j , where $\langle \cdot \rangle$ denotes averaging over time (Silva Pereira et al., 2017). One may note that the PLI takes values between 0 (random phase relations or perfect synchrony) and 1 (perfect phase locking). In this work we report the mean PLI over all time epochs for excitatory and inhibitory firing rates of each region pair for each adaptation value.

4.5. Parameter space exploration

A model such as the TVB-AdEx contains many parameters whose impact on the dynamics needs to be understood. Additionally, it is necessary to have reasonable, physiological ranges determined for them. As described above, most of them have already been fixed *via* biological or mathematical arguments, but there is still a subset of parameters whose impact needed to be studied to have a deeper and general understanding of the model. In Table 1, one can find the characteristics of the parameters chosen and the reason for their choice. For each parameter, 16 evenly spaced values were obtained inside the described range. Preliminary results allowed to reduce the explored parameter space to a total of 675,840 different configurations to be analyzed. Using High Performance Computing, the simulation of each parameter combination was parallelized. For each configuration, a seven second simulation was run and, afterwards, multiple features were extracted (mean and standard deviation of the excitatory and inhibitory firing rates, mean value of functional connectivity, etc.). By plotting the value of these features as a function of the parameter values, one can observe the influence of the latter on the model's dynamics (as is shown in Figure 6).

TABLE 1 Name, description, reason of choice, range and units of the parameters chosen for the parameter scan.

Parameter	Description	Reason of choice	Range	Units
S	Coupling strength between nodes. Has to be chosen phenomenologically.	Has to be chosen phenomenologically.	[0, 0.5]	Adimensional
$E_{L,i}$	Leakage reversal potential of AdEx inhibitory neurons.	Resting membrane potential of a neuron might vary depending on external conditions.	[-80, -60]	mV
$E_{L,e}$	Leakage reversal potential of AdEx excitatory neurons.	Resting membrane potential of a neuron might vary depending on external conditions.	[-80, -60]	mV
T	Timescale of the AdEx mean field model.	Has to be chosen phenomenologically.	[5, 40]	ms
b_e	Adaptation strength of excitatory AdEx neurons.	Models the change in neuromodulation that induces transition between AI and UD.	[0, 120]	pA

4.6. Evoked activity

We computed the Perturbational Complexity Index (PCI) in response to a localized square wave stimulus, over the firing rates in a given brain region of a TVB-AdEx simulation, following the method proposed by Casali et al. (2013). This stimulus was applied by augmenting the firing rate of the excitatory population during the pulse. This is done for multiple trials with the same stimulus delivered to the same node at a random point in time and with different realizations of noise. The PCI is the ratio of two quantities: the Lempel-Ziv algorithmic complexity and the source entropy (Casali et al., 2013). To compute both quantities, firing rates $v(t)$ must be binarized to produce significance vectors $s(t)$. First, the trials are aligned to stimulation time, considering only the 300 ms before and after stimulus onset. Then, each node's firing rate is re-scaled and mean and standard deviation given by pre-stimulus activity averaged over nodes. Afterwards, all pre-stimulus firing rates are randomized across time bins, this procedure being repeated 500 times. The threshold for significance T is then given by the one-tail percentile of the maximum absolute value over all repetitions within a series of 20 trials. For each trial of those 20 trials, we can then write $s(t) = 1$ whenever post-stimulus $v(t) > T$ and $s(t) = 0$ otherwise. For what follows, we concatenate all $s(t)$ vectors from all simulation nodes into one single significance vector $S(t)$ per trial.

The Lempel-Ziv complexity $LZ(S)$ is the length of the “zipped” vector $S(t)$, i.e., the number of possible binary “words” that make up the binary vector $S(t)$. Briefly, $S(t)$ is sectioned successively into consecutive words of between one and N_t characters where N_t is the total length of $S(t)$. Scanning sequentially through all words, each new encountered word is added to a “dictionary,” and $LZ(S)$ is the total number of words in the dictionary at the end of the procedure.

The spatial source entropy $H(S)$ is given by:

$$H(S) = -p(S = 0) \log_2(p(S = 0)) - p(S = 1) \log_2(p(S = 1)), \quad (14)$$

where \log_2 denotes the base-two logarithm.

The PCI can then be expressed as $PCI(S) = \frac{LZ(S)}{H(S)}$.

Code availability

A python-based open-access code to run the present whole-brain model will be accessible online in the EBRAINS platform (<https://ebrains.eu>) as a companion to the publication of the present article. The scripts are also accessible on Github at <https://gitlab.ebrains.eu/kancourt/tvb-adex-showcase3-git>.

Data availability statement

Publicly available datasets were analyzed in this study. This data can be found at: <https://gitlab.ebrains.eu/kancourt/tvb-adex-showcase3-git>.

Author contributions

JG and LK wrote the models. JG, LK, BY, DD, KA, and DA characterized the models and ran simulations. T-AN, LK, BY, DD, KA, DA, and JG performed analyzes. JG, T-AN, VJ, and AD designed the work and wrote the manuscript. AD and VJ supervised the work. All authors contributed to the article and approved the submitted version.

Funding

This research was funded by the European Community (Human Brain Project, H2020-785907, and H2020-945539), the Centre National de la Recherche Scientifique (CNRS, France), the ANR PARADOX, and the ICODE excellence network.

Acknowledgments

The authors would like to thank Arnau Manasanch, Michael Schirner, Petra Ritter, Nuria Tort-Colet, Matteo di Volo, Michele Farisco, Thierry Nieuws, and Mavi Sanchez-Vives for helpful discussion of the work in progress. Many thanks to EBRAINS for help with implementation. We thank Michiel Van der Vlag and Sandra Díaz for their advice and discussion, as well as EBRAINS and the Juelich Computer Center for providing HPC resources.

Conflict of interest

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Publisher's note

All claims expressed in this article are solely those of the authors and do not necessarily represent those of their affiliated organizations, or those of the publisher, the editors and the reviewers. Any product that may be evaluated in this article, or claim that may be made by its manufacturer, is not guaranteed or endorsed by the publisher.

Supplementary material

The Supplementary Material for this article can be found online at: <https://www.frontiersin.org/articles/10.3389/fncom.2022.1058957/full#supplementary-material>

SUPPLEMENTARY FIGURE 1

Inter-individual differences in model parameters predicted by personal connectomes from different healthy human subjects. High performance computing (HPC) grid search of parameters from a TVB-AdEx model constructed with a second unique human connectome (subject DH20120806 available from <https://zenodo.org/record/3497545#>).

Y4YybezMK3l, compared to the TVB-AdEx model featured in the main text (subject QL20120814). Spectral features are extracted and differences between personalized TVB-AdEx models are shown in heatmaps. In particular, the two subject models show differing dependencies on connectivity strength and spike-frequency adaptation parameters, transitioning between wake- and sleep-like states at different values. Spectral profiles of simulated neural activity vary between subjects with distinct connectomes, with delta, theta, alpha power, and power law scaling showing differences in parameter space. With easily interchangeable individual connectomes from subjects in which behavioral and biological data are also available, TVB-AdEx models will help make testable personalized predictions regarding individual variation in transitions between conscious and unconscious brain states in healthy and pathological conditions.

References

- Brette, R., and Gerstner, W. (2005). Adaptive exponential integrate-and-fire model as an effective description of neuronal activity. *J. Neurophysiol.* 94, 3637–3642. doi: 10.1152/jn.00686.2005
- Capone, C., di Volo, M., Romagnoni, A., Mattia, M., and Destexhe, A. (2019). A state-dependent mean-field formalism to model different activity states in conductance based networks of spiking neurons. *bioRxiv* 565127. doi: 10.1103/PhysRevE.100.062413
- Carlu, M., Chehab, O., Dalla Porta, L., Depannemaecker, D., Héricé, C., Jedynak, M., et al. (2020). A mean-field approach to the dynamics of networks of complex neurons, from nonlinear integrate-and-fire to hodgkin-huxley models. *J. Neurophysiol.* 123, 1042–1051. doi: 10.1152/jn.00399.2019
- Casali, A. G., Gosseries, O., Rosanova, M., Boly, M., Sarasso, S., Casali, K. R., et al. (2013). A theoretically based index of consciousness independent of sensory processing and behavior. *Sci. Transl. Med.* 5, 198ra105. doi: 10.1126/scitranslmed.3006294
- Chemla, S., Reynaud, A., di Volo, M., Zerlaut, Y., Perrinet, L., Destexhe, A., et al. (2019). Suppressive traveling waves shape representations of illusory motion in primary visual cortex of awake primate. *J. Neurosci.* 39, 4282–4298. doi: 10.1523/JNEUROSCI.2792-18.2019
- D'Andola, M., Rebollo, B., Casali, A. G., Weinert, J. F., Pigorini, A., Villa, R., et al. (2018). Bistability, causality, and complexity in cortical networks: an in vitro perturbational study. *Cereb. Cortex* 28, 2233–2242. doi: 10.1093/cercor/bhx122
- Dasilva, M., Camassa, A., Navarro-Guzman, A., Pazienti, A., Perez-Mendez, L., Zamora-López, G., et al. (2021). Modulation of cortical slow oscillations and complexity across anesthesia levels. *Neuroimage* 224, 117415. doi: 10.1016/j.neuroimage.2020.117415
- Depannemaecker, D., Destexhe, A., Jirsa, V., and Bernard, C. (2021). Modeling seizures: from single neurons to networks. *Seizure* 90, 4–8. doi: 10.1016/j.seizure.2021.06.015
- Destexhe, A. (2009). Self-sustained asynchronous irregular states and up-down states in thalamic, cortical and thalamocortical networks of nonlinear integrate-and-fire neurons. *J. Comput. Neurosci.* 27, 493. doi: 10.1007/s10827-009-0164-4
- Destexhe, A., Contreras, D., and Steriade, M. (1999). Spatiotemporal analysis of local field potentials and unit discharges in cat cerebral cortex during natural wake and sleep states. *J. Neuroscience* 19, 4595–4608. doi: 10.1523/JNEUROSCI.19-11-04595.1999
- di Volo, M., and Destexhe, A. (2020). Optimal responsiveness and collective oscillations emerging from the heterogeneity of inhibitory neurons. *arXiv*. doi: 10.48550/arXiv.2005.05596
- di Volo, M., Romagnoni, A., Capone, C., and Destexhe, A. (2019). Biologically realistic mean-field models of conductance-based networks of spiking neurons with adaptation. *Neural Comput.* 31, 653–680. doi: 10.1162/neco_a_01173
- El Boustani, S., and Destexhe, A. (2009). A master equation formalism for macroscopic modeling of asynchronous irregular activity states. *Neural Comput.* 21, 46–100. doi: 10.1162/neco.2009.02-08-710
- El Boustani, S., and Destexhe, A. (2010). Brain dynamics at multiple scales: can one reconcile the apparent low-dimensional chaos of macroscopic variables with the seemingly stochastic behavior of single neurons? *Int. J. Bifurcat. Chaos* 20, 1687–1702. doi: 10.1142/S0218127410026769
- Evers, K., and Salles, A. (2021). Epistemic challenges of digital twins and virtual brains: perspectives from fundamental neuroethics. *SCIO Rev. Filosofia* 27–53. doi: 10.46583/scio_2021.21.846
- Falcon, M. I., Jirsa, V., and Solodkin, A. (2016). A new neuroinformatics approach to personalized medicine in neurology: the virtual brain. *Curr. Opin. Neurol.* 29, 429. doi: 10.1097/WCO.0000000000000344
- Hashemi, M., Vattikonda, A., Sip, V., Guye, M., Bartolomei, F., Woodman, M., et al. (2020). The bayesian virtual epileptic patient: a probabilistic framework designed to infer the spatial map of epileptogenicity in a personalized large-scale brain model of epilepsy spread. *Neuroimage* 217, 116839. doi: 10.1016/j.neuroimage.2020.116839
- Jercog, D., Roxin, A., Bartho, P., Luczak, A., Compte, A., and de la Rocha, J. (2017). Up-down cortical dynamics reflect state transitions in a bistable network. *eLife* 6, e22425. doi: 10.7554/eLife.22425.018
- Jirsa, V. K., Proix, T., Perdikis, D., Woodman, M. M., Wang, H., Gonzalez-Martinez, J., et al. (2017). The virtual epileptic patient: individualized whole-brain models of epilepsy spread. *Neuroimage* 145, 377–388. doi: 10.1016/j.neuroimage.2016.04.049
- Jones, B. E. (2003). Arousal systems. *Front. Biosci.* 8, 1074. doi: 10.2741/1074
- Koch, C., Massimini, M., Boly, M., and Tononi, G. (2016). Neural correlates of consciousness: progress and problems. *Nat. Rev. Neurosci.* 17, 307–321. doi: 10.1038/nrn.2016.22
- Le Van Quyen, M., Muller, L. E., Telenczuk, B., Halgren, E., Cash, S., Hatsopoulos, N. G., et al. (2016). High-frequency oscillations in human and monkey neocortex during the wake-sleep cycle. *Proc. Natl. Acad. Sci. U.S.A.* 113, 9363–9368. doi: 10.1073/pnas.1523583113
- Massimini, M., Ferrarelli, F., Huber, R., Esser, S. K., Singh, H., and Tononi, G. (2005). Breakdown of cortical effective connectivity during sleep. *Science* 309, 2228–2232. doi: 10.1126/science.1117256
- McCormick, D. A. (1992). Neurotransmitter actions in the thalamus and cerebral cortex and their role in neuromodulation of thalamocortical activity. *Progr. Neurobiol.* 39, 337–388. doi: 10.1016/0301-0082(92)90012-4
- Melozzi, F., Bergmann, E., Harris, J. A., Kahn, I., Jirsa, V., and Bernard, C. (2019). Individual structural features constrain the mouse functional connectome. *Proc. Natl. Acad. Sci. U.S.A.* 116, 26961–26969. doi: 10.1073/pnas.1906694116
- Nghiem, T.-A., Telenczuk, B., Marre, O., Destexhe, A., and Ferrari, U. (2018). Maximum-entropy models reveal the excitatory and inhibitory correlation structures in cortical neuronal activity. *Phys. Rev. E* 98, 012402. doi: 10.1103/PhysRevE.98.012402
- Nghiem, T.-A. E., Tort-Colet, N., Górski, T., Ferrari, U., Moghimi-firoozabad, S., Goldman, J. S., et al. (2020). Cholinergic switch between two types of slow waves in cerebral cortex. *Cereb. Cortex* 30, 3451–3466. doi: 10.1093/cercor/bhz320
- Niedermeyer, E., and Lopes da Silva, F. H. (2005). *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Philadelphia, PA: Lippincott Williams & Wilkins.
- Olcese, U., Bos, J. J., Vinck, M., Lankelma, J. V., van Mourik-Donga, L. B., Schlumm, F., et al. (2016). Spike-based functional connectivity in cerebral cortex and hippocampus: loss of global connectivity is coupled to preservation of local connectivity during non-rem sleep. *J. Neurosci.* 36, 7676–7692. doi: 10.1523/JNEUROSCI.4201-15.2016

- Petkoski, S., and Jirsa, V. (2022). Normalizing the brain connectome for communication through synchronization. *bioRxiv* 2020–2021. doi: 10.1162/netn_a_00231
- Peyrache, A., Dehghani, N., Eskandar, E. N., Madsen, J. R., Anderson, W. S., Donoghue, J., et al. (2012). Spatiotemporal dynamics of neocortical excitation and inhibition during human sleep. *Proc. Natl. Acad. Sci. U.S.A.* 109, 1731–1736. doi: 10.1073/pnas.1109895109
- Rudolph, M., Pospischil, M., Timofeev, I., and Destexhe, A. (2007). Inhibition determines membrane potential dynamics and controls action potential generation in awake and sleeping cat cortex. *J. Neurosci.* 27, 5280–5290. doi: 10.1523/JNEUROSCI.4652-06.2007
- Sanz-Leon, P., Knock, S. A., Spiegler, A., and Jirsa, V. K. (2015). Mathematical framework for large-scale brain network modeling in the virtual brain. *Neuroimage* 111, 385–430. doi: 10.1016/j.neuroimage.2015.01.002
- Sarasso, S., Rosanova, M., Casali, A. G., Casarotto, S., Fecchio, M., Boly, M., et al. (2014). Quantifying cortical eeg responses to tms in (un) consciousness. *Clin. EEG Neurosci.* 45, 40–49. doi: 10.1177/1550059413513723
- Schirner, M., Domide, L., Perdakis, D., Triebkorn, P., Stefanovski, L., Pai, R., et al. (2022). Brain simulation as a cloud service: The virtual brain on ebrains. *Neuroimage* 251, 118973. doi: 10.1016/j.neuroimage.2022.118973
- Schirner, M., Rothmeier, S., Jirsa, V. K., McIntosh, A. R., and Ritter, P. (2015). An automated pipeline for constructing personalized virtual brains from multimodal neuroimaging data. *Neuroimage* 117, 343–357. doi: 10.1016/j.neuroimage.2015.03.055
- Silva Pereira, S., Hindriks, R., Mühlberg, S., Maris, E., van Ede, F., Griffa, A., et al. (2017). Effect of field spread on resting-state magneto encephalography functional network analysis: a computational modeling study. *Brain Connect.* 7, 541–557. doi: 10.1089/brain.2017.0525
- Skarda, C. A., and Freeman, W. J. (1987). How brains make chaos in order to make sense of the world. *Behav. Brain Sci.* 10, 161–173. doi: 10.1017/S0140525X00047336
- Spiegler, A., Abadchi, J. K., Mohajerani, M., and Jirsa, V. K. (2020). In silico exploration of mouse brain dynamics by focal stimulation reflects the organization of functional networks and sensory processing. *Netw. Neurosci.* 4, 807–851. doi: 10.1162/netn_a_00152
- Spiegler, A., Hansen, E. C., Bernard, C., McIntosh, A. R., and Jirsa, V. K. (2016). Selective activation of resting-state networks following focal stimulation in a connectome-based network model of the human brain. *eNeuro* 3, ENEURO.0068-16.2016. doi: 10.1523/ENEURO.0068-16.2016
- Spiegler, A., and Jirsa, V. (2013). Systematic approximations of neural fields through networks of neural masses in the virtual brain. *Neuroimage* 83, 704–725. doi: 10.1016/j.neuroimage.2013.06.018
- Steriade, M. (2003). *Neuronal Substrates of Sleep and Epilepsy*. Cambridge: Cambridge University Press.
- Steriade, M., Timofeev, I., and Grenier, F. (2001). Natural waking and sleep states: a view from inside neocortical neurons. *J. Neurophysiol.* 85, 1969–1985. doi: 10.1152/jn.2001.85.5.1969
- Tononi, G., and Edelman, G. M. (1998). Consciousness and complexity. *Science* 282, 1846–1851. doi: 10.1126/science.282.5395.1846
- Wilson, H. R., and Cowan, J. D. (1972). Excitatory and inhibitory interactions in localized populations of model neurons. *Biophys. J.* 12, 1–24. doi: 10.1016/S0006-3495(72)86068-5
- Zerlaut, Y., Chemla, S., Chavane, F., and Destexhe, A. (2018). Modeling mesoscopic cortical dynamics using a mean-field model of conductance-based networks of adaptive exponential integrate-and-fire neurons. *J. Comput. Neurosci.* 44, 45–61. doi: 10.1007/s10827-017-0668-2
- Zerlaut, Y., and Destexhe, A. (2017). Enhanced responsiveness and low-level awareness in stochastic network states. *Neuron* 94, 1002–1009. doi: 10.1016/j.neuron.2017.04.001
- Zerlaut, Y., Telerńczuk, B., Deleuze, C., Bal, T., Ouanounou, G., and Destexhe, A. (2016). Heterogeneous firing rate response of mouse layer v pyramidal neurons in the fluctuation-driven regime. *J. Physiol.* 594, 3791–3808. doi: 10.1113/JP272317

CHAPTER C

Brain simulation as a cloud service : The Virtual Brain on EBRAINS

C.1. Chapter overview

This chapter is a side work of this thesis. It describes an overview of the tools in EBRAINS using The Virtual Brain [299]. My participation in this work is the co-writing of the section about co-simulation and the creation of associated figures.

This chapter is based on the published article :

Brain simulation as a cloud service : The Virtual Brain on EBRAINS

Michael Schirner^{1,2,3,4,5}, Lia Domide⁶, Dionysios Perdikis^{1,2}, Paul Triebkorn^{1,2,7}, Leon Stefanovski^{1,2}, Roopa Pai^{1,2}, Paula Prodan⁶, Bogdan Valean⁶, Jessica Palmer^{1,2}, Chloë Langford^{1,2}, André Blickensdörfer^{1,2}, Michiel van der Vlag⁸, Sandra Diaz-Pier⁸, Alexander Peyser⁸, Wouter Klijn⁸, Dirk Pleiter^{9,10}, Anne Nahm⁹, Oliver Schmid¹¹, Marmaduke Woodman⁷, Lyuba Zehl¹², Jan Fousek⁷, Spase Petkoski⁷, Lionel Kusch⁷, Meysam Hashemi⁷, Daniele Marinazzo^{13,14}, Jean-François Mangin¹⁵, Agnes Flöel^{16,17}, Simisola Akintoye¹⁸, Bernd Carsten Stahl¹⁹, Michael Cepic²⁰, Emily Johnson²⁰, Gustavo Deco^{21,22,23,24}, Anthony R. McIntosh²⁵, Claus C. Hilgetag^{26,27}, Marc Morgan¹¹, Bernd Schuller⁹, Alex Upton²⁸, Colin McMurtrie²⁸, Timo Dickscheid¹², Jan G. Bjaalie²⁹, Katrin Amunts^{12,30}, Jochen Mersmann³¹, Viktor Jirsa⁷, Petra Ritter^{1,2,3,4,5}

¹ Berlin Institute of Health at Charité, Universitätsmedizin Berlin, Charitéplatz 1, 10117 Berlin, Germany

² Charité-Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt Universität zu Berlin, Department of Neurology with Experimental Neurology, Charitéplatz 1, 10117 Berlin, Germany

³ Bernstein Focus State Dependencies of Learning & Bernstein Center for Computational Neuroscience Berlin, Germany

⁴ Einstein Center for Neuroscience Berlin, Charitéplatz 1, 10117 Berlin, Germany

⁵ Einstein Center Digital Future, Wilhelmstraße 67, 10117 Berlin, Germany

⁶ Codemart, Str. Petofi Sandor, Cluj, Napoca, Romania

⁷ Institut de Neurosciences des Systèmes, Aix Marseille Université, Marseille, France

⁸ SimLab Neuroscience, Jülich Supercomputing Centre (JSC), Institute for Advanced Simulation, JARA, Forschungszentrum Jülich GmbH, Jülich, Germany

- ⁹ Jülich Supercomputer Centre, Forschungszentrum Jülich, Jülich, Germany
- ¹⁰ Institut für Theoretische Physik, Universität Regensburg, Regensburg, Germany
- ¹¹ Swiss Federal Institute of Technology Lausanne (EPFL), Campus Biotech, Geneva, Switzerland
- ¹² Institute of Neuroscience and Medicine (INM-1), Research Centre Jülich, Jülich, Germany
- ¹³ Department of Data-Analysis, Faculty of Psychology and Educational Sciences, Ghent University, Belgium
- ¹⁴ San Camillo IRCSS, Venice, Italy
- ¹⁵ Université Paris-Saclay, CEA, CNRS, Neurospin, Baobab, Gif-sur-Yvette, France
- ¹⁶ Department of Neurology, University Medicine Greifswald, Greifswald, Germany
- ¹⁷ German Center for Neurodegenerative Diseases (DZNE) Standort Greifswald, Greifswald, Germany
- ¹⁸ Leicester De Montfort Law School, De Montfort University, Leicester, United Kingdom
- ¹⁹ Centre for Computing and Social Responsibility, De Montfort University, Leicester, United Kingdom
- ²⁰ Department of Innovation and Digitalisation in Law, Faculty of Law, University of Vienna, Austria
- ²¹ Center for Brain and Cognition, Computational Neuroscience Group, Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain
- ²² Institució Catalana de la Recerca i Estudis Avançats, Barcelona, Spain
- ²³ Department of Neuropsychology, Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany
- ²⁴ School of Psychological Sciences, Turner Institute for Brain and Mental Health, Monash University, Melbourne, Clayton, Australia
- ²⁵ Baycrest Health Sciences, Rotman Research Institute, Toronto, ON, Canada
- ²⁶ Institute of Computational Neuroscience, University Medical Center Hamburg-Eppendorf, Hamburg, Germany
- ²⁷ Department of Health Sciences, Boston University, 635 Commonwealth Ave., Boston, Massachusetts 02215, United States
- ²⁸ CSCS Swiss National Supercomputing Centre, Lugano, Switzerland
- ²⁹ Institute of Basic Medical Sciences, University of Oslo, Norway
- ³⁰ C. and O. Vogt Institute for Brain Research, University Hospital Düsseldorf, Heinrich-Heine University Düsseldorf, Germany
- ³¹ CodeBox GmbH, Stuttgart, Germany



Contents lists available at ScienceDirect

NeuroImage

journal homepage: www.elsevier.com/locate/neuroimage

Brain simulation as a cloud service: The Virtual Brain on EBRAINS

Michael Schirner^{a,b,c,d,e,*}, Lia Domide^f, Dionysios Perdikis^{a,b}, Paul Triebkorn^{a,b,g}, Leon Stefanovski^{a,b}, Roopa Pai^{a,b}, Paula Prodan^f, Bogdan Valean^f, Jessica Palmer^{a,b}, Chloë Langford^{a,b}, André Blickensdörfer^{a,b}, Michiel van der Vlag^h, Sandra Diaz-Pier^h, Alexander Peyser^h, Wouter Klijn^h, Dirk Pleiter^{i,j}, Anne Nahmⁱ, Oliver Schmid^k, Marmaduke Woodman^g, Lyuba Zehl^l, Jan Fousek^g, Spase Petkoski^g, Lionel Kusch^g, Meysam Hashemi^g, Daniele Marinazzo^{m,n}, Jean-François Mangin^o, Agnes Flöel^{p,q}, Simisola Akintoye^r, Bernd Carsten Stahl^s, Michael Cepic^t, Emily Johnson^t, Gustavo Deco^{u,v,w,x}, Anthony R. McIntosh^y, Claus C. Hilgetag^{z,aa}, Marc Morgan^k, Bernd Schullerⁱ, Alex Upton^{bb}, Colin McMurtrie^{bb}, Timo Dickscheid^l, Jan G. Bjaalie^{cc}, Katrin Amunts^{l,dd}, Jochen Mersmann^{ee}, Viktor Jirsa^g, Petra Ritter^{a,b,c,d,e,*}

^a Berlin Institute of Health at Charité, Universitätsmedizin Berlin, Charitéplatz 1, 10117 Berlin, Germany

^b Charité-Universitätsmedizin Berlin, corporate member of Freie Universität Berlin and Humboldt Universität zu Berlin, Department of Neurology with Experimental Neurology, Charitéplatz 1, 10117 Berlin, Germany

^c Bernstein Focus State Dependencies of Learning & Bernstein Center for Computational Neuroscience Berlin, Germany

^d Einstein Center for Neuroscience Berlin, Charitéplatz 1, 10117 Berlin, Germany

^e Einstein Center Digital Future, Wilhelmstraße 67, 10117 Berlin, Germany

^f Codemart, Str. Petoși Sandor, Cluj, Napoca, Romania

^g Institut de Neurosciences des Systèmes, Aix Marseille Université, Marseille, France

^h SimLab Neuroscience, Jülich Supercomputing Centre (JSC), Institute for Advanced Simulation, JARA, Forschungszentrum Jülich GmbH, Jülich, Germany

ⁱ Jülich Supercomputer Centre, Forschungszentrum Jülich, Jülich, Germany

^j Institut für Theoretische Physik, Universität Regensburg, Regensburg, Germany

^k Swiss Federal Institute of Technology Lausanne (EPFL), Campus Biotech, Geneva, Switzerland

^l Institute of Neuroscience and Medicine (INM-1), Research Centre Jülich, Jülich, Germany

^m Department of Data-Analysis, Faculty of Psychology and Educational Sciences, Ghent University, Belgium

ⁿ San Camillo IRCSS, Venice, Italy

^o Université Paris-Saclay, CEA, CNRS, Neurospin, Baobab, Gif-sur-Yvette, France

^p Department of Neurology, University Medicine Greifswald, Greifswald, Germany

^q German Center for Neurodegenerative Diseases (DZNE) Standort Greifswald, Greifswald, Germany

^r Leicester De Montfort Law School, De Montfort University, Leicester, United Kingdom

^s Centre for Computing and Social Responsibility, De Montfort University, Leicester, United Kingdom

^t Department of Innovation and Digitalisation in Law, Faculty of Law, University of Vienna, Austria

^u Center for Brain and Cognition, Computational Neuroscience Group, Department of Information and Communication Technologies, Universitat Pompeu Fabra, Barcelona, Spain

^v Institució Catalana de la Recerca i Estudis Avançats, Barcelona, Spain

^w Department of Neuropsychology, Max Planck Institute for Human Cognitive and Brain Sciences, Leipzig, Germany

^x School of Psychological Sciences, Turner Institute for Brain and Mental Health, Monash University, Melbourne, Clayton, Australia

^y Baycrest Health Sciences, Rotman Research Institute, Toronto, ON, Canada

^z Institute of Computational Neuroscience, University Medical Center Hamburg-Eppendorf, Hamburg, Germany

^{aa} Department of Health Sciences, Boston University, 635 Commonwealth Ave., Boston, Massachusetts 02215, United States

^{bb} CSCS Swiss National Supercomputing Centre, Lugano, Switzerland

^{cc} Institute of Basic Medical Sciences, University of Oslo, Norway

^{dd} C. and O. Vogt Institute for Brain Research, University Hospital Düsseldorf, Heinrich-Heine University Düsseldorf, Germany

^{ee} CodeBox GmbH, Stuttgart, Germany

* Corresponding authors.

E-mail addresses: michael.schirner@bih-charite.de (M. Schirner), petra.ritter@bih-charite.de (P. Ritter).

<https://doi.org/10.1016/j.neuroimage.2022.118973>.

Received 30 November 2021; Received in revised form 2 February 2022; Accepted 3 February 2022

Available online 4 February 2022.

1053-8119/© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>)



ARTICLE INFO

Keywords:

Brain modelling
 Cloud
 Connectome
 Neuroimaging
 Network model
 High performance computing
 Reproducibility
 Data protection

ABSTRACT

The Virtual Brain (TVB) is now available as open-source services on the cloud research platform EBRAINS (ebrains.eu). It offers software for constructing, simulating and analysing brain network models including the TVB simulator; magnetic resonance imaging (MRI) processing pipelines to extract structural and functional brain networks; combined simulation of large-scale brain networks with small-scale spiking networks; automatic conversion of user-specified model equations into fast simulation code; simulation-ready brain models of patients and healthy volunteers; Bayesian parameter optimization in epilepsy patient models; data and software for mouse brain simulation; and extensive educational material. TVB cloud services facilitate reproducible online collaboration and discovery of data assets, models, and software embedded in scalable and secure workflows, a precondition for research on large cohort data sets, better generalizability, and clinical translation.

1. Introduction

This paper introduces cloud services for brain simulation that are now being offered on the open brain research platform EBRAINS (European brain research infrastructures; ebrains.eu), which makes scientific data, tools, and results accessible to everyone within a protected environment that promotes reproducible work. Scientific studies depend on increasingly complex workflows that are often difficult to replicate and the produced findings are often not confirmed by additional data (Aarts et al., 2015; Ioannidis, 2005). The data and the computational steps that produced the findings as well as the explicit workflow describing how to generate the results were identified as the minimal components for independent reproduction of computational results (Stodden et al., 2016). EBRAINS addresses these challenges by offering modelling and simulation services for collaborative brain research, databases with annotated and curated data of many modalities, atlases of human and rodent brains, image processing workflows, supercomputing resources, neuromorphic systems, and virtual robots. EBRAINS was developed by the Human Brain Project, a research initiative funded by the European Commission with the mission to decode the human brain (Amunts et al., 2019, 2016). TVB cloud services (Tables 1, 2) were developed by the Human Brain Project subproject "The Virtual Brain" in collaboration with the two Human Brain Project partnering projects TVB-Cloud (virtualbraincloud-2020.eu) and TVB-CD (bit.ly/3ogLYtb). To provide supercomputing resources, the Human Brain Project offers as part of the Interactive Computing E-Infrastructure project access to compute and storage resources of the Fenix infrastructure (fenix-ri.eu), a network of six European supercomputing centres.

TVB cloud services are interlinked and make use of EBRAINS cloud services (Fig. 1), which we briefly introduce in the following before focussing on the TVB services. Please see Table 3 for a glossary of technical terms and abbreviations. The 'Collaboratory' (Supplementary Note: The EBRAINS Collaboratory) provides online workspaces, called 'collabs', where research teams can exchange data and work together on documents, secured with access control to restrict usage to authorized users. 'Lab' provides JupyterLab instances for developing applications and running code in a protected environment that cannot be accessed by other users. Jupyter notebooks provide a programmatic interface to EBRAINS services, allow to execute live code and to link processing steps with visualized results and documentation. Data can be found and accessed via the 'KnowledgeGraph', which provides a graphical user interface (GUI) and Application Programming Interface for searching, populating, and editing the data base. The KnowledgeGraph uses controlled vocabularies and ontologies that are mapped with existing neuroimaging and brain simulation ontologies to store data in a structured format, which enables to search the EBRAINS platform for data sets and to identify related information (Supplementary Methods: Data integration and TVB-ready data). In addition, EBRAINS offers services for professional curation of data sets including minting of persistent identifiers like Digital Object Identifiers (DOI; doi.org), licensing, versioning, and setting up of data sharing agreements. RESTful APIs are used for connecting different cloud components, as well as for authentication, data

transfer and control of supercomputers. Atlases provide common spatial reference spaces including a multilevel atlas of the human brain as well as the Waxholm Space rat brain atlas (Osen et al., 2019; Papp et al., 2014). The Multilevel Human Brain Atlas uses the Julich-Brain probabilistic cytoarchitectonic maps (Amunts et al., 2020) to link with template spaces such as BigBrain (Amunts et al., 2013) at the micrometre scale and MNI (Das et al., 2016) at millimetre scale, and combines them with imaging-based maps of function (Evans et al., 2012) and connectivity (Guevara et al., 2017). Linking a growing set of multimodal features, the Human Brain Atlas captures brain organization in its different facets.

What are the benefits of a cloud-based research platform? One important advantage are on-demand scalable computing resources. Neuroimaging and brain modelling workflows that are used to analyse large data sets (like the UK Biobank or the Human Connectome Project data sets) require processing power and storage beyond what personal computers can offer. On EBRAINS a network of powerful supercomputers enables to scale computing resources to the needs of a project. Another key advantage of cloud-based research is the ability for interoperable and reusable sharing of data and software, which is an urgent need as there is typically not one individual researcher doing all the work from data acquisition, analysis, hypotheses generation, model building, validation, up to writing and publishing. Rather, it is getting increasingly common that multiple teams, with team members being potentially scattered all around the planet, work together in large projects that require ongoing interaction and synchronization of data and code. Instead of frequently transmitting data sets via the internet and maintaining intricate software environments at multiple computing sites it is more efficient and practical to have a shared platform where teams can work together on datasets and run software in a common computing space. Problematically, sharing of and collaborative work on personal data raises privacy concerns: highly personal and detailed health data like MRI can be misused for malicious intents and must therefore be thoroughly protected, which is reflected in legislation like the General Data Protection Regulation (GDPR) of the European Union. With TVB on EBRAINS we created a software environment that globally implements state-of-the-art security mechanisms like encryption, access control and sandboxing to protect personal data, while at the same time workflows can be flexibly and reproducibly modified using containerized applications. These globally implemented measures for data protection make it easier for individual researchers to protect confidential data and to comply with the law. An additional benefit of TVB on EBRAINS workflows is that mechanisms for data management, provenance tracking and reproducible research are directly embedded using DataLad (Halchenko et al., 2021), which enables explicit tracking of all inputs, codes and processing steps that produced a result in a manner similar to how GitHub (github.com) is used for source code management. Having reproducibility already "built-in" makes it not only easier for the scientist to understand and re-use their own complex workflows years later. More importantly, it makes it also easier for everyone else to understand and use a complex workflow or just individual steps thereof. With simple commands a reviewer, a student, or another researcher can start the entire process or just individual steps and verify the consistency and correctness of the research, or use

Table 1
TVB cloud software, source codes and URLs leading to their main entry points.

Service	URLs
The Virtual Brain <i>Brain network simulation</i>	thevirtualbrain.apps.hbp.eu <i>Web-App</i>
	wiki.ebrains.eu/bin/view/Collabs/the-virtual-brain <i>Collab</i>
	wiki.ebrains.eu/bin/view/Collabs/user-story-tvb <i>End-to-end use case</i>
	github.com/the-virtual-brain/tvb-root <i>Source code</i>
	pypi.org/project/tvb-library <i>Python library</i>
	hub.docker.com/r/thevirtualbrain/tvb-run <i>Container image</i>
	zenodo.org/record/4,263,723#.YYRPg1BzXg <i>Demo brain network model data</i>
	thevirtualbrain.apps.hbp.eu <i>Web-App</i>
	wiki.ebrains.eu/bin/view/Collabs/tvb-pipeline <i>Collab</i>
	github.com/BrainModes/tvb-pipeline <i>Source code</i>
TVB Image Processing Pipeline <i>Connectome analysis</i>	hub.docker.com/r/thevirtualbrain/tvb_converter <i>Container images</i>
	thevirtualbrain.apps.hbp.eu <i>Web-App (TVB-Multiscale)</i>
	wiki.ebrains.eu/bin/view/Collabs/the-virtual-brain-multiscale <i>Collab (TVB-Multiscale)</i>
	wiki.ebrains.eu/bin/view/Collabs/co-simulation-tvb-and-nest-high-computer <i>Collab (Parallel CoSimulation)</i>
	github.com/the-virtual-brain/tvb-multiscale <i>Source code (TVB-Multiscale)</i>
Multiscale Co-Simulation <i>Two toolboxes for concurrent simulation of large-scale and spiking networks</i>	github.com/multiscale-cosim/TVB-NEST <i>Source code (Parallel CoSimulation)</i>
	hub.docker.com/r/thevirtualbrain/tvb-nest <i>Container image (TVB-Multiscale)</i>
	thevirtualbrain.apps.hbp.eu <i>Web-App (TVB-Multiscale)</i>
	wiki.ebrains.eu/bin/view/Collabs/the-virtual-brain-multiscale <i>Collab (TVB-Multiscale)</i>
	wiki.ebrains.eu/bin/view/Collabs/co-simulation-tvb-and-nest-high-computer <i>Collab (Parallel CoSimulation)</i>
	github.com/the-virtual-brain/tvb-multiscale <i>Source code (TVB-Multiscale)</i>
	github.com/multiscale-cosim/TVB-NEST <i>Source code (Parallel CoSimulation)</i>
	hub.docker.com/r/thevirtualbrain/tvb-nest <i>Container image (TVB-Multiscale)</i>
	wiki.ebrains.eu/bin/view/Collabs/ratempl-tvb/ <i>Collab</i>
	github.com/the-virtual-brain/tvb-root <i>Source code</i>
TVB-HPC <i>Automatic code generation</i>	github.com/the-virtual-brain/tvb-root <i>Collab</i>
	wiki.ebrains.eu/bin/view/Collabs/fast-tvb <i>Source code</i>
	github.com/BrainModes/fast_tvb <i>Container image</i>
Fast_TV <i>Parallelized simulation (multithreading)</i>	hub.docker.com/r/thevirtualbrain/fast_tvb <i>Collab</i>
	wiki.ebrains.eu/bin/view/Collabs/bayesian-virtual-epileptic-patient <i>Source code</i>
	github.com/ins-amu/BVEP <i>Collab</i>
Bayesian Virtual Epileptic Patient <i>Epilepsy modelling</i>	wiki.ebrains.eu/bin/view/Collabs/tvb-mouse-brains <i>Collabs</i>
	wiki.ebrains.eu/bin/view/Collabs/mouse-stroke-brain-network-model/ <i>Collab</i>
TVB Mouse Brains <i>Mouse brain simulation</i>	kg.ebrains.eu/search/instances/Dataset/a696cc7-e742-4301-8b43-d6814f3e5a44 <i>Collab</i>
	wiki.ebrains.eu/bin/view/Collabs/openminds-metadata-for-tvb-ready-data <i>Collab</i>
	github.com/HumanBrainProject/openMINDS <i>openMINDS schema</i>
TVB-ready dataset <i>SC, FC, and fMRI from tumour patients and controls</i> <i>openMINDS metadata for TVB-ready data</i> <i>Metadata in JSON-LD format</i>	wiki.ebrains.eu/bin/view/Collabs/sga3-d1-1-showcase-1 <i>Collab</i>
	github.com/FZJ-INM1-BDA/siibra-python ; github.com/FZJ-INM1-BDA/siibra-api <i>Source code</i>
	github.com/FZJ-INM1-BDA/siibra-python ; github.com/FZJ-INM1-BDA/siibra-api <i>Visualizer</i>
TVB atlas adapter <i>Brain atlas</i>	brainsimulation.org/atlasweb_multiscale
	training.incf.org/collection/virtual-brain-simulation-platform

and adapt it for another problem, without necessarily needing domain knowledge about the used software, which helps to make workflows and results more robust and easier to review and reproduce.

In the following we guide readers through the main components of TVB on EBRAINS, highlighting their main features and the respective advantages of cloud-based operation. Subsequently, we demonstrate an

end-to-end use case example including the implemented mechanisms for reproducibility and provenance tracking (please see additional use cases in the Supplementary Material). We conclude the main part with a description of data protection mechanisms, the TVB on EBRAINS shared responsibility model, and a discussion. Technical details about the services and their deployments can be found in the Methods section and

Table 2
Publications using software, workflows or data sets underlying different TVB cloud software.

Cloud service	Publications
The Virtual Brain	(Ritter et al., 2013; Sanz-Leon et al., 2013, 2015)
TVB Image Processing Pipeline	(Proix et al., 2016; Schirner et al., 2015a)
Fast_TV	(Costa-Klein et al., 2020; Schirner et al., 2018; Shen et al., 2019; Zimmermann et al., 2018)
Bayesian Virtual Epileptic Patient	(Hashemi et al., 2020; Jirsa et al., 2017)
TVB Mouse Brain	(Melozzi et al., 2019, 2017)
TVB ready datasets	(Aerts et al., 2020, 2018)
INCF TVB training space	(Matzke et al., 2015)

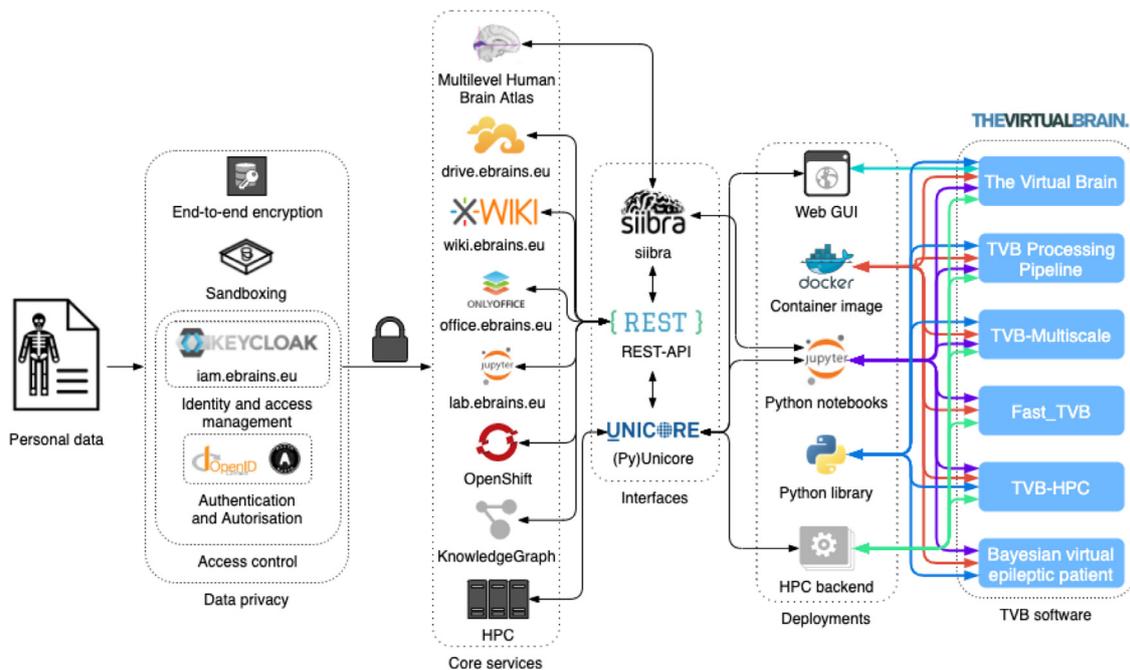


Fig. 1. TVB on EBRAINS cloud services. Human brain network modelling and neuroimaging require personal data applicable to data protection regulation. Encryption, sandboxing, and access control are used to protect personal data. EBRAINS provides core cloud services: the 'Multilevel Human Brain Atlas' provides maps of structure, function, and connectivity in multiple reference spaces; 'Drive' for storing and sharing files; 'Wiki' and 'Office' to create workspaces and documents for collaborative research; 'Lab' for running live code in sandboxed JupyterLab instances; 'OpenShift' for service and resource management; 'HPC' are supercomputers for resource-intensive computations. All software components interact via RESTful APIs and use UNICORE for communication with supercomputers. Software components exist in the form of web GUIs, container images, Python notebooks, Python libraries and high-performance machine codes. Curated scientific results, input and output data can be loaded from and stored into the EBRAINS KnowledgeGraph using openMINDS-compliant metadata annotations to enable efficient and robust sharing and reproducible re-use. The connectors show interactions between different components (colours group connectors according to different forms of software implementation).

exhaustive online documentation (Table 1). Supplementary material provides further information on the different components of TVB on EBRAINS.

2. Results

2.1. The Virtual Brain

TVB (thevirtualbrain.org) is an open-source software for simulating and analysing brain network models, which describe the brain as a graph that is composed of nodes that represent brain areas and edges that represent physical connections between these areas (Supplementary Note: Brain simulation with TVB) (Ritter et al., 2013; Sanz-Leon et al., 2013). TVB can be directly used on EBRAINS from a web GUI (Table 1), without the need to install further software or to have a specific operating system, computing environment or hardware. In addition, TVB can also be used as a Python library for programming in the EBRAINS Lab (Fig. 1). Via these interfaces users can upload brain network models, configure, and run simulations, as well as postprocess and export results. TVB usage is introduced through Jupyter notebooks, explanatory videos, and

technical documentation (Table 1). TVB's main documentation is hosted at docs.thevirtualbrain.org.

Importantly, TVB interfaces with supercomputers to rapidly perform simulations that require extensive processing time and storage space. For example, parameter space explorations with hundreds of parameter sets can be simulated in parallel. The web GUI simplifies the process of running high-performance simulations as no further knowledge about supercomputer usage is required: the entire process of sending encrypted data to a supercomputer, decrypting, sandboxed processing, encrypting of results and transmission to the web GUI is handled by the software automatically without any intervention by the user.

2.2. TVB Image Processing Pipeline

Brain network modelling requires a description of the anatomical network that connects brain areas, called structural connectivity, which can be estimated from diffusion-weighted MRI data using the TVB Image Processing Pipeline. The pipeline takes anatomical, functional and diffusion MRI as input and provides as output structural connectivity, region-average functional MRI time series, functional connectivity, brain sur-

Table 3
Glossary of technical terms and abbreviations.

Notation	Description
access control (computer security)	selective restriction to consume, enter or use a resource
annotation of data	categorization and labelling of data
authentication (computer security)	verifying the identity of a computer system user
authorisation (computer security)	specifying access rights and privileges to resources; access control rules are used to decide whether access requests from (authenticated) users shall be granted or not
API; application programming interface	interface that connects computers or software
BIDS	Brain Imaging Data Structure; a standard for organizing neuroscience data
brain network model	system of coupled differential equations for simulating brain activity
checksum	a small block of data that contains information about the contents of another block of data for the purpose of detecting errors
cloud computing	on-demand availability of computing power and storage over the internet
cloud service	infrastructure, platforms, or software made available through the internet
container image, containerization (software)	(creating) executable packages of software that include all dependencies needed to run an application reliably in different computing environments
controlled vocabulary	carefully selected list of words and phrases for unambiguous tagging of units of information
curation	organization and integration of data collected from various sources
data sharing agreement	legal contracts that detail what data are being shared and the appropriate use for the data
differential equation	equation that relates functions and their derivatives (rate at which the value of a function changes with respect to a change of its argument)
EBRAINS	European Brain Research InfrastructureS
encryption	converting information into secret code that hides the information's true meaning
functional connectivity	statistical relationships between brain signals represented as a network; often a matrix of pairwise correlation coefficients between region-average fMRI signals
General Data Protection Regulation	a regulation in European Union law on data protection and privacy with the aim to increase individual's control and rights over their personal data
GUI	graphical user interface
Jupyter notebooks	open-source web application to create and share documents that contain live code, equations, visualizations and narrative text
JupyterLab	web-based interactive development environment for Jupyter notebooks
key (computer security)	a piece of information, which, when processed through a cryptographic algorithm, can encode or decode cryptographic data
knowledge graph	a data model and database for linking, integrating, and storing information in a graph structure
licensing (software)	providing a software product with a legal statement (license) that governs its use and redistribution
metadata	data that provides information (annotations) about other data
metadata schema	a definition how metadata is structured
MRI	magnetic resonance imaging
neuromorphic systems	electronic analogue circuits to mimic neuro-biological architectures
ontology (information science)	a way to organize data, information, knowledge by defining concepts, categories and their relationships
openMINDS	specifications for structuring metadata in neuroscience (github.com/HumanBrainProject/openMINDS)
persistent identifiers	a long-lasting reference to an (often digital) object (e.g., document, file, web page); one example are digital object identifiers (DOI, doi.org), which are widely used to identify publications and data sets
public-key cryptography	a system that uses a different key for decryption than for encryption; this has the advantage that the decryption key needs not to be communicated via insecure channels, while the key for encryption can be known by everyone ("public") without compromising safety
RESTful API	an architectural style for APIs where resources are provided in a textual representation that can be read and modified with a predefined set of operations
sandbox (computer security)	security mechanism for separating running programs in an effort to protect computing systems from failure or attacks, often used to run untrusted programs and code
structural connectivity	aggregated descriptions of the networks that couple neurons, neural populations and brain areas
supercomputer	a computer that is shared by many users and that provides a high level of performance regarding processor time, memory and storage space
TVB	The Virtual Brain, a software to simulate brain network models
UNICORE	interface for exchanging data and commands between different computers in a network (unicore.eu)
versioning (software)	assigning unique version names or unique version numbers to unique states of computer software
version control	tracking and managing changes to software code or data sets
virtual robots	computer simulation of a physical robot

face triangulations, projection matrices for predicting EEG, and brain parcellations. The outputs can be directly uploaded to TVB for brain simulation and analysis. Users can configure and control pipeline steps from the TVB web GUI (Table 1), without needing to directly operate a supercomputer. A workflow orchestrator coordinates the execution of the pipeline and deals with privacy and reproducibility aspects. GUI and orchestrator ensure that the highly personal human brain data can only be accessed by authorized users, that they are always encrypted while at rest or in transit, and that they are only decrypted and processed inside a sandbox that is inaccessible by users of the cloud environment. In addition, the pipeline orchestrator supports provenance tracking and actionable reproducibility: the entire code, data, and all computational steps necessary to reproduce results starting from the raw data can be stored and re-run with a small set of simple commands on a

chosen level of granularity, which enables easy reproduction of research results. The pipeline supports flexible processing workflows as it consists of a sequence of container images that can be adapted, exchanged, added, or removed. Containerization makes the pipeline more platform-independent: it can be executed on all similar hardware platforms that support container runtimes like Docker or Singularity. Accordingly, the pipeline serves as a prototypical example for general-purpose protected and reproducible cloud workflows.

2.3. Multiscale Co-Simulation

Multiscale Co-Simulation are two new Python toolboxes for simulating large-scale brain networks with TVB that interact with spiking networks in NEST (Gewaltig and Diesmann, 2007). The toolboxes

provide interfaces to couple the two simulators by connecting the programmatic Python interface of TVB (Sanz-Leon et al., 2013) with PyNEST (Eppler et al., 2009), a Python wrapper for NEST. Multiscale Co-Simulation can be downloaded as standalone container image or used on EBRAINS from Jupyter notebooks (Table 1).

The need for a high-performance environment is for multiscale co-simulations even more important than for single-scale simulations: instead of one resource-demanding simulator there are two and they need to be executed in parallel. Critically, the two simulators need to synchronize to exchange their respective inputs, which is costly because the latency of network interaction is often orders of magnitude higher than the time needed to compute these inputs. To address the involved bottlenecks, the toolboxes implement routines that optimize communication and parallel execution. The Multiscale Co-Simulation project is under ongoing development currently focussing on postulating and validating coupling scenarios between the scales, optimizing the user interfaces as well as optimizing performance. See Supplementary Methods: Multiscale Co-Simulation for more information.

2.4. High-Performance implementations of TVB

Large software products like TVB are often designed with the goal to ease maintainability and long-term development, but that often comes at the cost of non-optimal execution speeds and resource consumption. Algorithms that are not optimized for speed can be orders of magnitude slower than optimized versions: instead of taking days or weeks, a simulation can be done in mere minutes, depending on how it is implemented. Problematically, optimizing computer code for speed is challenging and a task that is largely independent from scientific tasks like postulating and validating a new model: researchers must be put in a position where they can easily manipulate a given model in order to rapidly test hypotheses. To make it easier to simulate high-performance codes, two different strategies were realized. The first one, TVB-HPC (Table 1), automatically produces high-performance codes for CPUs and GPUs using an easy XML-based language called RateML for model specification. RateML is based on the domain-independent language 'LEMS' (Vella et al., 2014), which allows for the declarative description of computational models using a simple XML syntax. The already existing example implementations can be easily adapted to test different models, without requiring any knowledge about algorithmic optimization. The second one, Fast_TVB (Table 1), is a specialized high-performance implementation of the "Reduced Wong Wang" model (Deco et al., 2014; Sanz-Leon et al., 2015). Written in C it makes use of several optimization strategies and a sparse memory layout to efficiently use CPU resources, which makes it possible to simulate extremely large models with millions of nodes even on a standard computer in a reasonable time. Further information and benchmarks are provided in Supplementary Methods: High-performance implementations.

2.5. TVB atlas and data adapters

TVB on EBRAINS provides interfaces for interoperability with different components and services offered on EBRAINS, which enables researchers to plug in different analysis and modelling tools into their custom workflows. While the different TVB components are already interoperable by design, there is a need for 'adapters' that enable to interconnect with other EBRAINS services like the siibra toolbox, which connects TVB with the Human Brain Atlas (Table 1). The Human Brain Atlas characterizes brain regions with a growing set of multimodal features, including transmitter receptor densities (Palomero-Gallagher and Zilles, 2019), cell distributions, and physiological recordings, based on the Julich-Brain cytoarchitectonic maps (Amunts et al., 2020). Aligned with standard brain templates, the Human Atlas can be registered with individual brains to export multimodal microstructural "fingerprints" that can be used to set the parameters of brain models. The siibra adapter

gives direct programmatic access to EBRAINS atlas services like selecting a parcellation, browsing and searching brain region hierarchies, and obtaining maps of atlas features like the distributions of cell densities, neurotransmitters, or gene expression data. Internally, siibra connects with repositories like the EBRAINS KnowledgeGraph or the Allen Brain Atlas to retrieve the requested data, hiding the complexity of interacting with different services and minimizing common risks like misinterpretation of coordinates from different reference spaces. Complementary to siibra a viewer was implemented to visualize different atlas maps on the cortical surface (Table 1).

Additional adapters are under development that connect TVB with the Knowledge Graph and the Human Intracerebral EEG Platform to inform brain network model parameterization and to compare simulation results with empirical data. For example, it is planned to link intracranial electrophysiology recordings with the respective Julich-Brain regions to set model parameters based on direct measurements of effective connectivity and transmission delays from stimulation experiments (Trebaul et al., 2018). See Supplementary Methods: TVB atlas and data adapters for more information.

2.6. Data integration and TVB-ready data

Another advantage of cloud-based operation is that research results from different groups can be directly integrated into a central data record where they can be found and re-used by others. This functionality is provided by the EBRAINS KnowledgeGraph, an ontology-based graph data base where data sets are richly annotated with openMINDS metadata in order to ensure their interpretability in the future (Table 1). The openMINDS metadata annotations define an exact classification of research inputs and outputs (for example, empirical recordings, software, articles, books, imaging coordinate systems, reference atlases, models, projects) against a scientific ontology or knowledge framework. To ensure data quality EBRAINS employs a team of expert curators who assist in creating and verifying that data format and metadata annotations fulfil state of the art practices for provenance tracking and data management with regard to long-term availability and interpretability of the results. Data in the KnowledgeGraph is protected by the 'Human Data Gateway', which controls access to human datasets through regulatory compliant data use agreements and access policies. A first example of modelling results that were integrated into the KnowledgeGraph are TVB-ready connectivity data sets in BIDS format from tumour patients and matched control participants. The data set contains region-average fMRI time series, FC, and SC from 31 brain tumour patients before and after surgery, and 11 healthy controls (Aerts et al., 2019). See Supplementary Methods: Data integration and TVB-ready data for more information.

2.7. End-to-end use case with reproducible brain model construction

Upon introducing the individual components of TVB on EBRAINS we now exemplify how they may be combined. Additional use cases are described in the Supplementary Material, especially in the section 'Advanced use cases and training'. To get acquainted with TVB one may start by performing a few test simulations with TVB's default structural connectivity to learn usage of the web GUI and the Python interface; documentation and tutorials explain the steps (Table 1). Visualizing the outputs for different parameter settings and fitting simulation results with empirical data (for example, using functional connectivity) helps to create an intuitive understanding of brain network model dynamics. Next, researchers may want to perform a more detailed analysis, for example, comparing individuals in patient versus control groups to study mechanisms of pathological versus healthy brain dynamics. Here, the researchers can use the TVB Image Processing Pipeline to compute individual structural and functional connectivity from human MRI data. Estimating connectomes from MRI data consists of many complex steps,

making it hard to explicitly track all the necessary provenance data to robustly reproduce a particular configuration of processing steps. Just a minor update of a dependency or an untracked renaming of a file can break the entire workflow and make a result not reproducible. The pipeline uses DataLad (datalad.org) to make its workflow reproducible in an actionable manner: all software and data are tracked in a way that the entire workflow or just individual steps can be easily re-run, archived, published and shared. With DataLad all data and code files are version-controlled and managed in a manner that is comparable to how software is managed with GitHub (github.com), allowing to capture complex hierarchical project structures and all computational steps from raw data to final figures.

When large cohorts are modelled users may find the speed of standard brain model implementations insufficient and switch to TVB's high-performance implementations, which allow fast execution and easy generation of high-performance codes for custom models with TVB's XML-based modelling language RateML. To inform model parameters researchers may decide to include microstructural information from the EBRAINS Human Brain Atlas using the siibra interface (Wang, 2020). Or they may extend large-scale models to encompass finer scales using TVB Multiscale to study hypotheses about brain function that span spatial scales from individual point neurons over populations to whole brain models. In a recent preprint this novel approach was used to study the effect of deep brain stimulation on a spiking basal ganglia model (Meier et al., 2021). Finally, the resulting data outputs can be annotated with metadata, curated, and integrated into the KnowledgeGraph for future reuse by the community.

2.8. Advanced use cases and training

In addition to the introductory use cases described above, EBRAINS provides tutorials for several advanced use cases (Table 1). The Bayesian Virtual Epileptic Patient tutorials showcase how Bayesian inference can be used to compute posterior probability distributions for region-wise parameter settings of TVB's Epileptor model in order to study the spread of epileptic seizures (Jirsa et al., 2017, 2014). The approach makes use of prior distributions obtained from empirical data (for example, a patient's structural connectivity, or lesions detected in MRI) and model simulations to take into account the likelihood for these observations. For example, estimating excitability parameters of an Epileptor brain network model yields a map of region-wise epileptogenicity to guide clinical decision-making. The Virtual Mouse Brain extends TVB with tractography-based as well as tracer-based mouse SC (Melozzi et al., 2017), which was estimated from the Allen Mouse Brain Connectivity Atlas (Oh et al., 2014). Tutorials demonstrate how to export mouse connectivity at different resolutions and how to simulate strokes in mice (Allegra Mascaro et al., 2020). In addition to these notebook tutorials the INCF (International Neuroinformatics Coordination Facility) training space holds a dedicated collection for TVB with didactic use cases, video tutorials, Jupyter notebooks and example data sets (Table 1). See Supplementary Methods: Advanced use cases and training for more information.

2.9. What can go wrong? Common pitfalls of brain network modelling.

Although cloud services make it easier to run scalable modelling workflows there are several limitations to consider. Already one of the first steps, creating a brain network model from MRI data, involves several caveats. One major limitation of MRI tractography is that coupling strengths and time delays of nerve fibre tracts cannot be directly measured (Sotiropoulos and Zalesky, 2019). Identifying and quantifying fibre tracts is based on a mapping from water diffusion to fibre orientations, which is in general an ill-posed problem as MRI voxels are too large to resolve individual fibers. Neither the orientation of fibers in a voxel can be resolved, nor can different arrangements like bending,

fanning, crossing or kissing be distinguished. As a result, tractography provides only a model-based approximation of interregional coupling strengths and time delays. Problematically, these approximations are biased by factors like the distance of the regions, algorithmic choices, and individual anatomical properties (Jeurissen et al., 2019; Yeh et al., 2021). Furthermore, even if fibers could be reliably counted, there are several microstructural properties known to influence the strength of coupling that also cannot be directly measured like myelination, axon diameter and synaptic properties, which implicates that tractography results must be interpreted with caution (Jeurissen et al., 2019; Yeh et al., 2021). A related problem is node delineation and the question what is a meaningful parcellation of the brain to form the nodes of a network model? Unlike the microscale, where the mapping between nodes and neurons is obvious, defining nodes at the macroscale is less clear. An intuitive criterion would be functional homogeneity: voxels get grouped based on how similar their activity is, which is plausible, because one model node is usually governed by one type of dynamics. However, matters are complicated by individual structure-function variability. For example, the size of a well-characterized area like V1 can vary twofold in size across subjects (Amunts et al., 2000; Van Essen, 2013), which would be missed by group-level parcellations. Similarly, the scale and the number of nodes heavily impacts the resulting model and they must therefore be aligned with the goals of the research (Proix et al., 2016). For example, the parcellation must be fine enough to be able to represent and differentiate between the specific features of the system that are related to the aims of the research.

Probably one of the biggest challenges is to identify whether a given model can or cannot reproduce a set of observations, which is done in a process called 'inference' that works by comparing modelling outputs with the actual data and selecting the model that explains the observed phenomenon in a way that is deemed optimal. Problematically, already the related task of finding optimal parameter values for a given set of model equations suffers from the so-called 'curse of dimensionality': with each added dimension the space of possible model parameterizations increases exponentially (there is a combinatorial explosion in the possible values that the parameters can jointly take), making it harder to find models that generalize to the typically high-dimensional real-world scenarios in digital medicine (Berisha et al., 2021). Complex mechanistic models are poorly suited for inference, because computing the likelihood for a given observation is typically intractable (Cranmer et al., 2020), as this would require integrating over all potential outcomes of a simulation, the number of which increases exponentially with each model dimension. Likewise, complex systems are often degenerate, producing indistinguishable observations by infinitely many realizations of the same process. While new approaches for "likelihood-free" simulation-based inference are under development (Cranmer et al., 2020), in practical cases often recourse is made to traditional approaches like relying on the insights of scientists into the system to construct powerful summary statistics to effectively compare observed with simulated data. A related problem, especially regarding clinical application, is that models always involve (per definition) enormous simplifications and are often based on assumptions that are only weakly justified and might be very restrictive. Consequently, the conclusions that can be drawn are a function of the validity of the knowledge that was used to build the model and the efficiency with which the verbal knowledge was translated into mathematical equations and then into computer code. Especially in clinical applications false expectations, misinterpretations and overconfidence in simulated results can lead to significant real-life problems. Consequently, these workflows may not be used in a "turn-key" manner and with the expectation that they will automatically produce meaningful results. To produce meaningful results and to adequately interpret them knowledge about modelling and numerical methods as well as neuroscience domain knowledge are fundamentally necessary.

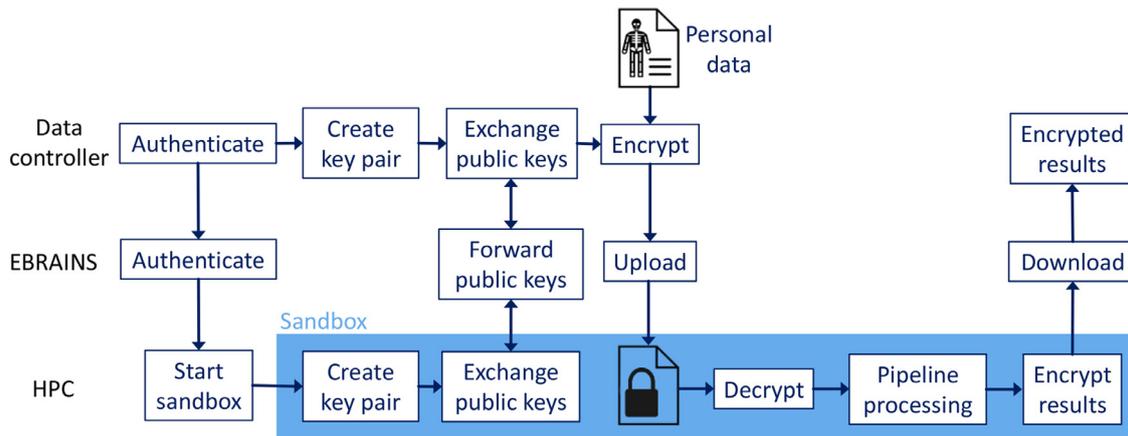


Fig. 2. Securing personal data processing workflows in shared environments. Access control ensures that only authorized users can access sensitive data. Sensitive data is encrypted with public-key cryptography on the data controller's computer before upload to the cloud. The key pair for upload is generated within a sandboxed process at the final processing site and the private key never leaves the sandbox. This ensures that the data can only be decrypted at the final processing site and personal data is never written outside the sandbox in unencrypted form. A public key generated by the data controller is used for returning encrypted results, which ensures that only the data controller can decrypt the data.

2.10. Data protection in the TVB on EBRAINS cloud

Biomedical research is facing challenges because many methods lack technical infrastructure to protect the privacy of sensitive data. Research often involves that teams exchange and process sensitive data on shared infrastructure like the internet and high-performance computers, which poses risks for illegitimate access. Consequently, an important requirement for privacy protection is to enable secure processing of sensitive data in shared infrastructures, as the involved networks and computers can be accessed by many human and non-human users with only logical separation between them. Cloud platforms have the advantage that privacy technology and legal compliance measures can be globally implemented and offered as a standardized and certified service, which makes it easier for the individual researchers to overcome technical and organizational hurdles for demonstrating compliance with data protection law. The European Union's General Data Protection Regulation (GDPR) and similar international and national laws impose restrictions on the processing of personal data including storage and sharing. Problematically, biomedical data cannot be easily anonymized or pseudonymized such that all potentially identifiable information are removed, and potential re-identification is excluded (Byrge and Kennedy, 2018; Gymrek et al., 2013; Rocher et al., 2019). A principle means of ensuring GDPR-compliant data processing is the implementation of technical and organizational measures to ensure a level of security appropriate to the risk of the processing (Article 32 GDPR). To protect data by design and default (Article 25 GDPR), TVB on EBRAINS implements access control, public-key cryptography, and sandboxing (Fig. 2).

Access control mechanisms, like the TVB web GUI, hide direct access to systems where sensitive data are actively processed: users need to log into the GUI with their password and can only access data that they uploaded or created themselves or that was made available to them through the role-based access control and permission management functionalities of the EBRAINS Collaboratory (see Supplementary Note: The EBRAINS Collaboratory). Sensitive data is encrypted before upload to EBRAINS and remains encrypted at all times with the only exception being the time when a processing job is actively executed. Cryptographic keys are created ad-hoc and independently for each processing job and the system is designed such that no human gets into possession of the decryption key while the data is in the cloud: the sensitive data can only be decrypted at their final processing site by an automatic procedure. During the actual processing sensitive data may exist in unencrypted

form, but only within isolated temporary memory locations that cannot be accessed by other users of the system (sandboxes). See Supplementary Methods: Data protection in the TVB on EBRAINS cloud for more information.

2.11. Shared responsibility & compliance

In addition to technical measures also organizational aspects must be considered for processing to be lawful. The GDPR describes two roles for lawful processing of personal data: data controllers and data processors. Data controllers are responsible for, and required to be able to demonstrate, compliance with GDPR (Art. 5, GDPR), by implementing technical and organisational measures that ensure appropriate security of the personal data (Art. 24, GDPR). In contrast, data processors process personal data only on behalf of data controllers, acting under the authority of the controller to carry out the processing (Art. 28/29, GDPR). When a user uses TVB on EBRAINS services to process personal data the user is always the data controller, while EBRAINS as a service provider is always the data processor, because the user is directing the processing through its interaction with the offered services, while EBRAINS is only executing the provided instructions. As data processor EBRAINS is responsible for protecting the global infrastructure with documented procedures and services on behalf of the user. As data controller a user maintains control over the data that it hosts or processes with TVB on EBRAINS, as mechanisms were put in place to prevent unauthorized access and to enable that data controllers can independently or jointly determine the means of the data processing. To use TVB cloud services a user must therefore agree to terms that clarify its personal responsibility regarding compliance with GDPR with respect to security precautions, access permissions, contact persons, personal responsibilities, monitoring, logging, and passing of information to third parties (ebrains.eu/terms).

3. Discussion

TVB cloud services were developed to lower the barriers to brain simulation and connectome analysis. They offer reproducible and protected workflows for collaborative computational neuroscience research. All codes are open source and available for download from EBRAINS and GitHub (Table 1). Software is packaged in container images that can be directly used without the need to install dependencies. Several software and data components have been peer-reviewed, and results were

published in academic journals (Table 2). To enable actionable reproducibility the image processing workflow is equipped with tools for data management and provenance tracking. All computational steps, inputs and software are tracked, and each step can be easily rerun and verified with a simple set of commands. Technical and organisational measures for protecting the privacy of personal data are globally implemented into the services offerings of the platform, making it easier for researchers to demonstrate compliance with data protection regulation. Access control, encryption and sandboxing ensure that sensitive data stays confidential. Comprehensive documentation in the form of manuals, tutorials, lectures, Jupyter notebooks, demo data, workshops, videos, use cases, mailing lists and support contacts provide efficient and didactic dissemination of knowledge and support. EBRAINS core services enable to map and organize complex projects by large remote teams into a persistent and replicable structure at a central and secure place, which makes it easier to pick up projects at a later time. The flexibility of the platform and its focus on community-driven research enable rapid adoption of advances in brain simulation and connectomics, as well as correction of errors. Technical and organisational security mechanisms are designed to provide highest data protection standards, while at the same time providing the required flexibility to enable state-of-the-art research. To keep the high quality of the cloud services, ongoing and future efforts are directed towards the continuous integration of improved community standards and best practices. The TVB on EBRAINS ecosystem can be transferred to other cloud environments within the European Open Science Cloud or beyond. Thus, it serves as a reference architecture for secure processing and simulation of neuroscience data in the cloud (Fig. 1 and Supplementary Discussion).

4. Methods

4.1. The Virtual Brain

The methods behind the main TVB neuroinformatics simulator are extensively described in several publications (Ritter et al., 2013; Sanz-Leon et al., 2015, 2013) and in online documentation (Table 1; docs.thevirtualbrain.org). To deploy TVB as cloud service it was implemented as container image executed on OpenShift, an open source container orchestration platform. This deployment serves TVB's GUI via the web and automatically scales the number of running instances of the TVB container depending on demand. The GUI is connected with the EBRAINS identity and access management system to perform access control: only registered EBRAINS user can access the GUI and they can only access the data for which they were given role-based permission. Depending on their complexity, simulation jobs are either directly computed in the running OpenShift instance that serves the web GUI or on a supercomputer. Currently users still have the responsibility to manually encrypt their data with a public key before upload, but in a next release it is planned that this will be automatically performed by the upload function. After upload every project is individually re-encrypted with a dedicated key. Decryption only happens when a user opens a project in the web GUI and the decrypted data is immediately deleted when the project is closed or the user logs out. The decrypted project is not directly written to a file system, but only stored inside the running container. For high-demand operations that run on the supercomputer data is only decrypted after the job gets started by the job scheduler and only inside the running TVB container. See Supplementary Methods: Brain simulation with TVB for more information.

4.2. TVB Image Processing Pipeline

The TVB Image Processing Pipeline (Schirner et al., 2015b) allows users to select and combine dedicated neuroimaging workflow containers, like BIDS Apps (see Supplementary Note: BIDS Apps), into reproducible workflows that process MRI data on supercomputers while protecting the privacy of personal data in compliance with data protec-

tion regulation. Containerization makes it easier to deploy neuroimaging workflows, as they often rely on a high number of dependencies and computational steps. Users can select amongst different neuroimaging containers like fmriprep for functional MRI processing (Esteban et al., 2019), Mrtrix3_connectome for diffusion MRI tractography (Smith and Connelly, 2019; Tournier et al., 2019), or the Human Connectome Project pipelines for both (Glasser et al., 2013). Like main TVB, the pipeline execution on the supercomputer can be controlled from the TVB web GUI without giving users direct access to the supercomputer. An orchestrator program on the supercomputer coordinates the execution of the container images and ensures that personal data is encrypted at all times, except for the duration of the processing and then only in the main memory of a sandboxed process (Fig. 2 and Supplementary Note: TVB Image Processing Pipeline for more details). To make workflow processing reproducible the open source distributed data management solution DataLad (datalad.org; (Halchenko et al., 2021)) was used for version control and provenance tracking: all files involved in a workflow (such as data, code and computational environment) are stored within nested directory trees, which allows to explicitly store the evolution of a data set from its raw state to the final result. Checksums allow the user to uniquely identify the contents of every file, which in turn allows to verify the correct execution of every computational step and thereby full computational reproducibility of the entire workflow. See Supplementary Methods: TVB Image Processing Pipeline for more information.

Declaration of competing interests

The authors declare no competing interests.

Credit authorship contribution statement

Michael Schirner: Conceptualization, Data curation, Formal analysis, Methodology, Writing – original draft, Writing – review & editing, Software. **Lia Domide:** Resources, Software, Writing – review & editing. **Dionysios Perdikis:** Methodology, Resources, Software, Writing – original draft, Writing – review & editing. **Paul Triebkorn:** Formal analysis, Methodology, Software, Writing – review & editing. **Leon Stefanovski:** Data curation, Formal analysis, Writing – review & editing. **Roopa Pai:** Conceptualization, Data curation, Formal analysis, Writing – review & editing. **Paula Prodan:** Software, Writing – review & editing. **Bogdan Valean:** Software, Writing – review & editing. **Jessica Palmer:** Visualization, Software, Writing – review & editing. **Chloë Langford:** Visualization, Software, Writing – review & editing. **André Blickensdörfer:** Visualization, Software, Writing – review & editing. **Michiel van der Vlag:** Methodology, Software, Writing – original draft, Writing – review & editing. **Sandra Diaz-Pier:** Methodology, Software, Writing – review & editing. **Alexander Peyser:** Methodology, Software, Writing – review & editing. **Wouter Klijn:** Methodology, Software, Writing – review & editing. **Anne Nahm:** Resources, Writing – review & editing. **Oliver Schmid:** Data curation, Methodology, Software, Writing – review & editing. **Marmaduke Woodman:** Methodology, Software, Writing – original draft, Writing – review & editing. **Lyuba Zehl:** Data curation, Methodology, Software, Writing – review & editing. **Jan Fousek:** Methodology, Software, Writing – review & editing. **Spase Petkoski:** Methodology, Software, Writing – original draft, Writing – review & editing. **Lionel Kusch:** Methodology, Software, Writing – original draft, Writing – review & editing. **Meysam Hashemi:** Methodology, Software, Writing – review & editing. **Daniele Marinazzo:** Data curation, Formal analysis, Writing – original draft, Writing – review & editing. **Jean-François Mangin:** Methodology, Resources, Writing – original draft, Writing – review & editing. **Agnes Flöel:** Resources, Writing – review & editing. **Simisola Akintoye:** Supervision, Writing – review & editing. **Bernd Carsten Stahl:** Supervision, Writing – review & editing. **Michael Cepic:** Supervision, Writing – original draft, Writing – review & editing. **Emily Johnson:** Supervision, Writing – orig-

inal draft, Writing – review & editing. **Gustavo Deco**: Conceptualization, Software, Supervision, Writing – review & editing. **Anthony R. McIntosh**: Conceptualization, Supervision, Writing – review & editing. **Claus C. Hilgetag**: Conceptualization, Resources, Supervision, Writing – review & editing. **Marc Morgan**: Resources, Supervision, Writing – review & editing. **Bernd Schuller**: Methodology, Software, Writing – review & editing. **Alex Upton**: Resources, Writing – review & editing. **Colin McMurtrie**: Resources, Supervision, Writing – review & editing. **Timo Dickscheid**: Conceptualization, Methodology, Resources, Software, Supervision, Writing – review & editing. **Jan G. Bjaalie**: Conceptualization, Resources, Supervision, Writing – review & editing. **Katrin Amunts**: Conceptualization, Resources, Supervision, Writing – review & editing. **Jochen Mersmann**: Conceptualization, Methodology, Resources, Software, Supervision, Writing – review & editing. **Viktor Jirsa**: Conceptualization, Resources, Supervision, Writing – review & editing. **Petra Ritter**: Funding acquisition, Methodology, Project administration, Resources, Visualization, Supervision, Writing – original draft, Writing – review & editing.

Data availability

The datasets generated and analysed during the current study are available in the EBRAINS KnowledgeGraph repository, search.kg.ebrains.eu. KnowledgeGraph is a DOI-minting repository where EBRAINS data and software services are indexed and referenced. To share personal data amongst researchers subject to national or international data protection laws (e.g., the General Data Protection Regulation of the European Union) protected environments and workflows as well as a shared responsibility model were created (Supplementary Methods: Data protection in the TVB on EBRAINS cloud).

Code availability

All software codes presented in this article have open-source licenses and can be downloaded from GitHub (Table 1). The EBRAINS database service KnowledgeGraph (search.kg.ebrains.eu) is a DOI-minting repository where all data and software services are indexed and archived. Source codes are deployed as cloud services that can be used on ebrains.eu and as standalone download versions that can be pulled as container images from Docker Hub (Table 1).

Acknowledgments

We gratefully acknowledge the Swiss National Supercomputing center CSCS for supporting this project by providing computing time through the Interactive Computing E-Infrastructure (ICEI) on the Supercomputer Piz Daint of the Fenix Infrastructure (projects ich10, ich12) and the Gauss Centre for Supercomputing e.V. (www.gauss-centre.eu) for supporting this project by providing computing time through the John von Neumann Institute for Computing (NIC) on the GCS Supercomputer JUWELS at Jülich Supercomputing Centre (JSC). We acknowledge support by H2020 Research and Innovation Action grants Human Brain Project SGA2 785907, SGA3 945539, ICEI 800858, VirtualBrainCloud 826421 and ERC 683049; Berlin Institute of Health & Foundation Charité, Johanna Quandt Excellence Initiative. Several computations have also been performed on the HPC for Research cluster of the Berlin Institute of Health. We acknowledge the use of Fenix Infrastructure resources, which are partially funded from the European Union's Horizon 2020 research and innovation programme through the ICEI project under the grant agreement No. 800858. German Research Foundation SFB 1436 (project ID 425899996); SFB 1315 (project ID 327654276); SFB 936 (project ID 178316478); SFB-TRR 295 (project ID 424778381); SPP Computational Connectomics RI 2073/6–1, RI 2073/10–2, RI 2073/9–1.

Supplementary materials

Supplementary material associated with this article can be found, in the online version, at doi:[10.1016/j.neuroimage.2022.118973](https://doi.org/10.1016/j.neuroimage.2022.118973).

References

- Aarts, A.A., Anderson, J.E., Anderson, C.J., Attridge, P.R., Attwood, A., Axt, J., Babel, M., Bahník, Š., Baranski, E., Barnett-Cowan, M., Bartmess, E., Beer, J., Bell, R., Bentley, H., Beyan, L., Binion, G., Borsboom, D., Bosch, A., Bosco, F.A., Bowman, S.D., Brandt, M.J., Braswell, E., Brohmer, H., Brown, B.T., Brown, K., Brüning, J., Calhoun-Sauls, A., Callahan, S.P., Chagnon, E., Chandler, J., Chartier, C.R., Cheung, F., Christopherson, C.D., Gillessen, L., Clay, R., Cleary, H., Cloud, M.D., Conn, M., Cohoon, J., Columbus, S., Cordes, A., Costantini, G., Alvarez, L.D.C., Cremata, E., Crusius, J., DeCoster, J., DeGaetano, M.A., Penna, N.D., 2015. Estimating the reproducibility of psychological science. *Science* (80–) doi:[10.1126/science.aac4716](https://doi.org/10.1126/science.aac4716).
- Aerts, H., Schirner, M., Dhollander, T., Jeurissen, B., Achten, E., Van Roost, D., Ritter, P., Marinazzo, D., 2020. Modeling brain dynamics after tumor resection using The Virtual Brain. *Neuroimage* doi:[10.1016/j.neuroimage.2020.116738](https://doi.org/10.1016/j.neuroimage.2020.116738).
- Aerts, H., Schirner, M., Jeurissen, B., Van Roost, D., Achten, E., Ritter, P., Marinazzo, D., 2019. TVB time series and connectomes for personalized brain modeling in brain tumor patients [Data set] doi:[10.25493/1ECN-6SM](https://doi.org/10.25493/1ECN-6SM).
- Aerts, H., Schirner, M., Jeurissen, B., Van Roost, D., Achten, E., Ritter, P., Marinazzo, D., 2018. Modeling brain dynamics in brain tumor patients using The Virtual Brain. *eNeuro* doi:[10.1523/ENEURO.0083-18.2018](https://doi.org/10.1523/ENEURO.0083-18.2018).
- Allegra Mascaro, A.L., Falotico, E., Petkoski, S., Pasquini, M., Vannucci, L., Tort-Colet, N., Conti, E., Resta, F., Spalletti, C., Ramalingasetty, S.T., von Arnim, A., Formento, E., Angelidis, E., Blixhavn, C.H., Leergaard, T.B., Caleo, M., Destexhe, A., Ijspeert, A., Micera, S., Laschi, C., Jirsa, V.K., Gewaltig, M.O., Pavone, F.S., 2020. Experimental and computational study on motor control and recovery after stroke: toward a constructive loop between experimental and virtual embodied neuroscience. *Front. Syst. Neurosci.* 14, 1–34. doi:[10.3389/fnsys.2020.00031](https://doi.org/10.3389/fnsys.2020.00031).
- Amunts, K., Ebell, C., Muller, J., Telefont, M., Knoll, A., Lippert, T., 2016. The human brain project: creating a european research infrastructure to decode the human brain. *Neuron* doi:[10.1016/j.neuron.2016.10.046](https://doi.org/10.1016/j.neuron.2016.10.046).
- Amunts, K., Knoll, A.C., Lippert, T., Pennartz, C.M.A., Rylvlin, P., Destexhe, A., Jirsa, V.K., D'Angelo, E., Bjaalie, J.G., 2019. The human brain project—synergy between neuroscience, computing, informatics, and brain-inspired technologies. *PLoS Biol* doi:[10.1371/journal.pbio.3000344](https://doi.org/10.1371/journal.pbio.3000344).
- Amunts, K., Lepage, C., Borgeat, L., Mohlberg, H., Dickscheid, T., Rousseau, M.É., Bludau, S., Bazin, P.L., Lewis, L.B., Oros-Peusquens, A.M., Shah, N.J., Lippert, T., Zilles, K., Evans, A.C., 2013. BigBrain: an ultrahigh-resolution 3D human brain model. *Science* (80) doi:[10.1126/science.1235381](https://doi.org/10.1126/science.1235381).
- Amunts, K., Malikovic, A., Mohlberg, H., Schormann, T., Zilles, K., 2000. Brodmann's areas 17 and 18 brought into stereotaxic space—where and how variable? *Neuroimage* 11, 66–84.
- Amunts, K., Mohlberg, H., Bludau, S., Zilles, K., 2020. Julich-Brain: a 3D probabilistic atlas of the human brain's cytoarchitecture. *Science* (80).
- Berisha, V., Krantsevich, C., Hahn, P.R., Hahn, S., Dasarath, G., Turaga, P., Liss, J., 2021. Digital medicine and the curse of dimensionality. *NPJ digital medicine* 4 (1), 1–8.
- Byrge, L., Kennedy, D.P., 2018. High-accuracy individual identification using a “thin slice” of the functional connectome. *Netw. Neurosci.* doi:[10.1162/netn_a_00068](https://doi.org/10.1162/netn_a_00068).
- Costa-Klein, P., Ettinger, U., Schirner, M., Ritter, P., Falkai, P., Koutsouleris, N., Kambeitz, J., 2020. Investigating the Effect of the Neuregulin-1 Genotype on brain function using brain network simulations. *Biol. Psychiatry* doi:[10.1016/j.biopsych.2020.02.118](https://doi.org/10.1016/j.biopsych.2020.02.118).
- Cranmer, K., Brehmer, J., Louppe, G., 2020. The frontier of simulation-based inference. *Proc. Natl. Acad. Sci.* 117, 30055–30062.
- Das, S., Glatar, T., MacIntyre, L.C., Madjar, C., Rogers, C., Rousseau, M.E., Rioux, P., MacFarlane, D., Mohades, Z., Gnanasekaran, R., Makowski, C., Kostopoulos, P., Adalat, R., Khalili-Mahani, N., Niso, G., Moreau, J.T., Evans, A.C., 2016. The MNI data-sharing and processing ecosystem. *Neuroimage* doi:[10.1016/j.neuroimage.2015.08.076](https://doi.org/10.1016/j.neuroimage.2015.08.076).
- Deco, G., Ponce-Alvarez, A., Hagmann, P., Romani, G.L., Mantini, D., Corbetta, M., 2014. How local excitation-inhibition ratio impacts the whole brain dynamics. *J. Neurosci.* 34, 7886–7898.
- Eppler, J.M., Helias, M., Muller, E., Diesmann, M., Gewaltig, M.O., 2009. PyNEST: a convenient interface to the NEST simulator. *Front. Neuroinform.* doi:[10.3389/neuro.11.012.2008](https://doi.org/10.3389/neuro.11.012.2008).
- Esteban, O., Markiewicz, C.J., Blair, R.W., Moodie, C.A., Isik, A.I., Erramuzpe, A., Kent, J.D., Goncalves, M., DuPre, E., Snyder, M., Oya, H., Ghosh, S.S., Wright, J., Durnez, J., Poldrack, R.A., Gorgolewski, K.J., 2019. fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nat. Methods.* doi:[10.1038/s41592-018-0235-4](https://doi.org/10.1038/s41592-018-0235-4).
- Evans, A.C., Janke, A.L., Collins, D.L., Baillet, S., 2012. Brain templates and atlases. *Neuroimage* doi:[10.1016/j.neuroimage.2012.01.024](https://doi.org/10.1016/j.neuroimage.2012.01.024).
- Gewaltig, M.-O., Diesmann, M., 2007. NEST (NEural Simulation Tool). *Scholarpedia* doi:[10.4249/scholarpedia.1430](https://doi.org/10.4249/scholarpedia.1430).
- Glasser, M.F., Sotiropoulos, S.N., Wilson, J.A., Coalson, T.S., Fischl, B., Andersson, J.L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J.R., Van Essen, D.C., Jenkinson, M., 2013. The minimal preprocessing pipelines for the human connectome project. *Neuroimage* doi:[10.1016/j.neuroimage.2013.04.127](https://doi.org/10.1016/j.neuroimage.2013.04.127).
- Guevara, M., Román, C., Houenou, J., Duclap, D., Poupon, C., Mangin, J.F., Guevara, P., 2017. Reproducibility of superficial white matter tracts using diffusion-weighted imaging tractography. *Neuroimage* 147, 703–725.

- Gymrek, M., McGuire, A.L., Golan, D., Halperin, E., Erlich, Y., 2013. Identifying personal genomes by surname inference. *Science* (80–) doi:10.1126/science.1229566.
- Halchenko, Y.O., Meyer, K., Poldrack, B., Solanky, D.S., Wagner, A.S., Gors, J., MacFarlane, D., Pustina, D., Sochat, V., Ghosh, S.S., 2021. DataLad: distributed system for joint management of code, data, and their relationship. *J. Open Source Softw.* 6, 3262.
- Hashemi, M., Vattikonda, A.N., Sip, V., Guye, M., Bartolomei, F., Woodman, M.M., Jirsa, V.K., 2020. The bayesian virtual epileptic patient: a probabilistic framework designed to infer the spatial map of epileptogenicity in a personalized large-scale brain model of epilepsy spread. *Neuroimage* doi:10.1016/j.neuroimage.2020.116839.
- Ioannidis, J.P.A., 2005. Why most published research findings are false. *PLoS Med* 2, e124.
- Jeurissen, B., Descoteaux, M., Mori, S., Leemans, A., 2019. Diffusion MRI fiber tractography of the brain. *NMR Biomed* 32, e3785. doi:10.1002/nbm.3785.
- Jirsa, V.K., Proix, T., Perdiks, D., Woodman, M.M., Wang, H., Bernard, C., Bénar, C., Chauvel, P., Bartolomei, F., Guye, M., Gonzalez-Martinez, J., 2017. The virtual epileptic patient: individualized whole-brain models of epilepsy spread. *Neuroimage* doi:10.1016/j.neuroimage.2016.04.049.
- Jirsa, V.K., Stacey, W.C., Quilichini, P.P., Ivanov, A.I., Bernard, C., 2014. On the nature of seizure dynamics. *Brain* 137, 2210–2230. doi:10.1093/brain/awu133.
- Matzke, H., Schirner, M., Vollbrecht, D., Rothmeier, S., Larena, A., Rojas, R., Triebkorn, P., Domide, L., Mersmann, J., Solodkin, A., 2015. TVB-edupack—an interactive learning and scripting platform for The Virtual Brain. *Front. Neuroinform* 9.
- Meier, J.M., Perdiks, D., Blickensdörfer, A., Stefanovski, L., Liu, Q., Maith, O.Ü., Dinkelbach, H., Baladron, J., Hamker, F.H., Ritter, P., 2021. Virtual deep brain stimulation: multiscale co-simulation of a spiking basal ganglia model and a whole-brain mean-field model with The Virtual Brain. *bioRxiv* doi:10.1101/2021.05.05.442704.
- Melozzi, F., Bergmann, E., Harris, J.A., Kahn, I., Jirsa, V.K., Bernard, C., 2019. Individual structural features constrain the mouse functional connectome. *Proc. Natl. Acad. Sci. U. S. A.* 116, 26961–26969. doi:10.1073/pnas.1906694116.
- Melozzi, F., Woodman, M.M., Jirsa, V.K., Bernard, C., 2017. The virtual mouse brain: a computational neuroinformatics platform to study whole mouse brain dynamics. *eNeuro* doi:10.1523/ENEURO.0111-17.2017.
- Oh, S.W., Harris, J.A., Ng, L., Winslow, B., Cain, N., Mihalas, S., Wang, Q., Lau, C., Kuan, L., Henry, A.M., Mortrud, M.T., Ouellette, B., Nguyen, T.N., Sorensen, S.A., Slaughterbeck, C.R., Wakeman, W., Li, Y., Feng, D., Ho, A., Nicholas, E., Hirokawa, K.E., Bohn, P., Joines, K.M., Peng, H., Hawrylycz, M.J., Phillips, J.W., Hohmann, J.G., Wahnoutka, P., Gerfen, C.R., Koch, C., Bernard, A., Dang, C., Jones, A.R., Zeng, H., 2014. A mesoscale connectome of the mouse brain. *Nature* 508, 207–214. doi:10.1038/nature13186.
- Osen, K.K., Imad, J., Wennberg, A.E., Papp, E.A., Leergaard, T.B., 2019. Waxholm Space atlas of the rat brain auditory system: three-dimensional delineations based on structural and diffusion tensor magnetic resonance imaging. *Neuroimage* doi:10.1016/j.neuroimage.2019.05.016.
- Palomero-Gallagher, N., Zilles, K., 2019. Cortical layers: cyto-, myelo-, receptor- and synaptic architecture in human cortical areas. *Neuroimage* doi:10.1016/j.neuroimage.2017.08.035.
- Papp, E.A., Leergaard, T.B., Calabrese, E., Johnson, G.A., Bjaalie, J.G., 2014. Waxholm Space atlas of the sprague dawley rat brain. *Neuroimage* doi:10.1016/j.neuroimage.2014.04.001.
- Proix, T., Spiegler, A., Schirner, M., Rothmeier, S., Ritter, P., Jirsa, V.K., 2016. How Do Parcellation Size and Short-Range Connectivity Affect Dynamics in Large-Scale Brain Network models?. *Neuroimage*.
- Ritter, P., Schirner, M., McIntosh, A.R., Jirsa, V.K., 2013. The Virtual Brain integrates computational modeling and multimodal neuroimaging. *Brain Connect* 3, 121–145. doi:10.1089/brain.2012.0120.
- Rocher, L., Hendrickx, J.M., de Montjoye, Y.A., 2019. Estimating the success of re-identifications in incomplete datasets using generative models. *Nat. Commun.* doi:10.1038/s41467-019-10933-3.
- Sanz-Leon, P., Knock, S.A., Spiegler, A., Jirsa, V.K., 2015. Mathematical framework for large-scale brain network modeling in The Virtual Brain. *Neuroimage* 111, 385–430.
- Sanz-Leon, P., Knock, S.A., Woodman, M.M., Domide, L., Mersmann, J., McIntosh, A.R., Jirsa, V.K., 2013. The Virtual Brain: a simulator of primate brain network dynamics. *Front. Neuroinform* 7.
- Schirner, M., McIntosh, A.R., Jirsa, V.K., Deco, G., Ritter, P., 2018. Inferring multi-scale neural mechanisms with brain network modelling. *Elife* doi:10.7554/eLife.28927.
- Schirner, M., Rothmeier, S., Jirsa, V.K., McIntosh, A.R., Ritter, P., 2015a. An automated pipeline for constructing personalized virtual brains from multimodal neuroimaging data. *Neuroimage*.
- Schirner, M., Rothmeier, S., Jirsa, V.K., McIntosh, A.R., Ritter, P., 2015b. An automated pipeline for constructing personalized virtual brains from multimodal neuroimaging data. *Neuroimage* 117, 343–357.
- Shen, K., Bezgin, G., Schirner, M., Ritter, P., Everling, S., McIntosh, A.R., 2019. A macaque connectome for large-scale network simulations in TheVirtualBrain. *Sci. data.* doi:10.1038/s41597-019-0129-z.
- Smith, R., Connelly, A., 2019. MRtrix3 connectome: A BIDS Application for Quantitative Structural Connectome Construction. *OHBM*, p. W610.
- Sotiropoulos, S.N., Zalesky, A., 2019. Building connectomes using diffusion MRI: why, how and but. *NMR Biomed* 32, e3752.
- Stodden, V., McNutt, M., Bailey, D.H., Deelman, E., Gil, Y., Hanson, B., Heroux, M.A., Ioannidis, J.P.A., Tauber, M., 2016. Enhancing reproducibility for computational methods. *Science* (80) doi:10.1126/science.aah6168.
- Tournier, J.D., Smith, R., Raffelt, D., Tabbara, R., Dhollander, T., Pietsch, M., Christiaens, D., Jeurissen, B., Yeh, C.H., Connelly, A., 2019. MRtrix3: a fast, flexible and open software framework for medical image processing and visualisation. *Neuroimage* doi:10.1016/j.neuroimage.2019.116137.
- Trebaul, L., Deman, P., Tuyisenge, V., Jedynak, M., Hugues, E., Rudrauf, D., Bhattacharjee, M., Tadel, F., Chanteloup-Foret, B., Saubat, C., Reyes Mejia, G.C., Adam, C., Nica, A., Pail, M., Dubeau, F., Rheims, S., Trébuchon, A., Wang, H., Liu, S., Blauwblomme, T., Garcés, M., De Palma, L., Valentin, A., Metsähonkala, E.L., Petrescu, A.M., Landré, E., Szurhaj, W., Hirsch, E., Valton, L., Rocamora, R., Schulze-Bonhage, A., Mindruta, I., Francione, S., Maillard, L., Taussig, D., Kahane, P., David, O., 2018. Probabilistic functional tractography of the human cortex revisited. *Neuroimage* doi:10.1016/j.neuroimage.2018.07.039.
- Van Essen, D.C., 2013. Cartography and connectomes. *Neuron* 80, 775–790.
- Vella, M., Cannon, R.C., Crook, S., Davison, A.P., Ganapathy, G., Robinson, H.P.C., Angus Silver, R., Gleeson, P., 2014. libNeuroML and PyLEMS: using python to combine procedural and declarative modeling approaches in computational neuroscience. *Front. Neuroinform.* doi:10.3389/fninf.2014.00038.
- Wang, X.-J., 2020. Macroscopic gradients of synaptic excitation and inhibition in the neocortex. *Nat. Rev. Neurosci.* 21, 169–178.
- Yeh, C.-H., Jones, D.K., Liang, X., Descoteaux, M., Connelly, A., 2021. Mapping structural connectivity using diffusion MRI: challenges and opportunities. *J. Magn. Reson. Imaging* 53, 1666–1682.
- Zimmermann, J., Perry, A., Breakspear, M., Schirner, M., Sachdev, P., Wen, W., Kochan, N.A., Mapstone, M., Ritter, P., McIntosh, A.R., Solodkin, A., 2018. Differentiation of Alzheimer's disease based on local and global parameters in personalized virtual brain models. *Neuroimage Clin* doi:10.1016/j.nicl.2018.04.017.

CHAPTER D

Supplementary material of chapter [1](#)

D.1. Parameters for classification of neuron dynamic

Model of the neurons		
Type	adaptive exponential leaky integrator and fire	
subthreshold dynamics	$C_m \frac{dV_m}{dt} = -g_L(V_m - E_L) + g_L \Delta_T e^{\frac{V_m - V_{th}}{\Delta_T}} - W + I_e$ $\tau_w \frac{dW}{dt} = a(V_m - E_L) - W$	
reset condition	<p>For $t^{(f)} = \{t V_m(t) \geq V_{peak}\}$</p> <ul style="list-style-type: none"> — $V_m([t^{(f)}; t^{(f)} + t_{ref}]) = V_{reset}$ — $W([t^{(f)}]) = W([t^{(f)}]) + b$ 	
Default Model Parameters		
C_m	Capacity of the membrane	201.0 pF
t_{ref}	Duration of refractory period	2.0 ms
V_{reset}	Reset value for V_m after a spike	-60.0 mV
E_L	Leak reversal potential	-70.6 mV
g_L	Leak conductance	30.0 nS
Δ_T	Slope factor	2.0 mV
V_{peak}	Spike detection threshold	10.0 mV
a	Subthreshold adaptation	0.0 nS
b	Spike-triggered adaptation	80.5 pA
τ_w	Adaptation time constant	144.0 ms
V_{th}	Spike initiation threshold	-50.4 mV
I_e	Constant external input current	0.0 pA
V_m	Initialization of the voltage membrane	-70.6 mV
W	Initialization of adaptation current	0.0 pA
Simulator		
simulator configuration	The simulator is NEST with a resolution of 0.001 ms. The implementation model is the default implementation of "aeif_cond_alpha"	

D. Supplementary material of chapter 1 – D.1. Parameters for classification of neuron dynamic

Classification of excitability from ramp current	
excitability class 1	$I_{maximum} : 800.0\text{pA}$
excitability class 2	$a : 2.0\text{nS}, b : 0.0\text{pA}$ and $I_{maximum} : 800.0\text{pA}$
Classification from step current (time simulation :5.s and time stimulation :4.s)	
tonic spike	$I_{stim} : 580.0\text{pA}$
phasic spike	$a : 2.\text{nS}, \tau_w : 200.0\text{ms}$ and $I_{stim} : 580.\text{pA}$
tonic burst	$b : 30.\text{pA}, V_{reset} : -48.\text{mV}$ and $I_{stim} : 580.\text{pA}$
phasic burst	$b : 30.\text{pA}, a : 2.\text{nS}, V_{reset} : -48.\text{mV}$ and $I_{stim} : 580.\text{pA}$
spike frequency adaptation	$b : 3.\text{pA}, a : 1.6\text{nS}$ and $I_{stim} : 580.\text{pA}$
mixed model	$b : 5.\text{pA}, a : 1.6\text{nS}, V_{reset} : -50.\text{mV}$ and $I_{stim} : 580.\text{pA}$
irregular spiking	$b : 15.\text{pA}, \tau_w : 130.\text{ms}, V_{reset} : -49.\text{mV}$ and $I_{stim} : 580.\text{pA}$
delay spike	$b : 15.\text{pA}, a : -3.\text{nS}, E_L : -72.5\text{mV}$ and $I_{step\ current} : 580.\text{pA}$
delay burst	$b : 7.\text{pA}, a : -1.1\text{nS}, E_L : -72.5\text{mV}, V_{reset} : -49.\text{mV}$ and $I_{stim} : 580.\text{pA}$
delay acceleration	$b : 9.\text{pA}, a : -2.5\text{nS}, \tau_w : 200.\text{ms}, E_L : -72.5\text{mV}, V_{reset} : -60.\text{mV}, V_m : -72.5\text{mV}$ and $I_{stim} : 580.\text{pA}$
Classification from impulse current (time simulation :500.ms and time stimulation :10.ms)	
single spike	$I_{stim} : 1250.\text{pA}$
spike latency	$\Delta_T : 19.\text{mV}, V_m : -57.\text{mV}$ and $I_{stim} : 500.\text{pA}$
after potential	$b : -300.\text{pA}, a : 10.\text{nS}, \tau_w : 50.\text{ms}, \Delta_T : 20.\text{mV}, V_{reset} : -50.\text{mV}, V_m : -62.\text{mV}, W : 70.\text{pA}$ and $I_{stim} : 1000.\text{pA}$
rebound spike	$a : 8.\text{nS}, E_L : -52.\text{mV}, V_m : -52.\text{mV}, W : 13.\text{pA}$ and $I_{stim} : -580.\text{pA}$
rebound burst	$a : 8.\text{nS}, b : 4.\text{pA}, E_L : -52.\text{mV}, V_{reset} : -49.\text{mV}, V_m : -52.\text{mV}, W : 13.\text{pA}$ and $I_{stim} : -580.\text{pA}$
bi-stability	$a : 90.\text{nS}, b : 5.\text{pA}, E_L : -51.\text{mV}, V_{reset} : -54.\text{mV}, V_m : -50.\text{mV}, W : 42.\text{pA}$ and $I_{stim} : 1000.\text{pA}$
Classification from perturbation (time simulation :500.ms and time stimulation :5.ms)	
integration	$I_{stim} : 160.\text{pA}$
resonator	$a : 300.\text{nS}, E_L : -61.\text{mV}, V_{th} : -60.\text{mV}, V_m : -61.\text{mV}, W : 34.5\text{pA}$ and $I_{stim} : 160.\text{pA}$
Example of dynamic	
demonstration	$\Delta_T : 10.\text{mV}, t_{ref} : 10.\text{ms}, a : 5.\text{nS}, \tau_w : 100.\text{ms}, V_m : -69.0\text{mV}, E_L : -69.\text{mV}, I_{stim} : 250.\text{pA}$

TABLEAU D.1. – Parameter for classification of neuron dynamic

CHAPTER E

Supplementary material of chapter 3

Contents of Chapter E

E.1. Table of content of Supplementary Notes, Figures and Tables	223
E.1. Supplementary Notes	225
E.1.1. Supplementary Note 1 Guidelines Input/Output (I/O) interface	225
E.1.1.1. NEST I/O interface	225
E.1.1.2. TVB I/O interface	225
E.1.2. Supplementary Note 2 Guideline for implementation of transfer module	226
E.1.3. Supplementary Note 3 Detail characterization of the workflow TVB-NEST	227
E.1.3.1. Non-Functional Requirements	227
E.1.3.2. Simulator Requirements	227
E.1.3.3. Framework Requirements	228
E.1.3.4. Additional : characterization of the coupling [59, 337]	229
E.2. Supplementary Table	230
E.3. Supplementary Figures	249

E.1. Table of content of Supplementary Notes, Figures and Tables

Supplementary Material	Title
Supplementary Note E.1.1	Guidelines Input/Output (I/O) interface
Supplementary Note E.1.2	Guideline for implementation of transfer module
Supplementary Note E.1.3	Detail characterization of the workflow TVB-NEST
Supplementary Table E.1	Table describing the co-simulation
Supplementary Figure E.1	Zoom on 1s for the spiking neural network with regular bursting state
Supplementary Figure E.2	Mouse Brain activity for Asynchronous Irregular
Supplementary Figure E.3	Mouse Brain activity for Synchronous Irregular
Supplementary Figure E.4	Mouse Brain activity for Regular Bursting
Supplementary Figure E.5	Detail sequence diagram of the co-simulation
Supplementary Figure E.6	Sequence diagram of the communication protocol with NEST
Supplementary Figure E.7	State diagram of NEST wrapper
Supplementary Figure E.8	State diagram of transfer components for interaction with NEST
Supplementary Figure E.9	Sequence diagram of the communication protocol with the wrapper of TVB
Supplementary Figure E.10	State diagram of wrapper of TVB modules
Supplementary Figure E.11	State diagram of transfer components for interaction with the wrapper of TVB
Supplementary Figure E.12	File organisation of the transformer module
Supplementary Figure E.13	State diagram of transfer components which transform data from one scale to another
Supplementary Figure E.14	Structure diagram of the two transfer modules with the description of each component
Supplementary Figure E.15	Class diagram of the transfer modules
Supplementary Figure E.16	Internal communication in the transfer modules between the components
Supplementary Figure E.17	Performance of the co-simulation on one computer with different number of neurons
Supplementary Figure E.18	Performance of the co-simulation on one computer with different time of synchronization between simulators
Supplementary Figure E.19	Performance of the co-simulation on one computer with different process and thread for NEST
Supplementary Figure E.20	Performance of the co-simulation on one supercomputer for different number of neurons
Supplementary Figure E.21	Performance of the co-simulation on one supercomputer for different time of synchronization between simulator

E. Supplementary material of chapter 3 – E.1. Table of content of Supplementary Notes, Figures and Tables

Supplementary Material	Title
Supplementary Figure E.22	Performance of the co-simulation on one supercomputer for different number of node for NEST
Supplementary Figure E.23	Details of the timer of one run for the reference configuration
Supplementary Figure E.24	Proof of concept of replacing NEST and TVB by other simulators

E.1. Supplementary Notes

E.1.1 Supplementary Note 1 | Guidelines Input/Output (I/O) interface

Before creating the I/O interface, the simulator must be analysed. The objective is to verify the existence of output and input devices, the paradigm of parallelisation, the tools for the parallelisation and the properties to keep. This analysis will help to modify the architecture of the simulator. The modification needs to follow the simulator development and its maintainability. The last part is the creation of a wrapper to communicate with the transfer module if required. Two paragraphs will provide further details about how to transfer data from NEST [152] to TVB [300] and vice versa.

E.1.1.1 NEST I/O interface

NEST has two types of devices : stimulating and recording devices. These devices receive or send messages mainly based on spikes times. The parallelisation uses MPI and/or threading depending on its parametrisation, and it is based principally on event transfer (spikes between neurons). The critical property to conserve is its scalability. From this statement, a new interface was implemented into version 3 of NEST and uses MPI communication. The modification architecture of NEST is creating a specific back end of the recording and stimulating devices and reformatting input devices to include the usage of a specific back end. Each back-end uses a particular communication protocol (see supplementary figures E.6), which includes transmitting the NEST state using tags and transferring data. The transfer module directly uses this interface (see supplementary figures E.13).

E.1.1.2 TVB I/O interface

The TVB simulator presents monitor classes for recording values and stimulating classes. There is no parallelisation optimisation, and all the data is stored in memory (recording simulated data, stimulating and transferring data between Neural Mass). This simulator does not have specific properties to keep for the optimisation simulation. From this statement, the prototype uses a new monitor that dynamically modifies the simulator. This new monitor adds an extra buffer to delay the simulated data. Its advantage is the possibility of using proxy nodes and an IO interface to include external data during this delay in communication between nodes. Following the simulator development and its maintainability, this new monitor is not included in the code of TVB. The main reason is that it is difficult to maintain and debug dynamic modifications. This interface is not enough for communication with the transfer module because there is a need to use MPI communication. A wrapper around this interface is implemented to overcome this requirement(see supplementary figure E.10 for details). A bug present in the implementation monitor does not take into

account the time of synchronisation between the simulator, but it does not have an impact on the co-simulation dynamic. The official release of TVB includes a new class of 'co-simulator' that handles external communication.

E.1.2 Supplementary Note 2 | Guideline for implementation of transfer module

This section focuses on the intention behind the implementation of the transfer modules. In the future, two types of scientists will improve and use transfer modules. The neuroscientist or physician will modify it to create new models and adapt them to their scientific questions. In parallel, computational scientists will work to improve communication speed between all modules and components. Furthermore, in the future, there will be a need to add other simulators such as NEURON [52], Arbor [3], Neurolib [48]. The architecture design needs to simplify the addition of other simulators and other types of data (membrane voltage, current, ...).

The separation of the neuroscience research and computer science research is done by the separation of the functions of the transfer module in three components/ objects/ processes : two for the I/O interface with simulators and one for transformation functions (see Supplementary Figure E.14). A neuroscientist will principally modify the transformation components where the meaning of the data transformation is required and important for his work. A computational scientist will focus on optimising the communication with the interface with a simulator, the internal communication and the management of the data flux.

To avoid conflict between this type of research, there is a simple API for receiving and sending data in each component (see examples of activity diagram of the transfer components in the Supplementary Figure E.16). The only constraint to the neuroscientist is to respect the buffering of data in the transformation function by releasing the input connection before accessing the output connection. Moreover, this simple API is implemented following the abstract factory pattern. This design pattern is chosen to help the comparison of different implementations of communication and the integration of new simulators.

The API address partially the constraint of the simplification for adding a new simulator because only one missing part is a component for the interface with the simulator; the rest can be reused. The other architecture element for this constraint is separating files for each simulator and encapsulating the interface in an abstract class following a composite pattern.

The second constraint is the simplification of adding a new type of data respected by the imposition of a convention for data management. This convention comprises four functions and one Boolean for sending and receiving data. The functions are "ready for transfer data?", "transfer the data", "end of transfer data" and "release the connection". The Boolean contains information about the connection statement from the other side (0 :open or 1 :closed).

E.1.3 Supplementary Note 3 | Detail characterization of the workflow TVB-NEST

This characterisation is based on the taxonomies proposed in Gomes et al. 2019 [141]. However, this taxonomy is not the best for this workflow because the transformation modules are not considered. Additionally, one of the hypotheses of this taxonomy is the presence of an orchestrator, which is not the case for the workflow.

E.1.3.1 Non-Functional Requirements

- Fault tolerance : No (NEST does not store the previous state and the communication spike, which creates the impossibility of coming back in the future)
- Configuration reused : Yes (the configuration of each simulator is independent and defined during the initialisation)
- Performance : Yes and No (the simulator’s scalability and parallelisation are kept, but there is no modulation of the integration step or signal extrapolation).
- IP Protection : No protection (NEST and TVB do not use protected models.)
- Parallelism : Yes (the communication use MPI and each simulator is run in individual processes)
- Distributed : Yes (the workflow keeps the properties of NEST to be simulated in a distributed way)
- Hierarchy : Yes (the workflow is independent of the model for each simulator and the transformation function. There is some requirement for the connection between modules which creates the dependencies.)
- Scalability : No (it is dependent on the simulators)
- Platform independent : Yes and No (it requires some dependence on the platform, but the usage of docker or singularity can pass it)
- Extensibility : Yes and No (some extra modules such as NESTML or TVB can create models for each simulator but not a specific extension for the transformation and all the simulations.)
- Accuracy : No (there are any simulators which provide the errors or the convergence of the simulations.)
- Open Source : Yes (each simulator is open source, and the workflow is also open source)

E.1.3.2 Simulator Requirements

Information Exposed :

- Frequency of State : No (the frequency of the state for the simulator and the co-simulation is fixed during the initialisation)
- Frequency of Outputs : No (same as before. Moreover, TVB can have an output frequency lower than this internal integration frequency)
- Detailed Model : Yes (the code for all the models is available)

E. Supplementary material of chapter 3 – E.1. Supplementary Notes

- Nominal Values of Outputs : dependent on the output and the models used
- Nominal Values of State : dependent on the model
- I/O Signal Kind : No (there is not a master algorithm but NEST has some internal statement about the signal communication between devices and nodes.)
- Time Derivative : Output only
- Jacobian : No
- Discontinuity Indicator : No (the transformation modules handles this part)
- Deadreckoning model : No
- Preferred Step Size : No (the step size is fixed at the beginning)
- Next Step Size : No (there is not an orchestrator for managing the step size and the step size are fixed)
- Order of Accuracy : No (there is no extrapolation function)
- I/O Causality : Propagation Delay (the delay is used for the parallelization. However this delay is fixed during the simulation)
- Input Extrapolation : No (there is no extrapolation function)
- State Variables : Values
- Micro-Step Outputs : Yes (TVB and NEST give the output of each micro-step but it can be modulated)
- Worst Case Execution Time : Yes (the worst case is when the minimum delay is equal to the micro-time step (see Performance section))

Causality : Causal

Time Constraints :

- Analytic Simulation : False (there does no analytic solution to this co-simulation)
- Scaled Real Time Simulation : Fixed for TVB and NEST
- Rollback Support : No (there is no rollback support for NEST and TVB)

Availability : local

E.1.3.3 Framework Requirements

- Standard : No standard (ad-hock solution)
- Coupling : Input/Output Assignments (Transformation modules between the two simulators take the role to synchronize the I/O of the simulators)
- Number of Simulation Units : Two simulators
- Domain : Hybrid
- Dynamic structure : No (all the dependency is defined at the beginning)
- Co-simulation Rate : Single (unique size of the synchronization step between simulators and micro-step is fixed during the simulation)
- Communication Step Size : Fixed
- Strong Coupling Support : None – Explicit Method (the transformation module contains the information on the coupling of the simulators)
- Results Visualization : It can be in live or postmortem
- Communication Approach : Jacobi (however, the delays allow the separation of micro-steps without creating errors)

E.1.3.4 Additional : characterization of the coupling [59, 337]

The previous characterization is focusing more on the technical details but it is missing the characterization of the transformation modules. For the workflow of TVB-NEST, the scales are separate in space (micro- and macro-scale). The coupling between the simulators is a tightly coupled or cyclic coupling using a fixed number of simulators instance. The workflows allow sequential or parallel execution depending on the number of initial conditions.

E.2. Supplementary Table

A1 Co-simulator environment		
Simulator	NEST [152]	TVB [299]
Version	3.0	2.0
Integrator method	4th order Runge-Kutta-Fehlberg method	Heun method
Integration step size	0.1 ms	0.1 ms
Synchronization time step	2.0 ms	
Simulated time	60.0 s	
Analyzed time	between 42.5 s and 53.5	
Type of I/O interface	proxy input	proxy region
A2 Co-simulator architecture		
Reference model	The mouse brain with 104 regions	
Simulator	NEST	TVB
Number of simulated region	2	102
Number of MPI processes	3	1
Number of thread per process	6	1
Number of random seeds	1	
Transfer module	NEST to TVB	TVB to NEST
number of transfer module	2	2
Number of MPI processes	2	2
Number of threads or processes	6	6
Number of random seeds	1	1
B1	NEST : Model Summary	
Topology	left and right CA1 connected to TVB	
Population	2 by regions : excitatory and inhibitory	
Connectivity	random convergent connection	
Neuron Model	adaptive exponential leaky integrate and fire neurons [37], fixed threshold and fixed absolute refractory time	
Synapse Model	conductance-based exponential shape	
Plasticity	—	
Input	Independent fixed rate Poisson generator spike trains to all neurons and spike trains from TVB	
Measurement	Voltage, Adaptation Current, Spike Activity and Model of Local Field Potential signal	

E. Supplementary material of chapter 3 – E.2. Supplementary Table

B2		NEST : Topology	
regions	2 regions (left CA1 and right CA1)		
number of neurons by regions	N=10000		
percentage of inhibitory neurons	$g_{inh}=20\%$		
B3		NEST : Population by regions	
Name	Elements	Size	
E	aeif_cond_exp	$N_e = (1-g_{inh})N = 8000$	
I	aeif_cond_exp	$N_i = g_{inh}N = 2000$	
P_{ext}	Poisson generator	1	
I_{ext}	spike generator (input from TVB)	N	
B4		NEST : Neuron Model	
Name	aeif		
Type	adaptive exponential leaky integrator [37] and fire with conductance synapse		
subthreshold dynamics	$C_m \frac{dV_m}{dt} = -g_L(V_m - E_L) + g_L \Delta_T e^{\frac{V_m - V_{th}}{\Delta_T}}$ $-g_e(t)(V_m - E_{ex}) - g_i(t)(V_m - E_{in})$ $-W + I_e$ $\tau_w \frac{dW}{dt} = a(V_m - E_L) - W$		
reset condition	<p>For $t^{(f)} = \{t \mid V_m(t) \geq V_{peak}\}$</p> <ul style="list-style-type: none"> — $V_m([t^{(f)}; t^{(f)} + t_{ref}]) = V_{reset}$ — $W([t^{(f)}]) = W([t^{(f)}]) + b$ 		
B5		NEST : Synapse Model	
Name	cond_exp		
Type	post-synaptic conductance in the form of truncated exponentials		
Coupling equation	$g_e(t) = \sum_{t_j^{(f)}} w_j \exp_trunc(t - t_j, \tau_{ex}) \text{ with } w_j > 0.0$ $g_i(t) = \sum_{t_j^{(f)}} w_j \exp_trunc(t - t_j, \tau_{in}) \text{ with } w_j < 0.0$ $\exp_trunc(t, \tau) = e^{1 - \frac{t}{\tau}} Heaviside(t)$		

E. Supplementary material of chapter 3 – E.2. Supplementary Table

B6		NEST : Excitatory Neuron Model Parameters		
case		Asynchronous	Irregular Synchronous	Regular bursting
C_m	Capacity of the membrane		200.0 pF	
t_{ref}	Duration of refractory period		5.0 ms	
V_{reset}	Reset value for V_m after a spike	-64.5 mV	-64.5 mV	-47.5 mV
E_L	Leak reversal potential	-64.5 mV	-64.5 mV	-74.0 mV
g_L	Leak conductance		10.0 nS	
Δ_T	Slope factor		2.0 mV	
V_{peak}	Spike detection threshold		0.0 mV	
a	Subthreshold adaptation		0.0 nS	
b	Spike-triggered adaptation	10.0 pA	100.0 pA	50.0 pA
τ_w	Adaptation time constant	500.0 ms	500.0 ms	150.0 ms
V_{th}	Spike initiation threshold		-50.0 mV	
I_e	Constant external input current		0.0 pA	
E_{ex}	Excitatory reversal potential		0.0 mV	
E_{in}	Inhibitory reversal potential		-80.0 mV	
V_m	Initialization of the voltage membrane	-64.5 mV	-64.5 mV	-47.5 mV
W	Initialization of adaptation current		0.0 pA	

E. Supplementary material of chapter 3 – E.2. Supplementary Table

B7		NEST : Inhibitory Neuron Model Parameters		
case		Asynchronous	Irregular Synchronous	Regular bursting
C_m	Capacity of the membrane		200.0 pF	
t_{ref}	Duration of refractory period		5.0 ms	
V_{reset}	Reset value for V_m after a spike	-65.0 mV	-65.0 mV	-75.0 mV
E_L	Leak reversal potential	-65.0 mV	-65.0 mV	-75.0 mV
g_L	Leak conductance		10.0 nS	
Δ_T	Slope factor		0.5 ms	
V_{peak}	Spike detection threshold		0.0 mV	
a	Subthreshold adaptation		0.0 nS	
b	Spike-triggered adaptation		0.0 pA	
τ_w	Adaptation time constant		1.0 ms	
V_{th}	Spike initiation threshold		-50.0 mV	
I_e	Constant external input current		0.0 pA	
E_{ex}	Excitatory reversal potential		0.0 mV	
E_{in}	Inhibitory reversal potential		-80.0 mV	
V_m	Initialization of the voltage membrane	-65.0 mV	-65.0 mV	-75.0 mV
W	Initialization of adaptation current		0.0 pA	

E. Supplementary material of chapter 3 – E.2. Supplementary Table

B8 NEST : Connectivity between regions				
parameter synapses				
τ_{ex}	Rise time of excitatory synaptic conductance			5.0ms
τ_{in}	Rise time of inhibitory synaptic conductance			5.0ms
Name	Source	Target	Weights	Pattern
EE_global	E	E I	1.0	Fixed total number of connections from one to another region. The number of synapses to another region is A : 1150000, IS : 3000000 and RB : 800000. The delay (161.6 ms) is defined by the multiplication of velocity (3.0 mm/ms) and distance between regions (53.855 mm). See for more details in the section TVB : connectivity because delays and the weights are extracted from the connectivity of TVB.

E. Supplementary material of chapter 3 – E.2. Supplementary Table

B9 NEST : Connectivity inside the regions				
Name	Source	Target	Weights	Pattern
EE	E	E	1.0	Fixed number of input synapses ($N_e * p_{connect}$: A and SI 400 = 8000 * 0.05 and RB 40 = 8000 * 0.005). Neuron can connect to itself and can have multiple connections with another neuron.
EI	E	I	1.0	Fixed number of input synapses ($N_e * p_{connect}$: A and SI 400 = 8000 * 0.05 and RB 40 = 8000 * 0.005). Neuron can have multiple connections with another neuron.
IE	I	E	g	Fixed number of input synapses ($N_i * p_{connect}$: A and SI 100 = 2000 * 0.05 and RB 10 = 2000 * 0.005). Neuron can have multiple connections with another neuron. The weight equals 10.0 for A, 5.0 for SI and 10.0 for RB.
II	I	I	g	Fixed number of input synapses ($N_i * p_{connect}$: A and SI 100 = 2000 * 0.05 and RB 10 = 2000 * 0.005). Neuron can connect to itself and can have multiple connections with another neuron. The weight equals 10.0 for A, 5.0 for SI and 10.0 for RB.

E. Supplementary material of chapter 3 – E.2. Supplementary Table

B10	NEST : Input		
Poisson generator			
equation	$p(n) = \frac{\lambda^n}{n!} \exp(-\lambda)$		
implementation algorithm	Ahrens and Dieter 1982		
case	A	IS	RB
excitatory firing rate λ_{ex}	1.0	0.0	0.0
inhibitory firing rate λ_{in}	0.0	0.0	0.0
weight connection	1.0	1.0	1.0
Spike generator			
Proxy for the input of the region simulated with TVB. (see the section transformation TVB to NEST)			

E. Supplementary material of chapter 3 – E.2. Supplementary Table

B11	NEST : Measurement (part 1)		
state variable	Voltage membrane, adaptation current	precision	0.1
		number of recorded neurons	10 excitatory and 10 inhibitory
	spike time	precision	0.1 ms
		number of recorded neurons	all
spike activities	raster plot	precision	0.1 ms
	histogram of instantaneous firing rate	bins	0.1ms
	simple moving average	windows size	T (20ms)
	spectrogram	method	Welch's method
		sampling frequencies	10^4 Hz
		window shape	Hann window
		length of each segment	10^4
		length of the FFT	10^4
		number of points overlapping	$5 \cdot 10^3$
		detrend	removing the mean
sides	only real part		

E. Supplementary material of chapter 3 – E.2. Supplementary Table

B11		NEST : Measurement (part 2)		
Micro -electrodes Local Field Potential	software	HybridLFPy [148]		
	number of MPI	2		
	number random seed	2		
	number of segment by neuron	defined by the method lambda100 of Neuron		
	resolution	0.1 ms		
	soma position	random in a cylinder of radius 2000 mm and height of 100mm with a minimal distance of 1mm. The centre of the cylinder is (0,-400).		
	excitatory neurons			
	morphology	pyramidal cell of Shuman 2020 [322] without biophysics and synapses mechanisms		
	initial membrane potential	V_m (-64.5 mV or -47.5 mV)		
	axial resistance	150.0 Ohm		
	membrane capacitance	C_m (200pF)		
	passive mechanism	yes		
	passive reversal potential	E_L (-64.5 mV or -74.0 mV))		
	passive conductance	g_L (10 nS)		
	inhibitory neurons			
	morphology	basket cell of Shuman 2020 [322] without biophysics and synapses mechanisms		
	initial membrane potential	V_m (-64.5 mV or -75.0 mV)		
	axial resistance	150.0 Ohm		
	membrane capacitance	C_m (200pF)		
	passive mechanism	yes		
passive reversal potential	E_L (-65.0 mV or -75.0 mV))			
passive conductance	g_L (10 nS)			

E. Supplementary material of chapter 3 – E.2. Supplementary Table

B11		NEST : Measurement(part 3)						
Micro -electrodes : Local Field Potential	connectivity							
	layers	2 : [[300,-100],[-100,-600]]						
	synapse shape	truncated exponential						
	delay and weight distribution	homogeneous values by population						
	excitatory connection by layers and populations	$\frac{N_e * p_{connect} * 0.5}{N_e * p_{connect} * 0.5}$		$\frac{0}{N_i * p_{connect}}$				
	excitatory weight	1.0						
	excitatory delay	dt (0.1 ms)						
	inhibitory connection by layers and populations	$\frac{0}{N_e * p_{connect}}$		$\frac{0}{N_i * p_{connect}}$				
	inhibitory weight	g (5.0 or 10.0)						
	inhibitory delay	dt (0.1 ms)						
	electrodes							
	extracellular conductivity	0.3 S						
	electrode positions and contacts surface normal	positions			normal			
		x	y	z	x	y	z	
		1273	1273	1273	1	1	0	
		1273	-1273	-1273	1	1	0	
		-1273	-1273	15	1	1	0	
		-15	15	-15	1	1	0	
		1288	1258	1288	1	-1	0	
		1258	1288	1258	1	-1	0	
	1288	1258	-1800	1	-1	0		
	-1800	-1800	-1800	1	-1	0		
	-385	-385	-415	1	0	0		
	-415	-385	-385	1	0	0		
	-415	-415	-385	1	0	0		
	-385	-415	-415	1	0	0		
contact shape	circle of radius 20 mm							
number of discrete point for compute the average potential	20							
assumption method	soma as point							

E. Supplementary material of chapter 3 – E.2. Supplementary Table

C1 TVB : Model Summary	
Neural Mass model	Mean Adaptive Exponential
Connectivity	Mouse connectome with 104 regions
Coupling	linear coupling
stimulus	—
Monitors	ECoG (Electrocorticography) and state variable of the mean-field
C2 TVB : Coupling	
Name	Linear
Type	Linear coupling
equations	$v_{ext_k} = a * \left(\sum_{j=1}^{104} u_{kj} v_{e_j}(t - \tau_{kj}) \right) + b$ <p>where u_{kj} are the elements of the weights matrix, τ_{kj} are the elements of the delay matrix and v_{e_j} are the mean excitatory firing rate of the regions j.</p>
parameters	$a = 1.0$ and $b = 0.0$
C3 TVB : Connectivity	
Connectivity is extracted from tracer data as explained by the paper TVBM [240]	
number of region	104
tract lengths	maximum : 115.46 and mean : 53.58
speed	3 ms
weights	The weights are normalized such as the sum of the input weight to one region equals 1. (maximum : 0.73 and mean : 0.02)
centers	average center of mouse brain : [57., 74.97, 42.53]
orientation	the orientation is defined by a vector from the average centre of a mouse brain to the centre of the regions

E. Supplementary material of chapter 3 – E.2. Supplementary Table

C3	TVB : Connectivity
region name	<p>the region names are extracted from Allen Mouse Brain Connectivity Atlas (17/01/2017)[263]</p>
	<p>Right Primary motor area, Right Secondary motor area, Right Primary somatosensory area nose, Right Primary somatosensory area barrel field, Right Primary somatosensory area lower limb, Right Primary somatosensory area mouth, Right Primary somatosensory area upper limb, Right Supplemental somatosensory area, Right Gustatory areas, Right Visceral area, Right Dorsal auditory area, Right Primary auditory area, Right Ventral auditory area, Right Primary visual area, Right Anterior cingulate area dorsal part, Right Anterior cingulate area ventral part, Right Agranular insular area dorsal part, Right Retrosplenial area dorsal part, Right Retrosplenial area ventral part, Right Temporal association areas, Right Perirhinal area, Right Ectorhinal area, Right Main olfactory bulb, Right Anterior olfactory nucleus, Right Piriform area, Right Cortical amygdalar area posterior part, Right Field CA1, Right Field CA3, Right Dentate gyrus, Right Entorhinal area lateral part, Right Entorhinal area medial part dorsal zone, Right Subiculum, Right Caudoputamen*, Right Nucleus accumbens*, Right Olfactory tubercle*, Right Substantia innominata*, Right Lateral hypothalamic area*, Right Superior colliculus sensory related*, Right Inferior colliculus*, Right Midbrain reticular nucleus*, Right Superior colliculus motor related*, Right Periaqueductal gray*, Right Pontine reticular nucleus caudal part*, Right Pontine reticular nucleus*, Right Intermediate reticular nucleus*, Right Central lobule*, Right Culmen*, Right Simple lobule*, Right Ansiform lobule*, Right Paramedian lobule*, Right Copula pyramidis*, Right Paraflocculus*, Left Primary motor area, Left Secondary motor area, Left Primary somatosensory area nose, Left Primary somatosensory area barrel field, Left Primary somatosensory area lower limb, Left Primary somatosensory area mouth, Left Primary somatosensory area upper limb, Left Supplemental somatosensory area, Left Gustatory areas, Left Visceral area, Left Dorsal auditory area, Left Primary auditory area, Left Ventral auditory area, Left Primary visual area, Left Anterior cingulate area dorsal part, Left Anterior cingulate area ventral part, Left Agranular insular area dorsal part, Left Retrosplenial area dorsal part, Left Retrosplenial area ventral part, Left Temporal association areas, Left Perirhinal area, Left Ectorhinal area, Left Main olfactory bulb, Left Anterior olfactory nucleus, Left Piriform area, Left Cortical amygdalar area posterior part, Left Field CA1, Left Field CA3, Left Dentate gyrus, Left Entorhinal area lateral part, Left Entorhinal area medial part dorsal zone, Left Subiculum, Left Caudoputamen*, Left Nucleus accumbens*, Left Olfactory tubercle*, Left Substantia innominata*, Left Lateral hypothalamic area*, Left Superior colliculus sensory related*, Left Inferior colliculus*, Left Midbrain reticular nucleus*, Left Superior colliculus motor related*, Left Periaqueductal gray*, Left Pontine reticular nucleus caudal part*, Left Pontine reticular nucleus*, Left Intermediate reticular nucleus*, Left Central lobule*, Left Culmen*, Left Simple lobule*, Left Ansiform lobule*, Left Paramedian lobule*, Left Copula pyramidis*, Left Paraflocculus*</p>
cortical region	all the region name ending by a '*' are not cortical regions

E. Supplementary material of chapter 3 – E.2. Supplementary Table

C4	TVB : Neural Mass Model (part 1)
Name	Mean Ad Ex [356]
Type	Neural Mass model of a network of adaptive exponential integrate and fire excitatory and inhibitory neurons of second statistical order with adaptation
equation	$T \frac{\partial v_e}{\partial t} = (\mathcal{F}_e - v_e) + \frac{1}{2} c_{ee} \frac{\partial^2 \mathcal{F}_e}{\partial v_e \partial v_e} + \frac{1}{2} c_{ei} \frac{\partial^2 \mathcal{F}_e}{\partial v_e \partial v_i} + \frac{1}{2} c_{ie} \frac{\partial^2 \mathcal{F}_e}{\partial v_i \partial v_e} + \frac{1}{2} c_{ii} \frac{\partial^2 \mathcal{F}_e}{\partial v_i \partial v_i}$ $T \frac{\partial v_i}{\partial t} = (\mathcal{F}_i - v_i) + \frac{1}{2} c_{ee} \frac{\partial^2 \mathcal{F}_i}{\partial v_e \partial v_e} + \frac{1}{2} c_{ei} \frac{\partial^2 \mathcal{F}_i}{\partial v_e \partial v_i} + \frac{1}{2} c_{ie} \frac{\partial^2 \mathcal{F}_i}{\partial v_i \partial v_e} + \frac{1}{2} c_{ii} \frac{\partial^2 \mathcal{F}_i}{\partial v_i \partial v_i}$ $T \frac{\partial c_{ee}}{\partial t} = (\mathcal{F}_e - v_e) (\mathcal{F}_e - v_e) + c_{ee} \frac{\partial \mathcal{F}_e}{\partial v_e} + c_{ee} \frac{\partial \mathcal{F}_e}{\partial v_e} + c_{ei} \frac{\partial \mathcal{F}_i}{\partial v_e} + c_{ie} \frac{\partial \mathcal{F}_i}{\partial v_e} - 2c_{ee} + \frac{\mathcal{F}_e (1/T - \mathcal{F}_e)}{N_e}$ $T \frac{\partial c_{ei}}{\partial t} = (\mathcal{F}_e - v_e) (\mathcal{F}_i - v_i) + c_{ee} \frac{\partial \mathcal{F}_e}{\partial v_e} + c_{ei} \frac{\partial \mathcal{F}_e}{\partial v_i} + c_{ei} \frac{\partial \mathcal{F}_i}{\partial v_e} + c_{ii} \frac{\partial \mathcal{F}_i}{\partial v_i} - 2c_{ei}$ $T \frac{\partial c_{ie}}{\partial t} = (\mathcal{F}_i - v_i) (\mathcal{F}_e - v_e) + c_{ie} \frac{\partial \mathcal{F}_e}{\partial v_i} + c_{ee} \frac{\partial \mathcal{F}_e}{\partial v_e} + c_{ii} \frac{\partial \mathcal{F}_i}{\partial v_i} + c_{ie} \frac{\partial \mathcal{F}_i}{\partial v_e} - 2c_{ie}$ $T \frac{\partial c_{ii}}{\partial t} = (\mathcal{F}_i - v_i) (\mathcal{F}_i - v_i) + c_{ie} \frac{\partial \mathcal{F}_e}{\partial v_i} + c_{ei} \frac{\partial \mathcal{F}_e}{\partial v_i} + c_{ii} \frac{\partial \mathcal{F}_i}{\partial v_i} + c_{ii} \frac{\partial \mathcal{F}_i}{\partial v_i} - 2c_{ii} + \frac{\mathcal{F}_i (1/T - \mathcal{F}_i)}{N_i}$ $\tau_{W_e} \frac{\partial W_e}{\partial t} = -W_e + b_e v_e + a_e (\mu_V(v_e, v_i, v_{ext}, W_e) - EL_e)$ $\tau_{W_i} \frac{\partial W_i}{\partial t} = -W_i + b_i v_i + a_i (\mu_V(v_e, v_i, v_{ext}, W_i) - EL_i)$

C4	TVB : Neural Mass Model (part 2)
noise equation	<p>Ornstein-Uhlenbeck process :</p> $\tau_{ou} \frac{dou_t}{dt} = (\mu - ou_t) + \sigma dW_t$ <p>with W_t is a Wiener process</p>
transfer function	$\mathcal{F}_e = \mathcal{F}((v_e + 1e - 6) + w_\sigma ou_t, v_{ext}, v_i, W_e)$ $\mathcal{F}_i = \mathcal{F}((v_e + 1e - 6) + w_\sigma ou_t, v_{ext}, v_i, W_i)$ $\mathcal{F} = \frac{1}{2\tau_V} \cdot \text{Erfc}\left(\frac{V_{thre}^{eff} - \mu_V}{\sqrt{2}\sigma_V}\right)$ $V_{thre}^{eff}(\mu_V, \sigma_V, \tau_V^N = \tau_V \frac{g_L}{Cm}) = P'_0 + \sum_{x \in \{\mu_V, \sigma_V, \tau_V^N\}} P_x \cdot \left(\frac{x - x^0}{\delta x^0}\right)$ $+ \sum_{x, y \in \{\mu_V, \sigma_V, \tau_V^N\}^2} P_{xy} \cdot \left(\frac{x - x^0}{\delta x^0}\right) \left(\frac{y - y^0}{\delta y^0}\right)$ $\mu_G(v_e, v_{ext}, v_i) = ((v_e K_e + v_{ext} K_{ext}) \tau_e Q_e) + (v_i K_i \tau_i Q_i) + g_L$ $\mu_{V_s}(v_e, v_{ext}, v_i, w, \mu_G) = \frac{((v_e K_e + v_{ext} K_{ext}) \tau_e Q_e) E_e}{\mu_G} + \frac{(v_i K_i \tau_i Q_i) E_i + g_L E L_s - w}{\mu_G}$ $\sigma_V(\mu_V, \mu_G) = \sqrt{\sum_{s \in \{e, i\}} K_s v_s \frac{\left(\frac{Q_s}{\mu_G} (E_s - \mu_V) \tau_s\right)^2}{2 \frac{Cm}{\mu_G} + \tau_s}}$ $\tau_V(\mu_V, \mu_G) = \frac{\sum_{s \in \{e, i\}} K_s v_s \left(\frac{Q_s}{\mu_G} (E_s - \mu_V) \tau_s\right)^2}{\sum_{s \in \{ex, in\}} K_s v_s \frac{\left(\frac{Q_s}{\mu_G} (E_s - \mu_V) \tau_s\right)^2}{2 \frac{Cm}{\mu_G} + \tau_s}}$

E. Supplementary material of chapter 3 – E.2. Supplementary Table

C5		TVB : Neural Mass Model Parameters(part 1)		
case		Asynchronous	Irregular Synchronize	Regular bursting
T	time resolution of the mean-field		20.0ms	
C_m	Capacity of the membrane		200.0 pF	
E_{L_e}	Leak reversal potential excitatory(E_L)	-64.5 mV	-64.5 mV	-74.0 mV
E_{L_i}	Leak reversal potential inhibitory(E_L)	-65.0 mV	-65.0 mV	-75.0 mV
g_L	Leak conductance		10.0 nS	
a_e	Subthreshold adaptation of excitatory neurons(a)		0.0 nS	
b_e	Spike-triggered adaptation of excitatory neurons(b)	10.0 pA	100.0 pA	50.0 pA
τ_{w_e}	Adaptation time constant of excitatory neurons(τ_w)	500.0 ms	500.0 ms	150.0 ms
a_i	Subthreshold adaptation of inhibitory neurons(a)		0.0 nS	
b_i	Spike-triggered adaptation inhibitory neurons(b)		0.0 pA	
τ_{w_i}	Adaptation time constant of inhibitory neurons(τ_w)		1.0 ms	
E_e	Excitatory reversal potential(E_{ex})		0.0 mV	
τ_e	Rise time of excitatory synaptic conductance(τ_{ex})		5.0 ms	
Q_e	excitatory quantal conductance		1.0 nS	
E_i	Inhibitory reversal potential(E_{in})		-80.0 mV	
τ_i	Rise time of inhibitory synaptic conductance(τ_{in})		5.0 ms	
Q_i	inhibitory quantal conductance	10.0 nS	5.0 nS	10.0 nS
$p_{connect}$	probability of connection	0.05	0.05	0.005
N_{tot}	Number of total neurons		10000	
p_i	percentage of inhibitory neurons		0.2	

E. Supplementary material of chapter 3 – E.2. Supplementary Table

C5		TVB : Neural Mass Model Parameters(part 2)			
N_e	Number of excitatory neurons	$N_{tot}(1 - p_i) = 8000$			
N_i	Number of inhibitory neurons	$N_{tot}p_i = 2000$			
K_e	mean number of input excitatory synapses : $N_e p_{connect}$	400	400	40	
K_i	mean number of input inhibitory synapses : $N_i p_{connect}$	100	100	10	
K_{ext_e}	number of external excitatory synapse	115	300	80	
P_e	second order polynomial of the phenomenological threshold for excitatory neurons in mV	P_0	P_{μ_V}	P_{σ_V}	$P_{\tau_V^N}$
		-0.0498	0.00506	-0.025	0.0014
		$P_{\mu_V^2}$	$P_{\sigma_V^2}$	$P_{(\tau_V^N)^2}$	
		-0.00041	0.0105	-0.036	
P_i	second order polynomial of the phenomenological threshold for inhibitory neurons in mV	P_0	P_{μ_V}	P_{σ_V}	$P_{\tau_V^N}$
		-0.0514	0.004	-0.0083	0.0002
		$P_{\mu_V^2}$	$P_{\sigma_V^2}$	$P_{(\tau_V^N)^2}$	
		-0.0005	0.0014	-0.014	
v_{ext}	external input	$P_{\mu_V \sigma_V}$	$P_{\mu_V \tau_V^N}$	$P_{\sigma_V \tau_V^N}$	
		0.0074	-0.0012	-0.0407	
		P_0	P_{μ_V}	P_{σ_V}	$P_{\tau_V^N}$
		-0.0045	0.0028	-0.00153	
w_σ	weight of the noise	see coupling section			
σ	variation of the noise	0.0002	0.0006	0.002	
μ	mean of the noise	0.2			
τ_{ou}	mean of the noise	0.0			
	initial condition (random between maximum and minimum)	20.0			
		$\mu_E(kHz) : (0., 0.)$	$\mu_i(kHz) : (0., 0.)$		
		$c_{ee} : (0., 0.)$	$c_{ei} : (0., 0.)$		
		$c_{ii} : (0., 0.)$			
		$W_e(pA) : (0., 5.)$	$W_i(pA) : (0., 0.)$		

E. Supplementary material of chapter 3 – E.2. Supplementary Table

C6		TVB : Monitor																																																						
state variable	proxy node	node																																																						
	Only the mean firing rate of the excitatory population because it is the coupling variable and it is extracted from NEST simulation	mean firing rate of excitatory and inhibitory population, the variation of excitatory and inhibitory firing rate, the co-variation between excitatory and inhibitory firing rate, mean adaptive current of excitatory and inhibitory firing rate																																																						
	precision	dt (0.1ms)																																																						
SEEG	equation	$\Psi_{ECOG}(channel, t) = P \cdot v_e + noise$ where P is the gain matrix and N is the mean firing rate of excitatory population $P_{ij} = scaling_factor * region_volume_j / r_i - r_j $ where r_i is the position of the contact point of the channel and r_j is the centre of region j.																																																						
	contact position	<table border="1"> <thead> <tr> <th></th> <th>x</th> <th>y</th> <th>z</th> </tr> </thead> <tbody> <tr> <td rowspan="8">left hemisphere</td> <td>40.0</td> <td>80.0</td> <td>79.5</td> </tr> <tr> <td>20.0</td> <td>80.0</td> <td>72.0</td> </tr> <tr> <td>30.0</td> <td>70.0</td> <td>76.5</td> </tr> <tr> <td>30.0</td> <td>90.0</td> <td>75.5</td> </tr> <tr> <td>22.5</td> <td>72.5</td> <td>73.0</td> </tr> <tr> <td>22.5</td> <td>87.5</td> <td>72.5</td> </tr> <tr> <td>37.5</td> <td>72.5</td> <td>78.5</td> </tr> <tr> <td>37.5</td> <td>87.5</td> <td>78.5</td> </tr> <tr> <td rowspan="8">right hemisphere</td> <td>94.0</td> <td>80.0</td> <td>69.</td> </tr> <tr> <td>74.0</td> <td>80.0</td> <td>78.5</td> </tr> <tr> <td>84.0</td> <td>70.0</td> <td>74.5</td> </tr> <tr> <td>84.0</td> <td>90.0</td> <td>74.0</td> </tr> <tr> <td>76.5</td> <td>72.5</td> <td>77.5</td> </tr> <tr> <td>76.5</td> <td>87.5</td> <td>77.5</td> </tr> <tr> <td>91.5</td> <td>72.5</td> <td>70.5</td> </tr> <tr> <td>91.5</td> <td>87.5</td> <td>70.</td> </tr> </tbody> </table>		x	y	z	left hemisphere	40.0	80.0	79.5	20.0	80.0	72.0	30.0	70.0	76.5	30.0	90.0	75.5	22.5	72.5	73.0	22.5	87.5	72.5	37.5	72.5	78.5	37.5	87.5	78.5	right hemisphere	94.0	80.0	69.	74.0	80.0	78.5	84.0	70.0	74.5	84.0	90.0	74.0	76.5	72.5	77.5	76.5	87.5	77.5	91.5	72.5	70.5	91.5	87.5	70.
		x	y	z																																																				
	left hemisphere	40.0	80.0	79.5																																																				
		20.0	80.0	72.0																																																				
30.0		70.0	76.5																																																					
30.0		90.0	75.5																																																					
22.5		72.5	73.0																																																					
22.5		87.5	72.5																																																					
37.5		72.5	78.5																																																					
37.5		87.5	78.5																																																					
right hemisphere	94.0	80.0	69.																																																					
	74.0	80.0	78.5																																																					
	84.0	70.0	74.5																																																					
	84.0	90.0	74.0																																																					
	76.5	72.5	77.5																																																					
	76.5	87.5	77.5																																																					
	91.5	72.5	70.5																																																					
	91.5	87.5	70.																																																					
<i>scaling_factor</i>	1.0																																																							
<i>region_volume</i>	The volume is extracted from the volume mapping of Allen Mouse Brain Connectivity Atlas mean : 3712.29 max : 16245.0 min : 957.0																																																							

E. Supplementary material of chapter 3 – E.2. Supplementary Table

D Transformation NEST to TVB : model	
Name	SMFR : sliding mean firing rate
Type	sliding mean over the histogram of the instantaneous firing rate
Input	spike trains of excitatory neurons from one brain region for synchronized time (2ms)
Output	mean firing rate of the excitatory population of a brain region for synchronized time (2ms)
equation	$\forall t \geq T, SMFR(t) = \frac{\sum_{s=t-T}^t \sum_{n=1}^{N_e} spike(n, s)}{N_e T} *$ <p>10³ (KHz)</p> <p>where $spike(n, s) =$</p> $\begin{cases} 1 & \text{if neuron } n \text{ create spike at time } s \\ & \text{with a precision of } dt \\ 0 & \text{else} \end{cases}$
parameters	size of the windows T : 20.0 ms (same as TVB)
parameters	number of neurons N_e : 8000 (same as NEST)
parameters	precision of the integration dt : 0.1 ms (same as TVB and NEST)
initialization	The transfer module does not have initialized because TVB used its initialization for starting the communication.

E. Supplementary material of chapter 3 – E.2. Supplementary Table

E		Transformation TVB to NEST : model
	Name	MIP
	Type	Multiple Interaction Process [209]
	Input	incoming excitatory firing rate of a brain region for synchronized time (2ms)
	Output	spike trains to individual neurons with a correlation of p for synchronized time (2ms)
	equation	reference spike train : $x_{ref}(t) = InhomogenousPoissonProcess((v_{input}(t)nb_{synapse} + 1e - 12)/p)$ input individual spike train to the neuron n : $x_n(t) = x_{ref}B(size(x_{ref}), p)$
where	v_{input}	mean external excitatory firing rate computed by TVB for the NEST population.
	$nb_{synapse}$	number of external input synapse (A :115, IS :300, RB :80)
	p	percentage of shared neurons (A :0.01, IS :0.1, RB :0.01)
	B	binomial law
	The implementation of the Inhomogeneous Poisson Process	Dedicated function from the python library elephant (version 0.9), which : 1) generates spike trains with homogeneous Poisson generator for the highest rate ; 2) removes some spikes for having variation of rate based on the input rates. The homogeneous Poisson generator computes the time interval between each spike using the exponential random generator of numpy.
	initialization	The initial rate sent to TVB are zeros during the first t_{synch} (2ms).

TABLEAU E.1. – Parameter for co-simulation

E.3. Supplementary Figures

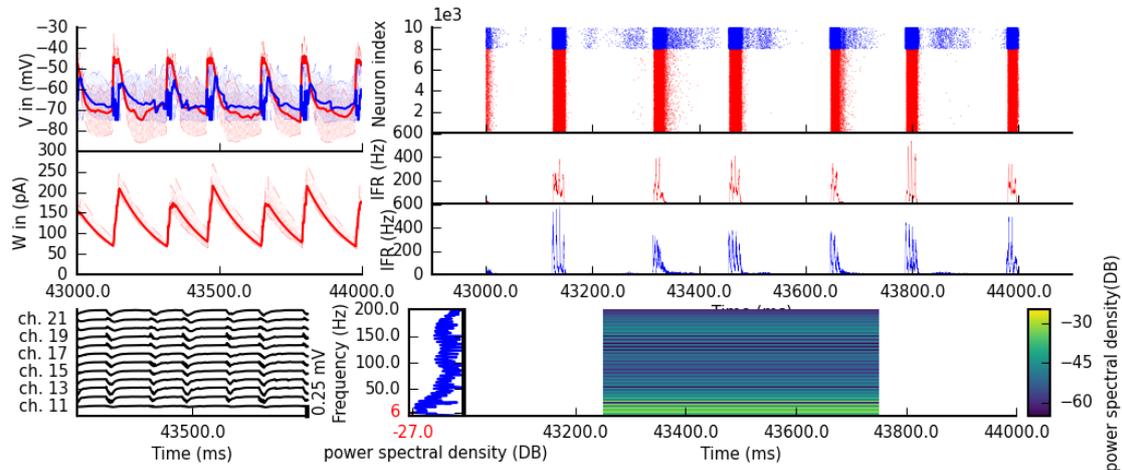


FIGURE E.1. – Zoom on 1s for the spiking neural network with regular bursting state. This figure is a zoom of the figure 3.3 between 43s and 44s. **top-left** Example of time series from 10 adaptive exponential leaky and integrator neurons. The red lines are excitatory and the blue curve are inhibitory neurons. The mean excitatory time series is shown with a thick red line and the inhibitory time series is shown with a thick blue line. **middle-left** The adaptation currents of 10 neurons are shown. The thick line is the mean adaptive currents. **bottom-left** The figure shows local field potential from the 12 sites of in the middle line of the polytrode. The local field potential is computed from the spike trains of all neurons by the software HybridLFPY [148]. **top-right** The figure shows spike trains of 10000 neurons for 1s. **bottom-right** The figure shows respectively the excitatory and inhibitory instantaneous firing rate of the population in panel middle-right in red and blue. **bottom** Spectrogram and power spectrum example of the instantaneous firing rate for 1s.

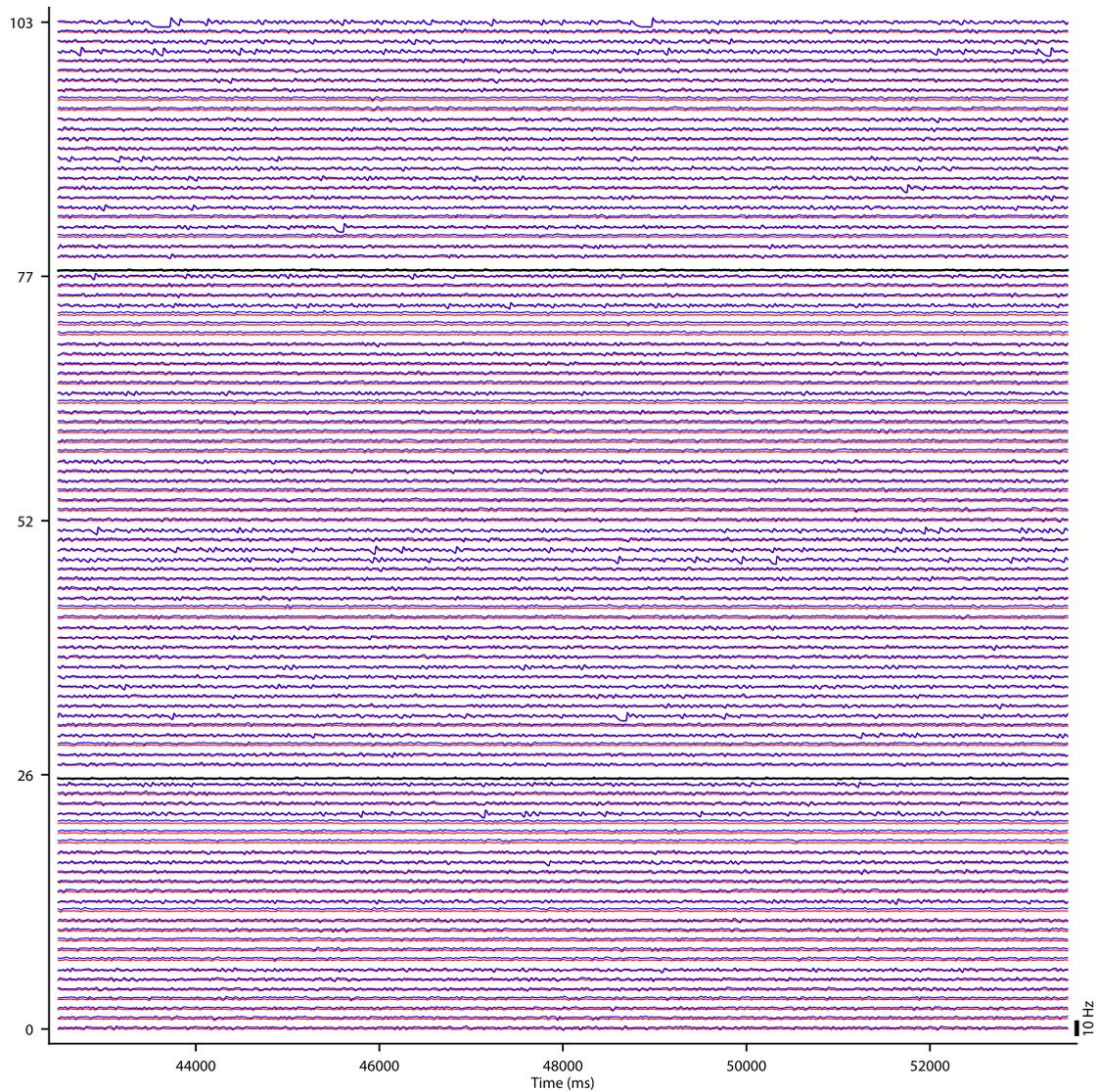


FIGURE E.2. – Mouse Brain activity for Asynchronous Irregular
For the state of the Asynchronous Irregular, an overview of the mean firing rates of excitatory, in red, and inhibitory, in blue, populations from the model of Mean Adaptive Exponential for all mouse brain regions. The two black curves are the mean firing rate of the two populations of excitatory neurons simulated with NEST [152].

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

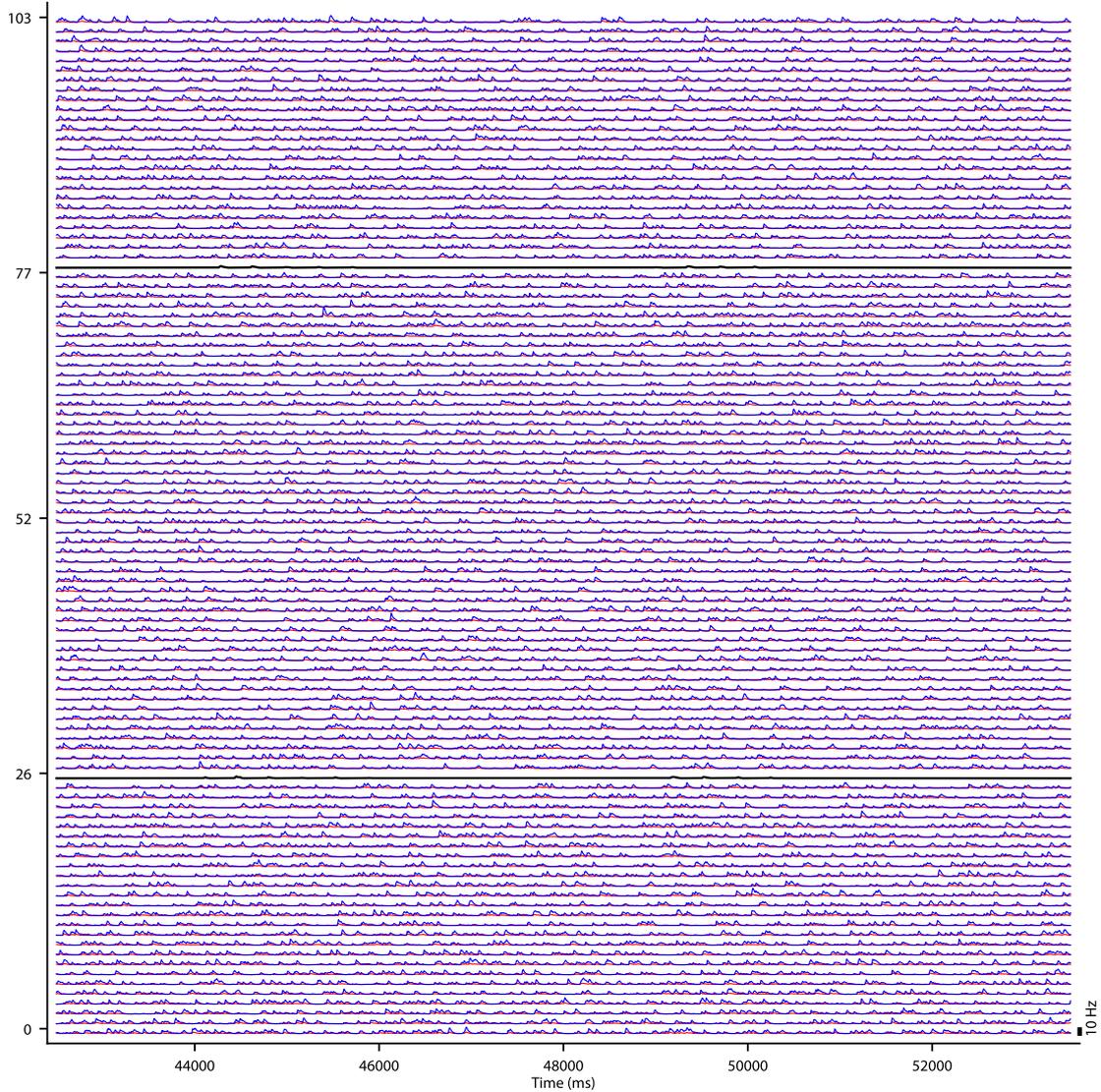


FIGURE E.3. – Mouse Brain activity for Synchronize Irregular

For the state of the Synchronize Irregular, an overview of the mean firing rates of excitatory, in red, and inhibitory, in blue, populations from the model of Mean Adaptive Exponential for all mouse brain regions. The two black curves are the mean firing rate of the two populations of excitatory neurons simulated with NEST [152].

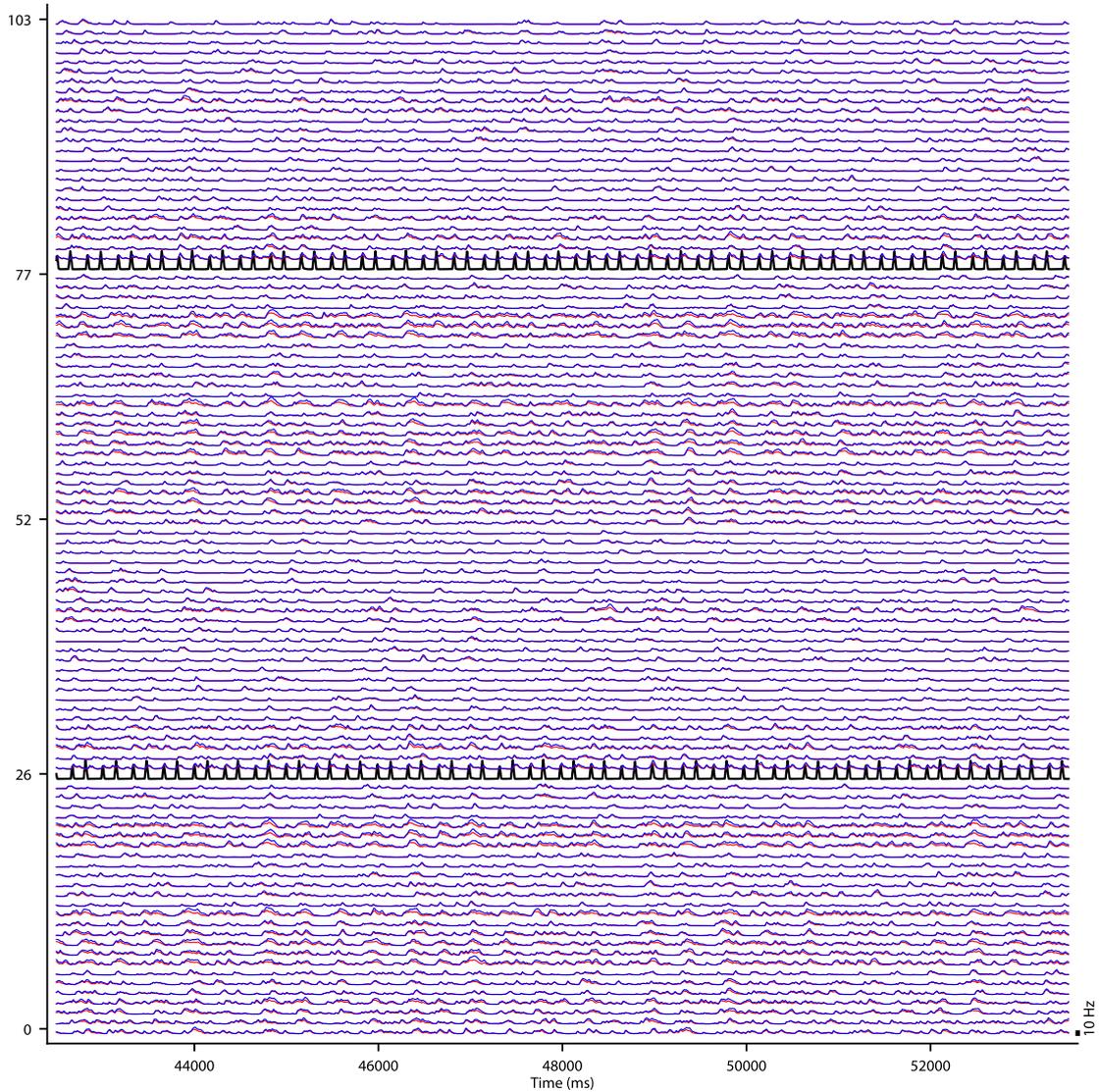


FIGURE E.4. – Mouse Brain activity for Regular Bursting

For the state of the Regular Bursting, an overview of the mean firing rates of excitatory, in red, and inhibitory, in blue, populations from the model of Mean Adaptive Exponential for all mouse brain regions. The two black curves are the mean firing rate of the two populations of excitatory neurons simulated with NEST [152].

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

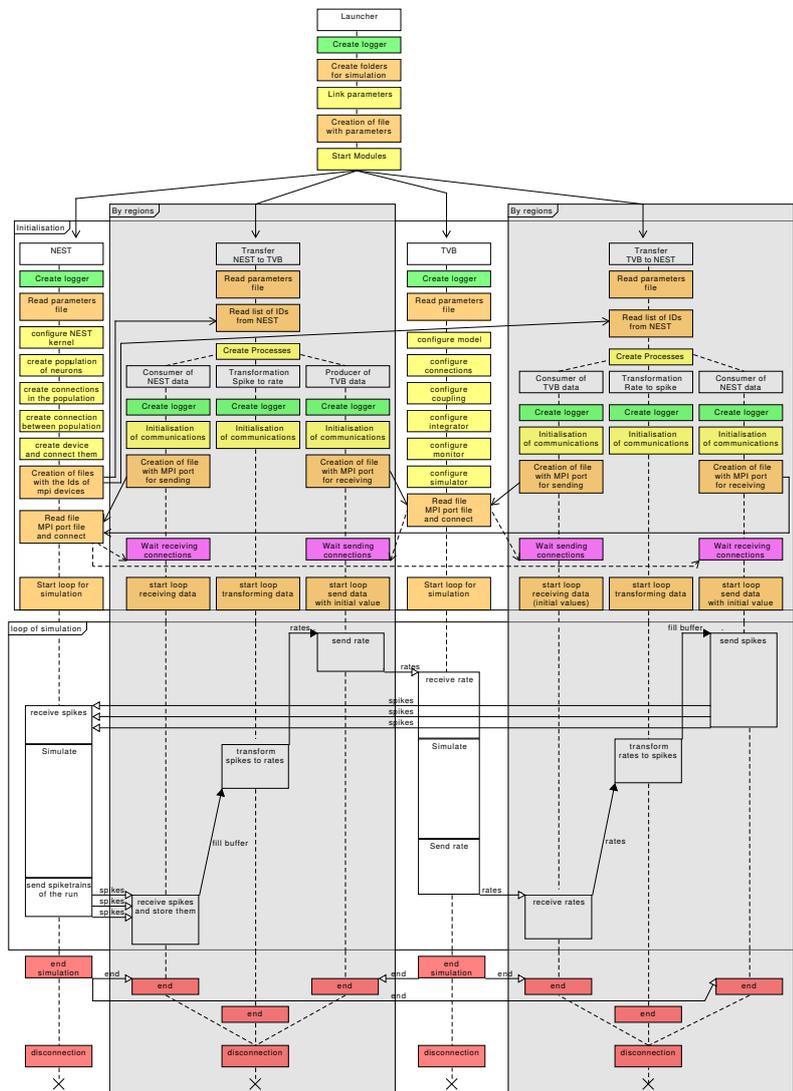


FIGURE E.5. – Detail sequence diagram of the co-simulation

This figure represents the interaction among the different modules during the co-simulation and the different exchanged data. The co-simulation is separated in 3 steps : initialisation and configuration, simulation and termination.

The colour code of the boxes :

- green for the creation of a logger, one by components and modules
- orange for access to file systems (the creation of a folder or a file, the reading of files, ...) and the start of the simulation with initial condition.
- yellow for initialization and configuration of modules and components
- magenta for MPI waiting connections
- white for the simulation step and the name of modules or components
- red for the termination of the simulation.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

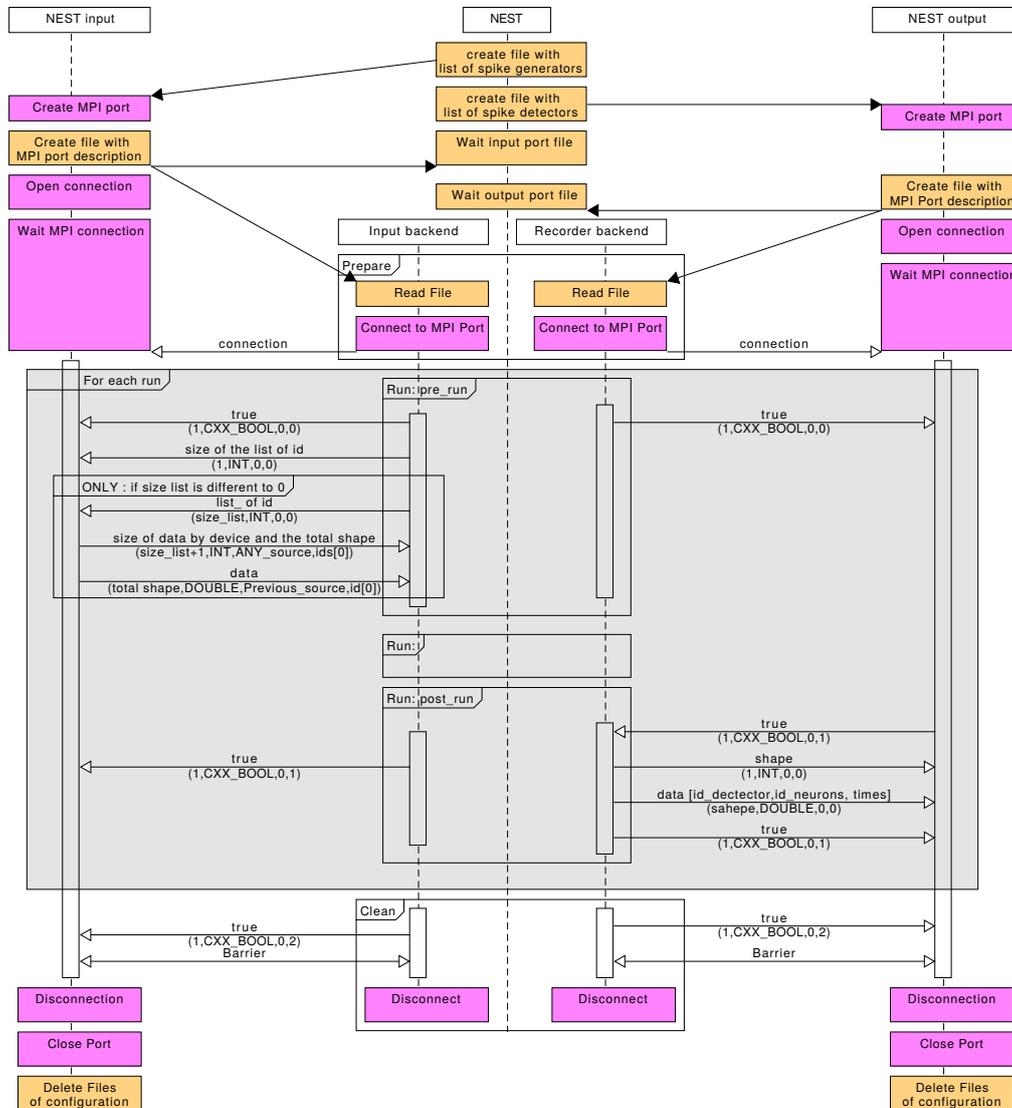


FIGURE E.6. – Sequence diagram of the communication protocol with NEST
 The communication with NEST [152] is separated in 3 steps : creation of the MPI connection, simulation and termination.

The colour code of the boxes :

- orange for access to file systems (the creation of files, the reading of files, ...).
- magenta for management of MPI port
- white for the name of the modules or components

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

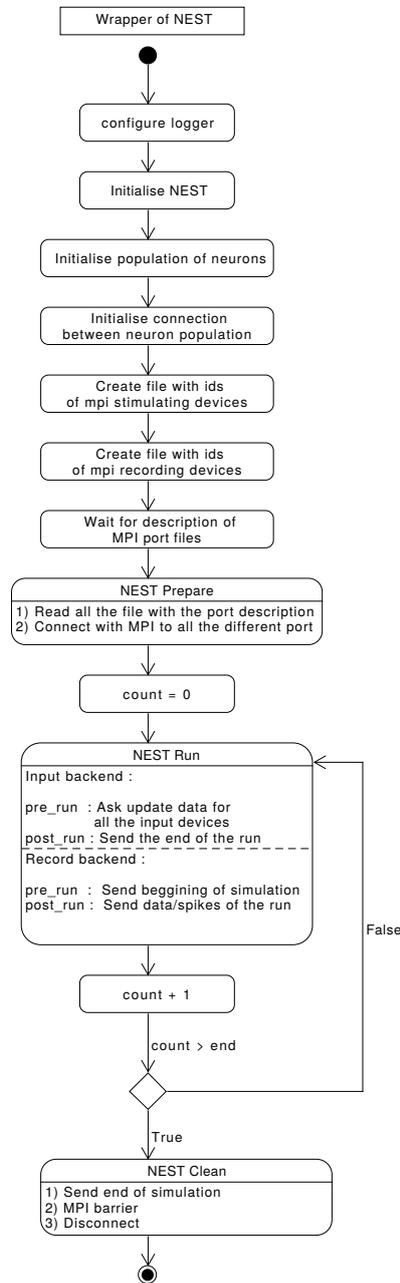


FIGURE E.7. – State diagram of NEST wrapper

The diagram describes all the different states of the NEST wrapper during the co-simulation. The beginning is the set-up of the network (the creation of neurons, their connection and the creation of devices). After, the loop of simulation and the termination.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

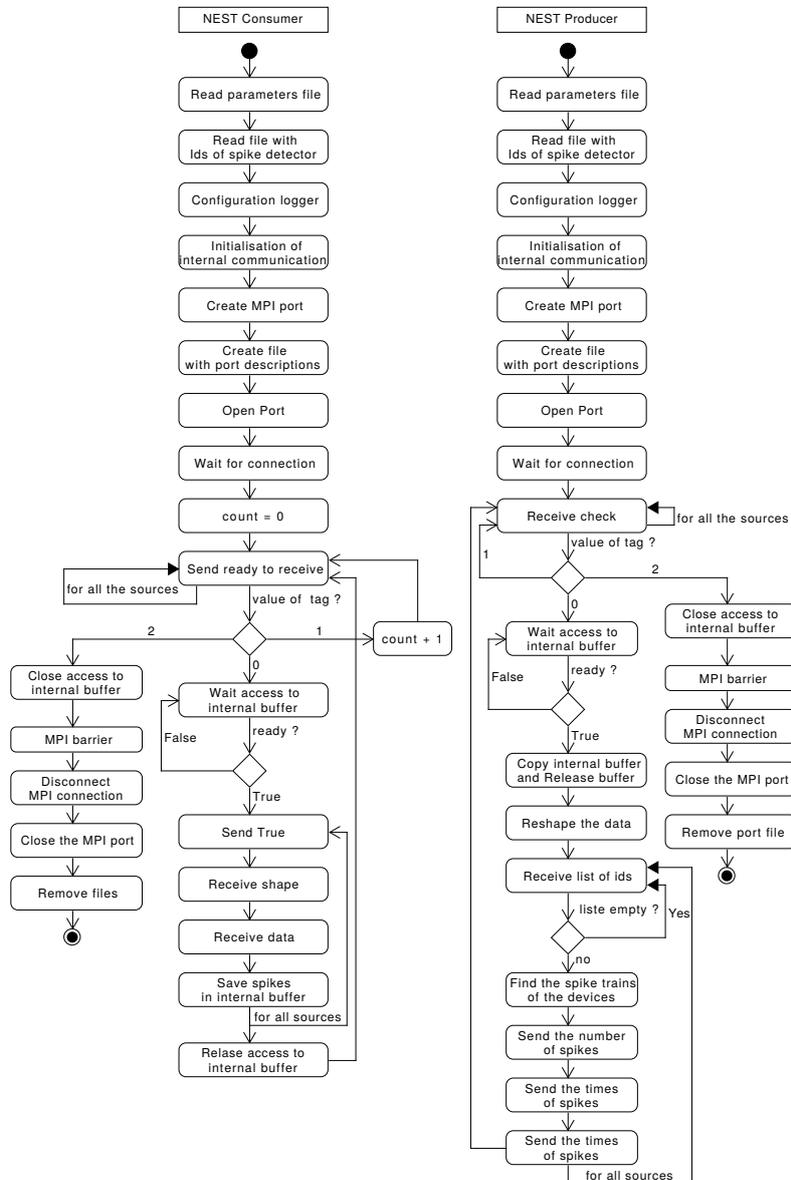


FIGURE E.8. – State diagram of transfer components for interaction with NEST
 The diagram describes all the state of the component of the transfer module which communicates with NEST. The beginning is the configuration of itself and the creation of the MPI connection. Once the MPI connection is made, there is the loop of the simulation. The centre of the simulation loop is the value of the tag receive by the component to identify if NEST is ready to receive or send messages. If this tag equals 2, the components go in the sequence for the termination phase.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

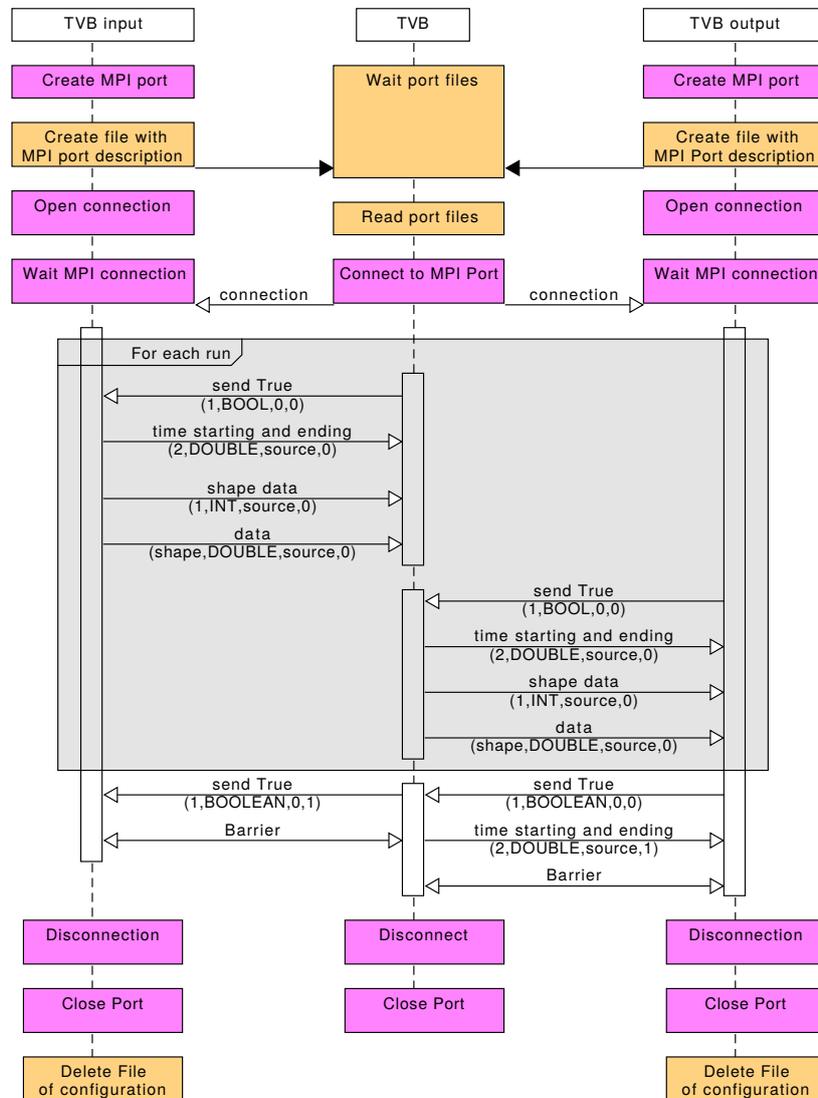


FIGURE E.9. – Sequence diagram of the communication protocol

The communication with TVB [299] is separated into 3 steps : creation of the MPI connection, simulation and termination.

The colour code of the boxes :

- orange for access to file systems (create files, read files, ...).
- magenta for management of MPI port
- white for the name of the modules or components

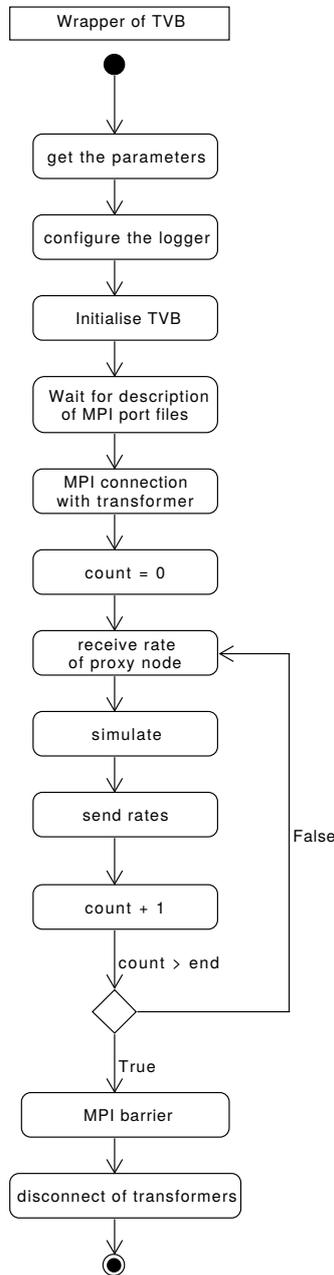


FIGURE E.10. – State diagram of wrapper of TVB modules

The diagram describe all the different state of the TVB wrapper during the co-simulation. The beginning is the set-up of the network (the creation of neurons, their connection and the creation of devices). After, the loop of simulation and the termination.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

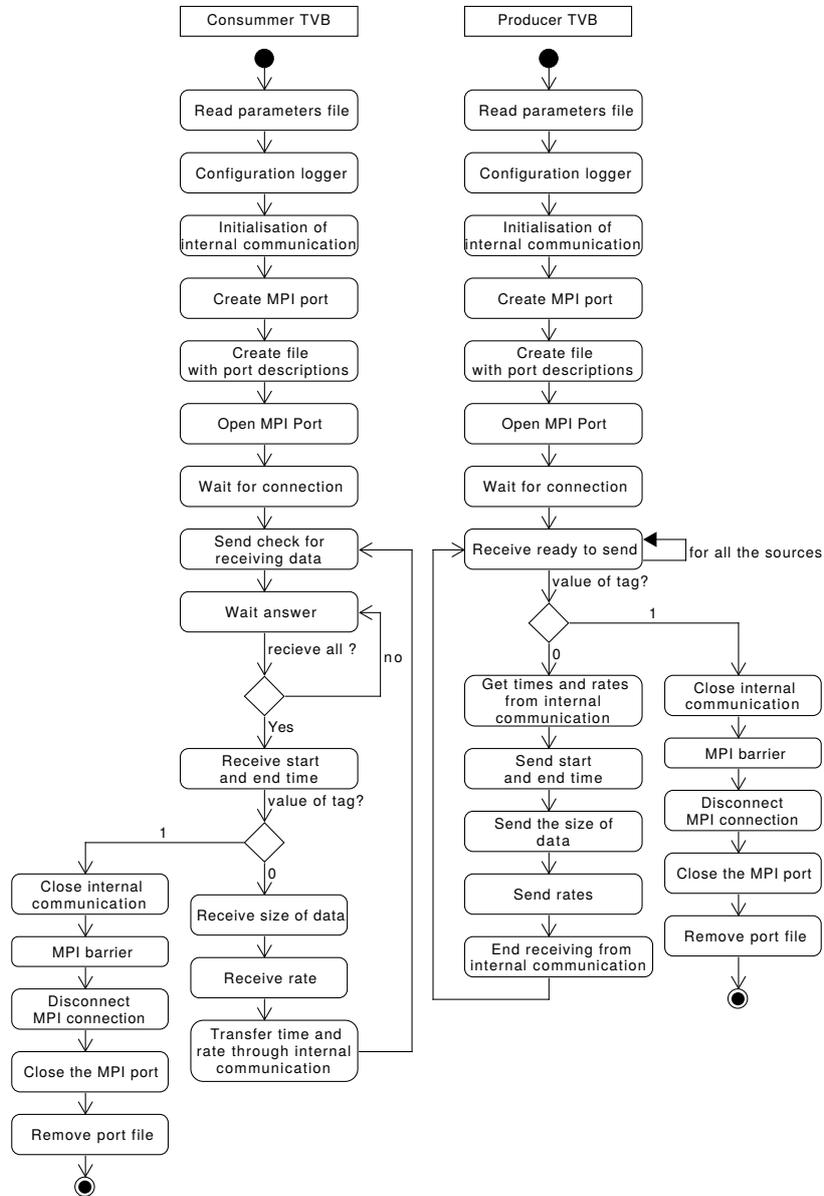


FIGURE E.11. – State diagram of transfer components

The diagram describes all the state of the component of the transfer module which communicates with TVB. The beginning is the configuration of itself and the creation of the MPI connection. Once the MPI connection is made, there is the loop of the simulation. The centre of the simulation loop is the value of the tag receive by the component to identify if NEST is ready to receive or send messages. If this tag equals 1, the components go in the sequence for the termination phase.

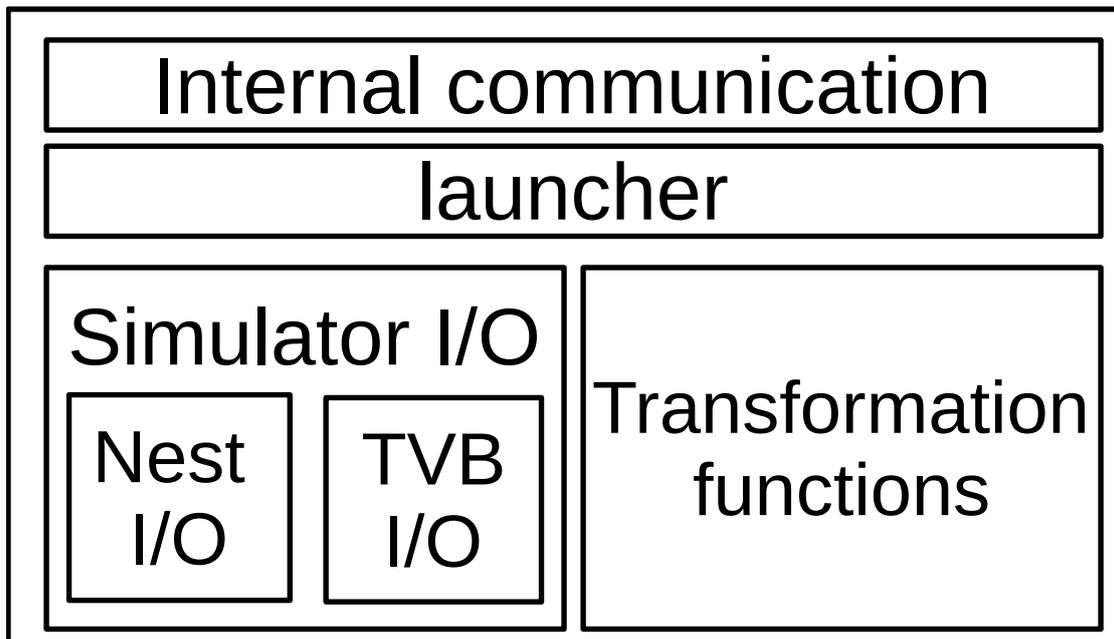


FIGURE E.12. – File organisation of the transformer module

The file of transfer modules are organised following the modularity of the module. The folder internal communication contains all the functions of the communication between components of the modules. The launcher regroups the files used to start the modules of transformation between specific simulators. The Simulator I/O contains the function for the interface of each simulator. The figure shows that interface for each simulator (TVB and NEST) is separated and independent. Transformation functions is a folder which contains the abstract class for the transformations and the implementation of specific transformation.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

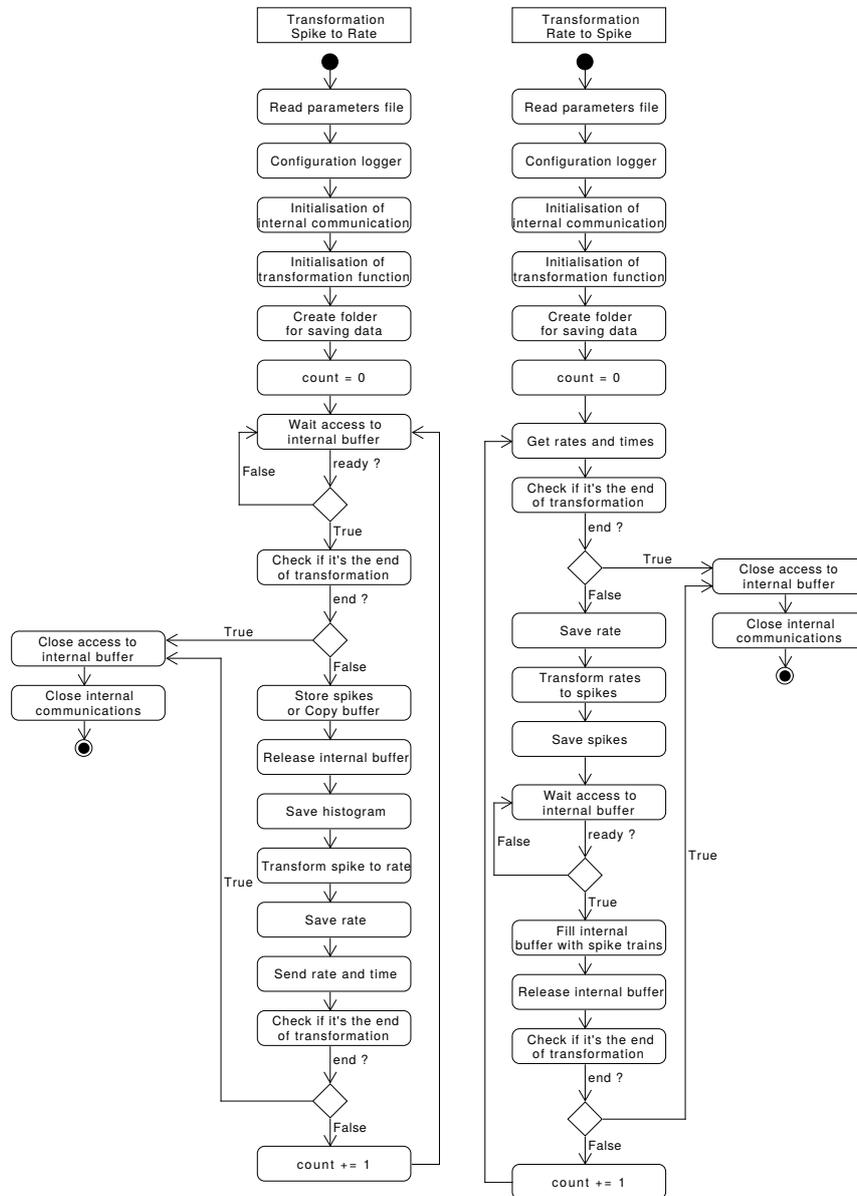


FIGURE E.13. – State diagram of transfer components which transform data from one scale to another

The diagram describes all the state of the transfer components. The beginning is the configuration of itself. The component is waiting to access to the data from one buffer for the transformation. After accessing to the data, it transforms them and awaits the access for writing in a second buffer. When it receives the termination from one side or the other, it goes in the sequence of termination.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

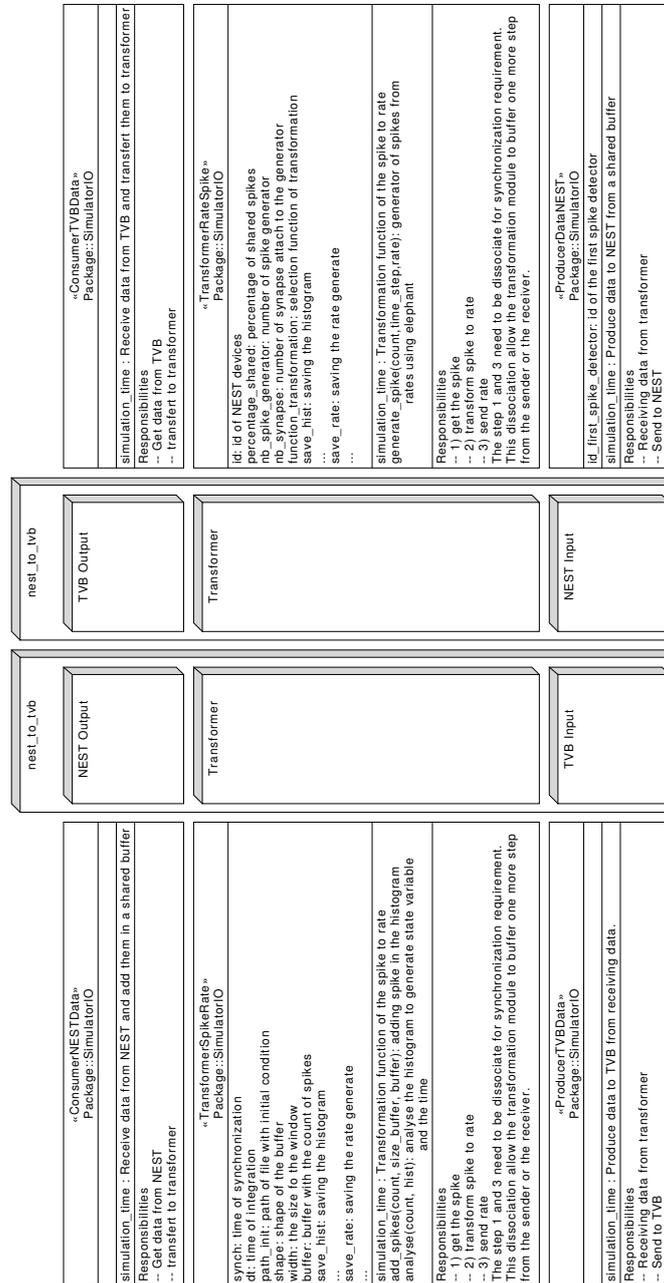


FIGURE E.14. – Structure diagram of the two transfer modules with the description of each component
Each component is based on a class. The diagram gives a short description of each class and their contents.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

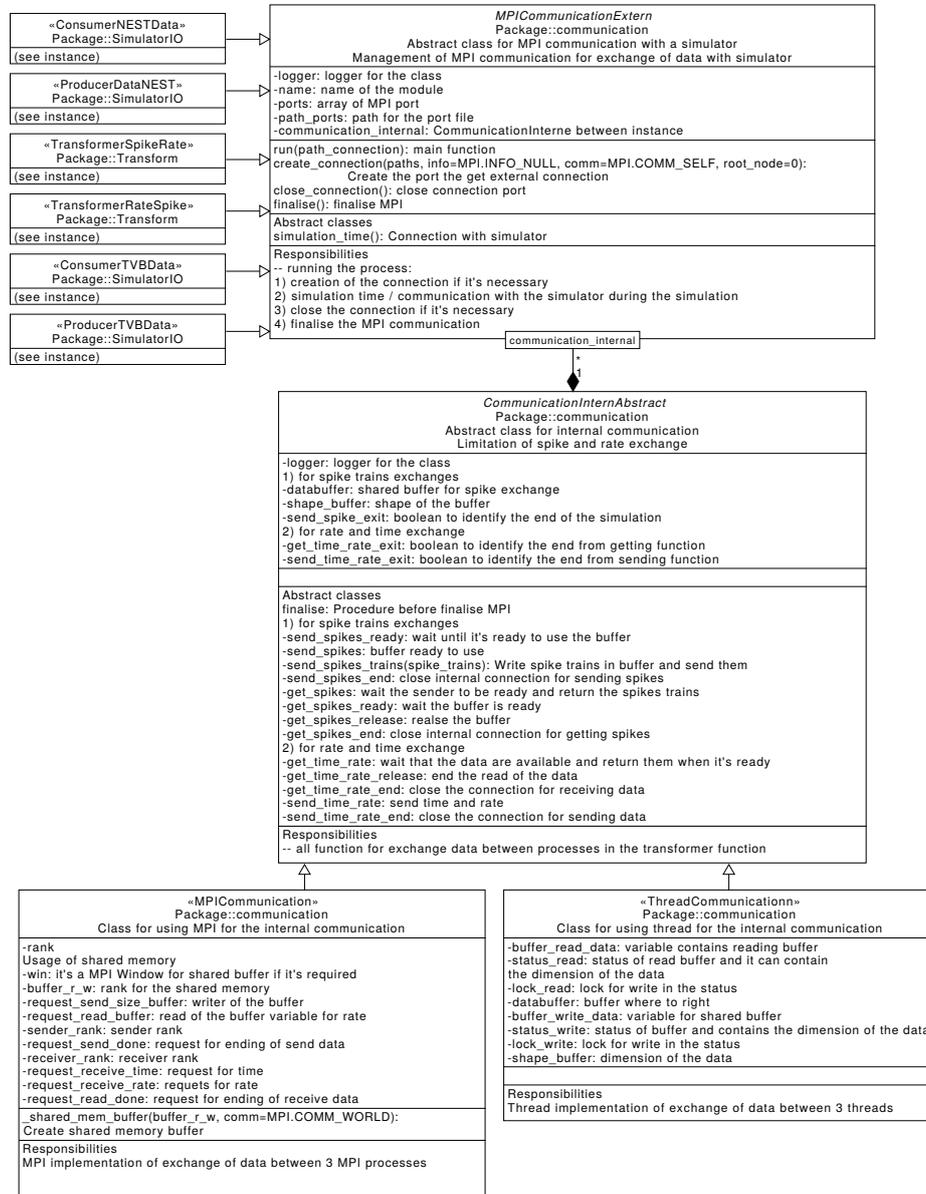
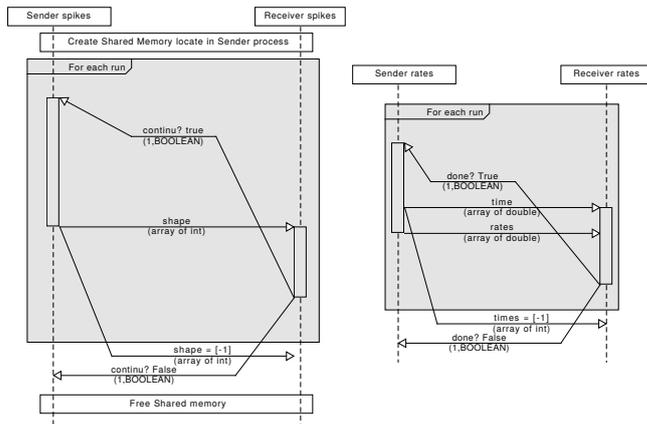


FIGURE E.15. – Class diagram of the transfer modules.

The description of the class on the right part is described in the supplementary figure E.14. All this class inherit from an abstract class which manage MPI communication. This abstract class manage the MPI connection and has an internal communicator. This internal communicator is an abstract class for the communication between the transfer components. This internal communication can be melting process or multithreading as the diagram shows.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

a



b

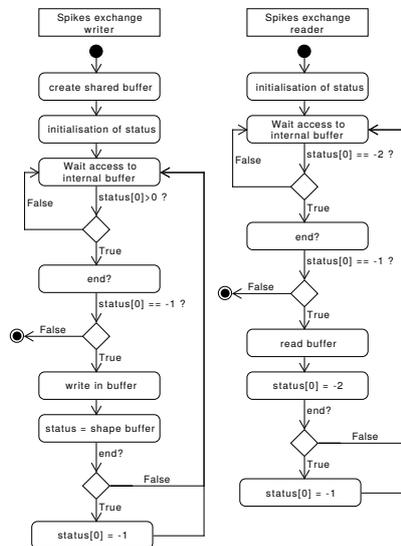


FIGURE E.16. – Communication between components in the transfer module
As described in the supplementary figure E.15, the internal communication has 2 im-
plementations. The panel a is a sequence diagram of the communication of spikes and
rates using MPI communication. The panel b is a state diagram of the management of
a shared buffer in the case of multithreading communication for transfer spike data.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

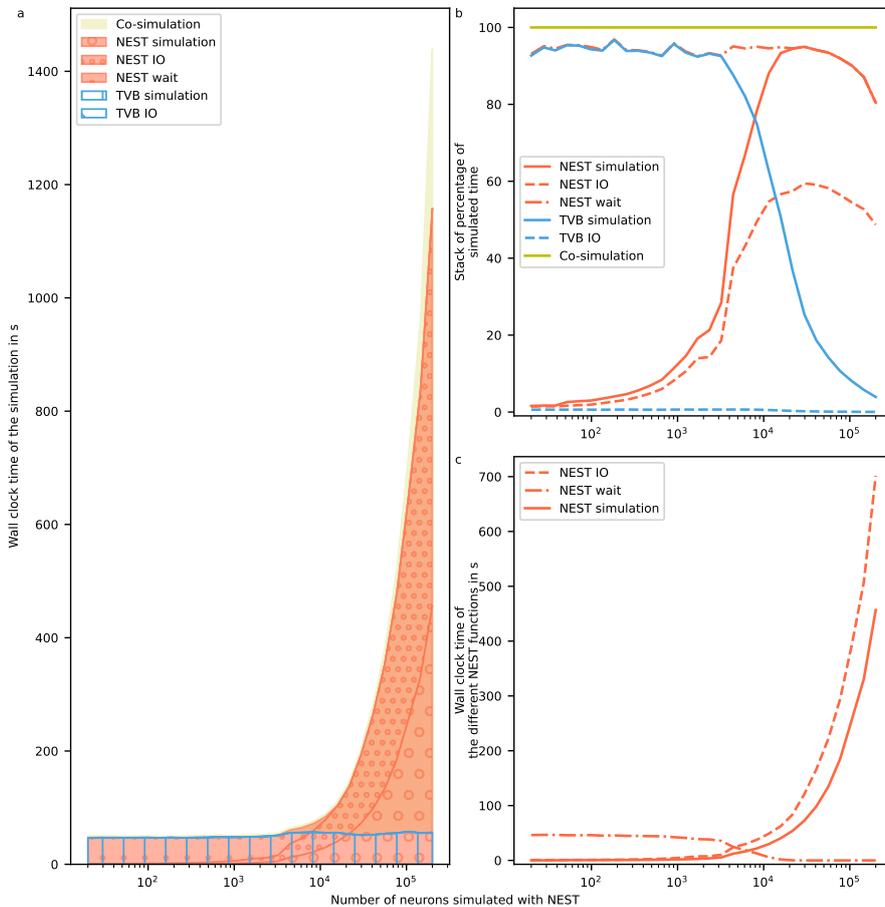


FIGURE E.17. – Details of the performance with the increase of neurons
 Performance is obtained for 1 second of simulated time on a computer (see Materials and Methods for more details). The reference implementation use 1 MPI process, 6 virtual processes/threads, 2.0 ms to synchronize time between simulator for the simulation of 20000 neurons. Simulation time depending on the number of neurons simulated with NEST. **a** The wall clock time of the simulator depending on the number of neurons. The total time of the co-simulation is represented in yellow. The "simulation", "IO" and "wait" times of NEST are represented in red surface with respectively hatches with big circles, small circles and points. The "simulation" and "IO" times of TVB are represented in the blue surface with respectively hatches horizontal lines and oblique lines. **b** The wall clock time for the co-simulation (yellow curve), NEST (red curves) and TVB (blue curves) by the total wall clock time. The solid, dashed and dashed dotted curves are associated with "simulation", "IO" and "wait" time of NEST. The solid and dashed line is associate to "simulation" and "IO" time of TVB. **c** The different timer for NEST simulator. Each contribution is reported as red curve and for increasing numbers of neurons. The solid, dashed and dashed dotted curves represent "simulation", "IO" and "wait" time of NEST respectively.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

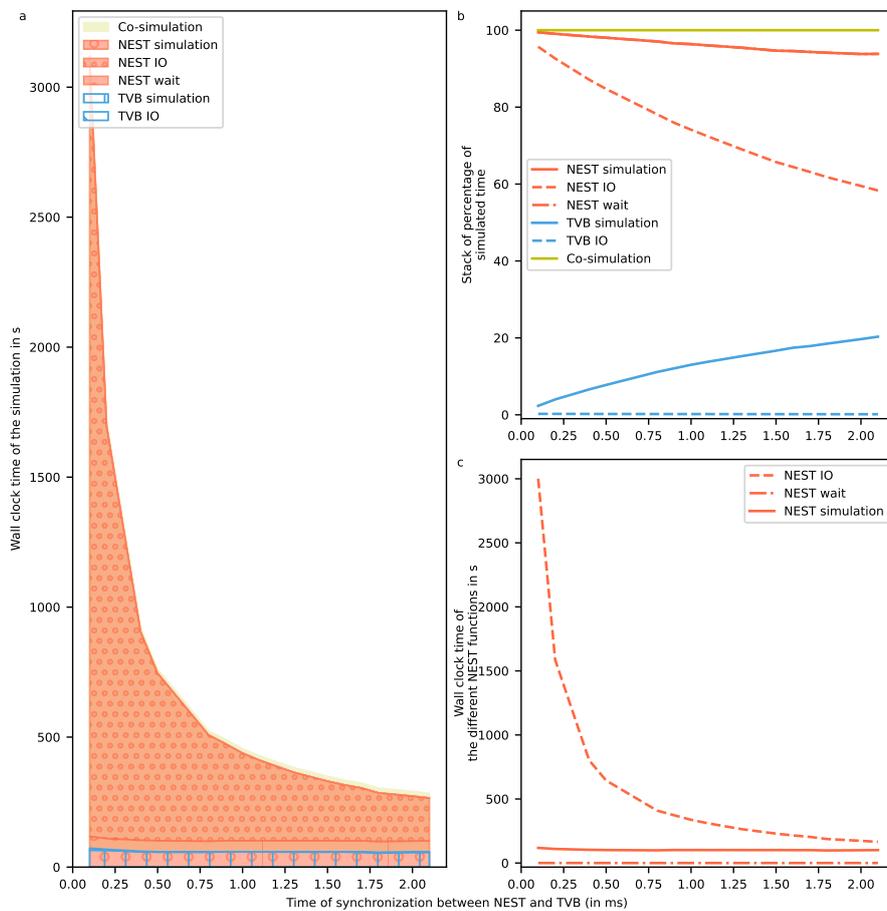


FIGURE E.18. – Details of the performance with the increase of synchronize time Performance is obtained for 1 second of simulated time on a computer (see Materials and Methods for more details). The reference implementation use 1 MPI process, 6 virtual processes/threads, 2.0 ms to synchronize time between simulator for the simulation of 20000 neurons. Simulation time depending on the synchronized time between NEST and TVB. **a** The wall clock time of the simulator depending on time of synchronization between the two simulators. The total time of the co-simulation is represented in yellow. The "simulation", "IO" and "wait" times of NEST are represented in red surface with respectively hatches with big circles, small circles and points. The "simulation" and "IO" times of TVB are represented in the blue surface with respectively hatches horizontal lines and oblique lines. **b** The wall clock time for the co-simulation (yellow curve), NEST (red curves) and TVB (blue curves) by the total wall clock time. The solid, dashed and dashed dotted curves are associated with "simulation", "IO" and "wait" time of NEST. The solid and dashed line is associate to "simulation" and "IO" time of TVB. **c** The different timer for NEST simulator. Each contribution is reported as red curve and for increasing numbers of neurons. The solid, dashed and dashed dotted curves represent "simulation", "IO" and "wait" time of NEST respectively.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

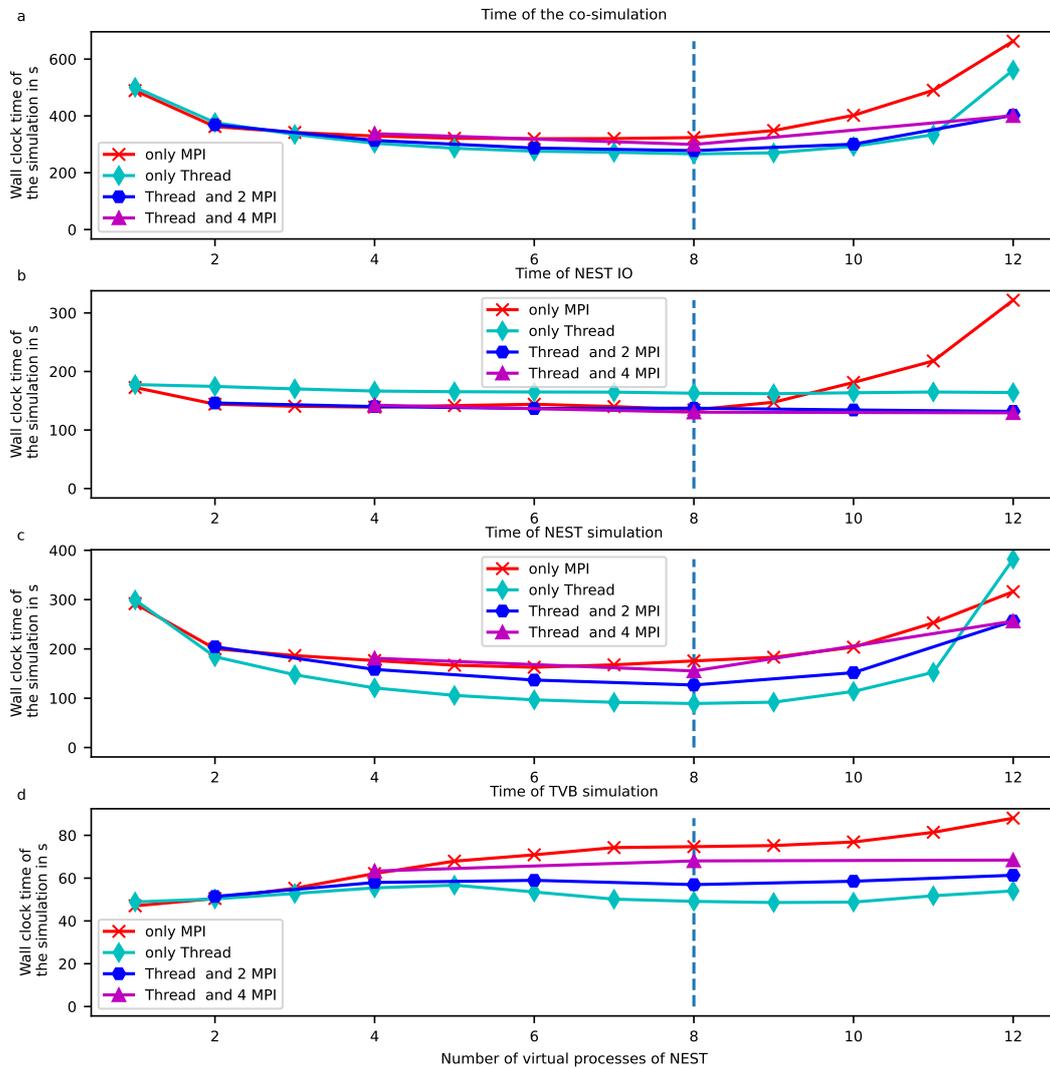


FIGURE E.19. – Details of the performance depending of the number of process and thread for NEST

Performance is obtained for 1 second of simulated time on a computer (see Materials and Methods for more details). The reference implementation use 1 MPI process, 6 virtual processes/threads, 2.0 ms to synchronize time between simulator for the simulation of 20000 neurons. Simulation time depending on the number of virtual process used by NEST. The green, blue, purple, red curves are associated with different parallelization strategy of NEST, respectively, only multithreading, 2 MPI processes with threads, 4 MPI processes with thread and only MPI processes. The horizontal green line represents the number of cores of the computer **a** The total time of the co-simulation. **b** The "IO" time of NEST **c** The "simulation" time of NEST **d** The "simulation" time of TVB

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

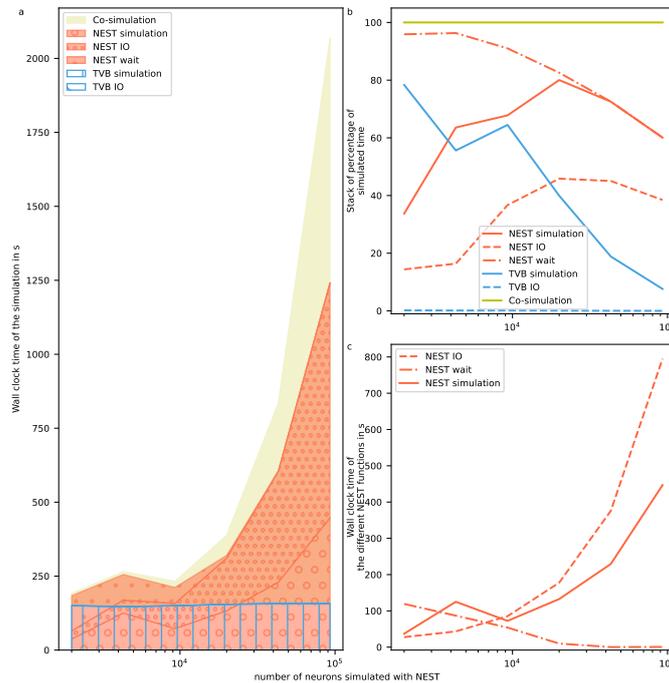


FIGURE E.20. – Performance of the co-simulation on one supercomputer
 Performance is obtained for 1 second of simulated time on a computer. The reference implementation use 1 MPI process, 6 virtual processes/threads, 2.0 ms to synchronize time between simulator for the simulation of 20000 neurons. The node of Jusuf, the supercomputer, content 2 AMD EPYC 7742 @ 2.25 GHz * 64 cores * 2 threads, 256 (16x16) GB DDR4 with 3200 MHz, connected by InfiniBand HDR100 (Connect-X6). The transfer modules and TVB has on one node and NEST on one or multiple other nodes. Simulation time depending on the number of neurons simulated with NEST. **a** The wall clock time of the simulator depending on the number of neurons. The total time of the co-simulation is represented in yellow. The "simulation", "IO" and "wait" times of NEST are represented in the red surface with respectively hatches with big circles, small circles and points. The "simulation" and "IO" times of TVB are represented in the blue surface with respectively hatches horizontal lines and oblique lines. The "simulation" time for TVB is constant. The sum of "simulation" and "IO" time of NEST is higher than the TVB "simulation". **b** The wall clock time for the co-simulation (yellow curve), NEST (red curves) and TVB (blue curves) by the total wall clock time. The solid, dashed and dashed dotted curves are associated with "simulation", "IO" and "wait" time of NEST. The solid and dashed line is associate to "simulation" and "IO" time of TVB. The initialisation and configuration time increase with the number of neurons. **c** The contribution of NEST module to the total amount of the wall clock time normalizes between 0 and 100. Each contribution is reported as red curve and for increasing numbers of neurons. The solid, dashed and dashed dotted curves represent "simulation", "IO" and "wait" time of NEST respectively. The "IO" time of NEST increases exponentially with the number of neurons and is higher than the "simulation time when the number of neurons is higher than $6 \cdot 10^4$ of neurons.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

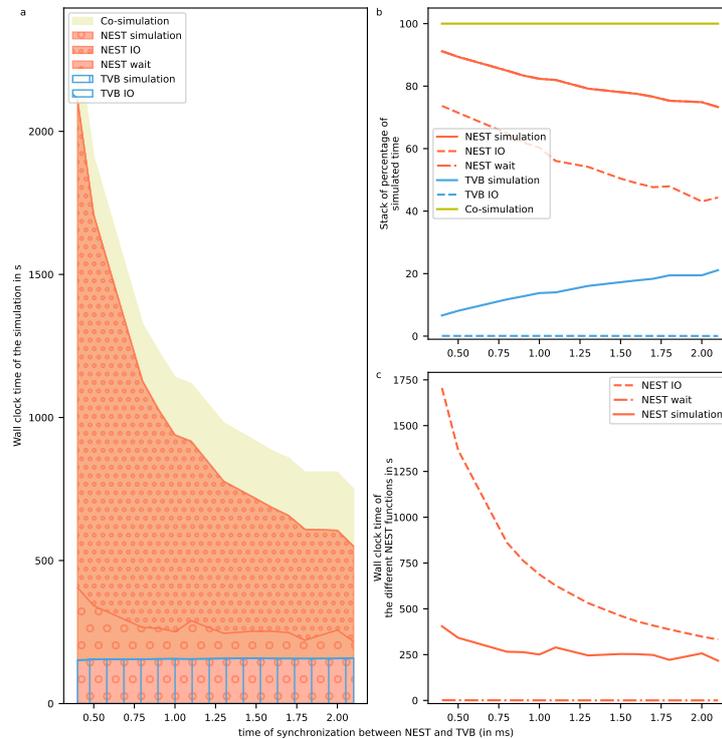


FIGURE E.21. – Performance of the co-simulation on one supercomputer
 Performance is obtained for 1 second of simulated time on a computer. The reference implementation use 1 MPI process, 6 virtual processes/threads, 2.0 ms to synchronize time between simulator for the simulation of 20000 neurons. The node of Jusuf, the supercomputer, content 2 AMD EPYC 7742 @ 2.25 GHz * 64 cores * 2 threads, 256 (16x16) GB DDR4 with 3200 MHz, connected by InfiniBand HDR100 (Connect-X6). The transfer modules and TVB has on one node and NEST on one or multiple other nodes. Simulation time depending on the synchronized time between simulator. **a** The wall clock time of the simulator. The simulation time reduces with the increase of the synchronization time between simulators. This reduction is due to the reduction of NEST "IO" time. The total time of the co-simulation is represented in yellow. The "simulation", "IO" and "wait" times of NEST are represented in the red surface with respectively hatches with big circles, small circles and points. The "simulation" and "IO" times of TVB are represented in the blue surface with respectively hatches horizontal lines and oblique lines. The "simulation" time for TVB is constant. The sum of "simulation" and "IO" time of NEST is higher than the TVB "simulation". **b** The wall clock time for different co-simulation modules normalized by the total wall clock time. All the curves are shown for an increase in the synchronized time between simulators. **c** The contribution of NEST module to the total amount of the wall clock time normalizes between 0 and 100. Each contribution is reported as red curve and for increasing synchronized time. The reduction follows a logarithm function.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

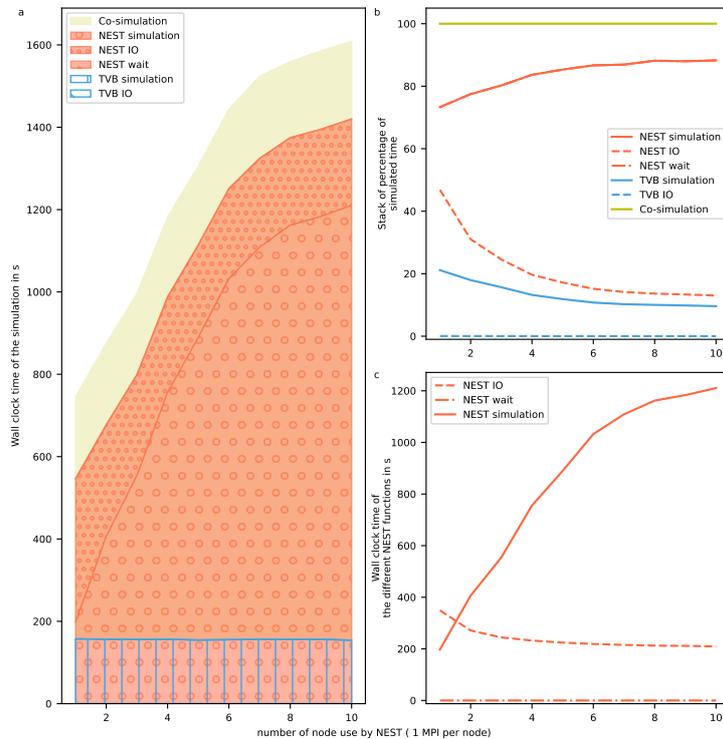


FIGURE E.22. – Performance of the co-simulation on one supercomputer
 Performance is obtained for 1 second of simulated time on a computer. The reference implementation use 1 MPI process, 6 virtual processes/threads, 2.0 ms to synchronize time between simulator for the simulation of 20000 neurons. The node of Jusuf, the supercomputer, content 2 AMD EPYC 7742 @ 2.25 GHz * 64 cores * 2 threads, 256 (16x16) GB DDR4 with 3200 MHz, connected by InfiniBand HDR100 (Connect-X6). The transfer modules and TVB has on one node and NEST on one or multiple other nodes. Simulation depending on the number of nodes used by NEST. **a** The wall clock time of the simulator as a function of the number of nodes used by NEST. The increase of the nodes create overhead communication in side NEST because the network is small. Moreover, the minimum delay in the network is the same as the integration step which creates an overhead of communication in NEST simulation. The wall clock time of the simulator depending on the number of neurons. The total time of the co-simulation is represented in yellow. The "simulation", "IO" and "wait" times of NEST are represented in the red surface with respectively hatches with big circles, small circles and points. The "simulation" and "IO" times of TVB are represented in the blue surface with respectively hatches horizontal lines and oblique lines. The "simulation" time for TVB is constant. The sum of "simulation" and "IO" time of NEST is higher than the TVB "simulation". **b** The wall clock time for different co-simulation modules normalized by the total wall clock time. **c** The contribution of NEST module to the total amount of the wall clock time normalized between 0 and 100. The NEST "IO" time remains constant with the increase of the number of nodes.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures



FIGURE E.23. – Details of the timer of one run for the reference configuration This tree-map represents the timer for each component of the transfer module and for the modules NEST and TVB at the top. The orange bar under each box represents the time required for the initialisation. Each rectangle represents the time spent on each specific piece of code.

E. Supplementary material of chapter 3 – E.3. Supplementary Figures

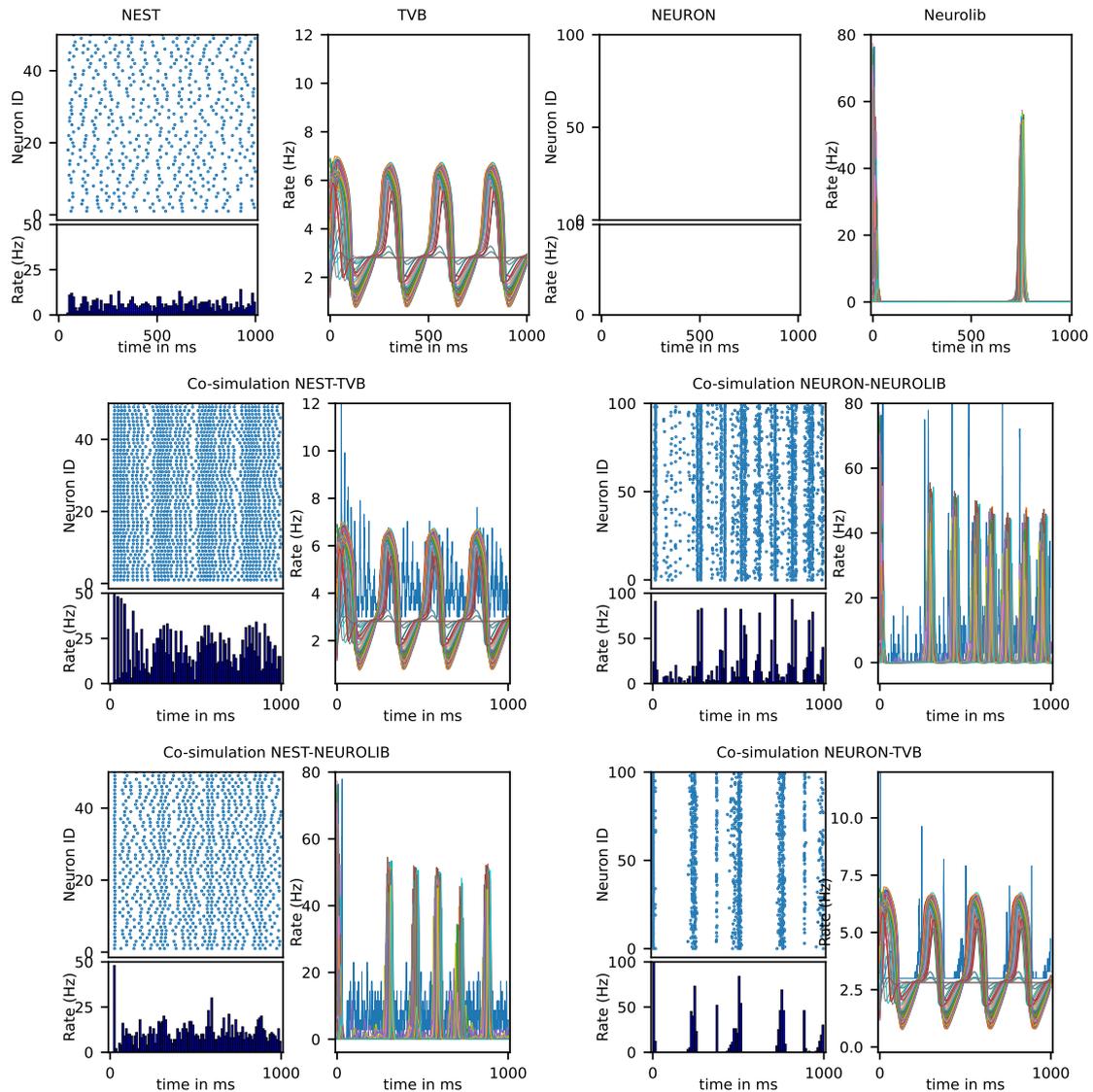


FIGURE E.24. – Proof of concept of replacing NEST and TVB by other simulators
 For spiking neuron simulators (NEST and NEURON), the simulation output is the spike train of neurons (top graphic) and the firing rate of the population (bottom graphic : histogram of spike count with a bin of 10 ms). For Neural Mass simulators (TVB and Neurolib), the simulation output is the mean firing rate of the excitatory population.

The first row shows different examples without co-simulation on other simulators. One remark about the network used in NEURON, this network without external stimulation does not have any activities. The second and third rows display the result of the coupling example together simulated using co-simulation. The co-simulation results show the interaction of examples between them and the possibility to simulate these four different multiscale examples. The code is available here : <https://github.com/multiscale-cosim/TVB-NEST-demo/tree/proof-concept>

CHAPTER F

Supplementary material of chapter 4

Contents of Chapter F

F1. Table of content of Supplementary Notes, Figures and Tables	274
F2. Supplementary Note 1 : Databases description	276
F2.1. Quality of database	276
F3. Supplementary Note 2 : supplementary results	276
F3.1. Firing rate and associated measures	276
F3.2. Inter-spiking interval and associated measures	277
F3.3. Phase and associated measures	277
F3.4. Bursts and associated measures	278
F4. Supplementary Table	279
F5. Supplementary Figures	283

E.1. Table of content of Supplementary Notes, Figures and Tables

Supplementary File	Title
Supplementary Note F2	database description
Supplementary Note F3	supplementary material
Supplementary Table F1	Network description
Supplementary Figure F1	View the grid as a torus
Supplementary Figure F2	Spike trains of disconnected network
Supplementary Figure F3	Parameter space of weight and delay of the long-range connections without noise
Supplementary Figure F4	Parameter space of weight and delay of the long-range connections with white noise
Supplementary Figure F5	Delay of long-range influence firing rate without white noise
Supplementary Figure F6	Parameter space of standard deviation of the connection kernel (σ_{long}) and delay of long-range connections without noise
Supplementary Figure F7	Parameter space of standard deviation of the connection kernel (σ_{long}) and delay of the long-range connections with white noise
Supplementary Figure F8	Parameter space of position of the target cell and delay of the long-range connections without noise
Supplementary Figure F9	Parameter space of position of the target cell and delay of the long-range connections with white noise
Supplementary Figure F10	Parameter space of the effect of the noise and excitatory synaptic weight
Supplementary Figure F11	Three characteristics network dynamics without long-range connection for different excitatory synaptic weights in the presence of white noise
Supplementary Figure F12	Network and neurons dynamics characteristic in function of adaptation current properties
Supplementary Figure F13	Parameter space of subthreshold adaptation and delay of the long-range connections with white noise
Supplementary Figure F14	Parameter space of spike-triggered adaptation and delay of the long-range connections with white noise
Supplementary Figure F15	Parameter space of time constant adaptation and delay of the long-range connections with white noise
Supplementary Figure F16	Network and neurons dynamics characteristic in function of voltage membrane properties

E Supplementary material of chapter 4 – E1. Table of content of Supplementary Notes, Figures and Tables

Supplementary Figure F17	Parameter space of heterogeneity and delay of the long-range connections with white noise
Supplementary Figure F18	Parameter space of voltage reset and delay of the long-range connections with white noise
Supplementary Figure F19	Parameter space of refractory time and delay of the long-range connections with white noise
Supplementary Figure F20	Three characteristics network dynamics with uni-direction long-range connection for different refractory time
Supplementary Figure F21	Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 100 percent of active neurons
Supplementary Figure F22	Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 75 percent of active neurons
Supplementary Figure F23	Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 50 percent of active neurons
Supplementary Figure F24	Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 25 percent of active neurons
Supplementary Figure F25	Parameter space of ratio excitatory/inhibitory synaptic weights and delay of the long-range connections without noise
Supplementary Figure F26	Parameter space of ratio excitatory/inhibitory synaptic weight and delay of the long-range connections with white noise (variance :1800pA)
Supplementary Figure F27	Parameter space of ratio excitatory/inhibitory synaptic weight and delay of the long-range connections with white noise (variance :2600pA)
Supplementary Figure F28	Two characteristics initial condition with uni-direction long-range connection for different weight of long-range connection without noise
Supplementary Figure F29	Comparison of the uni and bi-direction of long-range connection
Supplementary Figure F30	Four characteristic network dynamics with bi-directional long-range connections
Supplementary Figure F31	Parameter space of weight and delay of the long-range connections with white noise and bi-directional
Supplementary Figure F32	Parameter space of the size of the grid on these two dimensions
Supplementary Figure F33	One simulation transition of state

F.2. Supplementary Note 1 : Databases description

There is one database by parameter space, i.e. 23 databases. Each database contains the result of the analysis of the spike trains for the analysis of all the networks, by type of neurons, by cell and by quarter of the grid. The quarters of the grid are the five columns of the grid (Q_{source} :30-79 cells, Q_{target} :130-179 cells, Q_2 :80-129, Q_4 :180-200 and 0-29). The analysis is done after a transient period of 12 s and over 10 s. The Supplementary Material (see Supplementary Note 2) describes the different measures realized over the network's firing rate, the inter-spiking interval, the phase, and the bursts of the neurons during the analysis.

F.2.1 Quality of database

Each database contains the date of the analysis, the path of the simulation, the explored parameters, and the name of the population. However, some databases do not contain all expected analyses due to simulation bugs. Specifically, the exploration databases with bi-directional long-range connections do not contain cell and quarter analyses, as the analysis does not consider the grid's dimension for the separation of the quarters. Additionally, the databases associated with modifying the target cell contain results of simulations with a bug, where the long-range connection is not created when the target cell's index is higher than 15 or lower than 5.

F.3. Supplementary Note 2 : supplementary results

F.3.1 Firing rate and associated measures

The firing rate is the number of generated spikes by a neuron during 1 s. One measure of population synchronization quantifies the fluctuation of the population mean firing rate (for windows of 1 and 3 ms) with the coefficient of variation [41]. Additionally, one estimation of neural population frequency uses the dominant frequency and its associated power of the population mean firing rate (for a window of 1 ms) using Welch's method [358] from the elephant package [89]. (Databases record average ('rates_average'), standard deviation ('rates_std'), minimum ('rate_min') and maximum ('rate_max') mean firing rates; for each (1 ms and 3 ms), the maximum of firing rate (max_IFR_1ms' and 'max_IFR_3ms') and the coefficient of variation ('cvs_IFR_1ms' and 'cvs_IFR_3ms'); and for a window of 1 ms, the dominant frequency and its power('frequency_hist_1_freq' and 'frequency_hist_1_val').

F.3.2 Inter-spiking interval and associated measures

The inter-spiking interval is the time between 2 spikes. The distribution of this interval characterizes the regularity or irregularity of neuron spike generation [41], which is quantified by two measures : coefficient of variation of firing rate [41] and local variation [321][320]. These measures are provided by Elephant python package [89]. For viable estimation, these measures are determined by considering only neurons with more than 20 spikes in 10 s for analysis. (Databases record average ('ISI_average'), standard deviation('ISI_std'), minimum ('ISI_min'), and maximum ('ISI_max') inter-spiking intervals; the average and standard deviation over the population of the regularity measures ('cvs_ISI_average', 'cvs_ISI_std', 'lvs_ISI_average' and 'lvs_ISI_std'); and the percentage of neurons with more than 20 spikes ('percentage').

F.3.3 Phase and associated measures

Kuramoto proposed a model for describing and analyzing synchronization of coupled intrinsic oscillators [211], based on the following equation :

$$\frac{\partial \theta_i}{\partial t} = \omega_i + \frac{K}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i), \quad i = 1 \dots N, \quad (\text{E.1})$$

where the system is composed of N limit-cycle oscillators, with phases θ_i and coupling constant K.

The phase of a neuron can be defined by approximating the voltage membrane as an intrinsic oscillator or by approximating the phase using a constant velocity between 2 spikes, similar to Borges et al. 2017 [32]. This study used the second approximation described in the following equation :

$$\psi(t) = 2\pi m + 2\pi \frac{t - t_m}{t_{m+1} - t_m}, \quad \forall t \in [t_m, t_{m+1}] \quad (\text{E.2})$$

where t_m is the m-th spike of the neuron.

One measure of synchronization is the estimation of Kuramoto order-parameter of a population, i.e. the absolute values of the mean phase. Additionally, one estimation of neural population frequency uses the dominant frequency and its associated power of the mean phase (for a window of 0.1 ms) using Welch's method [358] from the elephant package [89]. (Databases record average ('synch_Rs_average'), standard deviation ('synch_Rs_std'), minimum ('synch_Rs_min') and maximum ('synch_Rs_max') of the Kuramoto order-parameter; the start time and end time of the mean phase estimation ('synch_Rs_times_init' and 'synch_Rs_times_end'); and for a window of 0.1 ms, the dominant frequency and its power ('frequency_phase_freq' and 'frequency_phase_val').

F.3.4 Bursts and associated measures

Bursts are consecutive spikes with inter-spike intervals of less than 10 ms. Additionally, the regularity of bursts is analyzed similarly to inter-spike intervals. For neurons with more than 20 bursts, the regularity of the interval of the beginning time of bursts and also their ending time is quantified with the coefficient of variation and the local variation. (Databases record average ('burst_count_average', 'burst_nb_average' and 'burst_rate_average'), standard deviation ('burst_count_std', 'burst_nb_std' and 'burst_rate_std'), maximum ('burst_count_max', 'burst_nb_max' and 'burst_rate_max') and minimum ('burst_count_min', 'burst_nb_min' and 'burst_rate_min') of a number of spikes in the sequence of burst, number of burst by neurons during the analysis and mean bursting rates; the percentage of neurons which have one burst ('percentage_burst'); average ('burst_interval_average'), standard deviation ('burst_interval_std'), minimum ('burst_interval_min'), maximum ('burst_interval_max'); and for neurons with more than 20 bursts, the average and the standard deviation of the coefficient of variation ('burst_cv_begin_average', 'burst_cv_end_average', 'burst_cv_begin_std', 'burst_cv_end_std') and local variation ('burst_lv_begin_std', 'burst_lv_end_std', 'burst_lv_begin_average', 'burst_lv_end_average') of interval of the beginning time of bursts and also their ending time.

F.4. Supplementary Table

A1		Nest : Model Summary	
Software	NEST		
Topology	two dimensional grid		
Population	excitatory and inhibitory		
Connectivity	two-dimensional Gaussian probability function		
Neuron Model	adaptive exponential leaky integrate and fire neurons[37], fixed threshold and fixed absolute refractory time		
Synapse Model	conductance-based alpha shape		
Plasticity	—		
Input	white noise		
Measurement	Spike activity, histogram, coefficient of variation of firing rate		
A2		Nest : Software	
version	2.14		
integrator method	4th order Runge-Kutta-Fehlberg method		
integration step	0.1 ms		
number of seed	16		
simulation time	22000 ms		
A3		Nest : Topology	
Two dimensional regular grid composed of cells of dimension 3 millimeters by 3 millimeters. By default, the grid has 20 cells on the first dimension (x) and 10 cells on the second dimension (y). The grid has two periods conditions.			
A3		Nest : Populations	
Name	Elements	Size	
E	aeif_cond_exp	$N_e = (1 - g_{inh})N = 140$ by cell	
I	aeif_cond_exp	$N_i = g_{inh}N = 60$ by cell	

F Supplementary material of chapter 4 – F.4. Supplementary Table

A4 Nest : Neuron Model	
Name	aeif
Type	adaptive exponential leaky integrator and fire with conductance synapse
subthreshold dynamics	$C_m \frac{dV_m}{dt} = -g_L(V_m - E_L) + g_L \Delta_T e^{\frac{V_m - V_{th}}{\Delta_T}}$ $-g_e(t)(V_m - E_{ex}) - g_i(t)(V_m - E_{in})$ $-W + I_e$ $\tau_w \frac{dW}{dt} = a(V_m - E_L) - W$
reset condition	<p>For $t^{(f)} = \{t V_m(t) \geq V_{peak}\}$</p> <ul style="list-style-type: none"> — $V_m([t^{(f)}; t^{(f)} + t_{ref}]) = V_{reset}$ — $W([t^{(f)}]) = W([t^{(f)}]) + b$
Nest : Synapse Model	
Name	cond_alpha
Type	post-synaptic conductance in the form of alpha function
Coupling equation	$g_e(t) = \sum_{t_j^{(f)}} w_j \alpha(t - t_j, \tau_{ex}) \text{ with } w_j > 0.0$ $g_i(t) = \sum_{t_j^{(f)}} w_j \alpha(t - t_j, \tau_{in}) \text{ with } w_j < 0.0$ $\alpha(t, \tau) = \frac{t}{\tau} e^{1 - \frac{t}{\tau}} \text{Heaviside}(t)$

F. Supplementary material of chapter 4 – F.4. Supplementary Table

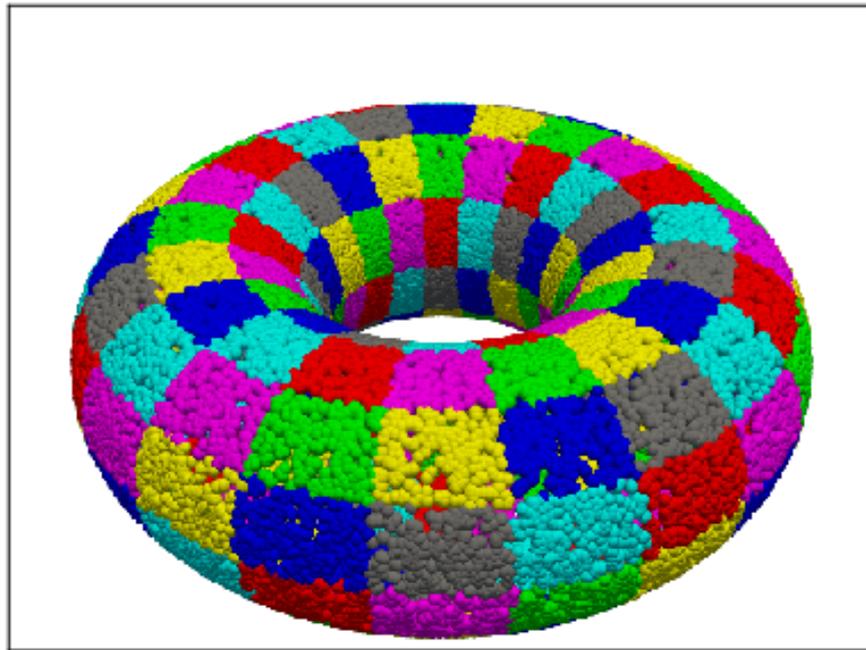
A5 Nest : Neuron Model Parameters		Default values
C_m	Capacity of the membrane	281.0 pF
t_{ref}	Duration of refractory period	2.0 ms
V_{reset}	Reset value for V_m after a spike	-60.0 mV
E_L	Leak reversal potential	-70.6 mV
g_L	Leak conductance	30.0 nS
Δ_T	Slope factor	2.0 mV
V_{peak}	Spike detection threshold	0.0 mV
a	Subthreshold adaptation	0.0 nS
b	Spike-triggered adaptation	80.5 pA
τ_w	Adaptation time constant	144.0 ms
V_{th}	Spike initiation threshold	-50.4 mV
E_{ex}	Excitatory reversal potential	0.0 mV
E_{in}	Inhibitory reversal potential	-80.0 mV
I_e	Constant external input current	mean : 0.0 pA sigma : 0.0 pA
V_m	Initialization of the voltage membrane	mean : -70.6 mV sigma : 100.0 mV
W	Initialization of adaptation current	mean : 200 pA sigma : 200 pA
g_e	Initialization of excitatory synapse conductance	0.0 nS
g_i	Initialization of inhibitory synapse conductance	0.0 nS
A6 Nest : Input		
approximation white noise		
equation	$I(t) = mean + std * N_j$ <p>where N_j are Gaussian random numbers with unit standard deviation. The noise is updated at intervals of dt, which give a piece-wise constant current with Gaussian distribution amplitude.</p>	
parameters		
$mean$	400.0 pA	
std	1800.0 pA	
dt	1.0 ms	

F. Supplementary material of chapter 4 – F.4. Supplementary Table

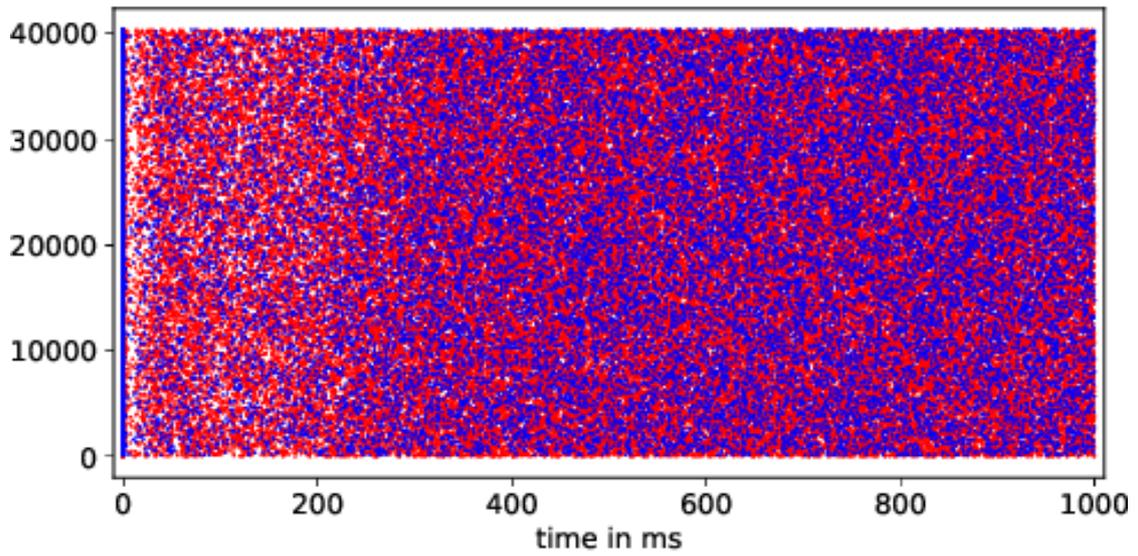
A7 Nest : Connectivity				
default parameter synapses				
τ_{ex}	Rise time of excitatory synaptic conductance			0.2 ms
τ_{in}	Rise time of inhibitory synaptic conductance			2.0 ms
g	Ratio excitatory/inhibitory synaptic weight			0.11
Name source population	σ	Weights	delay	Pattern
excitatory	0.8	$Q_e=10.0$ nS	dt=0.1	The neuron can not have multiple connections with the same neuron. The neuron can connect to itself.
inhibitory	1.2	$Q_i=g.Q_e=11.0$ nS	dt=0.1	The neuron cannot have multiple connections with the same neuron. The neuron can connect to itself.
long-range connection	2.0	$Q_{long}=15.0$ nS	dt=0.1	The neuron can not have multiple connections with the same neuron.
noise		$Q_{noise}=1$ nS	dt=0.1	The noise is connected to all the neurons and independents.
A8 Nest : Measurement				
state variable	spike time		precision	0.1 ms
			number of recorded neurons	all
spike activities	raster plot		precision	0.1 ms
	histogram of mean firing rate		bins	1.0 ms
	coefficient of variation of mean firing rate		bins	3.0 ms

TABLEAU F.1. – Parameter for parameter exploration

F.5. Supplementary Figures

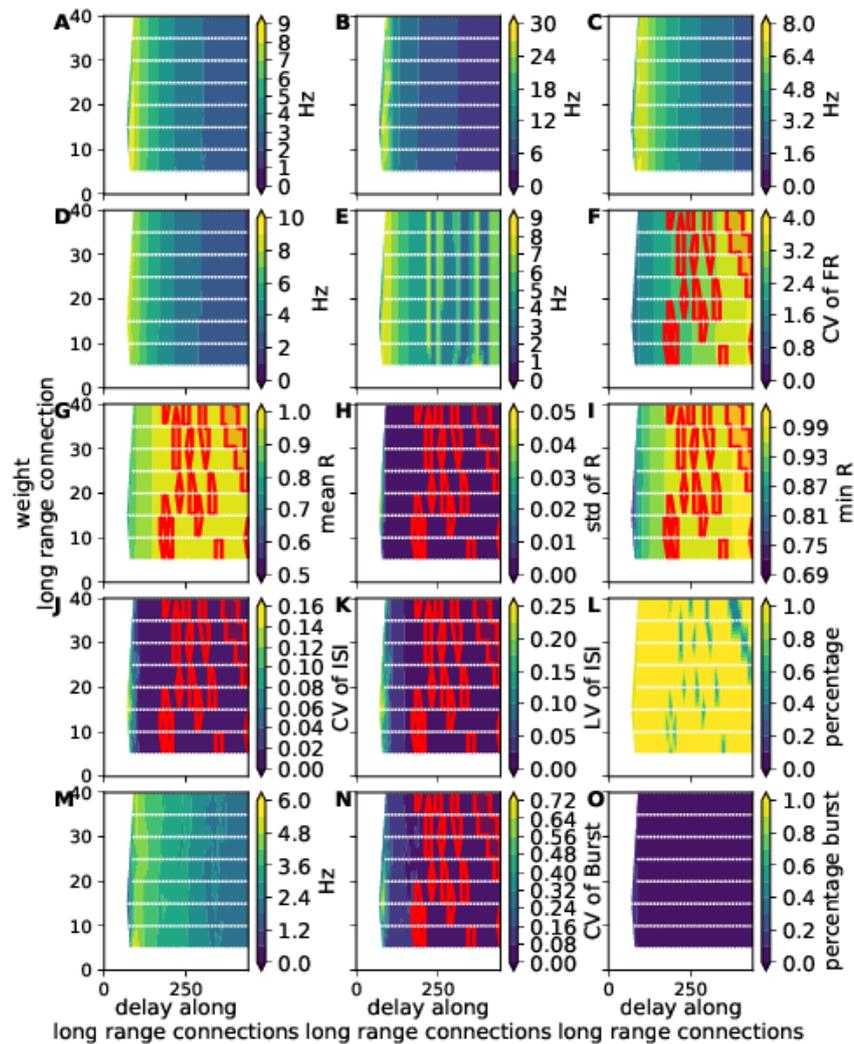


SUPPLEMENTARY FIGURE F.1. – View the grid as a torus
Representation of the grid in 3 dimensions with the position of the neurons and the color is defined for each grid cell.



SUPPLEMENTARY FIGURE F.2. – Spike trains of disconnected network
Spike trains of the neurons in the cases of disconnected neurons in the presence of noise with a variance of 1800 pA.

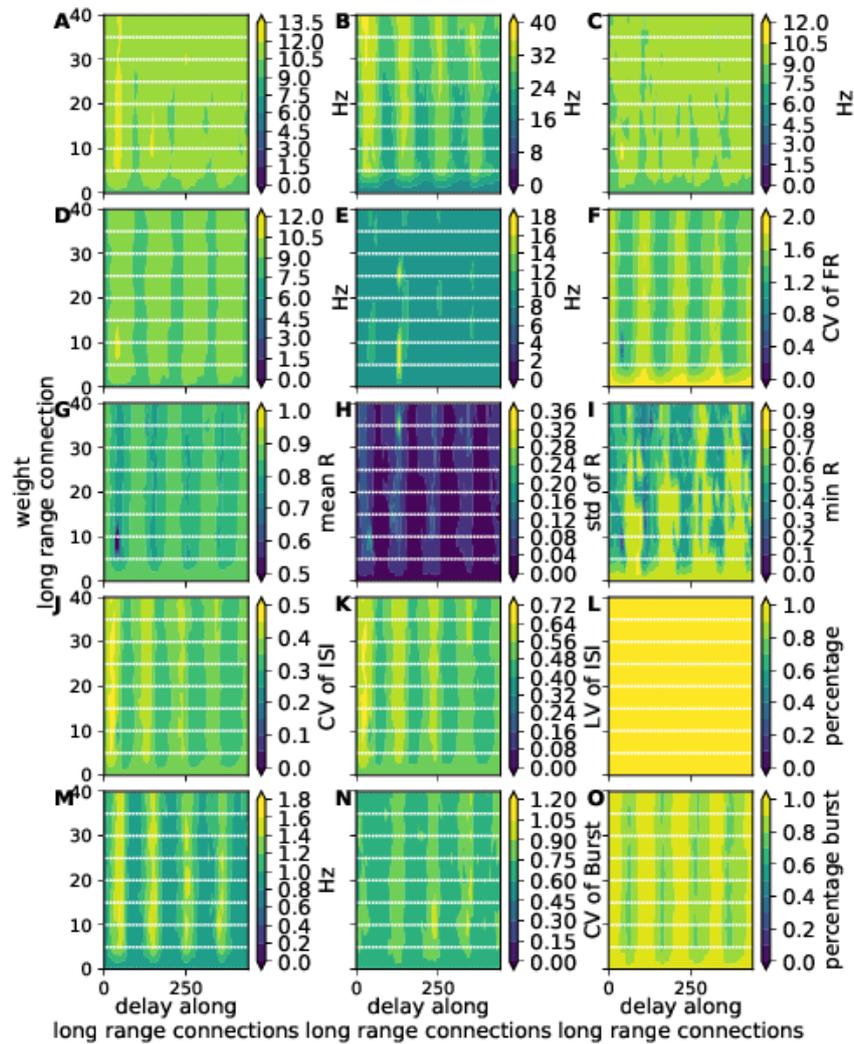
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.3. – Parameter space of weight and delay of the long-range connections without noise

Each graphic represents a measure of the network without noise. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the weight and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. Red areas represent simulations in which less than 95 % of neurons generate fewer than 20 spikes during the 10 s of analysis. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

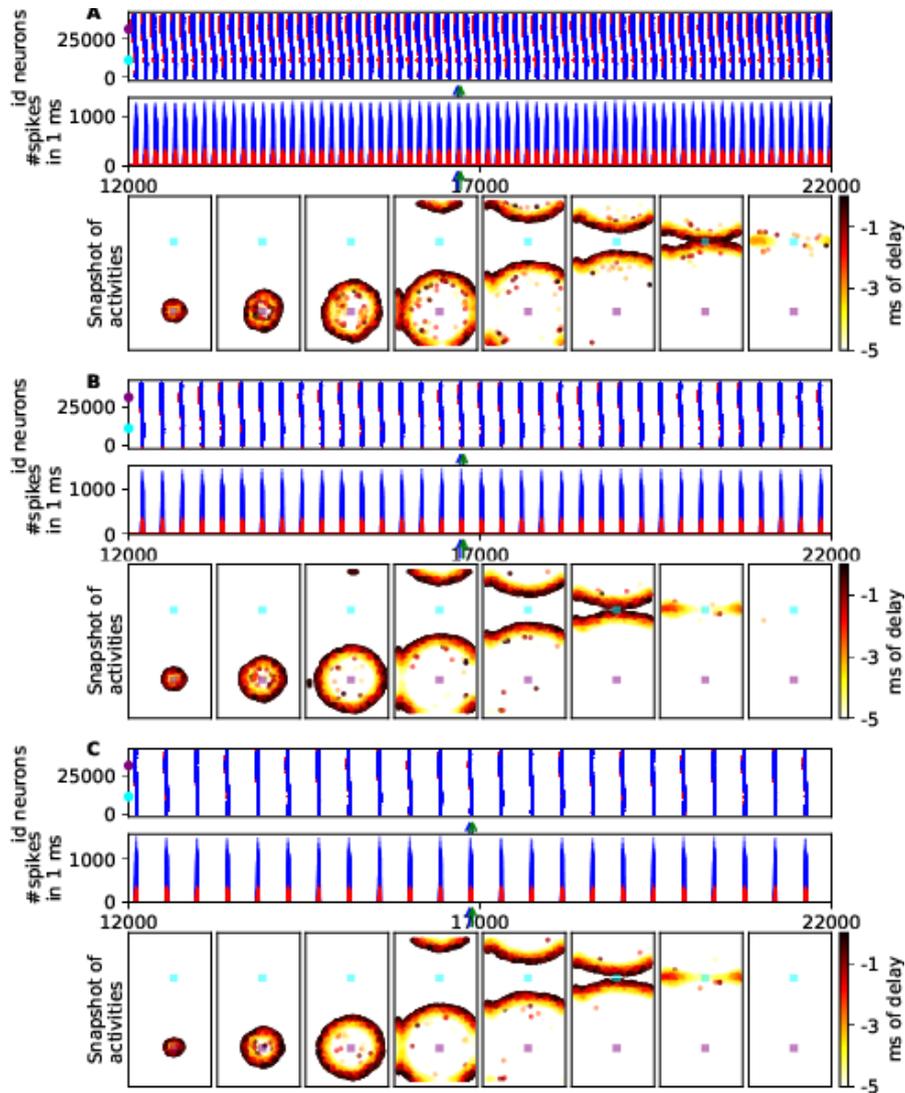
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.4. – Parameter space of weight and delay of the long-range connections with white noise

Each graphic represents a measure of the network with noise, which has a variance of 1800pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the weight and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum (B), and minimum (C) firing rate of the network, the dominant frequency of the mean phase (D) and the histogram with a bin of 1 ms (E), the coefficient of variance of the histogram with a bin of 3 ms (F), mean (G), standard deviation (H) and the minimum (I) of the order-parameter, coefficient of variation (J) and the local variation (K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes (L), the mean firing rate of the burst (M), coefficient of variation of the start of the burst (N), the percentage of the neurons with more than 20 bursts (O) (see supplementary note 1 for more details).

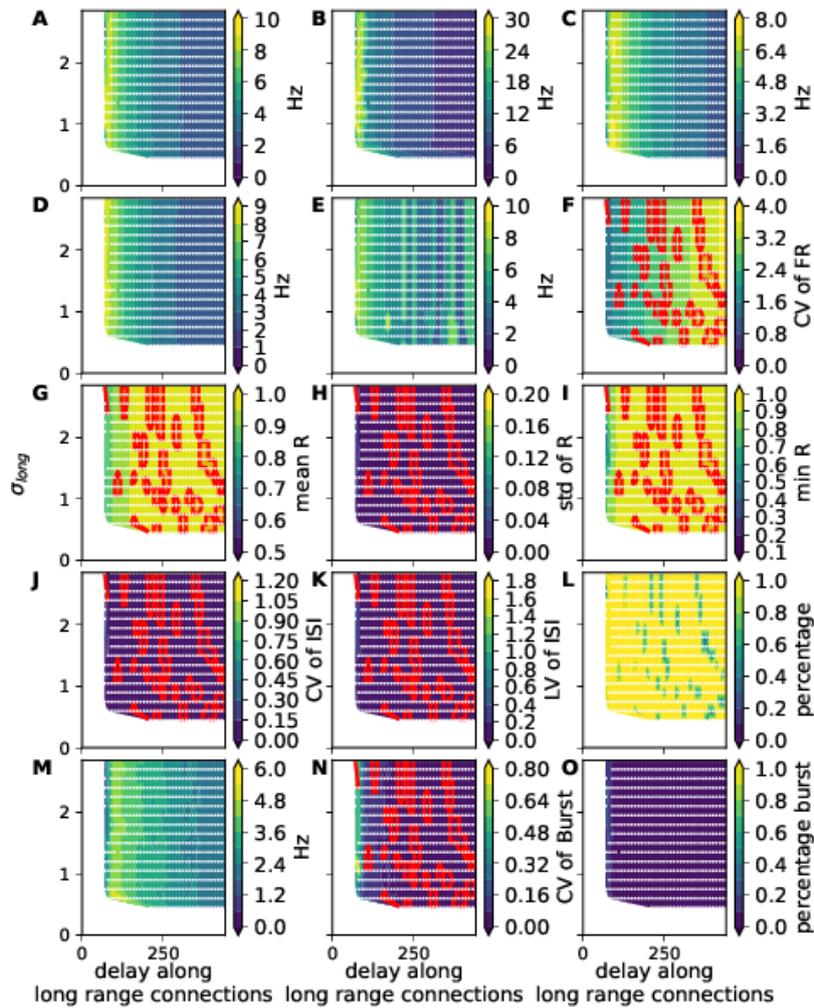
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE E.5. – Delay of long-range influence firing rate without white noise

Simulation of the network of 28000 excitatory neurons and 12000 inhibitory neurons without noise with different delays of long-range connections (A :100 ms, B :250 ms and C :400 ms). The index is ordered by cells of the grid, with excitatory neurons followed by inhibitory neurons in the cell. Each simulation is represented by spike trains of the neurons and the associated histogram with a bin of 1 ms between 12.0 s and 22.0 s. The blue and green arrows define the start and the end of the eight regular snapshots of the network activities (snapshot time : A :16680, 16685.7, 16691.4, 16697.1, 16702.9, 16708.6, 16714.3, 16720; B :16725, 16730.7, 16736.4, 16742.1, 16747.9, 16753.6, 16759.3, 16765; C :16855, 16860.7, 16866.4, 16872.1, 16877.9, 16883.6, 16889.3, 16895). On the histogram and eight snapshots, the blue represents the source population, and the purple represents the target. **A** Fast firing rate **B** Medium firing rate **C** Slow firing rate

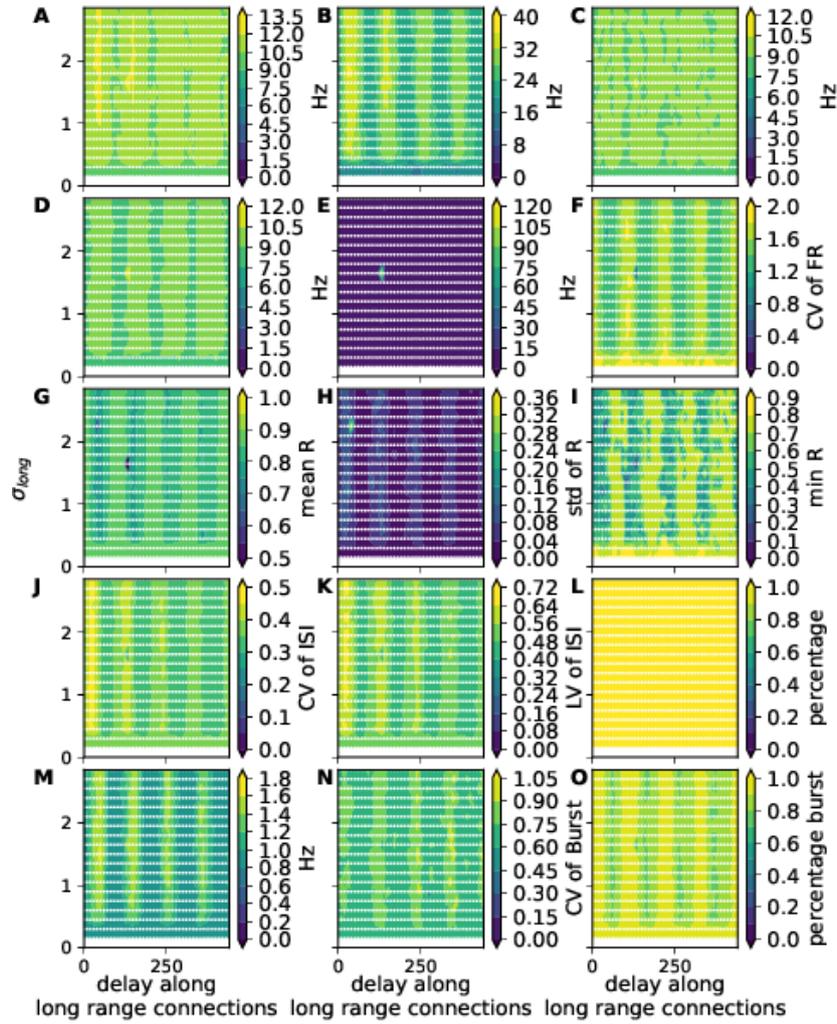
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.6. – Parameter space of standard deviation of the connection kernel (σ_{long}) and delay of the long-range connections without noise

Each graphic represents a measure of the network without noise. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the standard deviation of the connection kernel (σ_{long}) and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. Red areas represent simulations in which less than 95 % of neurons generate fewer than 20 spikes during the 10 s of analysis. The measures of the network are mean (A), maximum (B), and minimum (C) firing rate of the network, the dominant frequency of the mean phase (D) and the histogram with a bin of 1 ms (E), the coefficient of variance of the histogram with a bin of 3 ms (F), mean (G), standard deviation (H) and the minimum (I) of the order-parameter, coefficient of variation (J) and the local variation (K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes (L), the mean firing rate of the burst (M), coefficient of variation of the start of the burst (N), the percentage of the neurons with more than 20 bursts (O) (see supplementary note 1 for more details).

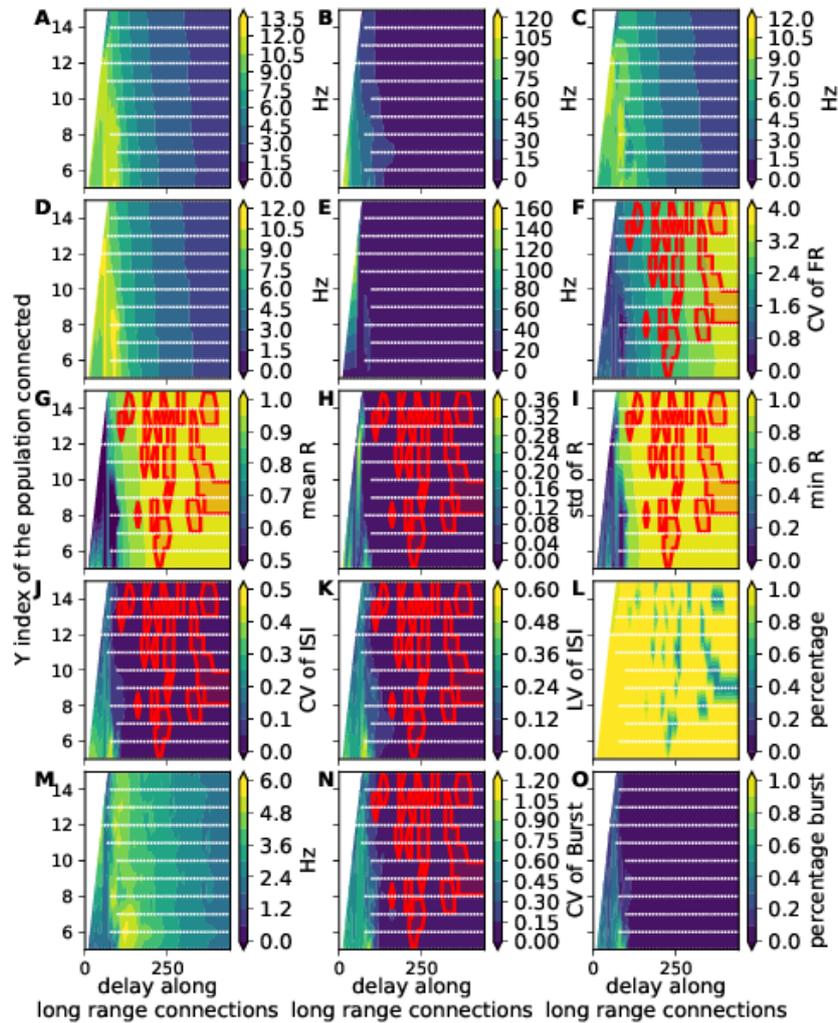
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.7. – Parameter space of standard deviation of the connection kernel (σ_{long}) and delay of the long-range connections with white noise

Each graphic represents a measure of the network with noise, which has a variance of 1800pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the standard deviation of the connection kernel (σ_{long}) and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

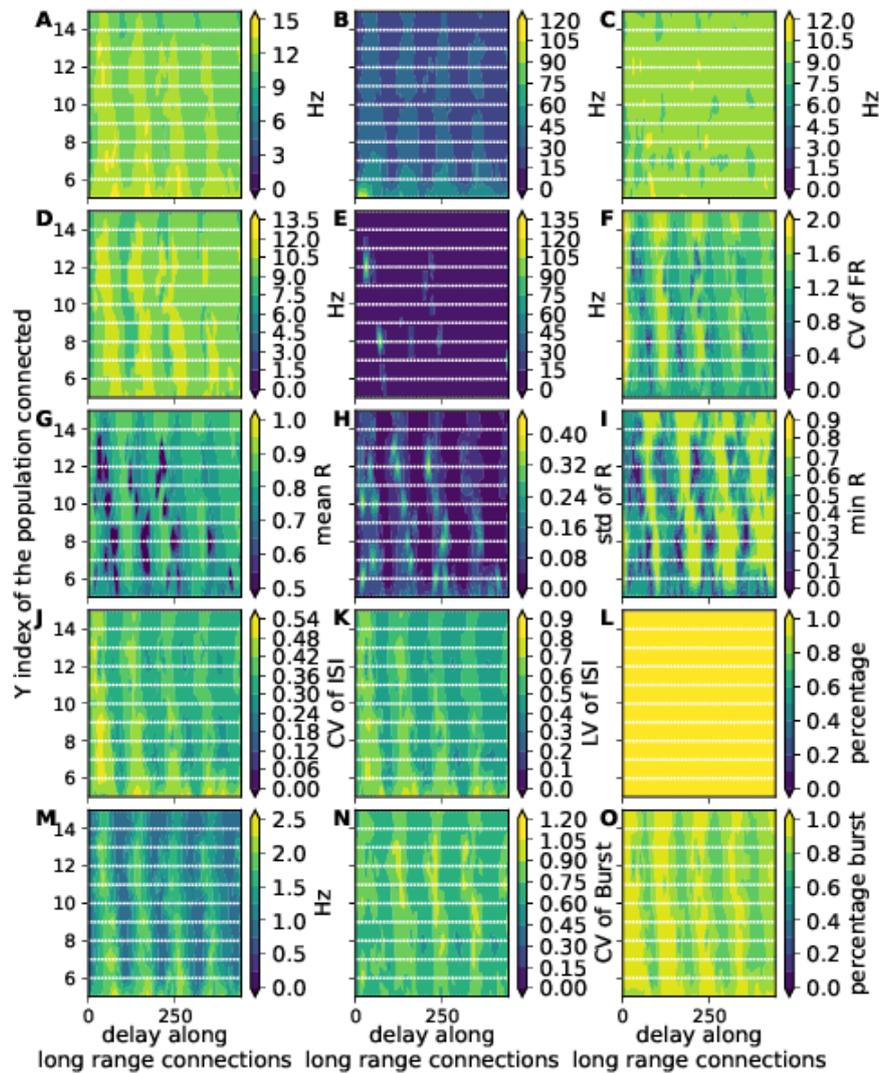
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.8. – Parameter space of position of the target cell and delay of the long-range connections without noise

Each graphic represents a measure of the network without noise. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the target cell and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. Red areas represent simulations in which less than 95 % of neurons generate fewer than 20 spikes during the 10 s of analysis. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

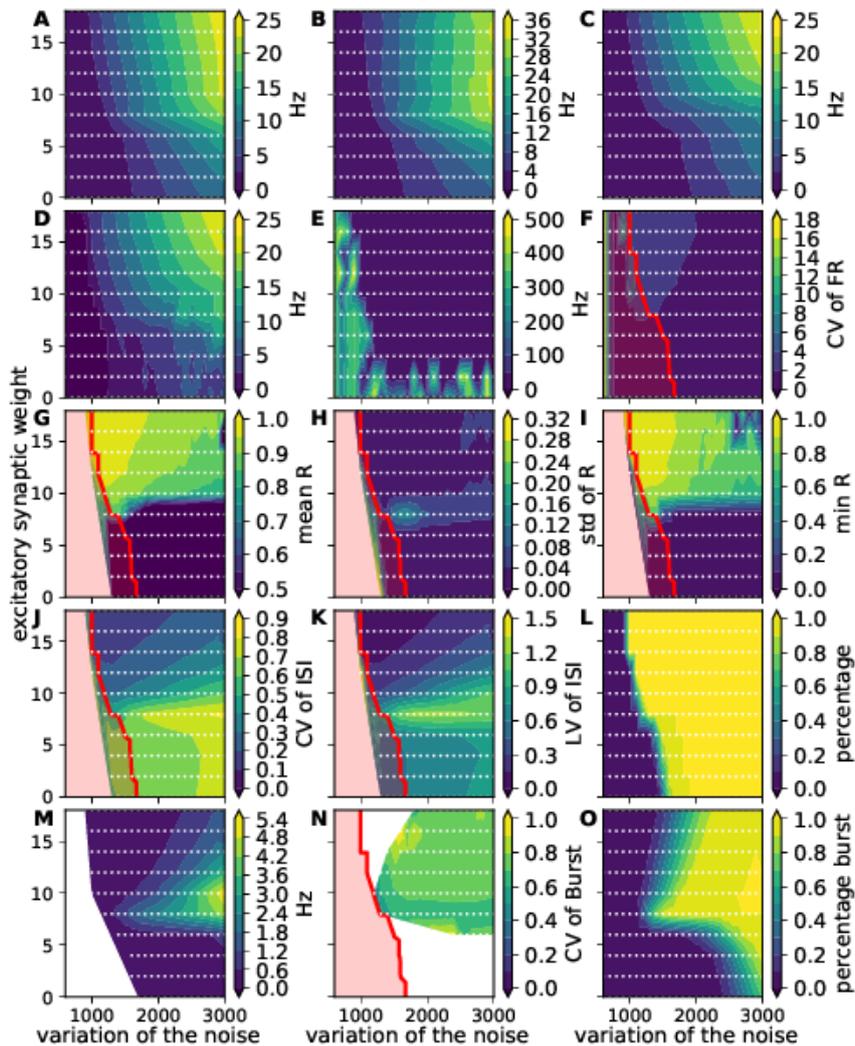
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.9. – Parameter space of position of the target cell and delay of the long-range connections with white noise

Each graphic represents a measure of the network with noise, which has a variance of 1800pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the target cell and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of a 1 ms(E), the coefficient of variance of the histogram with a bin of a 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

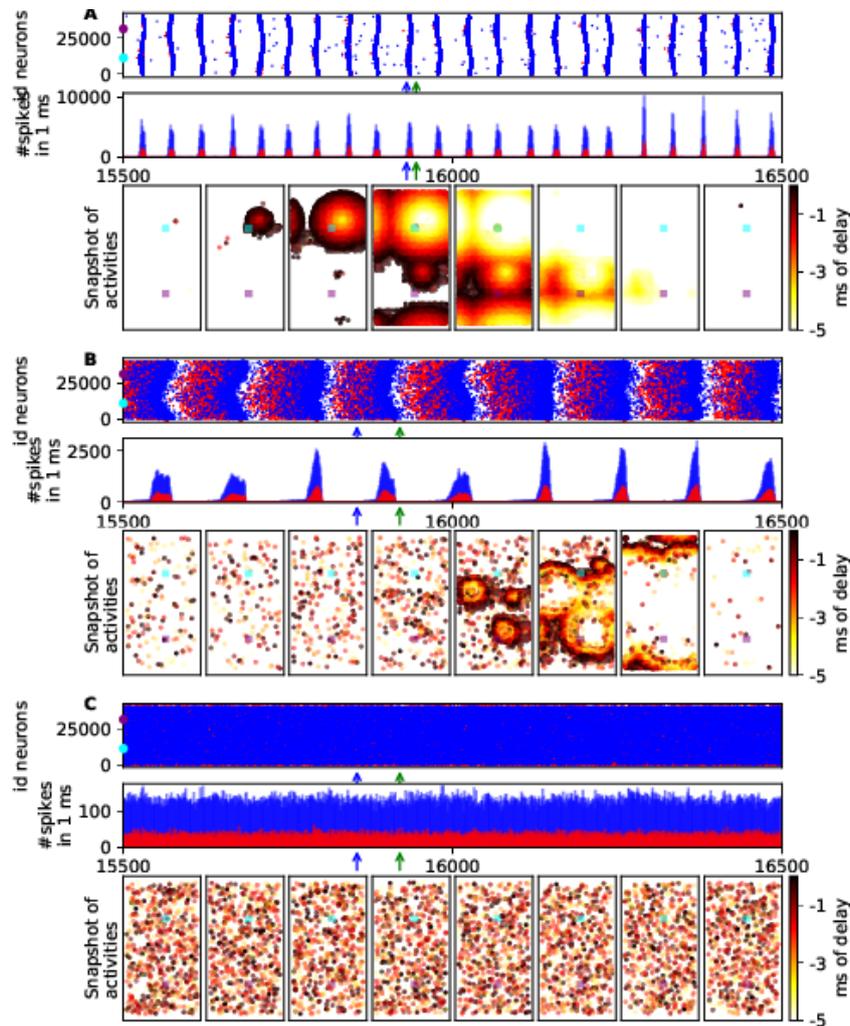
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.10. – Parameter space of the effect of the noise and excitatory synaptic weight

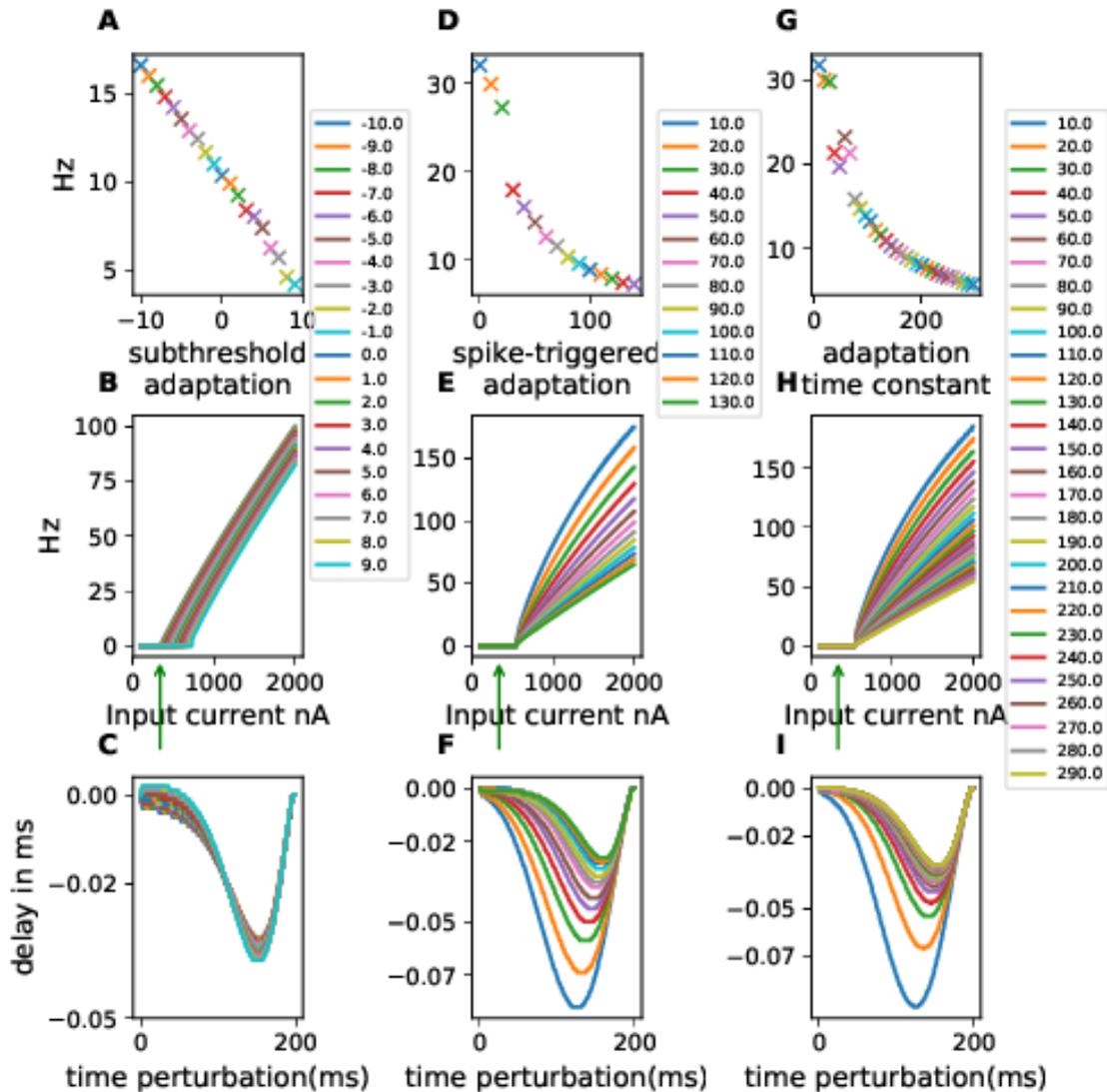
Each graphic represents a measure of the network, which does not have a long-range connection in this case. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the variance of the noise and synaptic weights). The gradient of colors is realized from a linear interpolation of the measures. Red areas represent simulations in which less than 95 % of neurons generate fewer than 20 spikes during the 10 s of analysis. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of a 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.11. – Three characteristics network dynamics without long-range connection for different excitatory synaptic weights in the presence of white noise

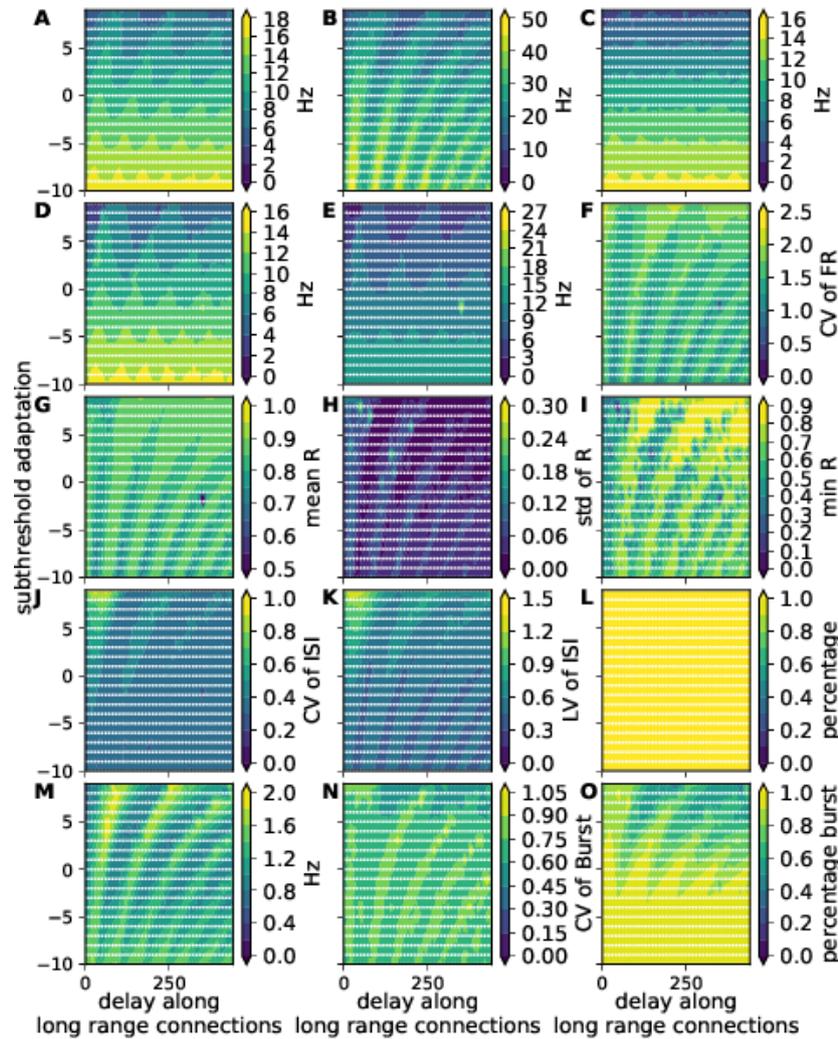
Simulation of the network of 28000 excitatory neurons and 12000 inhibitory neurons, with different excitatory synaptic weights (A :100.0 nS, B :10.0 nS and C :4.0 nS) and variance of white noise equals 1800 pA. The index is ordered by cells of the grid, with excitatory neurons followed by inhibitory neurons in the cell. Each simulation is represented by spike trains of the neurons and the associated histogram with a bin of 1 ms between 15.5 s and 16.5 s. The blue and green arrows define the starting and the end of the eight regular snapshots of the network activities (A :15930 ms, 15932.1 ms, 15934.3 ms, 15936.4 ms, 15938.6 ms, 15940.7 ms, 15942.9 ms, 15945 ms; B :15855 ms, 15864.3 ms, 15873.6 ms, 15882.9 ms, 15892.1 ms, 15901.4 ms, 15910.7 ms, 15920 ms; C :15855 ms, 15864.3 ms, 15873.6 ms, 15882.9 ms, 15892.1 ms, 15901.4 ms, 15910.7 ms, 15920 ms). On the histogram and eight snapshots, the blue represents the source population, and the purple represents the target. **A** Regular bumps. **B** Regular bumps start with random spikes. **C** asynchronous irregular activity.



SUPPLEMENTARY FIGURE F.12. – Network and neurons dynamics characteristic function of adaptation current properties

A, D, G The first row shows the mean firing rate of the network for long-range connection with a delay of 0.1 ms function of different parameters of adaptation current properties (A : subthreshold adaptation, D : spike-triggered adaptation, G : adaptation time constant). **B, E, H** The second row is frequency-current curves for different parameters of adaptation current properties (B : subthreshold adaptation, E : spike-triggered adaptation, H : adaptation time constant). **C, F, I** The third row is the phase response curve for different parameters of the adaptation current properties (C : subthreshold adaptation, F : spike-triggered adaptation, I : adaptation time constant).

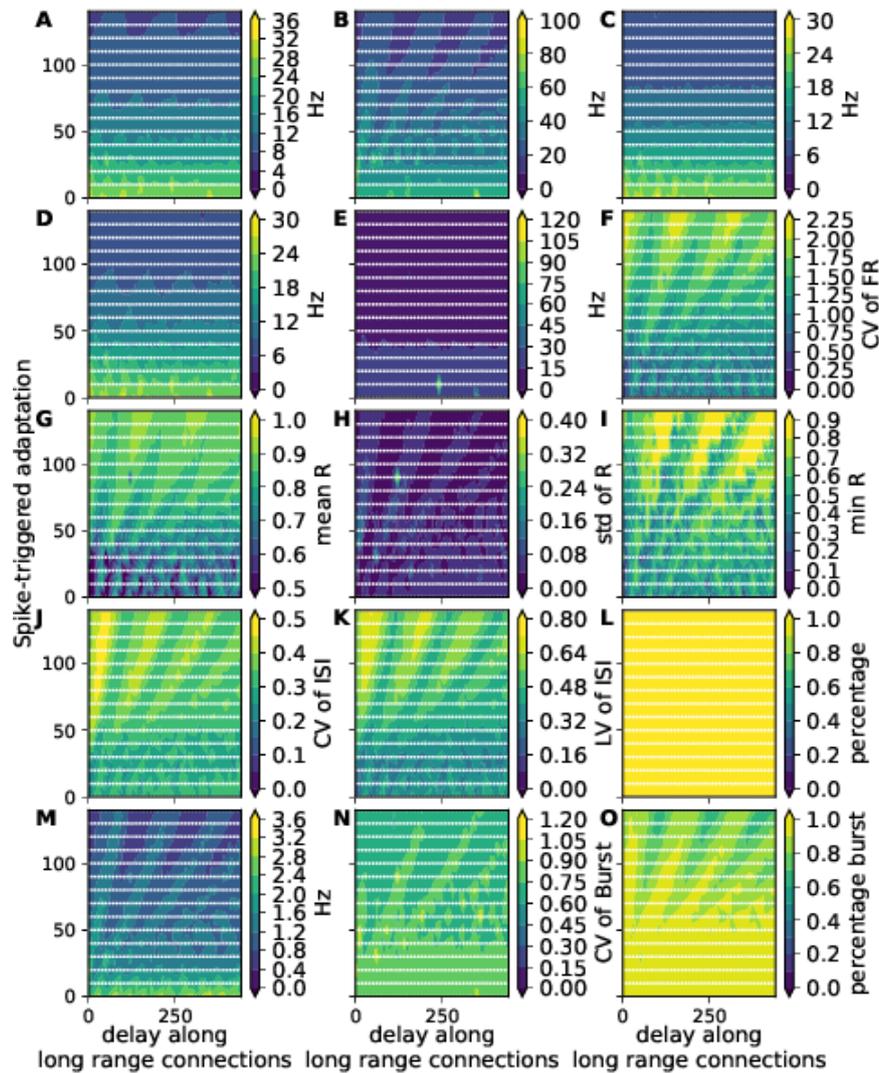
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.13. – Parameter space of subthreshold adaptation and delay of the long-range connections with white noise

Each graphic represents a measure of the network with noise, which has a variance of 1800pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the subthreshold adaptation and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

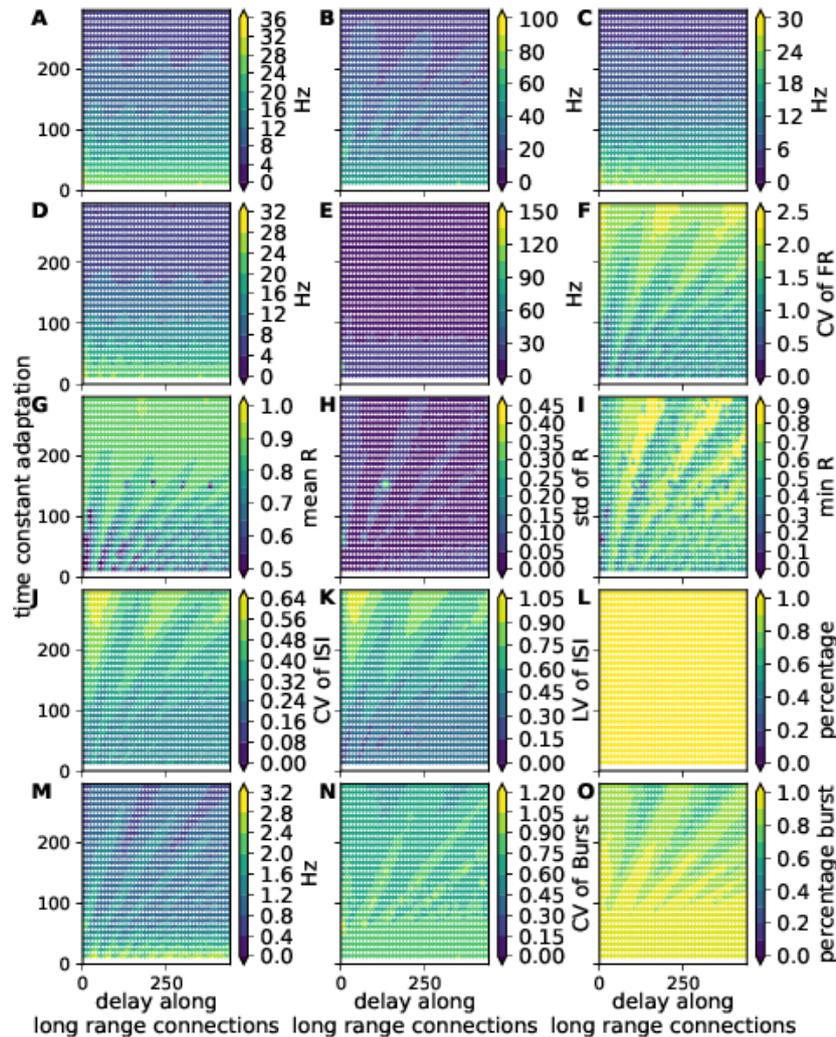
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.14. – Parameter space of spike-triggered adaptation and delay of the long-range connections with white noise

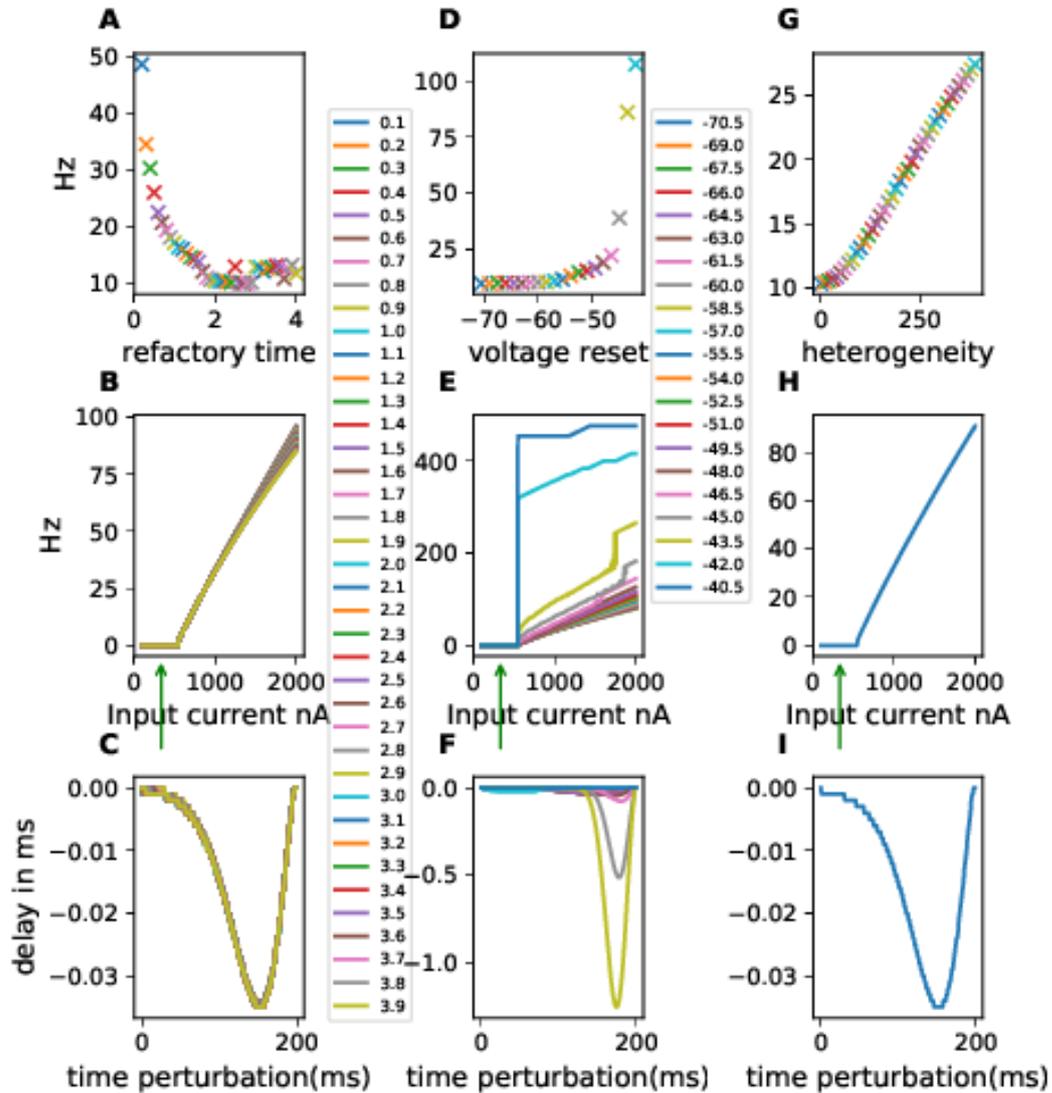
Each graphic represents a measure of the network with noise, which has a variance of 1800pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the spike-triggered adaptation and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum (B), and minimum (C) firing rate of the network, the dominant frequency of the mean phase (D) and the histogram with a bin of 1 ms (E), the coefficient of variance of the histogram with a bin of 3 ms (F), mean (G), standard deviation (H) and the minimum (I) of the order-parameter, coefficient of variation (J) and the local variation (K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes (L), the mean firing rate of the burst (M), coefficient of variation of the start of the burst (N), the percentage of the neurons with more than 20 bursts (O) (see supplementary note 1 for more details).

F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.15. – Parameter space of time constant adaptation and delay of the long-range connections with white noise

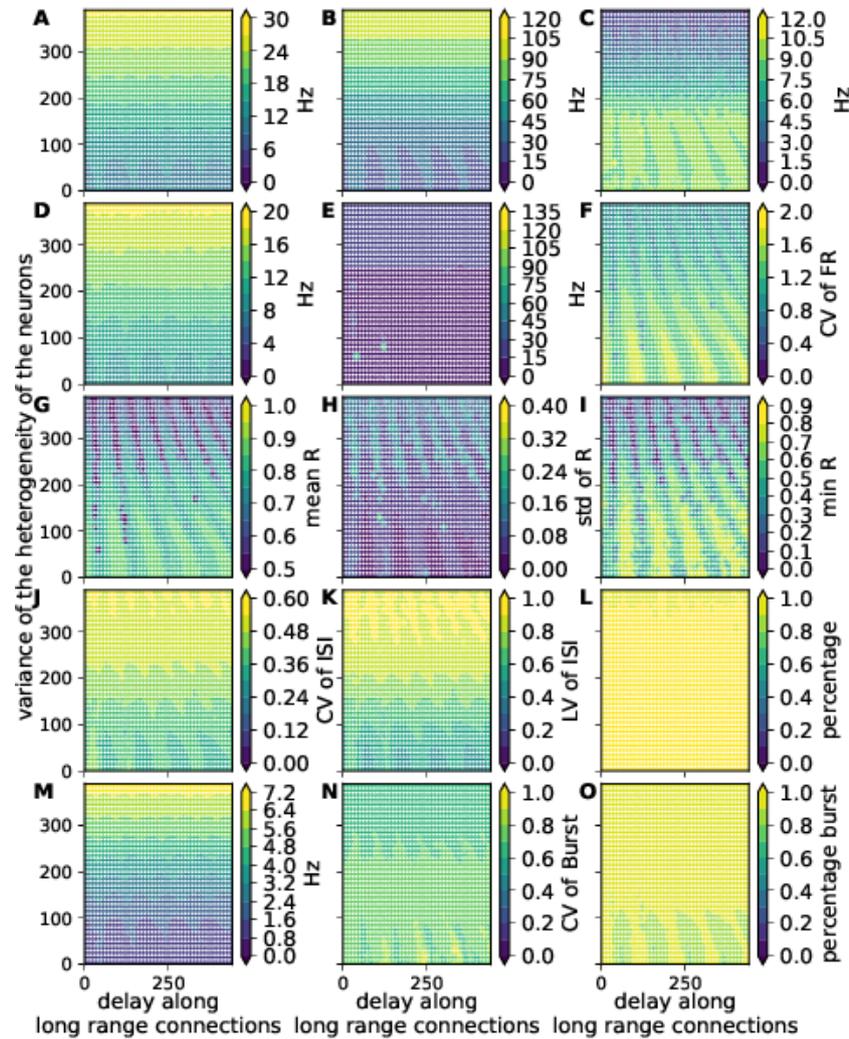
Each graphic represents a measure of the network, with noise, which has a variance of 1800pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the time constant adaptation and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).



SUPPLEMENTARY FIGURE F.16. – Network and neurons dynamics characteristic function of voltage membrane properties

A, D, G The first row shows the mean firing rate of the network for long-range connection with a delay of 0.1 ms function of different parameters of voltage membrane (A : refractory time, D : voltage reset, G : heterogeneity). **B, E, H** The second row is frequency-current curves for different parameters of voltage membrane (B : refractory time, E : voltage reset, H : heterogeneity). **C, F, I** The third row is the phase response curve for different parameters of the voltage membrane (C : refractory time, F : voltage reset, I : heterogeneity).

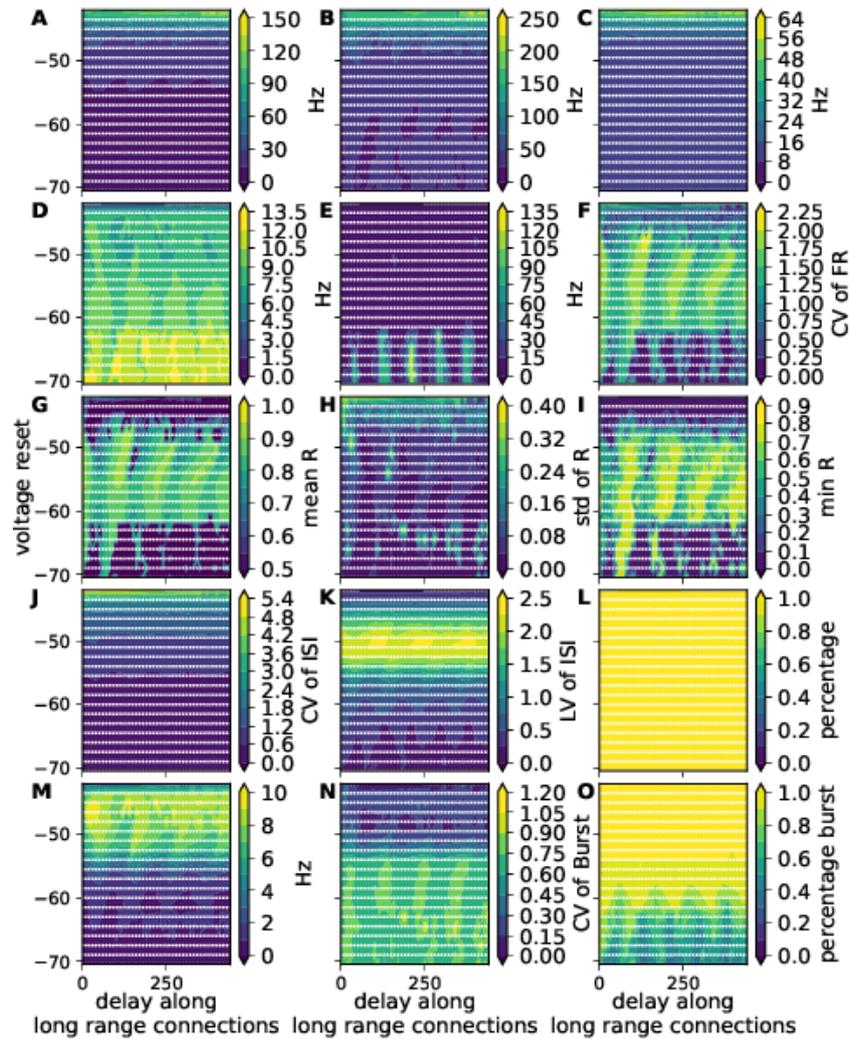
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.17. – Parameter space of heterogeneity and delay of the long-range connections with white noise

Each graphic represents a measure of the network with noise which has a variance of 1800pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the heterogeneity of neurons and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

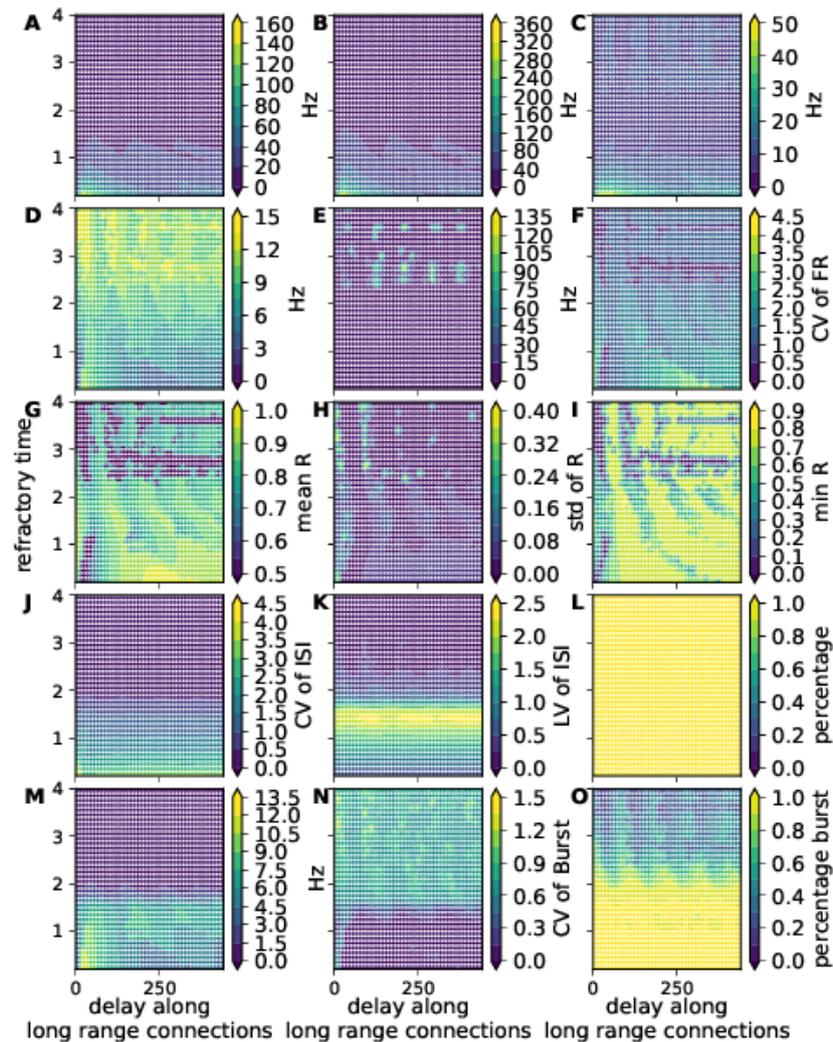
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.18. – Parameter space of voltage reset and delay of the long-range connections with white noise

Each graphic represents a measure of the network with noise, which has a variance of 1800pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the voltage reset of neurons and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

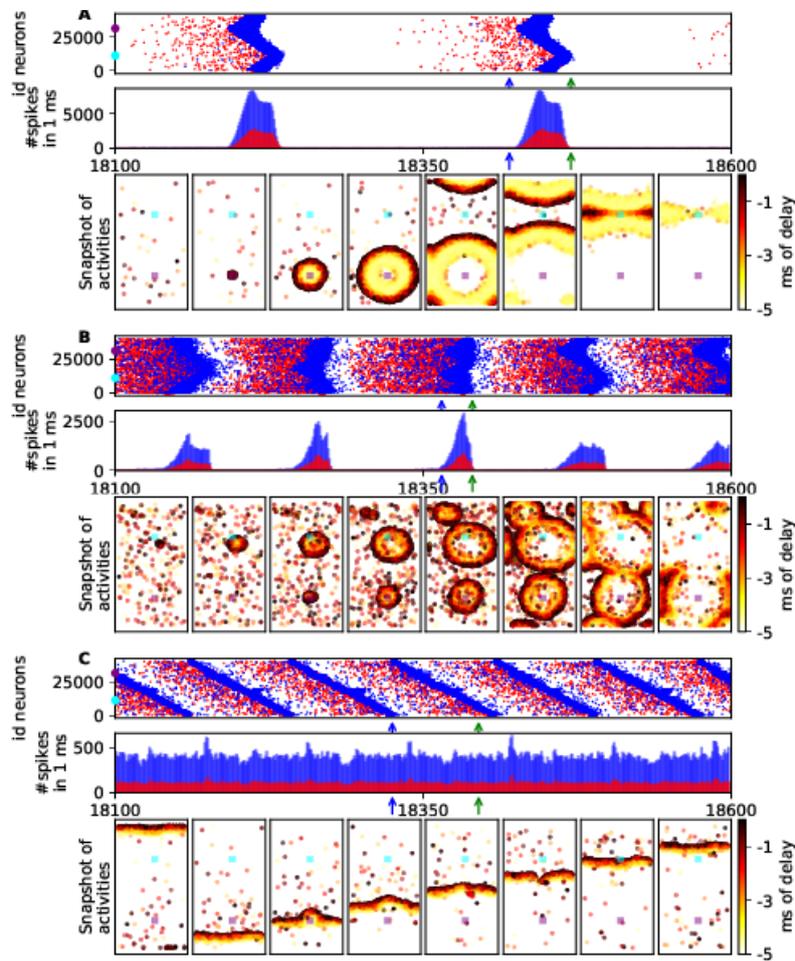
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.19. – Parameter space of refractory time and delay of the long-range connections with white noise

Each graphic represents a measure of the network with noise, which has a variance of 1800pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the refractory time of neurons and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

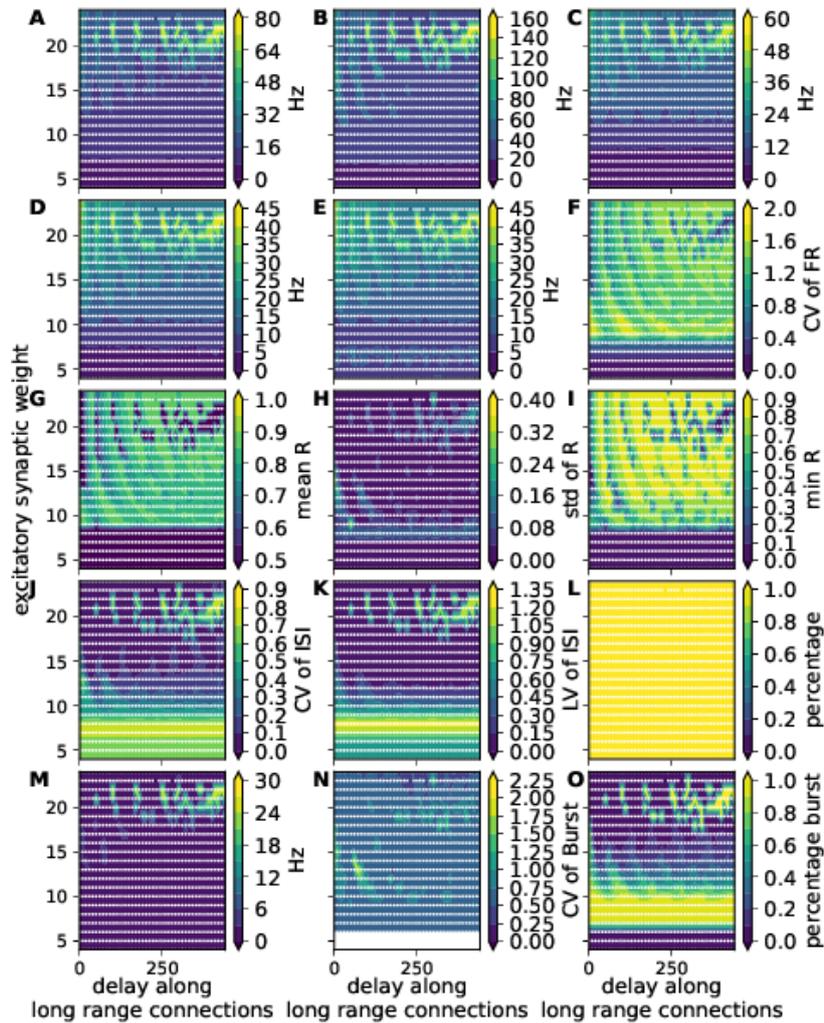
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.20. – Three characteristics network dynamics with uni-direction long-range connection for different refractory time

Simulation of the network of 28000 excitatory neurons and 12000 inhibitory neurons with white noise and different refractory times (A :0.5 ms, B :2.0 ms, and C :2.8 ms) and with a delay of the long-range connections equals 200 ms. The index is ordered by cells of the grid, with excitatory neurons followed by inhibitory neurons in the cell. Each simulation is represented by spike trains of the neurons and the associated histogram with a bin of 1 ms between 18.1 s and 18.6 s. The blue and green arrows define the start and the end of the eight regular snapshots of the network activities (A :18420 ms, 18427.1 ms, 18434.3 ms, 18441.4 ms, 18448.6 ms, 18455.7 ms, 18462.9 ms, 18470 ms; B :18365 ms, 18368.6 ms, 18372.1 ms, 18375.7 ms, 18379.3 ms, 18382.9 ms, 18386.4 ms, 18390 ms; C :18325 ms, 18335 ms, 18345 ms, 18355 ms, 18365 ms, 18375 ms, 18385 ms, 18395 ms). On the histogram and eight snapshots, the blue represents the source population, and the purple represents the target. **A** Large regular bump. **B** Regular bump with bumps from the noise. **C** Travelling waves.

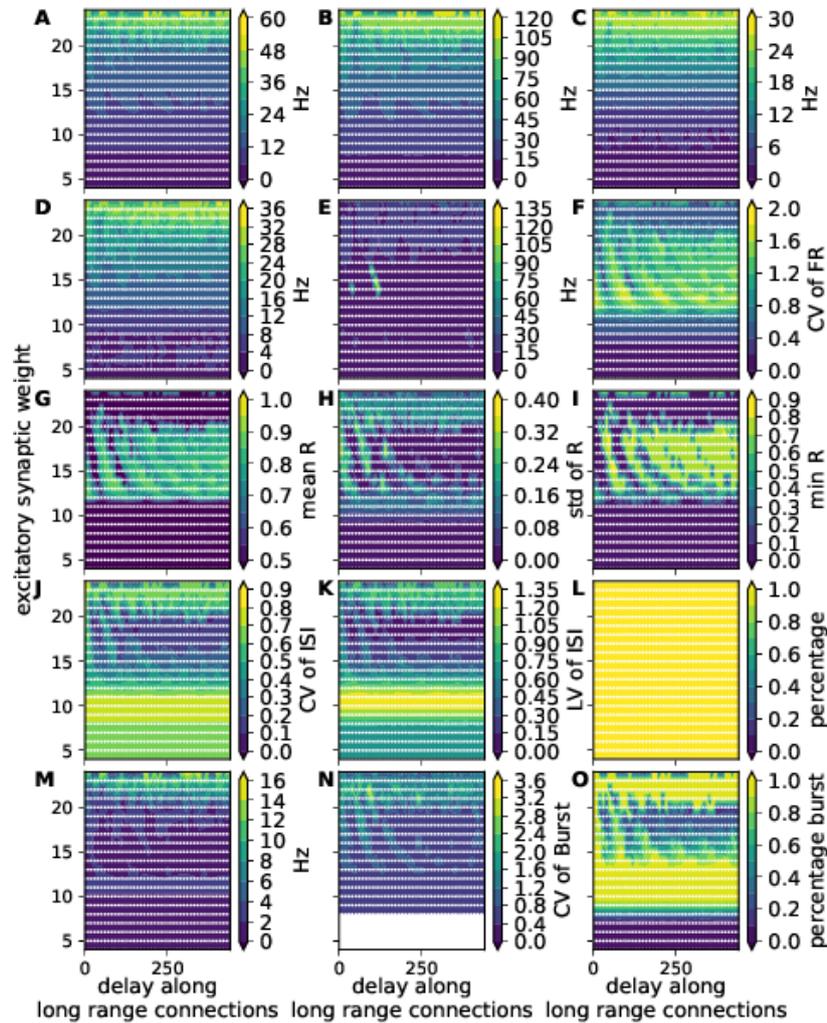
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.21. – Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 100 percent of active neurons

Each graphic represents a measure of the network with noise, which has a variance of 1800pA for 100 percent of active neurons. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the synaptic weights and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

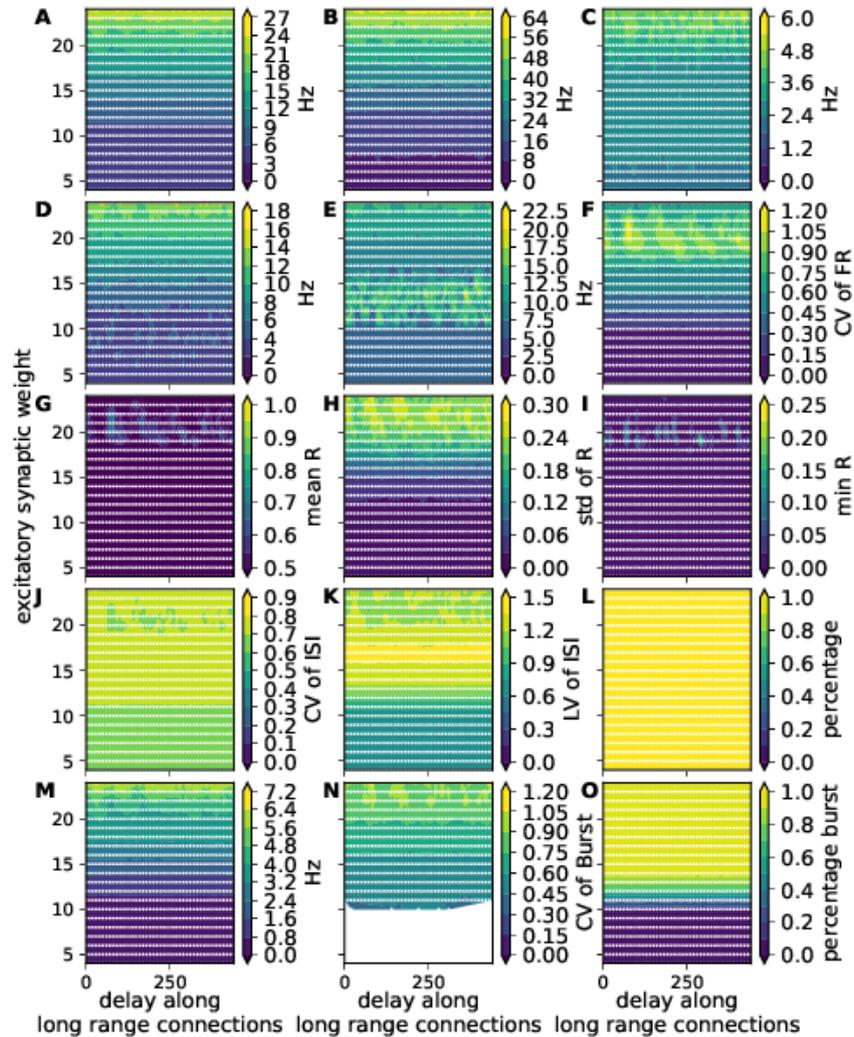
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.22. – Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 75 percent of active neurons

Each graphic represents a measure of the network with noise, which has a variance of 1800pA for 75 percent of active neurons. Each white point represents the analysis of one realization of 10 s after a transient period of t12 s for a modification of two parameters (the synaptic weights and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

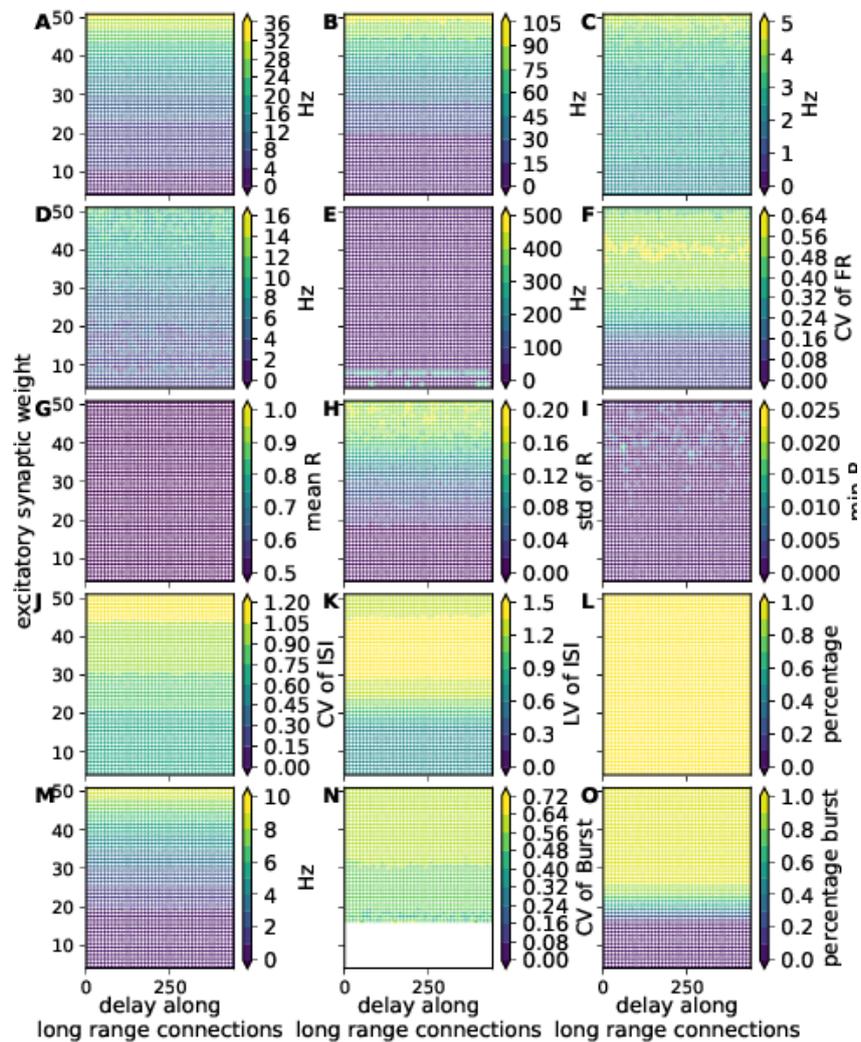
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.23. – Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 50 percent of active neurons

Each graphic represents a measure of the network with noise, which has a variance of 1800pA for 50 percent of active neurons. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the synaptic weights and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

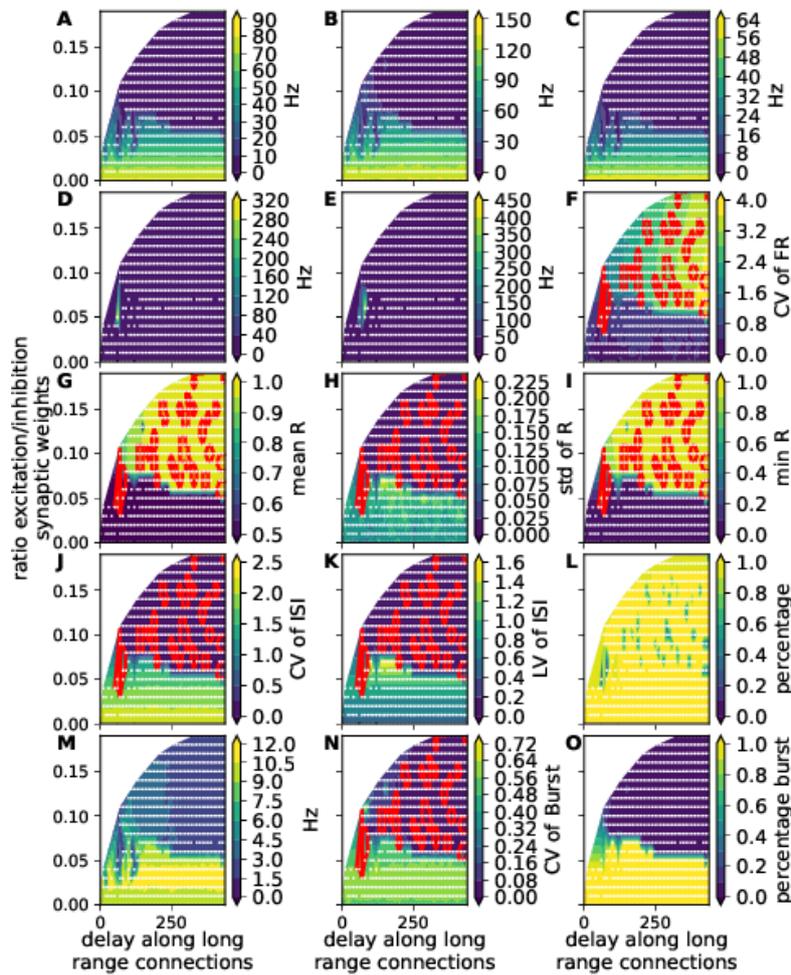
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.24. – Parameter space of excitatory synaptic weight and delay of the long-range connections with white noise for 25 percent of active neurons

Each graphic represents a measure of the network, with noise with a variance of 1800pA for 25 percent of active neurons. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the synaptic weights and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

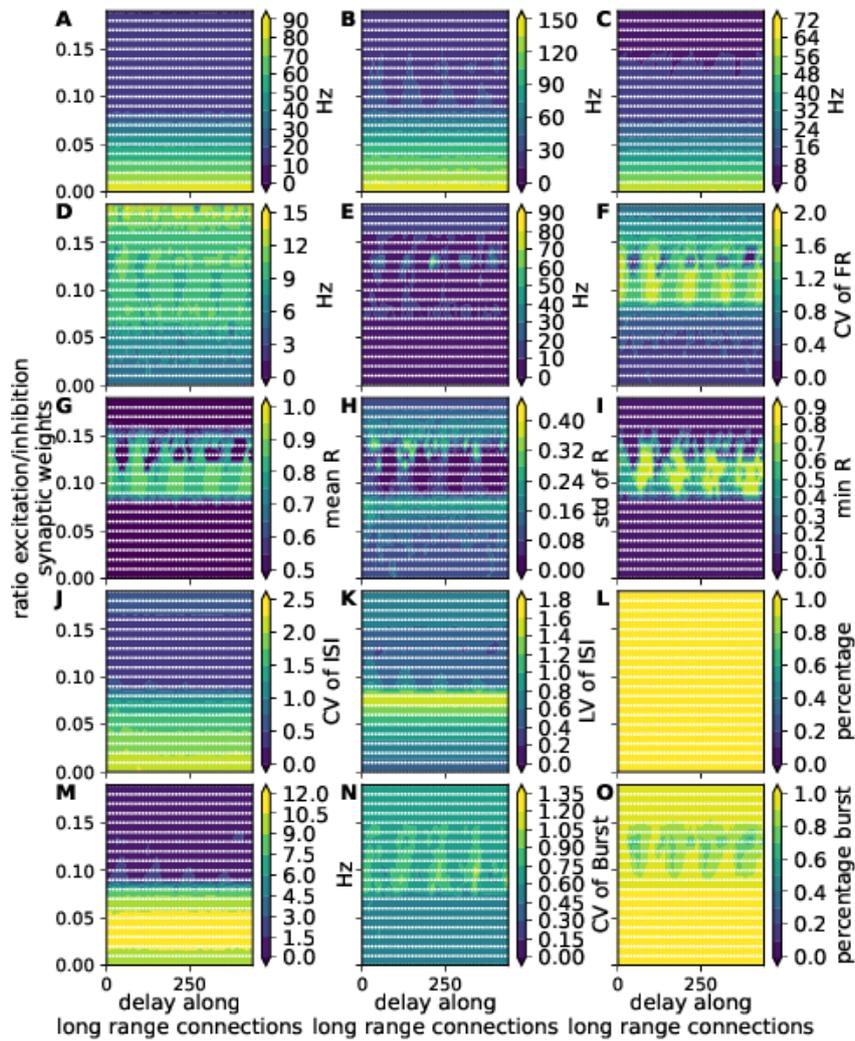
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.25. – Parameter space of ratio excitatory/inhibitory synaptic weights and delay of the long-range connections without noise

Each graphic represents a measure of the network without noise. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the ratio excitatory/inhibitory synaptic weight and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. Red areas represent simulations in which less than 95 % of neurons generate fewer than 20 spikes during the 10 s of analysis. The measures of the network are mean (A), maximum (B), and minimum (C) firing rate of the network, the dominant frequency of the mean phase (D) and the histogram with bin of 1 ms (E), the coefficient of variance of the histogram with bin of 3 ms (F), mean (G), standard deviation (H) and the minimum (I) of the order-parameter, coefficient of variation (J) and the local variation (K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes (L), the mean firing rate of the burst (M), coefficient of variation of the start of the burst (N), the percentage of the neurons with more than 20 bursts (O) (see supplementary note 1 for more details).

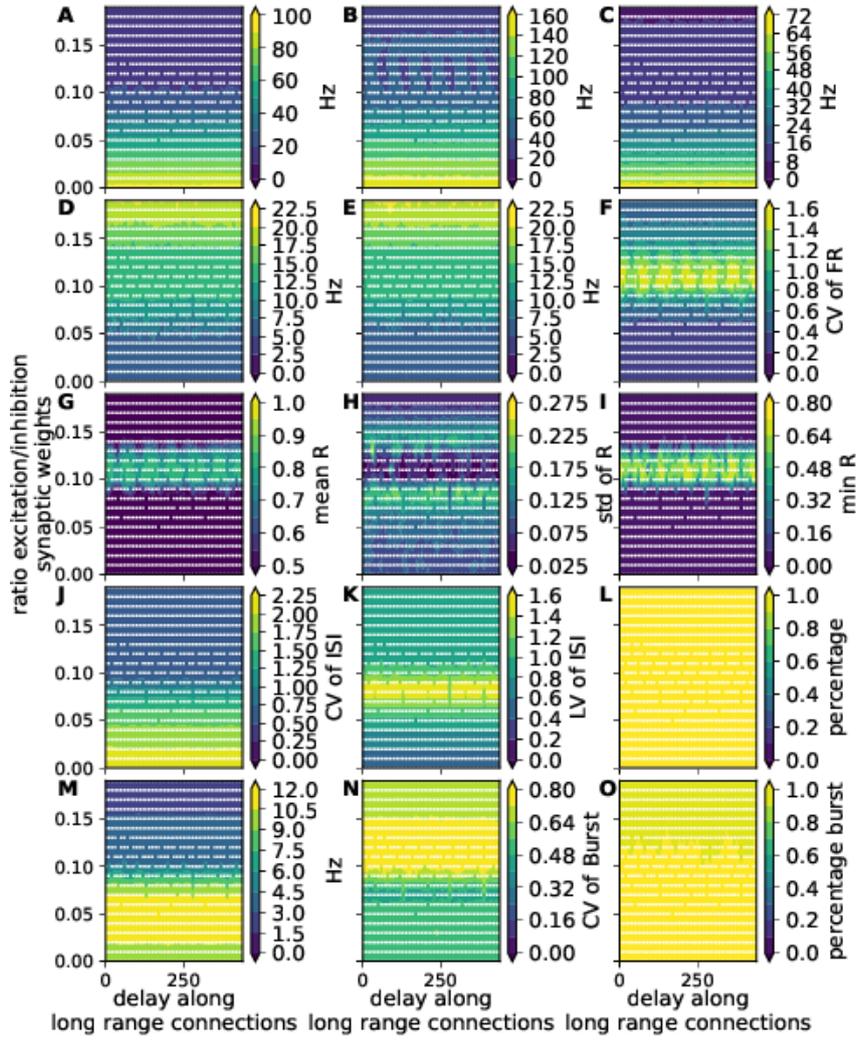
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.26. – Parameter space of ratio excitatory/inhibitory synaptic weight and delay of the long-range connections with white noise (variance :1800pA)

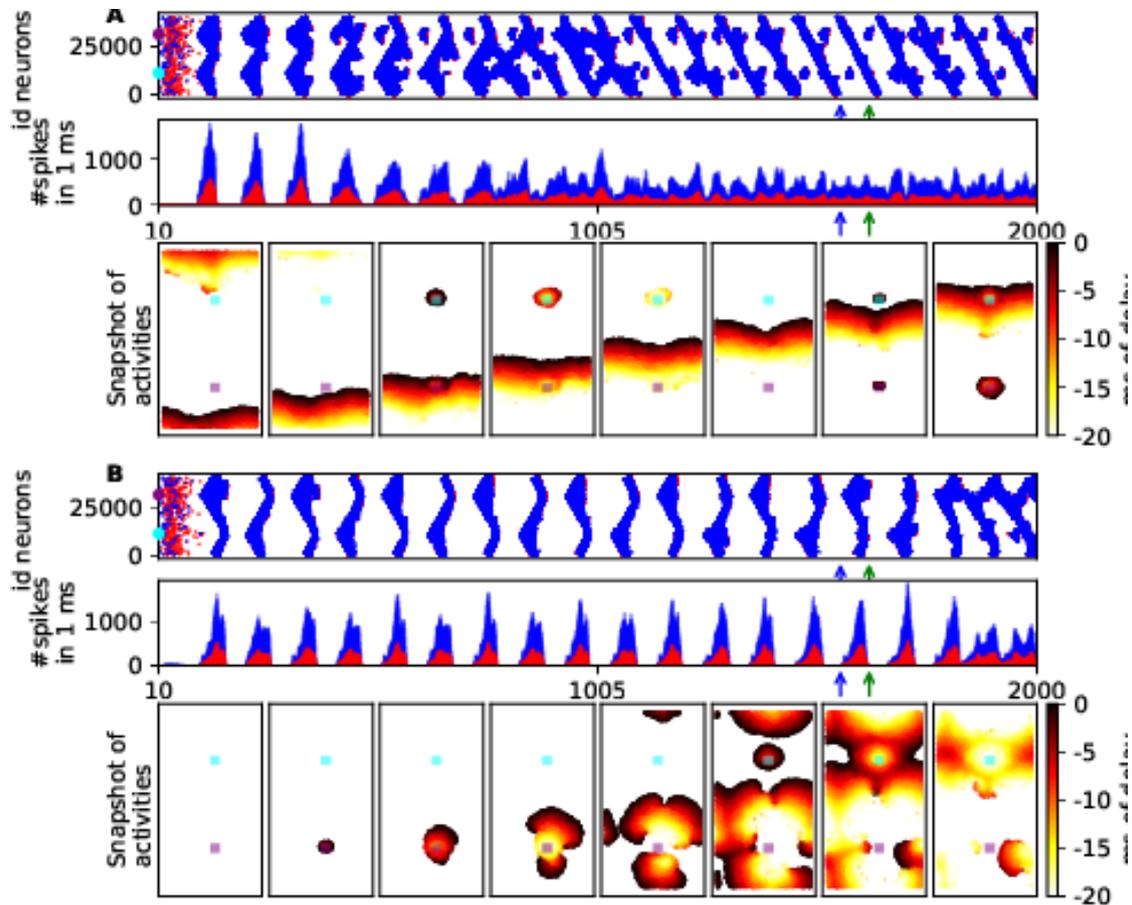
Each graphic represents a measure of the network with noise, which has a variance of 1800pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the ratio excitatory/inhibitory synaptic weight and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum (B), and minimum (C) firing rate of the network, the dominant frequency of the mean phase (D) and the histogram with a bin of 1 ms (E), the coefficient of variance of the histogram with a bin of 3 ms (F), mean (G), standard deviation (H) and the minimum (I) of the order-parameter, coefficient of variation (J) and the local variation (K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes (L), the mean firing rate of the burst (M), coefficient of variation of the start of the burst (N), the percentage of the neurons with more than 20 bursts (O) (see supplementary note 1 for more details).

F. Supplementary material of chapter 4 – F.5. Supplementary Figures



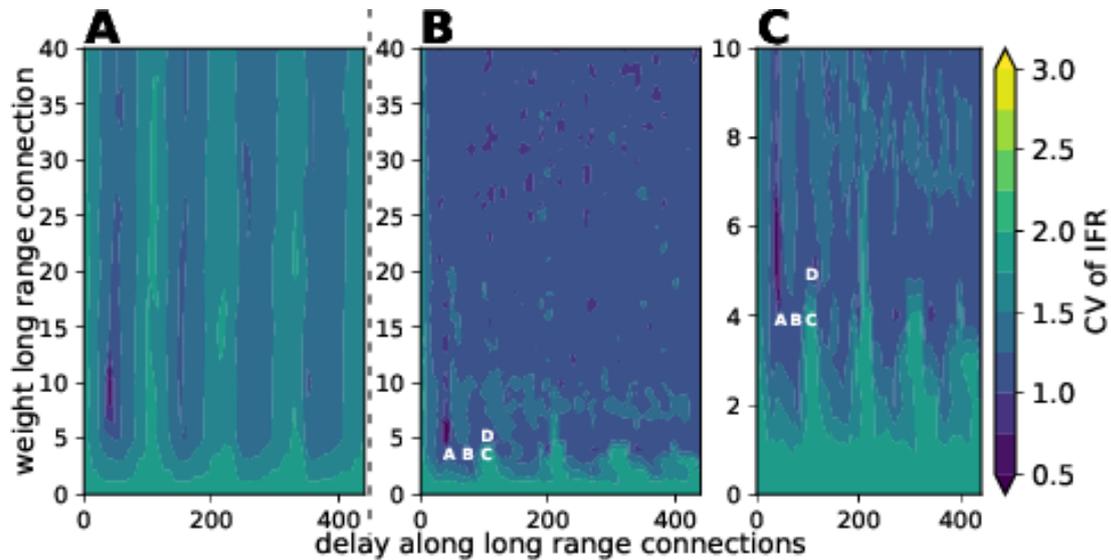
SUPPLEMENTARY FIGURE F.27. – Parameter space of ratio excitatory/inhibitory synaptic weight and delay of the long-range connections with white noise (variance :2600pA)

Each graphic represents a measure of the network with noise, which has a variance of 2600pA. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the ratio excitatory/inhibitory synaptic weight and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).



SUPPLEMENTARY FIGURE F.28. – Two characteristics initial condition with uni-direction long-range connection for different weight of long-range connection without noise

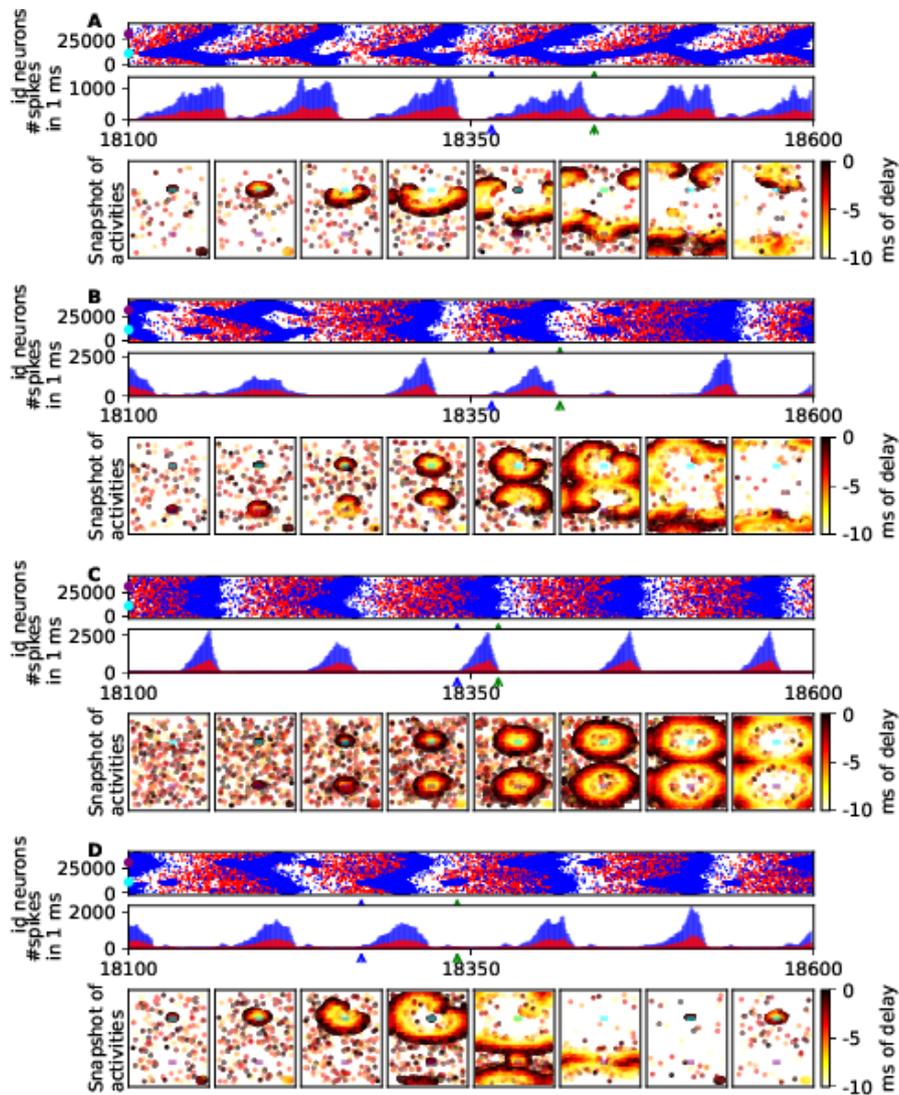
Simulation of the network of 28000 excitatory neurons and 12000 inhibitory neurons for different weights of long-range connection (A :40.0 nS and B :4.0 nS) and with a delay of the long-range connections equals 100 ms. The index is ordered by cells of the grid, with excitatory neurons followed by inhibitory neurons in the cell. Each simulation is represented by spike trains of the neurons and the associated histogram with a bin of 1 ms between 18.1 s and 18.6 s. The blue and green arrows define the start and the end of the eight regular snapshots of the network activities (A :1555 ms, 1564.3 ms, 1573.6 ms, 1582.9 ms, 1592.1 ms, 1601.4 ms, 1610.7 ms, 1620 ms; B :1555 ms, 1564.3 ms, 1573.6 ms, 1582.9 ms, 1592.1 ms, 1601.4 ms, 1610.7 ms, 1620 ms). On the histogram and eight snapshots, the blue represents the source population, and the purple represents the target. **A** Traveling wave **B** Synchronization initialisation



SUPPLEMENTARY FIGURE F.29. – Comparison of the uni and bi-direction of long-range connection

The network synchronization is quantified by the coefficient of variation of firing rate (no overlapping window of 3 ms) for 10 s after a transition period of 12 s. Parameter space network synchronization analysis is presented, showing the relationship between the delay of the long-range connection and the weight of long-range connections for uni-direction (A : same as the Figure 4.3C) and for bi-direction (B). C Aggrandisement of the graphic B on the low values of the weight. The A, B, C, and D in the graphic B and C are parameters set chosen for the simulation shown Figure F.30.

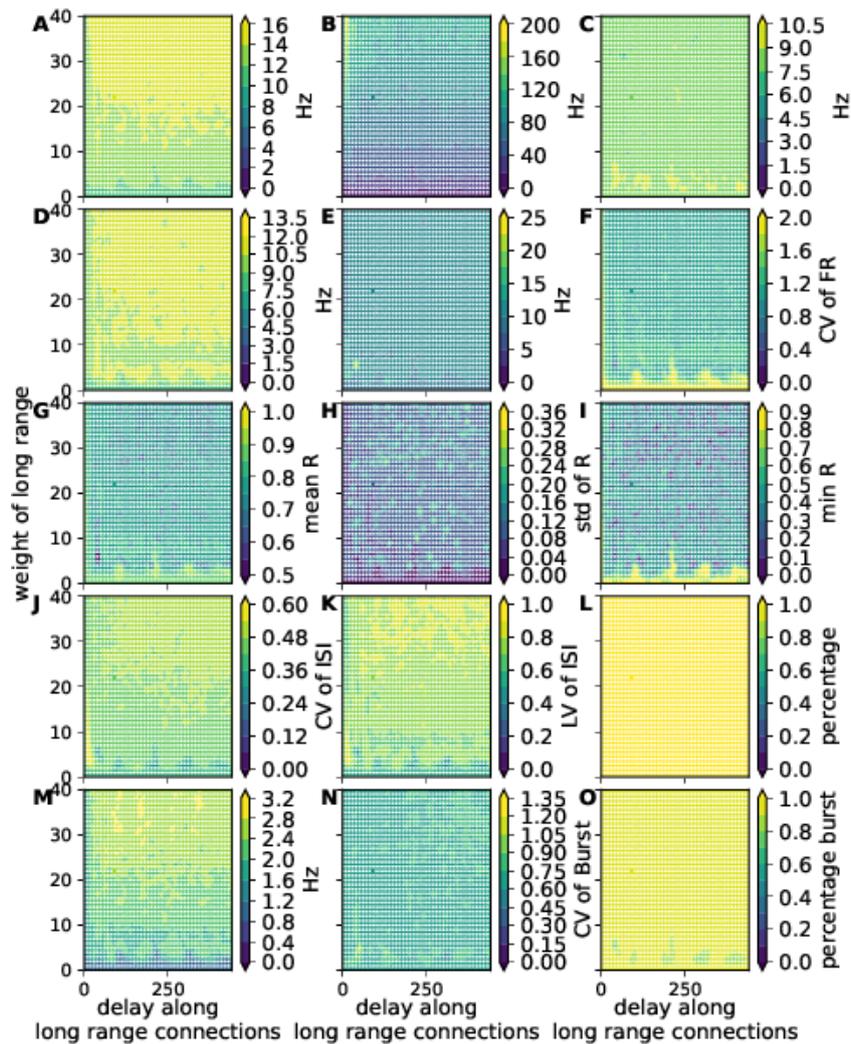
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE E.30. – Four characteristic network dynamics with bi-directional long-range connections

A network of 28,000 excitatory and 12,000 inhibitory adaptive exponential integrate and fire neurons was simulated in the presence of white noise, with different synaptic weights (A, B, C :4.0 nS, and D :5.0 nS) and different delays of the long-range connections (A, D : 40 ms, B : 70 ms, C : 100 ms). The index is ordered by cells of the grid, with excitatory neurons followed by inhibitory neurons of the cell. Each simulation is represented by spike trains of the neurons and an associated histogram with a bin size of 1 ms between 18.1 s and 18.6 s. The blue and green arrows indicate the start and end of the eight regular snapshots of the network activities, respectively. On the histogram and eight snapshots, blue represents the source population, and purple represents the target population. **A** shows Quasi-travelling waves; **B** illustrates two regular partial bumps; **C** displays two regular bumps; and **D** shows alternating diffuse bump and not diffuse bump.

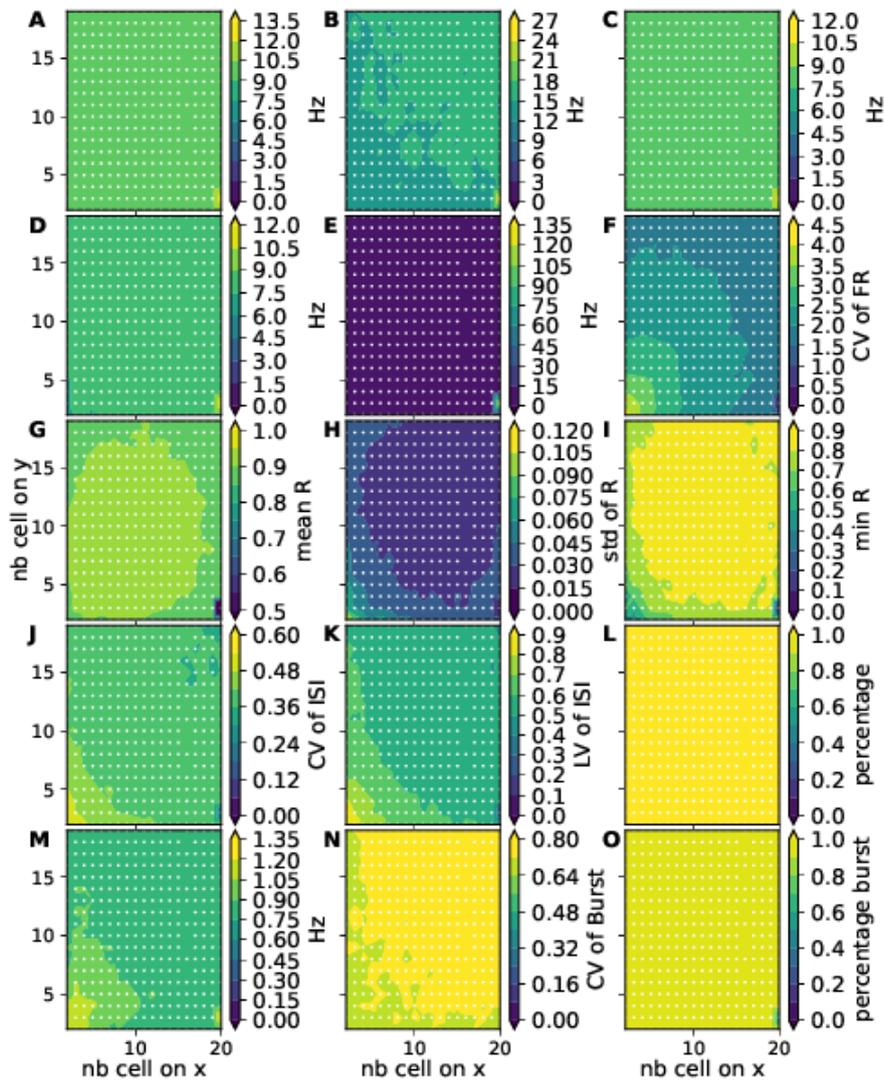
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.31. – Parameter space of weight and delay of the long-range connections with white noise and bi-directional

Each graphic represents a measure of the network with noise, which has a variance of 1800pA, and the long-range connection is bi-directional. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (the weight and the delay of the long-range connection). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum(B), and minimum(C) firing rate of the network, the dominant frequency of the mean phase(D) and the histogram with a bin of 1 ms(E), the coefficient of variance of the histogram with a bin of 3 ms(F), mean(G), standard deviation(H) and the minimum(I) of the order-parameter, coefficient of variation(J) and the local variation(K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes(L), the mean firing rate of the burst(M), coefficient of variation of the start of the burst(N), the percentage of the neurons with more than 20 bursts(O) (see supplementary note 1 for more details).

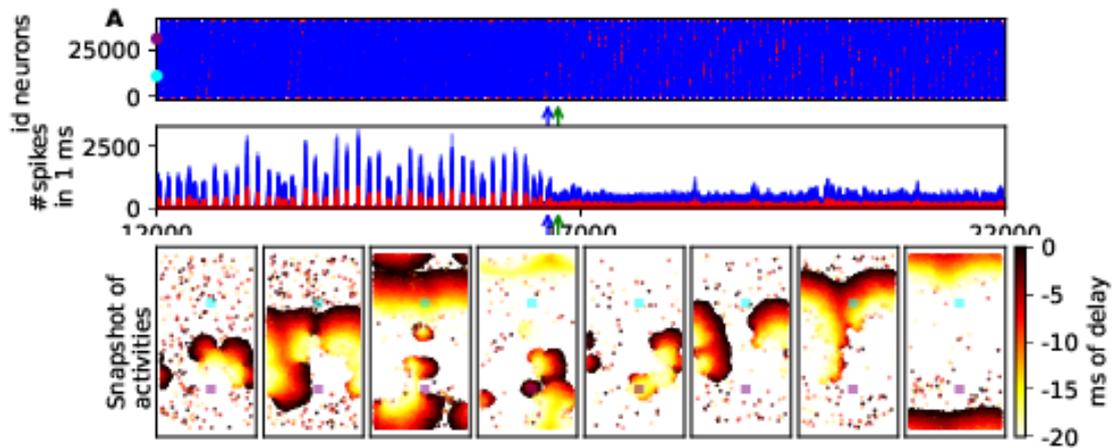
F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.32. – Parameter space of the size of the grid on these two dimensions

Each graphic represents a measure of the network, which does not have a long-range connection in this case. Each white point represents the analysis of one realization of 10 s after a transient period of 12 s for a modification of two parameters (number of cells for the first and second dimensions). The gradient of colors is realized from a linear interpolation of the measures. The measures of the network are mean (A), maximum (B), and minimum (C) firing rate of the network, the dominant frequency of the mean phase (D) and the histogram with a bin of 1 ms (E), the coefficient of variance of the histogram with a bin of 3 ms (F), mean (G), standard deviation (H) and the minimum (I) of the order-parameter, coefficient of variation (J) and the local variation (K) of the inter-spiking interval, the percentage of the neurons with more than 20 spikes (L), the mean firing rate of the burst (M), coefficient of variation of the start of the burst (N), the percentage of the neurons with more than 20 bursts (O) (see the supplementary note 1 for more details).

F. Supplementary material of chapter 4 – F.5. Supplementary Figures



SUPPLEMENTARY FIGURE F.33. – One simulation transition of state
 Simulation of the network of 28000 excitatory neurons and 12000 inhibitory neurons in the presence of white noise with spike-triggered adaptation b equals 90.0 pA and with a delay of the long-range connections equals 120 ms. The index is ordered by cells of the grid with excitatory neurons, followed by the cell's inhibitory neurons. Each simulation is represented by spike trains of the neurons and the associated histogram with a bin of 1 ms between 12.0 s and 22.0 s. The blue and green arrows define the start and the end of the eight regular snapshots of the network activities. On the histogram and eight snapshots, the blue represents the source population, and the purple represents the target. A State transition between partial bump and traveling waves.

CHAPTER G

Supplementary material of Annex A

Contents of Chapter G

G.1. Supplementary Notes	319
G.1.1. Supplementary Note 1 : Additional analysis	319
G.1.1.1. Variability of the measure of the firing rate	319
G.1.1.2. Estimation of coexistence of fixed-points	319
G.1.1.3. Sensitive analysis of polynomial coefficients of the phe- nomenal threshold of the transfer function	319
G.1.1.4. Existence of negative derivation for null firing rate	319
G.1.2. Supplementary Note 2 : Data available	320
G.1.3. Supplementary Note 3 : Supplementary results of spiking neural network	321
G.1.3.1. Firing rate and associated measures	321
G.1.3.2. Inter-spiking interval and associated measures	321
G.1.3.3. Phase and associated measures	321
G.1.3.4. Bursts and associated measures	322
G.1.3.5. Relation with input firing rate	323
G.1.4. Supplementary Note 4 : Supplementary results of mean-field	324
G.1.4.1. Firing rate	324
G.2. Supplementary Tables	325
G.2.1. Supplementary Table 1 : Description of the network simulation simulation	325
G.2.2. Supplementary Table 2 : Description of the mean-field simulation	331
G.3. Supplementary Figures	335

G. Supplementary material of Annex A –

Supplementary Note G.1.1	Additional analysis
Supplementary Note G.1.2	Data available
Supplementary Note G.1.3	supplementary results of spiking neural network
Supplementary Note G.1.4	Supplementary results of mean-field
Supplementary Table G.1	Description of the network simulation simulation
Supplementary Table G.2	Description of the mean-field simulation
Supplementary Figure G.1	Variability of fixed-point from network simulation with and without adaptation
Supplementary Figure G.2	Bifurcation diagram of external excitatory input for spike-trigger adaptation equals 0.0 pA
Supplementary Figure G.3	Bifurcation diagram of external excitatory input for spike-trigger adaptation equals 30.0 pA
Supplementary Figure G.4	Bifurcation diagram of external excitatory input for spike-trigger adaptation equals 60.0 pA
Supplementary Figure G.5	Variability of the measure of the firing rate
Supplementary Figure G.6	Variability of the measure of the firing rate
Supplementary Figure G.7	Estimation and comparison of the high fixed-point
Supplementary Figure G.8	Estimation and comparison of the low fixed-point
Supplementary Figure G.9	Fitting of Transfer functions full
Supplementary Figure G.10	Estimation of error of transfer function for neurons in the presence of negative current
Supplementary Figure G.11	Negative derivative of the transfer function for zeros firing rate
Supplementary Figure G.12	Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 7Hz and adaptation (b = 60.0pA)
Supplementary Figure G.13	Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 0Hz
Supplementary Figure G.14	Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 0Hz and adaptation (b = 60.0pA)

G.1. Supplementary Notes

G.1.1 Supplementary Note 1 : Additional analysis

Additional analysis was performed to complete the main analysis

G.1.1.1 Variability of the measure of the firing rate

The spiking network with a Poisson generator does not have a constant firing rate over time. Measuring the firing rate variability is crucial for comparing with a mean-field because it quantifies the tolerable range of difference with the mean-field. The variability of the mean and standard deviation of the firing rate is estimated with the measure's distribution of 5000 values taken on 30 s of simulation after a transient of 10 s. The only parameter for calculating the firing rate from a spike train using a rectangular window is the window's size. We analyze the influence of the window's size using a log scale between 0.2 ms and 3489.9 ms.

G.1.1.2 Estimation of coexistence of fixed-points

A first parameter exploration of the external input gives the values where the network is switching between a fixed-point around 50 Hz to a fixed-point around 200 Hz of excitatory firing rate. From a value of external firing close to this shift, a longer simulation is run where the external input is reduced or increased by 1Hz every 10s. The variation of the mean firing rate estimates the range of coexistence of fixed-points by considering that 10 s is long enough to demonstrate the existence of a steady-state.

G.1.1.3 Sensitive analysis of polynomial coefficients of the phenomenal threshold of the transfer function

The sensitivity analysis in this study quantifies the variation of a fixed-point depending on a number of significant digits of the coefficients. We choose to quantify the variability of the fixed-point for $b = 0$ and for an excitatory firing rate equal 10 Hz. A numerical continuation of this fixed-point is performed for each coefficient. From these numerical continuations, we quantify the variation of the polynomial coefficients for the variability of 0.1Hz, 1Hz and 10Hz of the firing rate of the fixed-point. By taking the higher variability of coefficients, we find the minimal number of digits for these three precision of the fixed-point. However, this analysis does not consider the variance due to the variation of the combination of two or more coefficients.

G.1.1.4 Existence of negative derivation for null firing rate

The transfer function is not defined for a negative firing rate and the mean-field should not define negative derivation for a null firing rate. However, for specific value

G. Supplementary material of Annex A – G.1. Supplementary Notes

of the mean-field is not the case. For identifying these values, we assume that the firing rate of a population is null but also its variance and the covariance. By consequence, the first order equation become :

$$0 < T \frac{\partial v_e}{\partial t} \quad (\text{G.1})$$

$$0 < (\mathcal{F}_e - v_e) + \frac{1}{2} c_{ee} \frac{\partial^2 \mathcal{F}_e}{\partial v_e \partial v_e} + \frac{1}{2} c_{ei} \frac{\partial^2 \mathcal{F}_e}{\partial v_e \partial v_i} + \frac{1}{2} c_{ie} \frac{\partial^2 \mathcal{F}_e}{\partial v_i \partial v_e} + \frac{1}{2} c_{ii} \frac{\partial^2 \mathcal{F}_e}{\partial v_i \partial v_i} \quad (\text{G.2})$$

$$0 < (\mathcal{F}_e - 0.) + \frac{1}{2} 0. \frac{\partial^2 \mathcal{F}_e}{\partial v_e \partial v_e} + \frac{1}{2} 0. \frac{\partial^2 \mathcal{F}_e}{\partial v_e \partial v_i} + \frac{1}{2} 0. \frac{\partial^2 \mathcal{F}_e}{\partial v_i \partial v_e} + \frac{1}{2} c_{ii} \frac{\partial^2 \mathcal{F}_e}{\partial v_i \partial v_i} \quad (\text{G.3})$$

$$0 < \mathcal{F}_e + \frac{1}{2} c_{ii} \frac{\partial^2 \mathcal{F}_e}{\partial v_i \partial v_i} \quad (\text{G.4})$$

$$c_{ii} < 2 \frac{\mathcal{F}_e}{\frac{\partial^2 \mathcal{F}_e}{\partial v_i \partial v_i}} \quad (\text{G.5})$$

The supplementary figure [G.11](#) displays the minimal positive value of the covariance where this condition is respected.

G.1.2 Supplementary Note 2 : Data available

In the Zenodo repository, the data from simulations and analyses are available. The available bifurcation analysis results from the numerical continuation of the mean-field for all different values of adaptation spike trigger b (0., 30., 60.). However, the bifurcations for the sensitivity analysis are not available. The repository contains the result of the negative derivative for a null firing rate (see supplementary figure [G.11](#)) for excitatory and inhibitory transfer functions. It also contains the data for fitting the transfer function and the result of the fit. The only available spike trains are the output of the short simulation with an external input rate of 10, 60 and 80 Hz used in the figure [A.1](#). The only additional data not saved in a database is the firing rate of the steady-state of the analysis of the coexistence of fixed-point for each value of b (see supplementary figure [G.7](#) and [G.8](#)). The Zenodo repository contains multiple databases : one database contains the analysis of the spike trains of the steady-state for different external input firing rates for each seed (see supplementary figure [G.1](#)), one database for each long simulation used in the analysis of the variability of firing rate (see supplementary figure [G.5](#) and [G.6](#)), one database for each parameter exploration of the oscillatory input parameters of neural network (see figure [A.4](#)) and a unique database for the parameters exploration of oscillatory input parameters for the mean-field (see figure [A.5](#)).

G.1.3 Supplementary Note 3 : Supplementary results of spiking neural network

G.1.3.1 Firing rate and associated measures

The firing rate is the number of generated spikes by a neuron during 1 s. One measure of population synchronization quantifies the fluctuation of the population mean firing rate (for windows of 0.1 and 1 ms) with the coefficient of variation [41]. Additionally, one estimation of neural population frequency uses the dominant frequency and its associated power of the population mean firing rate (for a window of 1 ms) using Welch's method [358] from the elephant package [89]. (Databases record average ('rates_average'), standard deviation ('rates_std'), minimum ('rate_min') and maximum ('rate_max') mean firing rates; for each (0.1 ms and 1 ms) the coefficient of variation ('cvs_IFR_0_1ms' and 'cvs_IFR_1ms'); and for a window of 1 ms, the dominant frequency and its power ('frequency_hist_1_freq' and 'frequency_hist_1_val').

G.1.3.2 Inter-spiking interval and associated measures

The inter-spiking interval is the time between 2 spikes. The distribution of this interval characterizes the regularity or irregularity of neuron spike generation [41], which is quantified by two measures : coefficient of variation of firing rate [41] and local variation [321][320]. These measures are provided by Elephant python package [89]. These measures are determined for viable estimation by considering only neurons with more than 20 spikes in 10 s for analysis. (Databases record average ('ISI_average'), standard deviation ('ISI_std'), minimum ('ISI_min'), and maximum ('ISI_max') inter-spiking intervals; the average and standard deviation over the population of the regularity measures ('cvs_ISI_average', 'cvs_ISI_std', 'lvs_ISI_average' and 'lvs_ISI_std'); and the percentage of neurons with more than 20 spikes ('percentage').

G.1.3.3 Phase and associated measures

Kuramoto proposed a model for describing and analyzing synchronization of coupled intrinsic oscillators [211], based on the following equation :

$$\frac{\partial \theta_i}{\partial t} = \omega_i + \frac{K}{N} \sum_{j=1}^N \sin(\theta_j - \theta_i), \quad i = 1 \dots N, \quad (\text{G.6})$$

where the system is composed of N limit-cycle oscillators, with phases θ_i and coupling constant K.

The phase of a neuron can be defined by approximating the voltage membrane as an intrinsic oscillator or by approximating the phase using a constant velocity between 2 spikes, similar to Borges et al. 2017 [32]. This study used the second approximation

described in the following equation :

$$\psi(t) = 2\pi m + 2\pi \frac{t - t_m}{t_{m+1} - t_m}, \forall t \in [t_m, t_{m+1}] \quad (\text{G.7})$$

where t_m is the m-th spike of the neuron.

One measure of synchronization is the estimation of Kuramoto order-parameter of a population, i.e. the absolute values of the mean phase. Additionally, one estimation of neural population frequency uses the dominant frequency and its associated power of the mean phase (for a window of 0.1 ms) using Welch's method [358] from the elephant package [89]. (Databases record average ('synch_Rs_average'), standard deviation ('synch_Rs_std'), minimum ('synch_Rs_min') and maximum ('synch_Rs_max') of the Kuramoto

order-parameter; the start time and end time of the mean phase estimation ('synch_Rs_times_init' and 'synch_Rs_times_end'); and for a window of 0.1 ms, the dominant frequency and its power ('frequency_phase_freq' and 'frequency_phase_val').)

G.1.3.4 Bursts and associated measures

Bursts are consecutive spikes with inter-spike intervals of less than 10 ms. Additionally, the regularity of bursts is analyzed similarly to inter-spike intervals. For neurons with more than 20 bursts, the regularity of the interval of the beginning time of bursts and also their ending time is quantified with the coefficient of variation and the local variation. (Databases record average ('burst_count_average', 'burst_nb_average' and 'burst_rate_average'), standard deviation ('burst_count_std', 'burst_nb_std' and 'burst_rate_std'), maximum ('burst_count_max', 'burst_nb_max' and 'burst_rate_max') and minimum ('burst_count_min', 'burst_nb_min' and 'burst_rate_min') of a number of spikes in the sequence of burst, number of burst by neurons during the analysis and mean bursting rates; the percentage of neurons which have one burst ('percentage_burst'); average ('burst_interval_average'), standard deviation ('burst_interval_std'), minimum ('burst_interval_min'), maximum ('burst_interval_max'); and for neurons with more than 20 bursts, the average and the standard deviation of the coefficient of variation ('burst_cv_begin_average', 'burst_cv_end_average', 'burst_cv_begin_std', 'burst_cv_end_std') and local variation ('burst_lv_begin_std', 'burst_lv_end_std', 'burst_lv_begin_average', 'burst_lv_end_average') of interval of the beginning time of bursts and also their ending time.

G.1.3.5 Relation with input firing rate

The mean phase difference between the oscillatory input and the oscillatory histogram of the spiking neural network :

$$meanphase = \frac{1}{N} \sum_{n=1}^N \exp(j(\theta_{input}(t, n) - \theta_{network}(t, n))) \quad (G.8)$$

where θ_{input} is the analytic signal given by the Hilbert transformation of the sinusoidal input, and $\theta_{network}$ is the analytic signal provided by the Hilbert transformation of the network oscillation signal. The absolute value of the *meanphase* corresponds to the phase locking value as defined by Lachaux et al. 1999 [217]. The angle of the *meanphase* corresponds to the phase shift [124]. This measure is calculated for a histogram with a bin of 0.1 ms, 1.0 ms and 5 ms and a smoothed histogram with an average sliding window of 5 ms. Additionally, the time scale of the network is only analyzed for networks with oscillatory input. The time scale is quantified using the auto-correlation function of elephant package [89]. The function is the implementation of the method proposed by Wieland et al. 2015 [361] :

$$\tau_{corr} = \int_{-\tau_{max}}^{\tau_{max}} \left[\frac{\hat{C}(\tau)}{\hat{C}(0)} \right]^2 d\tau \quad (G.9)$$

where $\hat{C}(\tau) = \langle x(t)x(t-\tau) \rangle \langle x \rangle^2 - \langle x \rangle \delta(\tau)$ with $x(t) = \sum_{n=1}^N \delta(t-t_n)$ for t_n is the n-th spike time of the spike trains.

(Databases with oscillatory input record phase locking value ('PLV_0_1ms', 'PLV_1ms', 'PLV_5ms', 'PLV_w5ms'), phase shift ('PLV_angle_0_1ms', 'PLV_angle_1ms', 'PLV_angle_5ms', 'PLV_angle_w5ms') and 'MeanPhaseShift_0_1ms', 'MeanPhaseShift_1ms', 'MeanPhaseShift_5ms', 'MeanPhaseShift_w5ms'), the coefficient of variation of the histogram ('cvs_IFR_0_1ms', 'cvs_IFR_1ms', 'cvs_IFR_5ms', 'cvs_IFR_w5ms'), maximum of firing rate (max_IFR_0_1ms, max_IFR_1ms, max_IFR_5ms, max_IFR_w5ms) for a bin of 0.1 ms, 1 ms, 5 ms and smoothing with windows of 5 ms.)

G.1.4 Supplementary Note 4 : Supplementary results of mean-field

G.1.4.1 Firing rate

The firing rate corresponds to the variable of the mean-field. This measure variable is quantified, using similar methods that for the histogram of spiking neural network, phase locking value, phase shift and dominant frequency. (Databases with oscillatory input record average ('mean_rates'), standard deviation ('std_rates'), minimum ('min_rates'), maximum ('max_rate') and dominant frequency ('frequency_dom') of firing rates and phase locking value ('PLV_value') and phase shift ('PLV_angle') between the oscillatory input and the firing rate)

G.2. Supplementary Tables

G.2.1 Supplementary Table 1 : Description of the network simulation simulation

A1		NEST : model summary	
software		NEST	
topology		Erdős - Rényi [108]	
population		excitatory and inhibitory	
connectivity		random connectivity without self-recurrent connections	
neuron model		adaptive exponential leaky integrate and fire neurons, fixed threshold and fixed absolute refractory time	
synapse model		conductance-based exponential shape	
plasticity		none	
input		independent fixed rate Poisson generator spike trains to all neurons	
measurement		spike activity	
A2		NEST : software	
version		alpha 3.0.0 (8f5a5f) [152]	
integrator method		4th order Runge-Kutta-Fehlberg method	
integration step		0.1 ms	
number of seeds		8 (master seed : 0-30 or 46)	
simulation time		2s (example), 5s(normal, transient :1s), 40s(long, transient :10s) or variable(test stability, transient :10s)	
A3		NEST : populations	
name	elements	size	
E	aeif_cond_exp	$N_e = (1 - g_{inh})N = 8000$	
I	aeif_cond_exp	$N_i = g_{inh}N = 2000$	
P_{ext}	Poisson generator	8000	
$P_{sin ext}$	Sinusoidal Poisson generator	1	

G. Supplementary material of Annex A – G.2. Supplementary Tables

A4 NEST : neuron model	
name	aeif
type	adaptive exponential leaky integrator and fire [37] with conductance synapse
subthreshold dynamics	$C_m \frac{dV_m}{dt} = -g_L(V_m - E_L) + g_L \Delta_T e^{\frac{V_m - V_{th}}{\Delta_T}}$ $-g_e(t)(V_m - E_{ex}) - g_i(t)(V_m - E_{in}) + I_e$ $-w + I_e$ $\tau_w \frac{dw}{dt} = a(V_m - E_L) - w$
reset condition	<p>for $t^{(f)} = \{t V_m(t) \geq V_{peak}\}$</p> <ul style="list-style-type: none"> — $V_m([t^{(f)}; t^{(f)} + t_{ref}]) = V_{reset}$ — $w([t^{(f)}]) = w([t^{(f)}]) + b$
NEST : synapse model	
name	cond_exp
type	post-synaptic conductance in the form of truncated exponentials
coupling equation	$g_e(t) = \sum_{t_j^{(f)}} w_j \exp_trunc(t - t_j, \tau_{syn}) \text{ with } w_j > 0.0$ $g_i(t) = \sum_{t_j^{(f)}} w_j \exp_trunc(t - t_j, \tau_{syn}) \text{ with } w_j < 0.0$ $\exp_trunc(t, \tau) = e^{1 - \frac{t}{\tau}} \text{Heaviside}(t)$

G. Supplementary material of Annex A – G.2. Supplementary Tables

A5 NEST : neuron model parameters		
excitatory neurons		
C_m	capacity of the membrane	200.0 pF
t_{ref}	duration of refractory period	5.0 ms
V_{reset}	reset value for V_m after a spike	-55.0 mV
E_L	leak reversal potential	-63.0 mV
g_L	leak conductance	10.0 nS
Δ_T	slope factor	2.0 mV
V_{peak}	spike detection threshold	0.0 mV
a	subthreshold adaptation	0.0 nS
b	spike-triggered adaptation	0.0/30.0/60.0pA
τ_w	adaptation time constant	500.0 ms
V_{th}	spike initiation threshold	-50.0 mV
I_e	constant external input current	0.0 pA
E_{ex}	excitatory reversal potential	0.0 mV
E_{in}	inhibitory reversal potential	-80.0 mV
V_m	initialization of the voltage membrane	-65.0 ± 100.0 mV
w	initialization of adaptation current	200.0 ± 200 pA
g_e	initialization of excitatory synapse input	0.0 pA
g_i	initialization of inhibitory synapse input	0.0 pA
inhibitory neurons		
C_m	capacity of the membrane	200.0 pF
t_{ref}	duration of refractory period	5.0 ms
V_{reset}	reset value for V_m after a spike	-65.0 mV
E_L	leak reversal potential	-65.0 mV
g_L	leak conductance	10.0 nS
Δ_T	slope factor	0.5 ms
V_{peak}	spike detection threshold	0.0 mV
a	subthreshold adaptation	0.0 nS
b	spike-triggered adaptation	0.0 pA
τ_w	Adaptation time constant	1.0 ms
V_{th}	spike initiation threshold	-50.0 mV
I_e	constant external input current	0.0 pA
E_{ex}	excitatory reversal potential	0.0 mV
E_{in}	inhibitory reversal potential	-80.0 mV
V_m	initialization of the voltage membrane	-65.0 ± 100.0 mV
w	initialization of adaptation current	200.0 ± 200 pA
g_e	initialization of excitatory synapse input	0.0 pA
g_i	initialization of inhibitory synapse input	0.0 pA

G. Supplementary material of Annex A – G.2. Supplementary Tables

A6 NEST : input	
Poisson generator	
equation	$p(n) = \frac{\lambda^n}{n!} \exp(-\lambda)$
implementation algorithm	Ahrens and Dieter 1982 [2]
excitatory firing rate λ_{ex}	0-100 Hz
Sinusoidal Poisson generator	
equation	$p(n) = \frac{(rate + a \sin(f * n))^n}{n!} \exp(-rate + a \sin(f * n))$
amplitude a	0-50 Hz
frequency f	0-50 Hz
average $rate$	0-50 Hz

G. Supplementary material of Annex A – G.2. Supplementary Tables

A7 NEST : connectivity				
parameter synapses				
τ_{ex}	rise time of excitatory synaptic conductance			5.0ms
τ_{in}	rise time of inhibitory synaptic conductance			5.0ms
name	source	target	weights	pattern
EE	E	E	$Q_e=1.5$ nS	probability of connections is 0.05 between pairs of neurons. Neurons cannot be directly connected to themselves nor have multiple connections with the same postsynaptic neuron.
EI	E	I	$Q_e=1.5$ nS	probability of connections is 0.05 between pairs of neurons. Neurons cannot be directly connected to themselves nor have multiple connections with the same postsynaptic neuron.
IE	I	E	$Q_i=5.0$ nS	probability of connections is 0.05 between pairs of neurons. Neurons cannot be directly connected to themselves nor have multiple connections with the same postsynaptic neuron.
II	I	I	$Q_i=5.0$ nS	probability of connections is 0.05 between pairs of neurons. Neurons cannot be directly connected to themselves nor have multiple connections with the same postsynaptic neuron.
$P_{ext}E$	P_{ext}	E	$Q_e=1.5$ nS	one generator by neuron.
$P_{sin ext}E$	$P_{sin ext}$	E	$Q_e=1.5$ nS	the generator is connected to a neurons with a probability of 0.05.
A8 NEST : measurement				
state variable	spike time		precision	0.1 ms
spike activities	raster plot		precision	0.1 ms
	simple moving average		windows size	T (5ms)

G. Supplementary material of Annex A – G.2. Supplementary Tables

TABLEAU G.1. – Description of the network simulation simulation

G.2.2 Supplementary Table 2 : Description of the mean-field simulation

B1		TVB : model summary	
software	The Virtual Brain [300]		
Neural Mass model	mean-field of Ad Ex [106, 356]		
connectivity	None		
input	sinusoidal input		
monitor	raw values		
B2		TVB : software	
version	2.6		
integrator method	Heun integrator		
integration step	0.1 ms		
simulation time	20.001 s (including a transient of 2.5 s)		
B3		TVB : input and monitor	
input	sinusoidal input	equation amplitude a frequency f rate $rate$	$rate + a \sin(f * t)$ 0-50 Hz 0-50 Hz 0 or 7 Hz
monitor	Raw monitor	state variable	0.1 ms

G. Supplementary material of Annex A – G.2. Supplementary Tables

B4	TVB : Neural Mass model
name	Mean-Field AdEx [106]
type	Neural Mass model network of adaptive exponential integrate and fire excitatory and inhibitory neurons of second statistical order
equation	$T \frac{\partial v_e}{\partial t} = (\mathcal{F}_e - v_e) + \frac{1}{2} c_{ee} \frac{\partial^2 \mathcal{F}_e}{\partial v_e \partial v_e} + \frac{1}{2} c_{ei} \frac{\partial^2 \mathcal{F}_e}{\partial v_e \partial v_i} + \frac{1}{2} c_{ie} \frac{\partial^2 \mathcal{F}_e}{\partial v_i \partial v_e} + \frac{1}{2} c_{ii} \frac{\partial^2 \mathcal{F}_e}{\partial v_i \partial v_i}$ $T \frac{\partial v_i}{\partial t} = (\mathcal{F}_i - v_i) + \frac{1}{2} c_{ee} \frac{\partial^2 \mathcal{F}_i}{\partial v_e \partial v_e} + \frac{1}{2} c_{ei} \frac{\partial^2 \mathcal{F}_i}{\partial v_e \partial v_i} + \frac{1}{2} c_{ie} \frac{\partial^2 \mathcal{F}_i}{\partial v_i \partial v_e} + \frac{1}{2} c_{ii} \frac{\partial^2 \mathcal{F}_i}{\partial v_i \partial v_i}$ $T \frac{\partial c_{ee}}{\partial t} = (\mathcal{F}_e - v_e) (\mathcal{F}_e - v_e) + c_{ee} \frac{\partial \mathcal{F}_e}{\partial v_e} + c_{ee} \frac{\partial \mathcal{F}_e}{\partial v_e} + c_{ei} \frac{\partial \mathcal{F}_e}{\partial v_i} + c_{ie} \frac{\partial \mathcal{F}_e}{\partial v_i} - 2c_{ee}$ $+ \frac{\mathcal{F}_e (1/T - \mathcal{F}_e)}{N_e}$ $T \frac{\partial c_{ei}}{\partial t} = (\mathcal{F}_e - v_e) (\mathcal{F}_i - v_i) + c_{ee} \frac{\partial \mathcal{F}_i}{\partial v_e} + c_{ie} \frac{\partial \mathcal{F}_i}{\partial v_i} + c_{ei} \frac{\partial \mathcal{F}_e}{\partial v_e} + c_{ii} \frac{\partial \mathcal{F}_e}{\partial v_i} - 2c_{ei}$ $T \frac{\partial c_{ie}}{\partial t} = (\mathcal{F}_i - v_i) (\mathcal{F}_e - v_e) + c_{ee} \frac{\partial \mathcal{F}_i}{\partial v_e} + c_{ie} \frac{\partial \mathcal{F}_i}{\partial v_i} + c_{ei} \frac{\partial \mathcal{F}_e}{\partial v_e} + c_{ii} \frac{\partial \mathcal{F}_e}{\partial v_i} - 2c_{ie}$ $T \frac{\partial c_{ii}}{\partial t} = (\mathcal{F}_i - v_i) (\mathcal{F}_i - v_i) + c_{ie} \frac{\partial \mathcal{F}_i}{\partial v_e} + c_{ei} \frac{\partial \mathcal{F}_i}{\partial v_e} + c_{ii} \frac{\partial \mathcal{F}_i}{\partial v_i} + c_{ii} \frac{\partial \mathcal{F}_i}{\partial v_i} - 2c_{ii}$ $+ \frac{\mathcal{F}_i (1/T - \mathcal{F}_i)}{N_i}$ $\tau_{W_e} \frac{\partial W_e}{\partial t} = -W_e + b_e v_e + a_e (\mu_V(v_e, v_i, W_e) - EL_e)$ $\tau_{W_i} \frac{\partial W_i}{\partial t} = -W_i + b_i v_i + a_i (\mu_V(v_e, v_i, W_i) - EL_i)$
transfer function	$\mathcal{F}_e = \mathcal{F}((v_e + 1e - 6), v_{ext}, v_i, W_e)$ $\mathcal{F}_i = \mathcal{F}((v_e + 1e - 6), v_{ext}, v_i, W_i)$ $\mathcal{F} = \frac{1}{2\tau_V} \cdot \text{Erfc}\left(\frac{V_{thre}^{eff} - \mu_V}{\sqrt{2}\sigma_V}\right)$ $V_{thre}^{eff}(\mu_V, \sigma_V, \tau_V^N = \tau_V \frac{g_L}{C_m}) = P'_0 + \sum_{x \in \{\mu_V, \sigma_V, \tau_V^N\}} P_x \cdot \left(\frac{x - x^0}{\delta x^0}\right)$ $+ \sum_{x, y \in \{\mu_V, \sigma_V, \tau_V^N\}^2} P_{xy} \cdot \left(\frac{x - x^0}{\delta x^0}\right) \left(\frac{y - y^0}{\delta y^0}\right)$ $\mu_G(v_e, v_{ext}, v_i) = ((v_e K_e + v_{ext} K_{ext}) \tau_e Q_e) + (v_i K_i \tau_i Q_i) + g_L$ $\mu_{V_s}(v_e, v_{ext}, v_i, W, \mu_G) = \frac{((v_e K_e + v_{ext} K_{ext}) \tau_e Q_e) E_e + (v_i K_i \tau_i Q_i) E_i + g_L EL_s - W}{\mu_G}$ $\sigma_V(\mu_V, \mu_G) = \sqrt{\sum_{s \in \{e, i\}} K_s v_s \frac{\left(\frac{Q_s}{\mu_G} (E_s - \mu_V) \tau_s\right)^2}{2 \frac{C_m}{\mu_G} + \tau_s}}$ $\tau_V(\mu_V, \mu_G) = \frac{\sum_{s \in \{e, i\}} K_s v_s \left(\frac{Q_s}{\mu_G} (E_s - \mu_V) \tau_s\right)^2}{332 \sum_{s \in \{ex, in\}} K_s v_s \frac{\left(\frac{Q_s}{\mu_G} (E_s - \mu_V) \tau_s\right)^2}{2 \frac{C_m}{\mu_G} + \tau_s}}$

G. Supplementary material of Annex A – G.2. Supplementary Tables

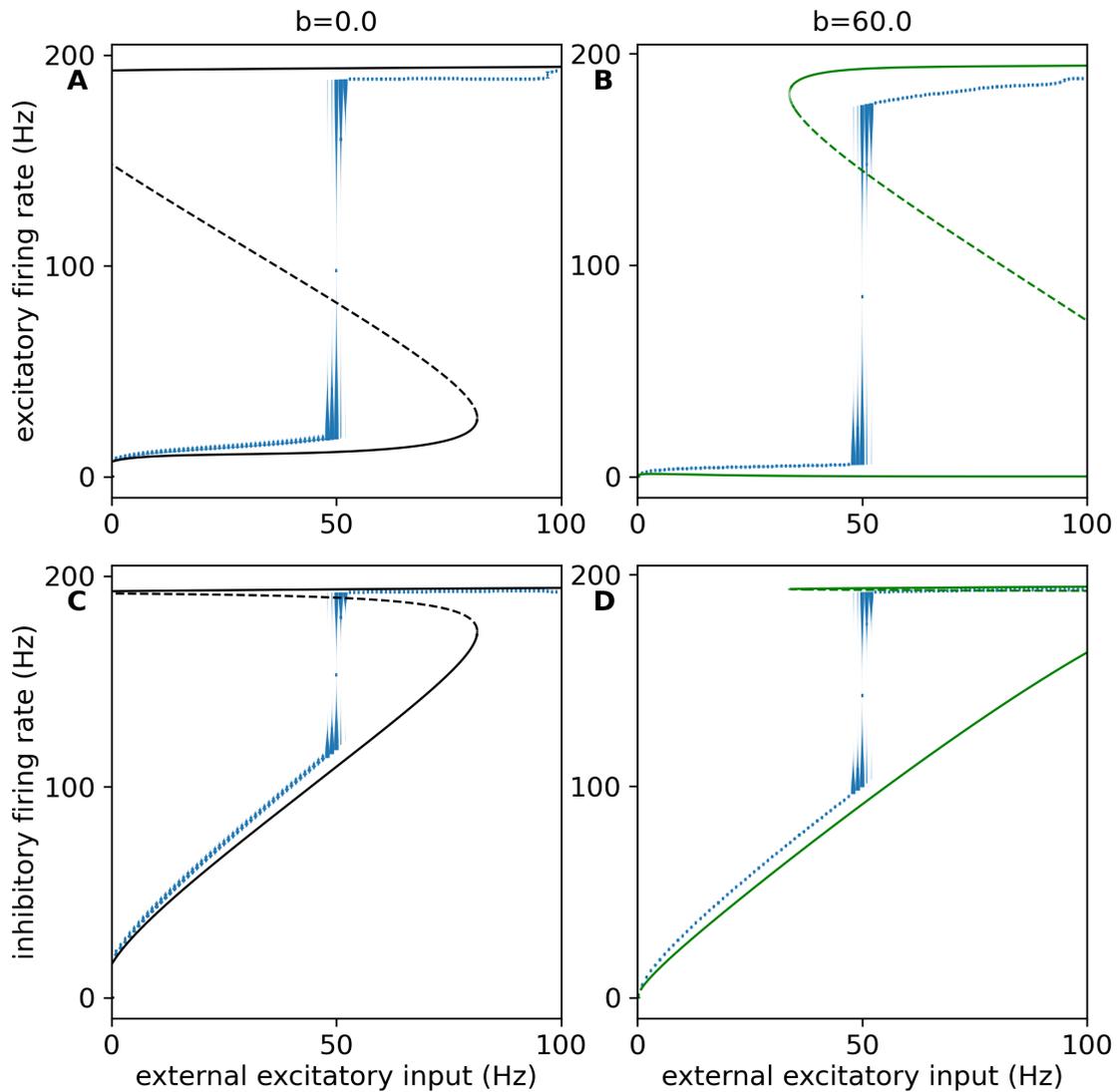
B5	TVB : Neural Mass model parameters	
T	time resolution of the mean-field	5.0 ms
C_m	capacity of the membrane	200.0 pF
EL_e	leak reversal potential excitatory(E_L)	-63.0 mV
EL_i	leak reversal potential inhibitory(E_L)	-65.0 mV
g_L	leak conductance	10.0 nS
a_e	subthreshold adaptation of excitatory neurons(a)	0.0 nS
b_e	spike-triggered adaptation of excitatory neurons(b)	0.0/30.0/60.0 pA
τ_{W_e}	adaptation time constant of excitatory neurons(τ_w)	500.0 ms
a_i	subthreshold adaptation of inhibitory neurons(a)	0.0 nS
b_i	spike-triggered adaptation inhibitory neurons(b)	0.0 pA
τ_{W_i}	adaptation time constant of inhibitory neurons(τ_w)	1.0 ms
E_e	excitatory reversal potential(E_{ex})	0.0 mV
τ_e	rise time of excitatory synaptic conductance(τ_{ex})	5.0 ms
Q_e	excitatory quantal conductance	1.5 nS
E_i	inhibitory reversal potential(E_{in})	-80.0 mV
τ_i	rise time of inhibitory synaptic conductance(τ_{in})	5.0 ms
Q_i	inhibitory quantal conductance	5.0 nS
$p_{connect}$	probability of connection	0.05
N_{tot}	number of total neurons	10000
p_i	percentage of inhibitory neurons	0.2
N_e	number of excitatory neurons	$N_{tot}(1 - p_i) = 8000$
N_i	number of inhibitory neurons	$N_{tot}p_i = 2000$
K_e	mean number of input excitatory synapses : $N_e p_{connect}$	400
K_i	mean number of input inhibitory synapses : $N_i p_{connect}$	100
v_{ext}	external input	see external input section
	initial condition	$\mu_E : 0.Hz; \mu_i : 0.Hz; c_{ee} : 0.; c_{ei} : 0.; c_{ii} : 0.; W_e : 1000.pA; W_i : 0.pA$

G. Supplementary material of Annex A – G.2. Supplementary Tables

P_e	second order polynomial of the phenomenological threshold for inhibitory neuron in mV	P_0	P_{μ_V}	P_{σ_V}	$P_{\tau_V^N}$
		-4.923163e-02	1.762790e-03	-7.677835e-04	-3.824880e-03
		$P_{\mu_V^2}$	$P_{\sigma_V^2}$	$P_{(\tau_V^N)^2}$	
		2.356120e-04	4.0210098e-03	1.812297e-03	
P_i	second order polynomial of the phenomenological threshold for inhibitory neuron in mV	$P_{\mu_V \sigma_V}$	$P_{\mu_V \tau_V^N}$	$P_{\sigma_V \tau_V^N}$	
		-3.723180e-05	1.929229e-04	3.974934e-03	
		P_0	P_{μ_V}	P_{σ_V}	$P_{\tau_V^N}$
		-5.079953e-2	2.139835e-3	-4.646189e-3	3.727148e-4
P_i	second order polynomial of the phenomenological threshold for inhibitory neuron in mV	$P_{\mu_V^2}$	$P_{\sigma_V^2}$	$P_{(\tau_V^N)^2}$	
		5.053228e-4	1.304294e-3	-1.073580e-2	
		$P_{\mu_V \sigma_V}$	$P_{\mu_V \tau_V^N}$	$P_{\sigma_V \tau_V^N}$	
		1.995937e-3	1.932031e-3	-1.015957e-2	

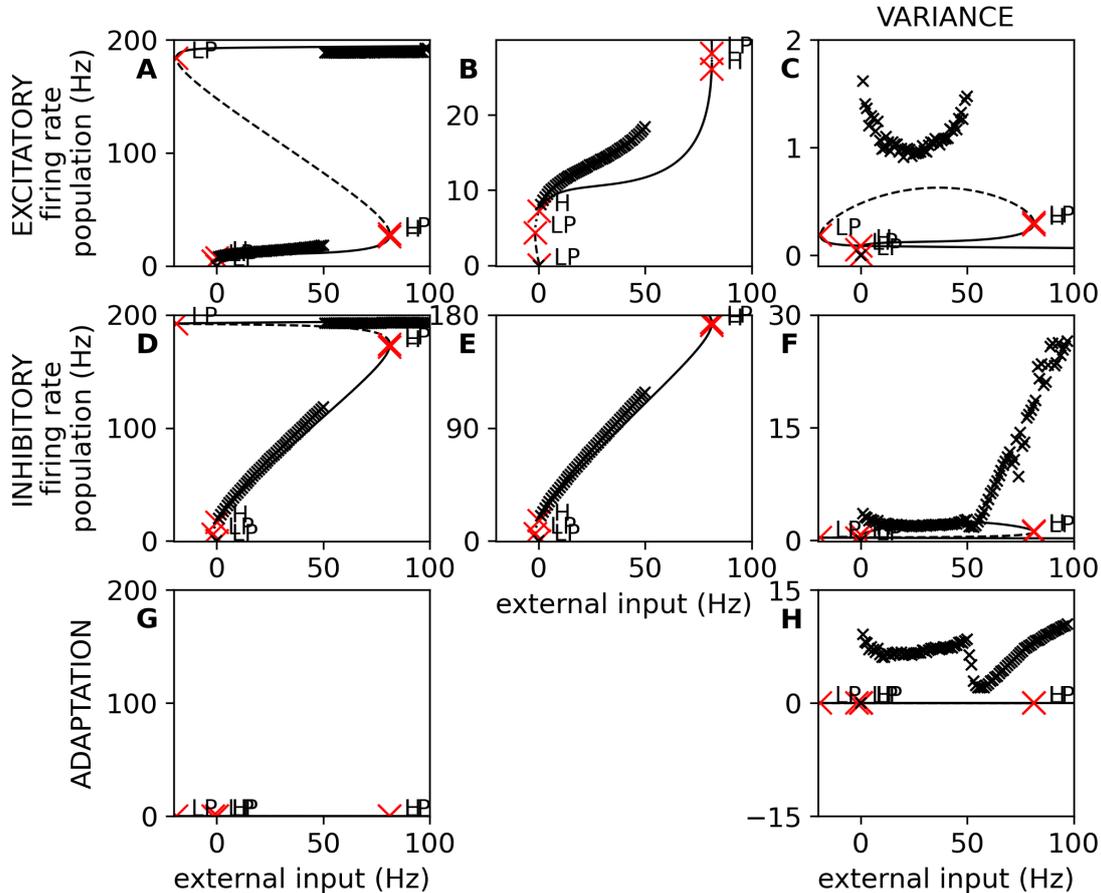
TABLEAU G.2. – Description of the mean-field simulation

G.3. Supplementary Figures



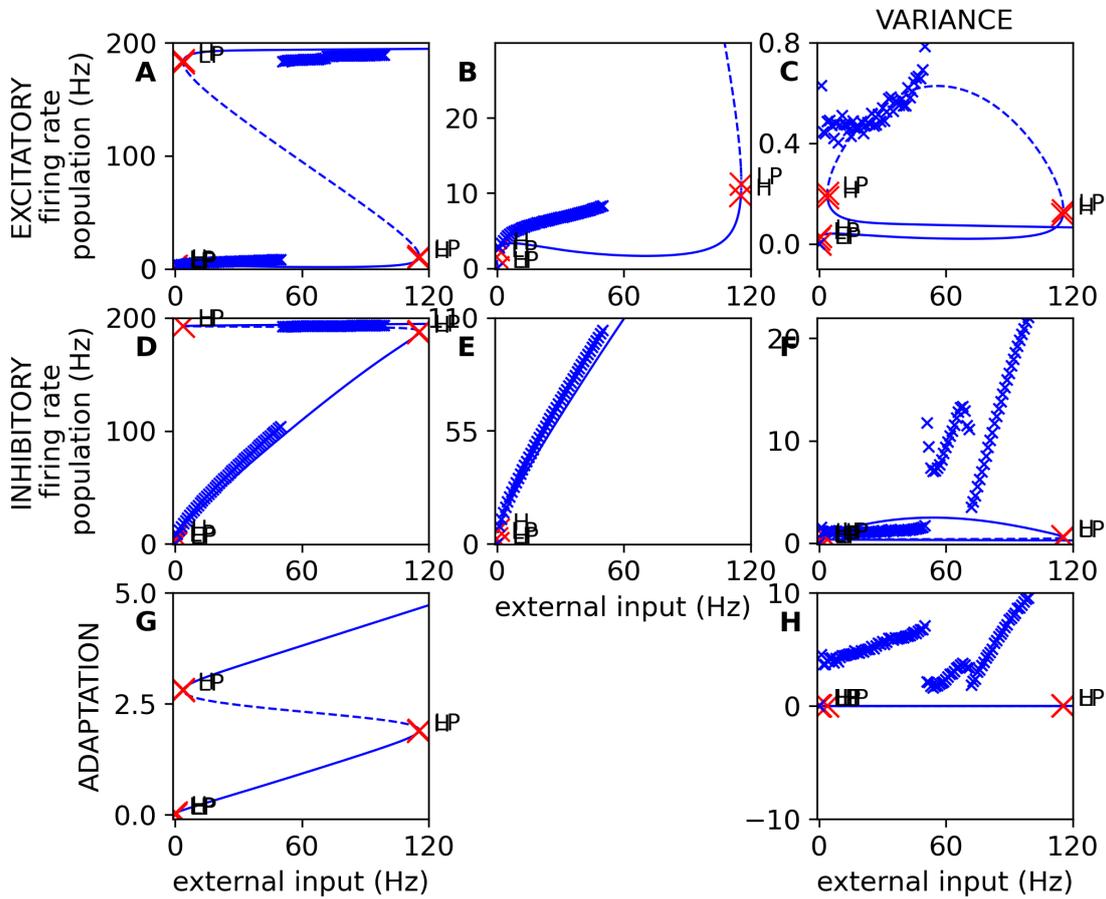
SUPPLEMENTARY FIGURE G.1. – Variability of fixed-point from network simulation with and without adaptation

Each violin represents the distribution of the mean firing rate of an excitatory or inhibitory population of 4 seconds after 1 second of transient for 30 networks. **A, B**: These violins in the top graphics represent the distribution of the estimated fixed-point for the mean firing rate of excitatory dependent on the excitatory external firing rate. **C, D**: These violins in the top graphics represent the distribution of the estimated fixed-point for the mean firing rate of excitatory dependent on the excitatory external firing rate. The additional curve represents the bifurcation diagram of the mean-field. The left graphics (A and C) is for cases without adaptation, and the right graphics (B and D) is for cases with spike-trigger adaptation equals 60 pA.



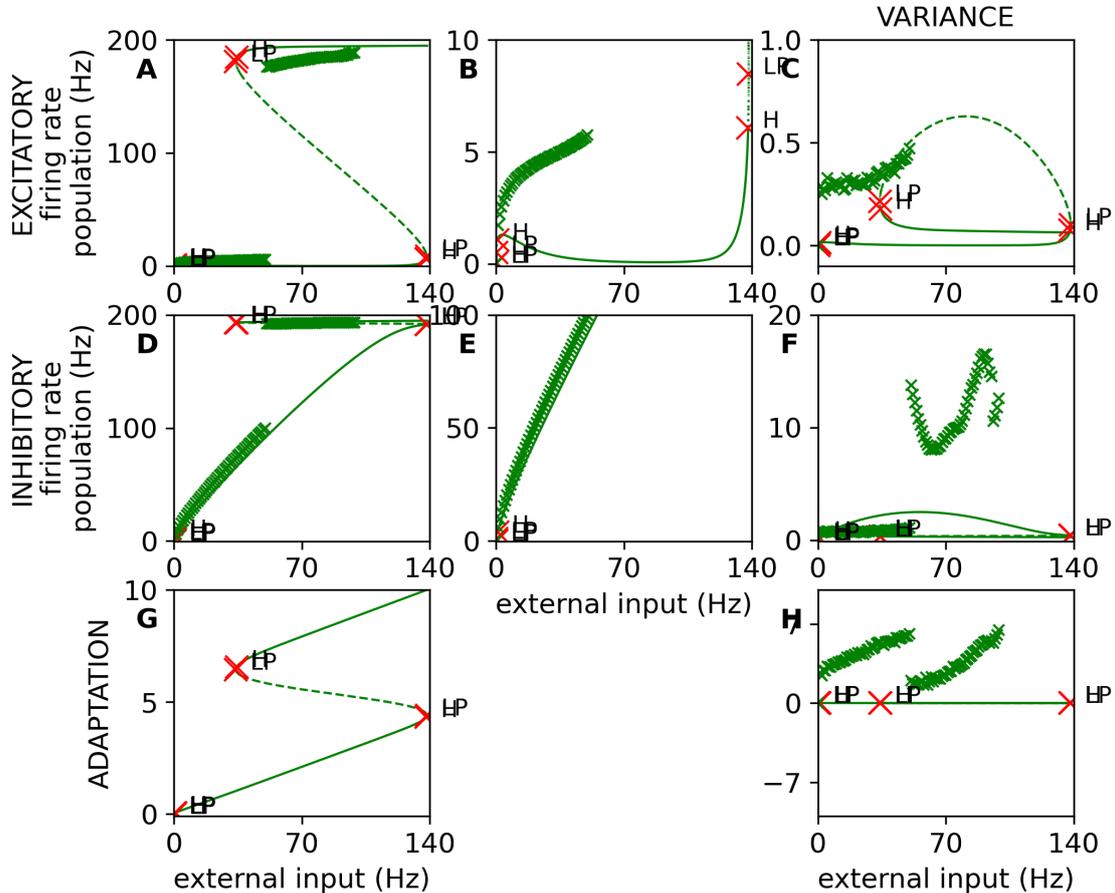
SUPPLEMENTARY FIGURE G.2. – Bifurcation diagram of external excitatory input for spike-trigger adaptation equals 0.0 pA

Each graphic represents the bifurcation diagram for each variable of the mean-field (A : firing rate of excitatory population, D : firing rate of inhibitory population, G : mean adaptation current, C : variance of excitatory firing rate, F : variance of inhibitory firing rate, H : covariance of the firing rate of the inhibitory and excitatory population). The black curve style represents the stability of the fixed-point (continuous line for stable fix points and dashed lines for unstable fix points). The curve style represents the stability of the fixed-point (continuous line for stable fix points and dashed lines for unstable fix points). The graphics B and E enlarge the bottom part of their left graphic (A and D). The black crosses are the estimation of the fixed-point using network simulation. The bifurcation points are indicated with a red cross and a letter that means the type of bifurcation (H : Hopf bifurcation and LP : limit point or saddle-node bifurcation).



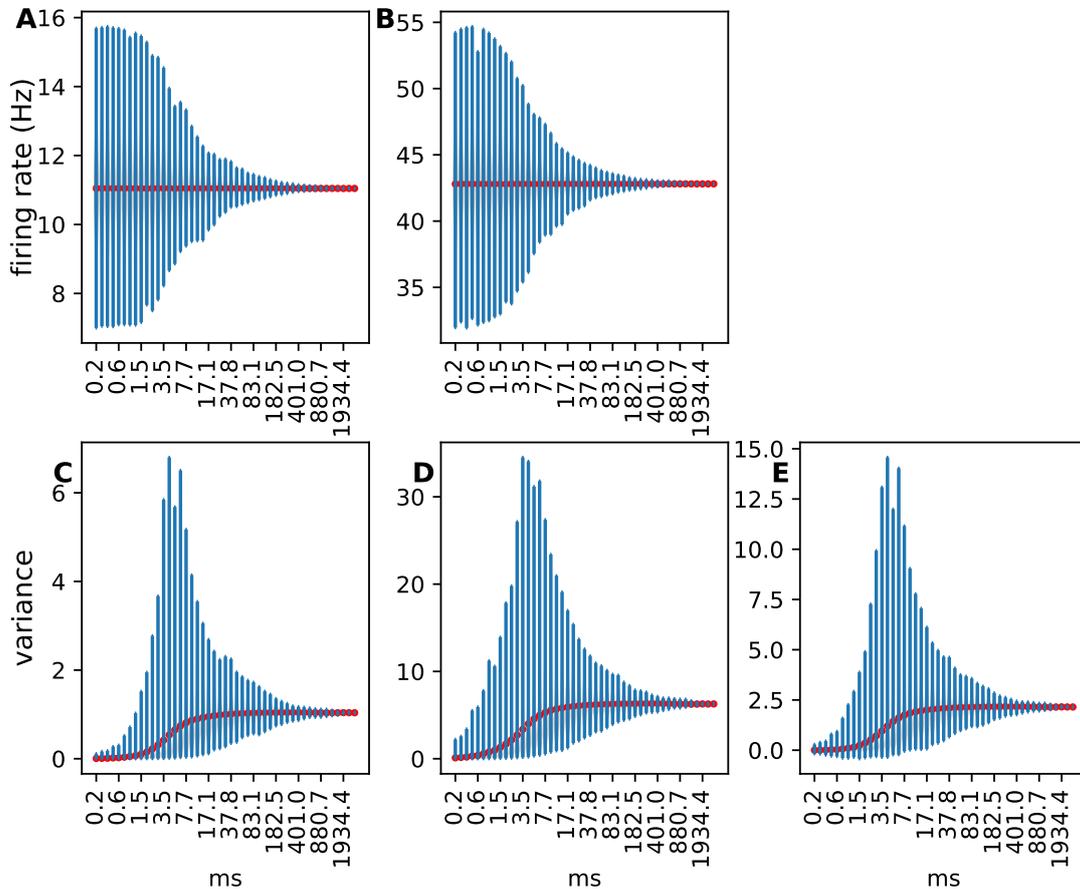
SUPPLEMENTARY FIGURE G.3. – Bifurcation diagram of external excitatory input for spike-trigger adaptation equals 30.0 pA

Each graphic represents the bifurcation diagram for each variable of the mean-field (A : firing rate of excitatory population, D : firing rate of inhibitory population, G : mean adaptation current, C : variance of excitatory firing rate, F : variance of inhibitory firing rate, H : covariance of the firing rate of the inhibitory and excitatory population). The blue curve style represents the stability of the fixed-point (continuous line for stable fix points and dashed lines for unstable fix points). The curve style represents the stability of the fixed-point (continuous line for stable fix points and dashed lines for unstable fix points). The blue crosses are the estimation of the fixed-point using network simulation. The bifurcation points are indicated with a red cross and a letter that means the type of bifurcation (H : Hopf bifurcation and LP : limit point or saddle-node bifurcation).



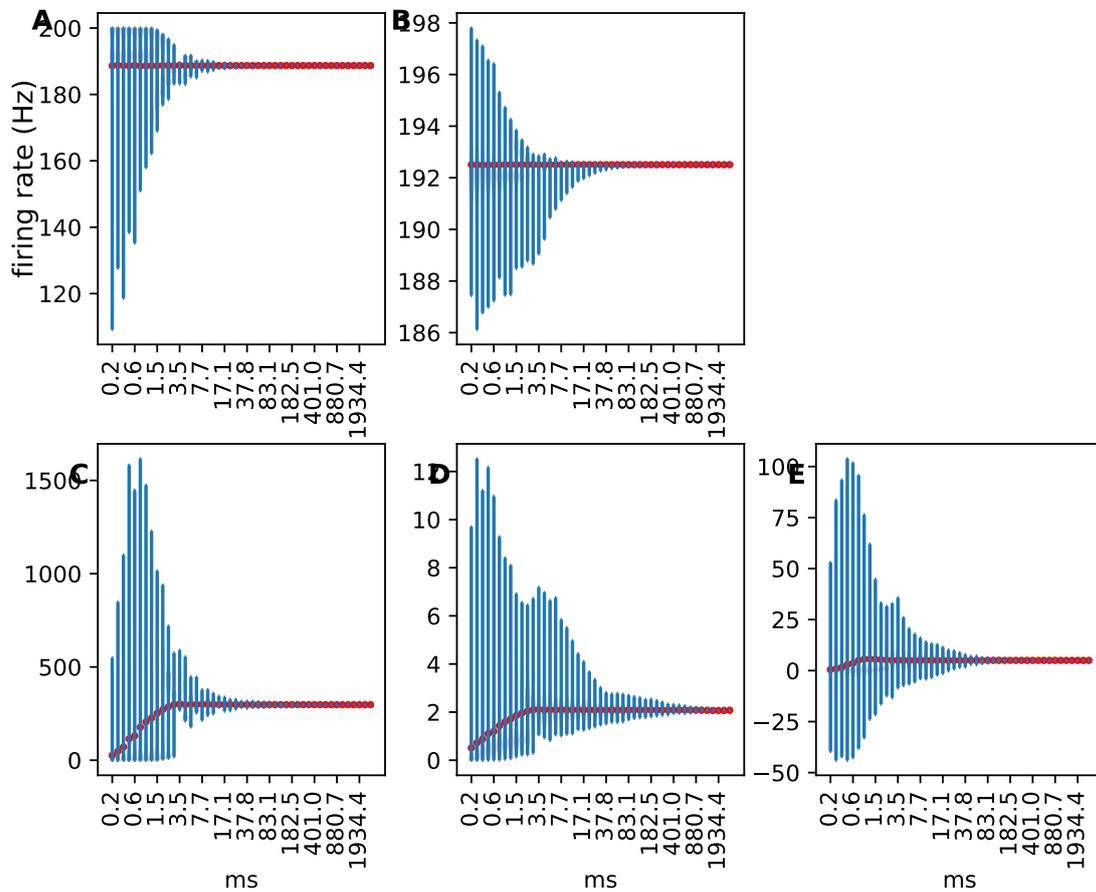
SUPPLEMENTARY FIGURE G.4. – Bifurcation diagram of external excitatory input for spike-trigger adaptation equals 60.0 pA

Each graphic represents the bifurcation diagram for each variable of the mean-field (A : firing rate of excitatory population, D : firing rate of inhibitory population, G : mean adaptation current, C : variance of excitatory firing rate, F : variance of inhibitory firing rate, H : covariance of the firing rate of the inhibitory and excitatory population). The green curve style represents the stability of the fixed-point (continuous line for stable fix points and dashed lines for unstable fix points). The graphics B and E enlarge the bottom part of their left graphic (A and D). The green crosses are the estimation of the fixed-point using network simulation. The bifurcation points are indicated with a red cross and a letter that means the type of bifurcation (H : Hopf bifurcation and LP : limit point or saddle-node bifurcation).

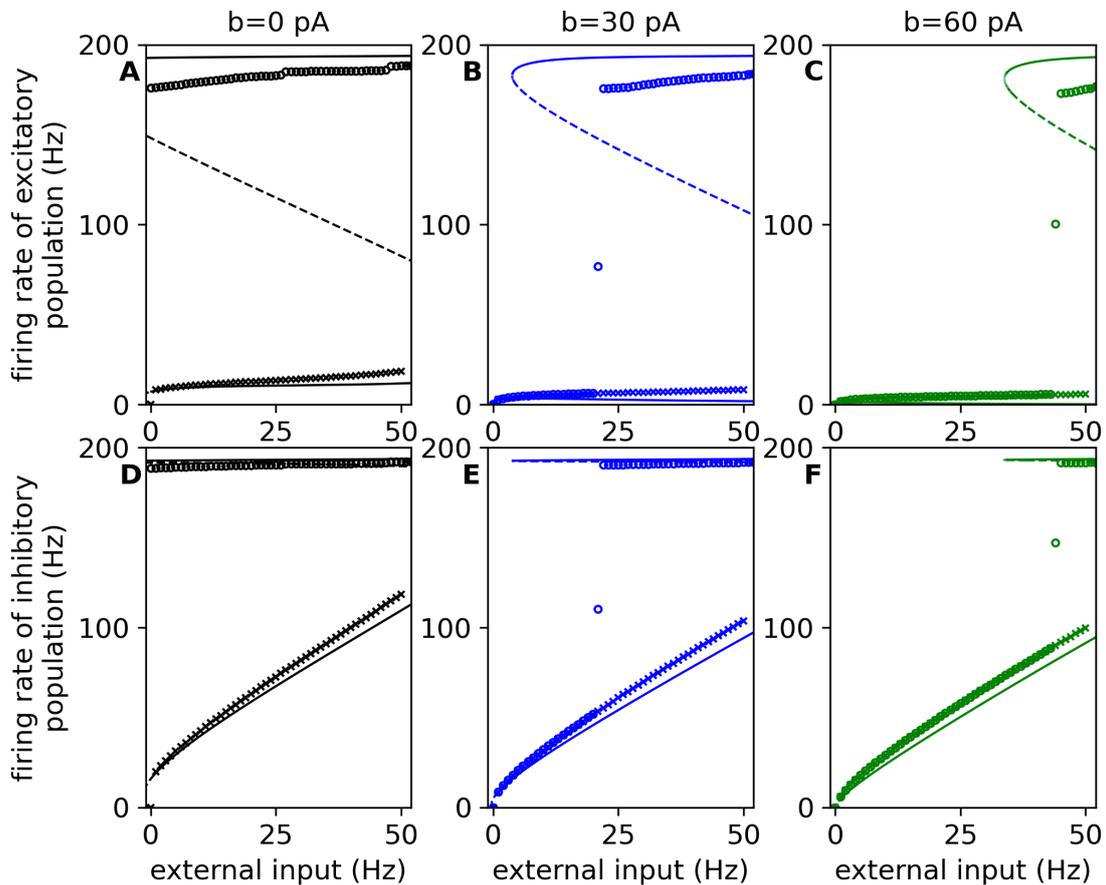


SUPPLEMENTARY FIGURE G.5. – Variability of the measure of the firing rate
 Each violin plot represents the distribution in function of the time windows for the measures associated with the variable of the mean-field, i.e. the mean and variance of firing rate for excitatory and inhibitory. The distribution represents the values of 5000 measures taken randomly over 30 seconds of smoothing instantaneous firing rate with sliding windows of 5 ms. The instantaneous firing rate is recorded from a unique spiking neural network with no adaptation and an excitatory external firing rate of 10 Hz. **A, B** : The top graphics represent the mean firing rate of each population. **C, D, E** : the bottom graphics represent the variance and covariance of these two populations. The left graphics (A and C) are associated with the excitatory population, and the central graphics (C and D) are associated with the inhibitory population. On each graphic, the red line represents the mean of each distribution, and the blue shape represents a rotated kernel density. We can see an exponential decrease in the variability. At the difference of the mean firing rate, the mean of the distribution of the variance and covariance change with the size of the window, and the distribution is not symmetric. The width of the distribution of the variance and covariance increase for a window size of 1.5 ms and 20 ms.

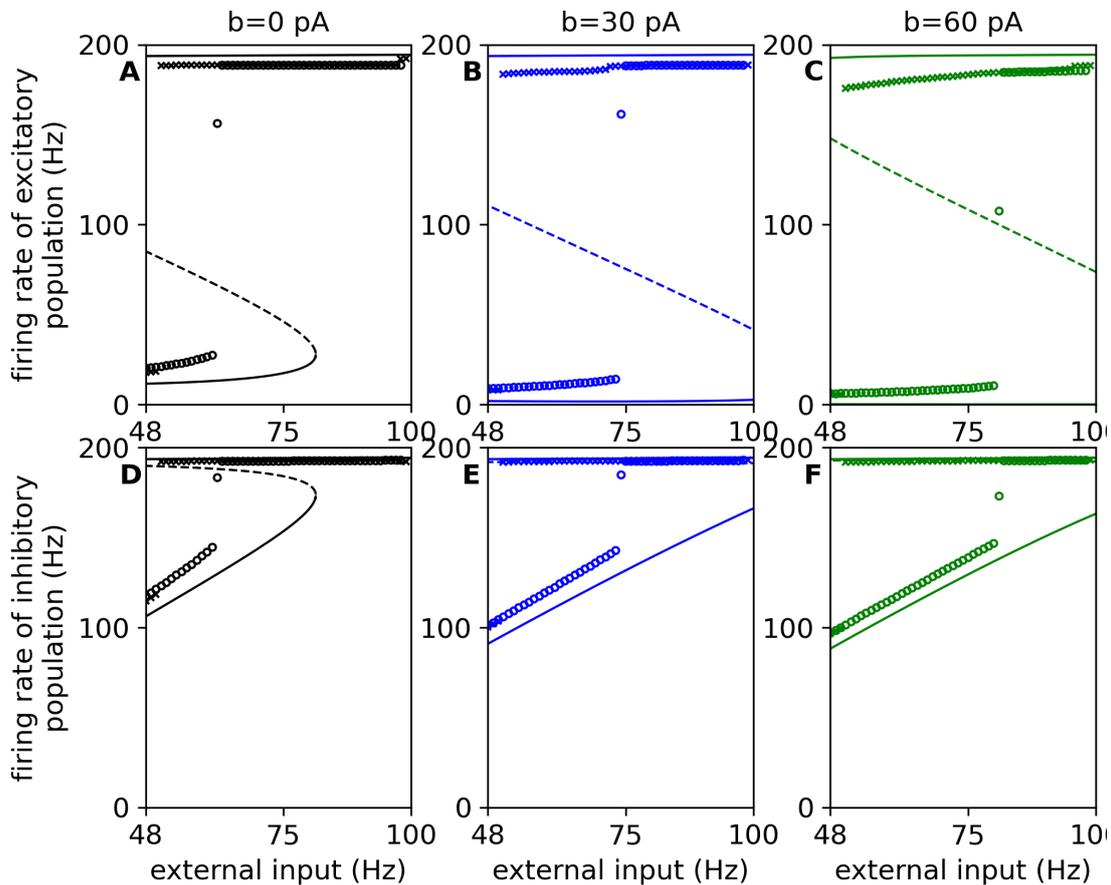
G. Supplementary material of Annex A – G.3. Supplementary Figures



SUPPLEMENTARY FIGURE G.6. – Variability of the measure of the firing rate
 Each violin plot represents the distribution in function of the time windows for the measures associated with the variable of the mean-field, i.e. the mean and variance of firing rate for excitatory and inhibitory. The distribution represents the values of 5000 measures taken randomly over 30 seconds of smoothing instantaneous firing rate with sliding windows of 5 ms. The instantaneous firing rate is recorded from a unique spiking neural network with no adaptation and an excitatory external firing rate of 60 Hz. **A, B** : The top graphics represent the mean firing rate of each population. **C, D, E** : the bottom graphics represent the variance and covariance of these two populations. The left graphics (A and C) are associated with the excitatory population, and the central graphics (C and D) are associated with the inhibitory population. On each graphic, the red line represents the mean of each distribution, and the blue shape represents a rotated kernel density. We can see the variance become smaller for the window's size smaller than the previous figure except for the covariance. At the difference of the mean firing rate, the mean of the distribution of the variance and covariance change with the size of the window. The large variance for small variance is due to the oscillation of the network at a frequency of around 200 Hz.

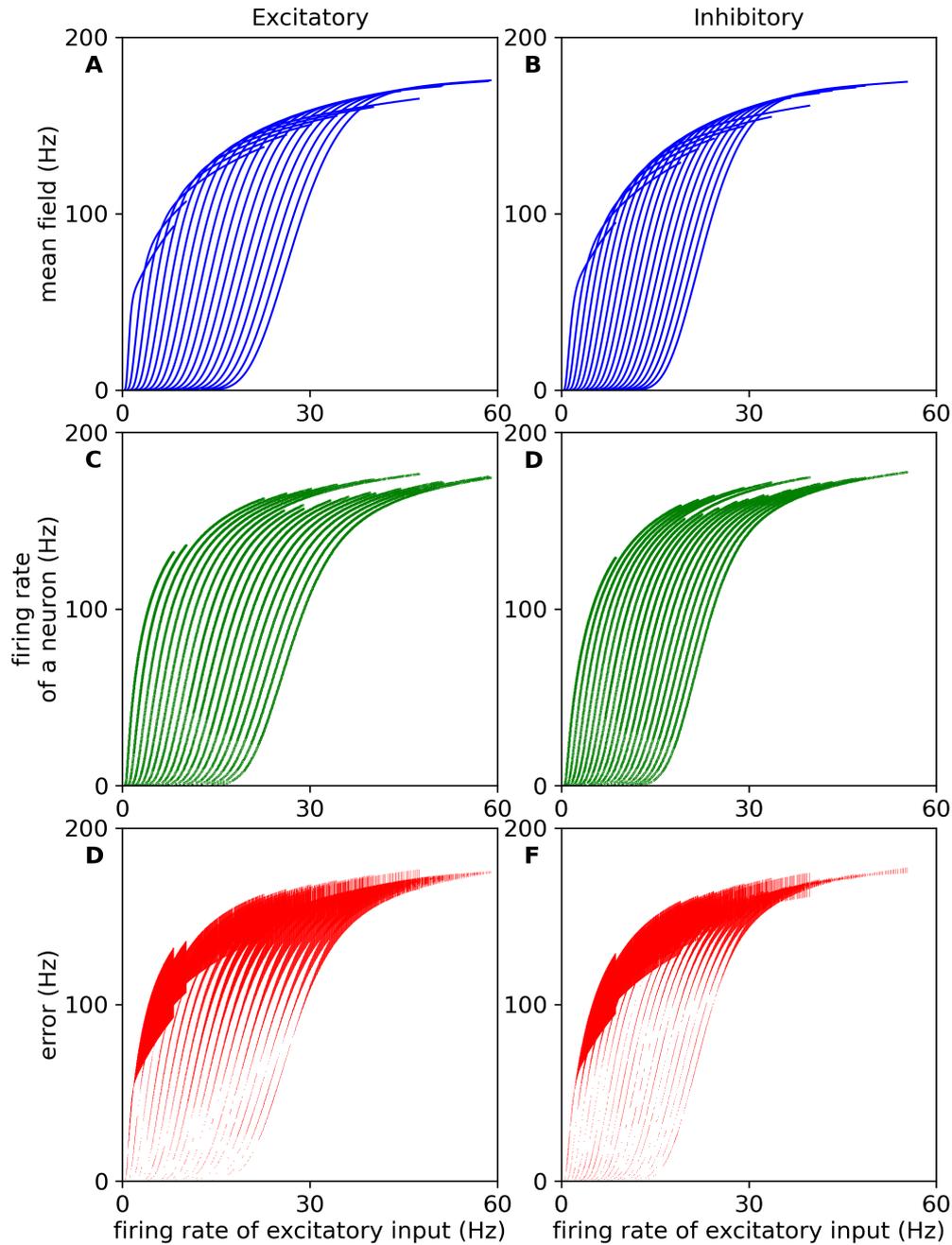


SUPPLEMENTARY FIGURE G.7. – Estimation and comparison of the high fixed-point **A, B, C** : The top graphics represent the mean firing rate of the excitatory population depending on the external excitatory input value. **D, E, F** : The bottom graphics represent the mean firing rate of the inhibitory population depending on the external excitatory input. The left graphics (A, D) are for neurons without adaptation, the central graphics (B, E) are for the excitatory population with a spike trigger at 30 pA, and the right graphics (C, F) are for the excitatory population with a spike trigger at 60 pA. The curve and the crosses are the same as the figure A.2 (see its caption for more details). The circles represent the mean firing rate over 10 seconds every 10 seconds for one network simulation. During these network simulations, the external excitatory input is reduced to 1 Hz every 10 seconds, from 51 Hz to 0 Hz.



SUPPLEMENTARY FIGURE G.8. – Estimation and comparison of the low fixed-point **A, B, C** : The top graphics represent the mean firing rate of the excitatory population depending on the external excitatory input value. **D, E, F** : The bottom graphics represent the mean firing rate of the inhibitory population depending on the external excitatory input. The left graphics (A, D) are for neurons without adaptation, the central graphics (B, E) are for the excitatory population with a spike trigger at 30 pA, and the right graphics (C, F) are for the excitatory population with a spike trigger at 60 pA. The curve and the crosses are the same as the figure A.2 (see its caption for more details). The circles represent the mean firing rate over 10 seconds every 10 seconds for one network simulation. During these network simulations, the external excitatory input is reduced to 1 Hz every 10 seconds, from 48 Hz to 99 Hz.

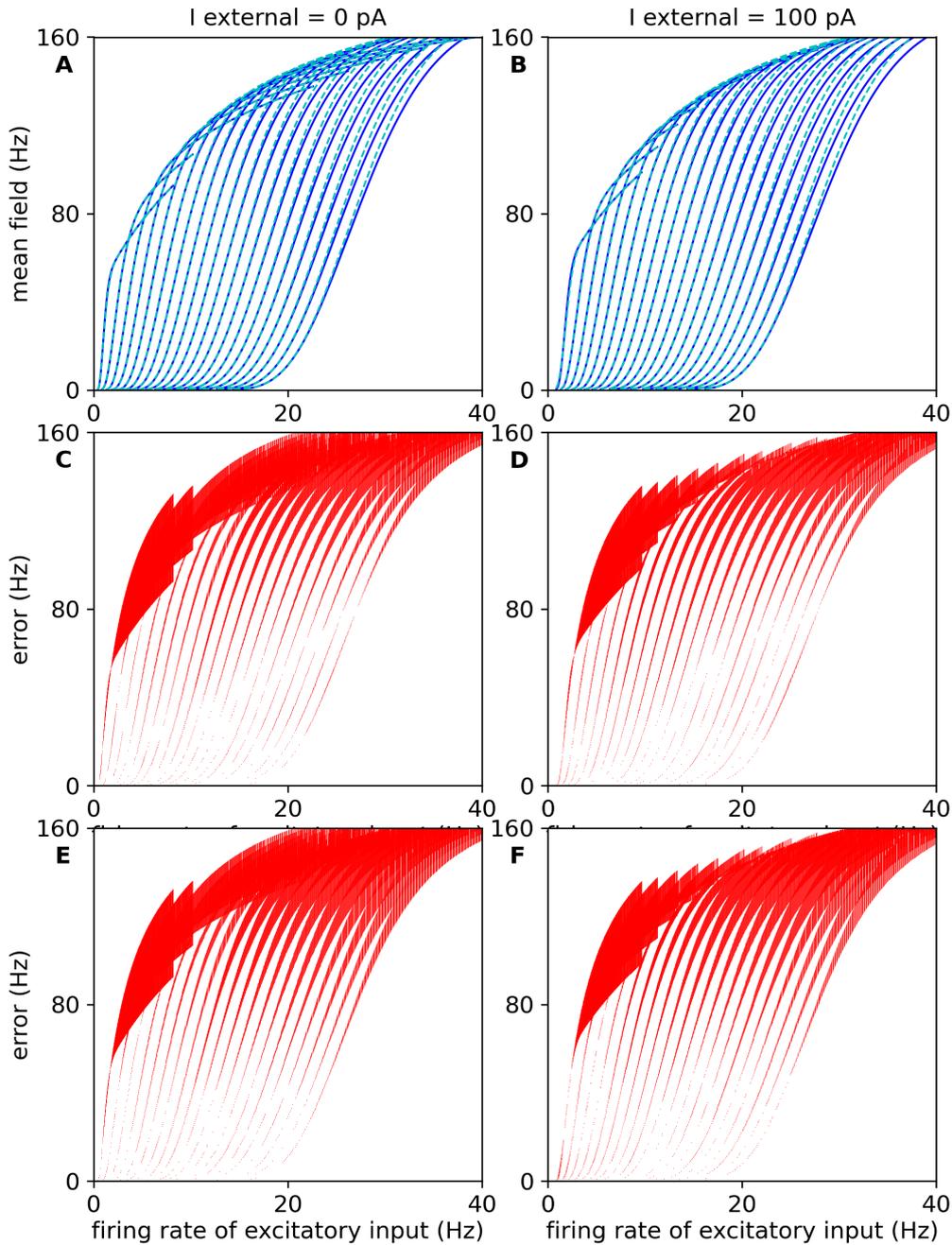
G. Supplementary material of Annex A – G.3. Supplementary Figures



SUPPLEMENTARY FIGURE G.9. – Fitting of Transfer functions full

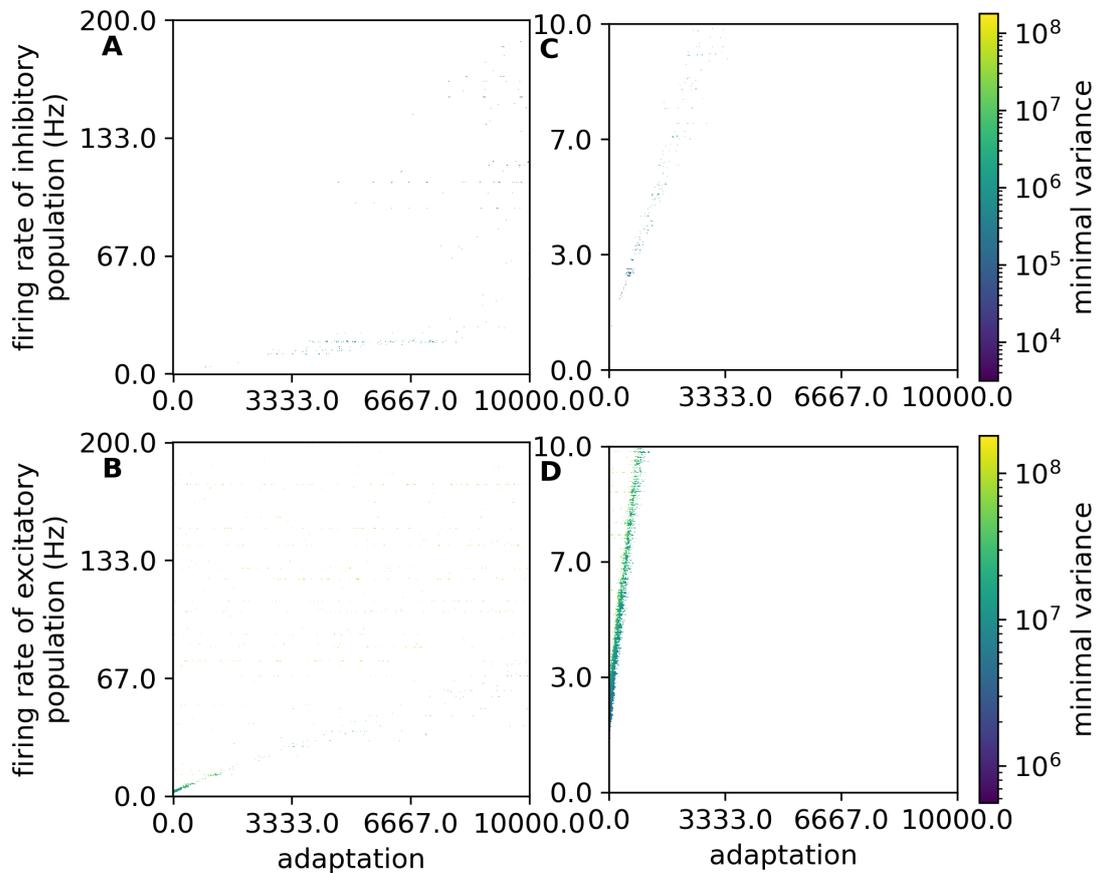
This figure enlarges the figure A.3. **A, C, E** : The left part is dedicated to the excitatory neuron. **B, D, F**, The right part is dedicated to the inhibitory neuron. The top graph (A and B) represents the transfer function of the mean-field fitted to the data. The middle graph (C and D) shows the data used for the fitting, which is the mean firing rate of 50 independent neurons. The bottom graph (E and F) displays the error between the transfer function and the data. Each line is associated with a different value of inhibitory firing rate (20 values evenly distributed between 0 Hz and 40 Hz).

G. Supplementary material of Annex A – G.3. Supplementary Figures



SUPPLEMENTARY FIGURE G.10. – Estimation of error of the transfer function for neurons in the presence of negative current

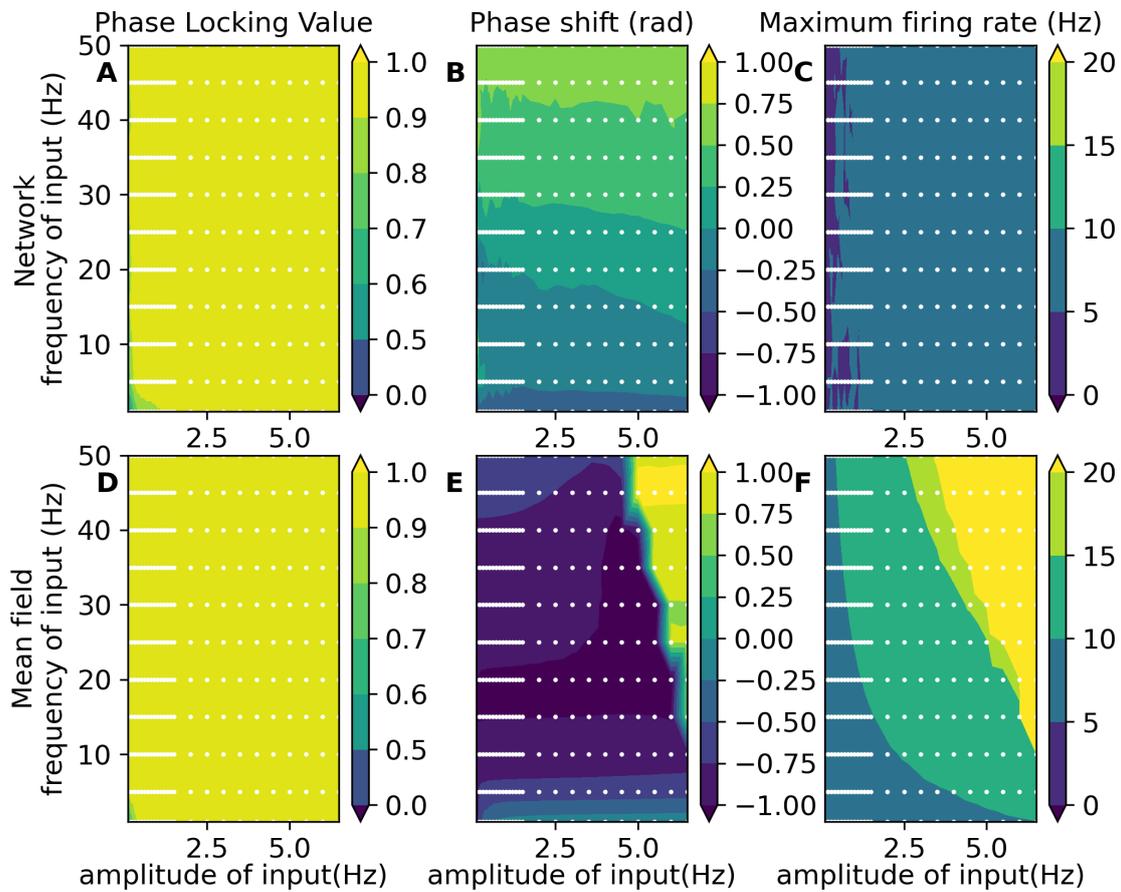
A, B : The top graphics represent the transfer function fit with (continuous dark blue line) and without (light dashed blue line) data of firing rate with a negative current. **C, D** : The middle graphics display the difference between the transfer function fitted with data of negative current. **E, F** : The bottom graphics display the difference between the transfer function fitted without the data of negative current. The left graphics is data of the mean firing rate



SUPPLEMENTARY FIGURE G.11. – Negative derivative of the transfer function for zeros firing rate

The graphics display the value of the variance of population A for which the mean firing rate of population B becomes a negative value for a specific value of the mean firing rate of population B and mean adaptive current. **A, C** : The top graphic illustrates the variance of the inhibitory population for getting a negative mean excitatory firing rate. **B, D** : The bottom graphic illustrates the variance of the excitatory population for getting a negative mean inhibitory firing rate. The right graphics (A and B) enlarge the left graphics (C and D).

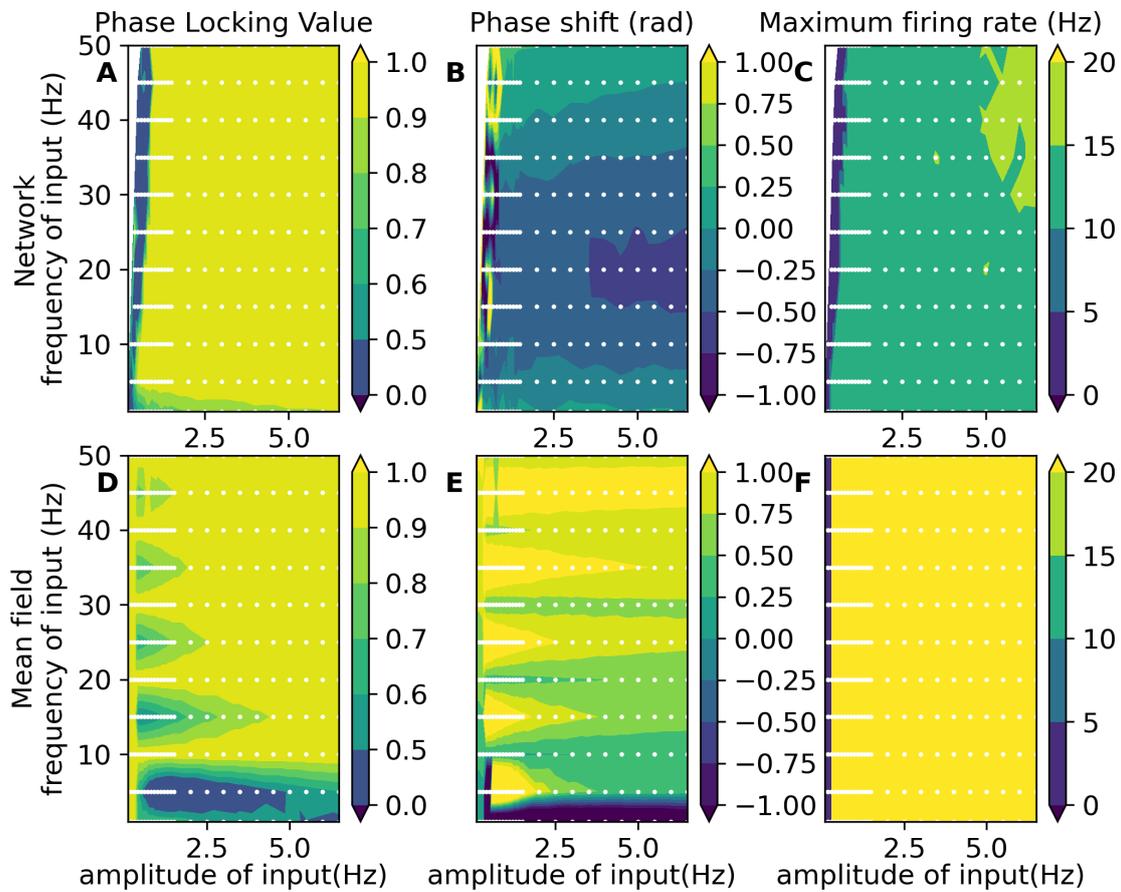
G. Supplementary material of Annex A – G.3. Supplementary Figures



SUPPLEMENTARY FIGURE G.12. – Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 7Hz and adaptation ($b = 60.0 pA$)

A, B, C : The top graphics represent the measure of the excitatory population in the spiking neural network. **D, E, F** : The bottom graphics illustrate the measure of the excitatory mean firing rate of the mean-field. The three measures displayed here are the phase locking values in rad (A and D), the phase shift (B and E) between the oscillatory input and the mean firing rate in rad, and the excitatory population's maximum firing rate in Hz (C and F).

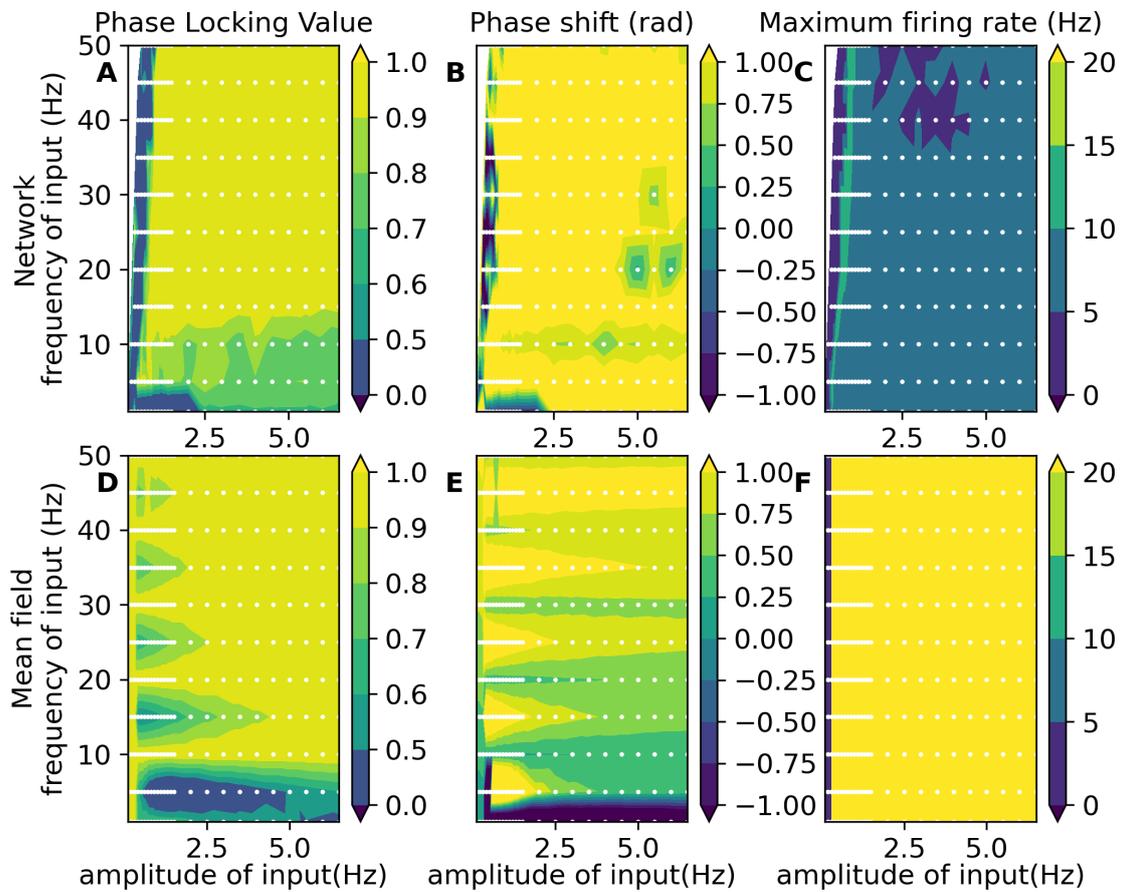
G. Supplementary material of Annex A – G.3. Supplementary Figures



SUPPLEMENTARY FIGURE G.13. – Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 0Hz

A, B, C : The top graphics represent the measure of the excitatory population in the spiking neural network. **D, E, F** : The bottom graphics illustrate the measure of the excitatory mean firing rate of the mean-field. The three measures displayed here are the phase locking values in rad (A and D), the phase shift (B and E) between the oscillatory input and the mean firing rate in rad, and the excitatory population's maximum firing rate in Hz (C and F).

G. Supplementary material of Annex A – G.3. Supplementary Figures



SUPPLEMENTARY FIGURE G.14. – Comparison between the mean-field and the spiking network in the presence of oscillatory input with a mean rate of 0Hz and adaptation ($b = 60.0 \mu A$)

A, B, C : The top graphics represent the measure of the excitatory population in the spiking neural network. **D, E, F :** The bottom graphics illustrate the measure of the excitatory mean firing rate of the mean-field. The three measures displayed here are the phase locking values in rad (A and D), the phase shift (B and E) between the oscillatory input and the mean firing rate in rad, and the excitatory population's maximum firing rate in Hz (C and F).

CHAPTER H

Supplementary material of Annex B

A1		Brian : Model Summary	
Software	Brian		
Topology	--		
Population	excitatory and inhibitory		
Connectivity	random connection without self recurrent connections		
Neuron Model	adaptive exponential leaky integrate and fire neurons, fixed threshold and fixed absolute refractory time		
Synapse Model	conductance-based exponential shape		
Plasticity	--		
Input	Independent fixed rate Poisson generator spike trains to all neurons		
Measurement	Voltage, Adaptation Current, Spike Activity		
A2		Brain : Software	
version	2.		
integrator method	Heun method		
integration step	0.1 ms		
number of seed	1		
simulation time	3000ms		
A3		Brian : Populations	
Name	Elements	Size	
E	aeif_cond_exp	$N_e = (1-g_{inh})N = 8000$	
I	aeif_cond_exp	$N_i = g_{inh}N = 2000$	
P_{ext}	poisson generator	8000	

A4 Brian : Neuron Model	
Name	aeif
Type	adaptive exponential leaky integrator and fire with conductance synapse
subthreshold dynamics	$C_m \frac{dV_m}{dt} = -g_L(V_m - E_L) + g_L \Delta_T e^{\frac{V_m - V_{th}}{\Delta_T}}$ $-g_e(t)(V_m - E_{ex}) - g_i(t)(V_m - E_{in})$ $-W + I_e$ $\tau_w \frac{dW}{dt} = a(V_m - E_L) - W$
reset condition	<p>For $t^{(f)} = \{t V_m(t) \geq V_{peak}\}$</p> <ul style="list-style-type: none"> • $V_m([t^{(f)}; t^{(f)} + t_{ref}]) = V_{reset}$ • $W([t^{(f)}]) = W([t^{(f)}]) + b$
Brian : Synapse Model	
Name	cond_exp
Type	post-synaptic conductance in the form of truncated exponentials
Coupling equation	$g_e(t) = \sum_{t_j^{(f)}} w_j \exp_trunc(t - t_j, \tau_{ex}) \text{ with } w_j > 0.0$ $g_i(t) = \sum_{t_j^{(f)}} w_j \exp_trunc(t - t_j, \tau_{in}) \text{ with } w_j < 0.0$ $\exp_trunc(t, \tau) = e^{1 - \frac{t}{\tau}} \text{Heaviside}(t)$

A5		Brian : Neuron Model Parameters
		excitatory neurons
C_m	Capacity of the membrane	200.0 pF
t_{ref}	Duration of refractory period	5.0 ms
V_{reset}	Reset value for V_m after a spike	-55.0 mV
E_L	Leak reversal potential	-63.0 mV
g_L	Leak conductance	10.0 nS
Δ_T	Slope factor	2.0 mV
V_{peak}	Spike detection threshold	-40.0 mV
a	Subthreshold adaptation	0.0 nS
b	Spike-triggered adaptation	1.0pA for AI and 60.0 for UDstate
τ_w	Adaptation time constant	500.0 ms
V_{th}	Spike initiation threshold	-50.0 mV
I_e	Constant external input current	0.0 pA
E_{ex}	Excitatory reversal potential	0.0 mV
E_{in}	Inhibitory reversal potential	-80.0 mV
V_m	Initialization of the voltage membrane	-65.0 mV
W	Initialization of adaptation current	1.35 pA
		inhibitory neurons
C_m	Capacity of the membrane	200.0 pF
t_{ref}	Duration of refractory period	5.0 ms
V_{reset}	Reset value for V_m after a spike	-65.0 mV
E_L	Leak reversal potential	-75.0 mV
g_L	Leak conductance	10.0 nS
Δ_T	Slope factor	0.5 ms
V_{peak}	Spike detection threshold	-47.5 mV
a	Subthreshold adaptation	0.0 nS
b	Spike-triggered adaptation	0.0 pA
τ_w	Adaptation time constant	1.0 ms
V_{th}	Spike initiation threshold	-50.0 mV
I_e	Constant external input current	0.0 pA
E_{ex}	Excitatory reversal potential	0.0 mV
E_{in}	Inhibitory reversal potential	-80.0 mV
V_m	Initialization of the voltage membrane	-65.0 mV
W	Initialization of adaptation current	0.0 pA
A6		Brian : Input
Poisson generator		
equation	$p(n) = \frac{\lambda^n}{n!} \exp(-\lambda)$	
implementation	approximation with the binomial law	
excitatory firing rate λ_{ex}	0.35	

A7 Brian : Connectivity				
parameter synapses				
τ_{ex}	Rise time of excitatory synaptic conductance			5.0ms
τ_{in}	Rise time of inhibitory synaptic conductance			5.0ms
Name	Source	Target	Weights	Pattern
EE	E	E	$Q_e=1.5$ nS	The probability of connections is 0.05 between pair of neurons. The neuron can't connected to itself and it can't have multiple connections with the same neurons.
EI	E	I	$Q_e=1.5$ nS	The probability of connections is 0.05 between pair of neurons. The neuron can't have multiple connections with the same neurons.
IE	I	E	$Q_i=5.0$ nS	The probability of connections is 0.05 between pair of neurons. The neuron can't have multiple connections with the same neurons.
II	I	I	$Q_i=5.0$ nS	The probability of connections is 0.05 between pair of neurons. The neuron can't connected to itself and it can't have multiple connections with the same neurons.
$P_{ext}E$	P_{ext}	E	$Q_e=1.5$ nS	The probability of connections is 0.05 between pair of neurons. The neuron can't have multiple connections with the same neurons.
$P_{ext}I$	P_{ext}	I	$Q_e=1.5$ nS	The probability of connections is 0.05 between pair of neurons. The neuron can't have multiple connections with the same neurons.
A8 Brain : Measurement				
state variable	spike time	precision	0.1 ms	
		number of recorded neurons	all	
spike activities	raster plot	precision	0.1 ms	
	histogram of instantaneous firing rate	bins	0.1ms	
	simple moving average	windows size	T (20ms)	

B1 TVB : Model Summary	
Software	TVB
Neural Mass model	Mean Adaptive Exponential
Connectivity	Mouse connectome with 104 regions
Coupling	linear coupling
stimulus	--
Monitors	state variable of the mean field
B2 TVB : Software	
version	2.1
integrator method	Heun method
integration step	0.1 ms
number of seed	1
simulation time	3000ms
B3 TVB : Coupling	
Name	Linear
Type	Linear coupling
equations	$\nu_{ext_k} = a * \left(\sum_{j=1}^{104} u_{kj} \nu_{e_j} (t - \tau_{kj}) \right) + b$ <p>where u_{kj} are the elements of the weights matrix, τ_{kj} are the elements of the delay matrix and ν_{e_j} are the mean excitatory firing rate of the regions j.</p>
parameters	$a = 1.0$ and $b = 0.0$
B4 TVB : Connectivity	
Connectivity is extracted from tracer data as explained by the paper TVBM	
number of region	104
tract lengths	maximum : 115.46 and mean : 53.58
speed	3 ms
weights	The weights are normalize such as the sum of the input weight to one region equals 1. (maximum : 0.73 and mean : 0.02)
centers	average center of mouse brain : [57., 74.97, 42.53]

B4	TVB : Connectivity
region name	the region name are extracted from Allen Mouse Brain Connectivity Atlas
	<p>Right Primary motor area, Right Secondary motor area, Right Primary somatosensory area nose, Right Primary somatosensory area barrel field, Right Primary somatosensory area lower limb, Right Primary somatosensory area mouth, Right Primary somatosensory area upper limb, Right Supplemental somatosensory area, Right Gustatory areas, Right Visceral area, Right Dorsal auditory area, Right Primary auditory area, Right Ventral auditory area, Right Primary visual area, Right Anterior cingulate area dorsal part, Right Anterior cingulate area ventral part, Right Agranular insular area dorsal part, Right Retrosplenial area dorsal part, Right Retrosplenial area ventral part, Right Temporal association areas, Right Perirhinal area, Right Ectorhinal area, Right Main olfactory bulb, Right Anterior olfactory nucleus, Right Piriform area, Right Cortical amygdalar area posterior part, Right Field CA1, Right Field CA3, Right Dentate gyrus, Right Entorhinal area lateral part, Right Entorhinal area medial part dorsal zone, Right Subiculum, Right Caudoputamen , Right Nucleus accumbens , Right Olfactory tubercle , Right Substantia innominata , Right Lateral hypothalamic area , Right Superior colliculus sensory related , Right Inferior colliculus , Right Midbrain reticular nucleus , Right Superior colliculus motor related , Right Periaqueductal gray , Right Pontine reticular nucleus caudal part , Right Pontine reticular nucleus , Right Intermediate reticular nucleus , Right Central lobule , Right Culmen , Right Simple lobule , Right Ansiform lobule , Right Paramedian lobule , Right Copula pyramidis , Right Paraflocculus , Left Primary motor area, Left Secondary motor area, Left Primary somatosensory area nose, Left Primary somatosensory area barrel field, Left Primary somatosensory area lower limb, Left Primary somatosensory area mouth, Left Primary somatosensory area upper limb, Left Supplemental somatosensory area, Left Gustatory areas, Left Visceral area, Left Dorsal auditory area, Left Primary auditory area, Left Ventral auditory area, Left Primary visual area, Left Anterior cingulate area dorsal part, Left Anterior cingulate area ventral part, Left Agranular insular area dorsal part, Left Retrosplenial area dorsal part, Left Retrosplenial area ventral part, Left Temporal association areas, Left Perirhinal area, Left Ectorhinal area, Left Main olfactory bulb, Left Anterior olfactory nucleus, Left Piriform area, Left Cortical amygdalar area posterior part, Left Field CA1, Left Field CA3, Left Dentate gyrus, Left Entorhinal area lateral part, Left Entorhinal area medial part dorsal zone, Left Subiculum, Left Caudoputamen , Left Nucleus accumbens , Left Olfactory tubercle , Left Substantia innominata , Left Lateral hypothalamic area , Left Superior colliculus sensory related , Left Inferior colliculus , Left Midbrain reticular nucleus , Left Superior colliculus motor related , Left Periaqueductal gray , Left Pontine reticular nucleus caudal part , Left Pontine reticular nucleus , Left Intermediate reticular nucleus , Left Central lobule , Left Culmen , Left Simple lobule , Left Ansiform lobule , Left Paramedian lobule , Left Copula pyramidis , Left Paraflocculus</p>

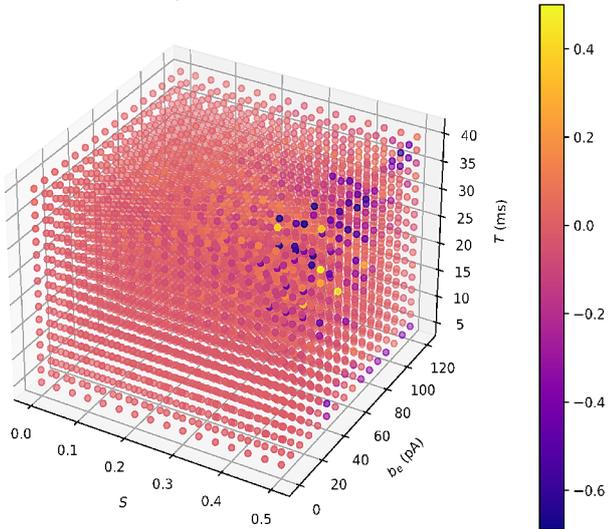
Name	Mean Ad Ex
Type	Neural mass model of network of adaptive exponential integrate and fire excitatory and inhibitory neurons of second statistical order
equation	$T \frac{\partial \nu_e}{\partial t} = (\mathcal{F}_e - \nu_e) + \frac{1}{2} c_{ee} \frac{\partial^2 \mathcal{F}_e}{\partial \nu_e \partial \nu_e} + \frac{1}{2} c_{ei} \frac{\partial^2 \mathcal{F}_e}{\partial \nu_e \partial \nu_i} + \frac{1}{2} c_{ie} \frac{\partial^2 \mathcal{F}_e}{\partial \nu_i \partial \nu_e} + \frac{1}{2} c_{ii} \frac{\partial^2 \mathcal{F}_e}{\partial \nu_i \partial \nu_i}$ $T \frac{\partial \nu_i}{\partial t} = (\mathcal{F}_i - \nu_i) + \frac{1}{2} c_{ee} \frac{\partial^2 \mathcal{F}_i}{\partial \nu_e \partial \nu_e} + \frac{1}{2} c_{ei} \frac{\partial^2 \mathcal{F}_i}{\partial \nu_e \partial \nu_i} + \frac{1}{2} c_{ie} \frac{\partial^2 \mathcal{F}_i}{\partial \nu_i \partial \nu_e} + \frac{1}{2} c_{ii} \frac{\partial^2 \mathcal{F}_i}{\partial \nu_i \partial \nu_i}$ $T \frac{\partial c_{ee}}{\partial t} = (\mathcal{F}_e - \nu_e)(\mathcal{F}_e - \nu_e) + c_{ee} \frac{\partial \mathcal{F}_e}{\partial \nu_e} + c_{ee} \frac{\partial \mathcal{F}_e}{\partial \nu_e} + c_{ei} \frac{\partial \mathcal{F}_i}{\partial \nu_e} + c_{ie} \frac{\partial \mathcal{F}_i}{\partial \nu_e} - 2c_{ee} + \frac{\mathcal{F}_e(1/T - \mathcal{F}_e)}{N_e}$ $T \frac{\partial c_{ei}}{\partial t} = (\mathcal{F}_e - \nu_e)(\mathcal{F}_i - \nu_i) + c_{ee} \frac{\partial \mathcal{F}_e}{\partial \nu_e} + c_{ei} \frac{\partial \mathcal{F}_e}{\partial \nu_i} + c_{ei} \frac{\partial \mathcal{F}_i}{\partial \nu_e} + c_{ii} \frac{\partial \mathcal{F}_i}{\partial \nu_i} - 2c_{ei}$ $T \frac{\partial c_{ie}}{\partial t} = (\mathcal{F}_i - \nu_i)(\mathcal{F}_e - \nu_e) + c_{ie} \frac{\partial \mathcal{F}_e}{\partial \nu_i} + c_{ee} \frac{\partial \mathcal{F}_e}{\partial \nu_e} + c_{ii} \frac{\partial \mathcal{F}_i}{\partial \nu_i} + c_{ie} \frac{\partial \mathcal{F}_i}{\partial \nu_e} - 2c_{ie}$ $T \frac{\partial c_{ii}}{\partial t} = (\mathcal{F}_i - \nu_i)(\mathcal{F}_i - \nu_i) + c_{ie} \frac{\partial \mathcal{F}_e}{\partial \nu_i} + c_{ei} \frac{\partial \mathcal{F}_e}{\partial \nu_i} + c_{ii} \frac{\partial \mathcal{F}_i}{\partial \nu_i} + c_{ii} \frac{\partial \mathcal{F}_i}{\partial \nu_i} - 2c_{ii} + \frac{\mathcal{F}_i(1/T - \mathcal{F}_i)}{N_i}$ $\tau_{W_e} \frac{\partial W_e}{\partial t} = -W_e + b_e \nu_e + a_e (\mu_V(\nu_e, \nu_i, W_e) - EL_e)$ $\tau_{W_i} \frac{\partial W_i}{\partial t} = -W_i + b_i \nu_i + a_i (\mu_V(\nu_e, \nu_i, W_i) - EL_i)$
noise equation	Ornstein-Uhlenbeck process : $\tau_{OU} \frac{dOU_t}{dt} = (\mu - OU_t) + \sigma dW_t$ with W_t is a Wiener process
transfer function	$\mathcal{F}_e = \mathcal{F}(\nu_e + 1e - 6, \nu_{ext} + w_\sigma OU_t, \nu_i, W_e)$ $\mathcal{F}_i = \mathcal{F}(\nu_e + 1e - 6) + w_\sigma OU_t, \nu_{ext}, \nu_i, W_i)$ $\mathcal{F} = \frac{1}{2\tau_V} \cdot \text{Erfc}\left(\frac{V_{thre}^{eff} - \mu_V}{\sqrt{2}\sigma_V}\right)$ $V_{thre}^{eff}(\mu_V, \sigma_V, \tau_V^N = \tau_V \frac{gL}{C_m}) = P'_0 + \sum_{x \in \{\mu_V, \sigma_V, \tau_V^N\}} P_x \cdot \left(\frac{x - x^0}{\delta x^0}\right) + \sum_{x,y \in \{\mu_V, \sigma_V, \tau_V^N\}^2} P_{xy} \cdot \left(\frac{x - x^0}{\delta x^0}\right) \left(\frac{y - y^0}{\delta y^0}\right)$ $\mu_G(\nu_e, \nu_{ext}, \nu_i) = ((\nu_e K_e + \nu_{ext} K_{ext})\tau_e Q_e) + (\nu_i K_i \tau_i Q_i) + g_L$ $\mu_{V_s}(\nu_e, \nu_{ext}, \nu_i, w, \mu_G) = \frac{((\nu_e K_e + \nu_{ext} K_{ext})\tau_e Q_e)E_e + (\nu_i K_i \tau_i Q_i)E_i + g_L EL_s - w}{\mu_G}$ $\sigma_V(\mu_V, \mu_G) = \sqrt{\sum_{s \in \{e, i\}} K_s \nu_s \frac{\left(\frac{Q_s}{\mu_G}(E_s - \mu_V)\tau_s\right)^2}{2\frac{C_m}{\mu_G} + \tau_s}}$ $\tau_V(\mu_V, \mu_G) = \frac{\sum_{s \in \{e, i\}} K_s \nu_s \left(\frac{Q_s}{\mu_G}(E_s - \mu_V)\tau_s\right)^2}{\sum_{s \in \{ex, in\}} K_s \nu_s \frac{\left(\frac{Q_s}{\mu_G}(E_s - \mu_V)\tau_s\right)^2}{2\frac{C_m}{\mu_G} + \tau_s}}$

case		Asynchronous	Irregular Synchronize	Regular bursting	
T	time resolution of the mean field		20.0ms		
C_m	Capacity of the membrane		200.0 pF		
EL_e	Leak reversal potential excitatory(E_L)	-64.5 mV	-64.5 mV	-74.0 mV	
EL_i	Leak reversal potential inhibitory(E_L)	-65.0 mV	-65.0 mV	-75.0 mV	
g_L	Leak conductance		10.0 nS		
a_e	Subthreshold adaptation of excitatory neurons(a)		0.0 nS		
b_e	Spike-triggered adaptation of excitatory neurons(b)	10.0 pA	100.0 pA	50.0 pA	
τ_{W_e}	Adaptation time constant of excitatory neurons(τ_w)	500.0 ms	500.0 ms	150.0 ms	
a_i	Subthreshold adaptation of inhibitory neurons(a)		0.0 nS		
b_i	Spike-triggered adaptation inhibitory neurons(b)		0.0 pA		
τ_{W_i}	Adaptation time constant of inhibitory neurons(τ_w)		1.0 ms		
E_e	Excitatory reversal potential(E_{ex})		0.0 mV		
τ_e	Rise time of excitatory synaptic conductance(τ_{ex})		5.0 ms		
Q_e	excitatory quantal conductance		1.0 nS		
E_i	Inhibitory reversal potential(E_{in})		-80.0 mV		
τ_i	Rise time of inhibitory synaptic conductance(τ_{in})		5.0 ms		
Q_i	inhibitory quantal conductance	10.0 nS	5.0 nS	10.0 nS	
$p_{connect}$	probability of connection	0.05	0.05	0.005	
N_{tot}	Number of total neurons		10000		
p_i	percentage of inhibitory neurons		0.2		
N_e	Number of excitatory neurons		$N_{tot}(1 - p_i) = 8000$		
N_i	Number of inhibitory neurons		$N_{tot}p_i = 2000$		
K_e	mean number of input excitatory synapses : $N_e p_{connect}$	400	400	40	
K_i	mean number of input inhibitory synapses : $N_i p_{connect}$	100	100	10	
K_{ext_e}	number of external excitatory synapse	115	300	90	
P_e	second order polynomial of the phenomenological threshold for inhibitory neuron in mV	P_0	P_{μ_V}	P_{σ_V}	$P_{\tau_V^N}$
		-0.0498	0.00506	-0.025	0.0014
		$P_{\mu_V^2}$	$P_{\sigma_V^2}$	$P_{(\tau_V^N)^2}$	
		-0.00041	0.0105	-0.036	
$P_{\mu_V \sigma_V}$	$P_{\mu_V \tau_V^N}$	$P_{\sigma_V \tau_V^N}$			
0.0074	-0.0012	-0.0407			
P_i	second order polynomial of the phenomenological threshold for inhibitory neuron in mV	P_0	P_{μ_V}	P_{σ_V}	$P_{\tau_V^N}$
		-0.0514	0.004	-0.0083	0.0002
		$P_{\mu_V^2}$	$P_{\sigma_V^2}$	$P_{(\tau_V^N)^2}$	
		-0.0005	0.0014	-0.014	
$P_{\mu_V \sigma_V}$	$P_{\mu_V \tau_V^N}$	$P_{\sigma_V \tau_V^N}$			
0.0045	0.0028	-0.00153			
ν_{ext}	external input		see coupling section		
w_σ	weight of the noise	8	1e-5		
σ	variation of the noise		0.2		
μ	mean of the noise		0.7		
τ_{OU}	mean of the noise		0.7		
	initial condition (random between maximum and minimum)	$\mu_E(kHz) : (0., 0.)$	$\mu_i(kHz) : (0., 0.)$		
		$c_{ee} : (0., 0.)$	$c_{ei} : (0., 0.)$		
		$c_{ii} : (0., 0.)$			
		$W_e(pA) : (5., 0.)$	$W_i(pA) : (0., 0.)$		

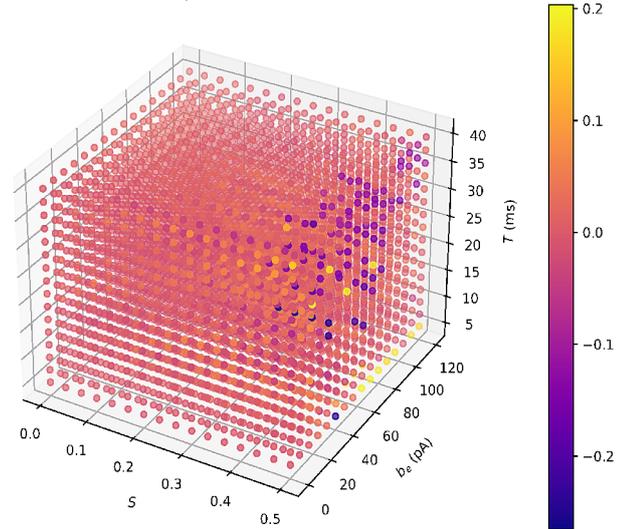
B7	TVB : Monitor
state variable	Mean firing rate of excitatory and inhibitory population, the variation of excitatory and inhibitory firing rate, the co-variation between excitatory and inhibitory firing rate, mean adaptive current of excitatory and inhibitory firing rate
	precision : dt (0.1ms)

A

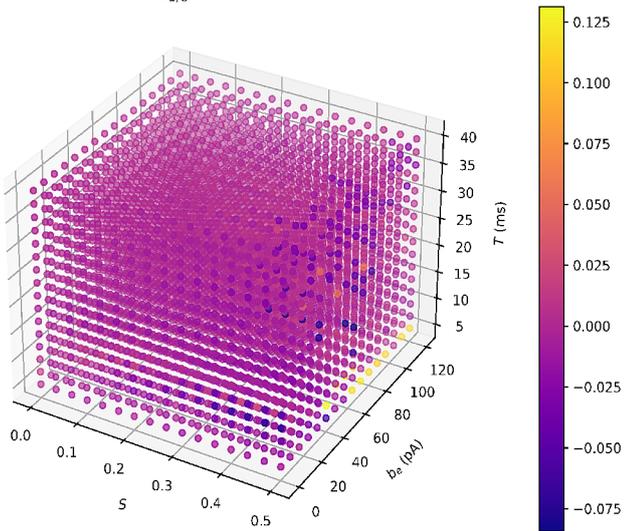
Difference of Relative power of PSD_e in δ band for $E_{L,i} = -64.0$ mV,
 $E_{L,e} = -64.0$ mV

**B**

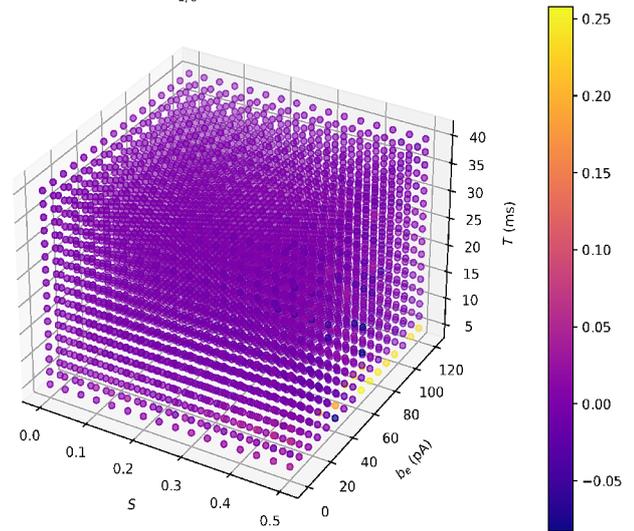
Difference of Relative power of PSD_e in θ band for $E_{L,i} = -64.0$ mV,
 $E_{L,e} = -64.0$ mV

**C**

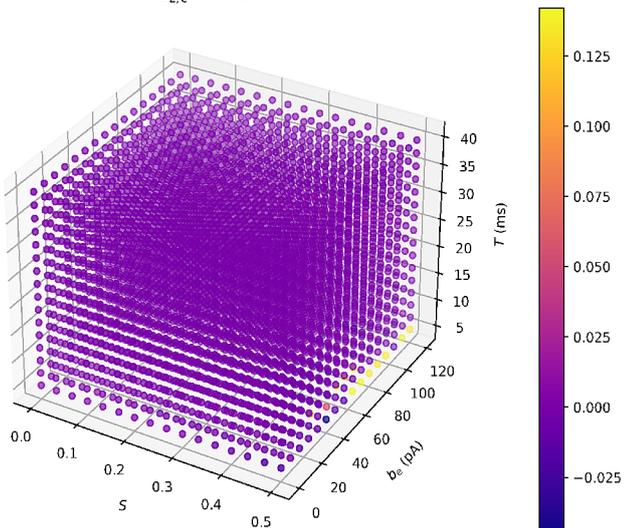
Difference of Relative power of PSD_e in α band for $E_{L,i} = -64.0$ mV,
 $E_{L,e} = -64.0$ mV

**D**

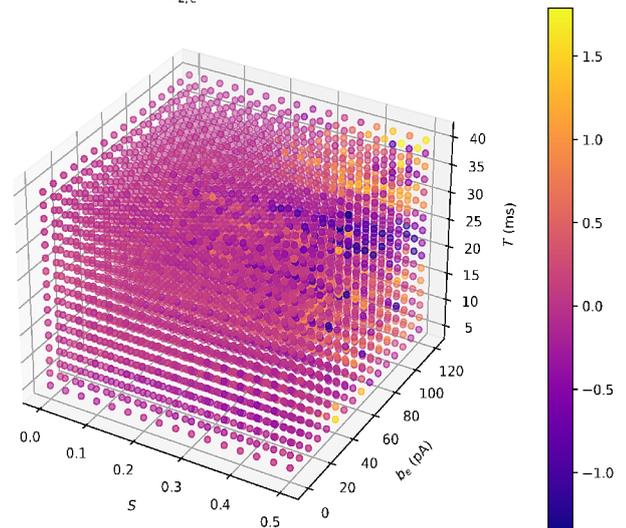
Difference of Relative power of PSD_e in β band for $E_{L,i} = -64.0$ mV,
 $E_{L,e} = -64.0$ mV

**E**

Difference of Relative power of PSD_e in γ band for $E_{L,i} = -64.0$ mV,
 $E_{L,e} = -64.0$ mV

**F**

Difference of Power Law of PSD_e for $E_{L,i} = -64.0$ mV,
 $E_{L,e} = -64.0$ mV



CHAPTER I

Supplementary material of Annex C

Brain simulation as a cloud service: The Virtual Brain on EBRAINS

SUPPLEMENTARY MATERIAL

SUPPLEMENTARY NOTE: THE EBRAINS COLLABORATORY	2
SUPPLEMENTARY METHODS: BRAIN SIMULATION WITH TVB	4
SUPPLEMENTARY METHODS: TVB IMAGE PROCESSING PIPELINE	6
Pipeline processing steps	7
Diffusion MRI tractography	7
Functional MRI processing	8
Brain network model construction	9
Exemplary input data	12
SUPPLEMENTARY NOTE: BIDS APPS	13
SUPPLEMENTARY METHODS: MULTISCALE CO-SIMULATION	14
SUPPLEMENTARY METHODS: HIGH-PERFORMANCE IMPLEMENTATIONS	16
TVB-HPC	16
Component type "derivatives"	17
Component type "coupling"	20
Component type "noise"	21
Benchmark Python and CUDA: ReducedWongWang parameter exploration	21
Fast_TVb	22
SUPPLEMENTARY METHODS: TVB ATLAS AND DATA ADAPTERS	24
SUPPLEMENTARY METHODS: DATA INTEGRATION AND TVB-READY DATA	25
openMINDS metadata annotation	25
BIDS metadata annotation	25
TVB-ready data	26

SUPPLEMENTARY METHODS: ADVANCED USE CASES AND TRAINING	26
Bayesian Virtual Epileptic Patient	26
TVB Mouse Brains	28
INCF training space	28
SUPPLEMENTARY METHODS: DATA PROTECTION IN THE TVB ON EBRAINS CLOUD	29
Cloud computing risks	30
Protective measures	31
Shared responsibility & compliance	32
SUPPLEMENTARY DISCUSSION	34
SUPPLEMENTARY REFERENCES	36

Supplementary Note: The EBRAINS Collaboratory

The EBRAINS Collaboratory provides protected workspaces for researchers to cooperate on projects via the internet, to host and exchange data, code and documentation. It consists of lightweight 'collabs', isolated workspaces with explicit access control--other EBRAINS users can be added to a collab with different roles that have specific permissions, in an environment that follows strict data protection guidelines (see section Supplementary Methods: Data protection in the TVB on EBRAINS cloud). For detailed and continuously updated documentation, please refer to wiki.ebrains.eu/bin/view/Collabs/the-collaboratory.

The Collaboratory is based on open-source tools:

- XWiki (wiki.ebrains.eu) is a wiki platform for collaborative content creation, version control, permission control, search, templating, file management, blogs, calendar functions, discussion forums, and task management.
- Keycloak (iam.ebrains.eu) provides identity and access management, single sign-on authentication, and permission control for all EBRAINS cloud services.
- Seafile (drive.ebrains.eu) is a cloud storage system with web interface to upload, download, manage, share, and synchronize files.
- OnlyOffice (office.ebrains.eu) is an office suite for collaborative word processing, spreadsheets, and presentations.
- JupyterLab (lab.ebrains.eu) is a web-based interactive development environment for Jupyter notebooks.

Seafile implements the EBRAINS Drive, which is the central data-sharing place of EBRAINS. Each collab has a Drive where all data belonging to that collab is centrally stored. A collab's Drive can be accessed by clicking on the link "Drive" in the left sidebar of the collab. With Drive, users can upload,

synchronize and share files with private or public groups and devices. Drive keeps track of the contributions from different authors and the history of the file during its evolution, which enables versioning with roll-back to previous versions and exact tracking of individual contributions. In private collabs data sharing is disabled per default and users have to either actively enable public sharing, or actively specify the permissions of the collab members. Sharing can also be restricted in time, and, e.g., expire after a set date. In addition to a web-interface, Seafile also provides standalone clients for file browsing and data synchronization. Drive enables data exchange not only between users, but also between EBRAINS services. A service can fetch data from a collab's drive and either use it directly or forward it to other services. As an example, an application may load input data from one collab, store intermediary outputs in a second collab, and store final results in a third collab, all with different access permissions. In this way it becomes possible to protect sensitive data at raw or intermediary processing stages from unintended disclosure, and only grant access to final results.

Drive implements role-based access control for permission management that can be individually specified for each collab. To share data or work collaboratively, users can add other users to a collab's team and then define a 'role' for each team member. Viewers have read-only access, Editors can read, modify, create and delete subpages, and Admins have permission to do all of the former, plus adding and removing team members. Permissions within a Collab are invariant across all pages in the wiki and all files in the Drive. For example, a user with Viewer Role will be able to execute a Jupyter notebook that is stored in the collab's Drive, but the user will not be able to save changes to the notebook or write out data. In order to make changes to read-only content, users have the option to copy the content into a collab where they have write permissions. To share results publicly, users can set up public collabs that are read-only accessible without EBRAINS account or share 'public links' for public read-only access to individual files in a private collab. 'Smart links' provide access for each user based on their individual roles. Roles can be defined for individual users or groups of users. Users can be grouped into 'Units' or 'Groups'. Units are used to map static relationships, like membership with an Institution, Country, or Project, and are managed by the administrators of the partnering institutions. Examples for Units are "Spain", "University of Aberdeen", "Institutions", "Projects", "Administrators", or "Sub-Project 1 - Mouse Brain Organisation". To become a member of a Unit, a user can navigate to the Unit's wiki page and formally "Request to Join", by clicking the respective button. The request will then be evaluated by the administrators of the Unit and the User notified via email. In contrast, Groups are used to map dynamic relationships. Groups can be created and managed ad hoc by every user with administration rights for that group and simply add other Users or Units to this Group. Administrators can be appointed at every level of the Groups tree and an administrator that was appointed for a certain parent node is automatically administrator for all child nodes and all descendants. Users and Units are managed in Keycloak, although therein the latter is called 'User groups', while Roles are called 'Client Roles'.

Lab is EBRAINS' main environment for code development. Lab hosts JupyterLab, which is a web-based interface for Jupyter notebooks. Jupyter notebooks are open-source web applications for creating and sharing richly structured documents that can simultaneously contain live code, equations, visualizations and narrative text, which helps keeping detailed track of complex workflows--a prerequisite for reproducible research and effective education. Lab allows users to run

code in the EBRAINS cloud in a private environment that is inaccessible by other users and that is reset after 24 hours of inactivity. As interface for data transfer and control of supercomputing resources EBRAINS uses PyUnicore, a Python wrapper for UNICORE, which contains a RESTful API for authentication, authorisation, user mapping, file transfer and job submission within supercomputing networks. Usage of JupyterLab is introduced with several notebooks that show how to use (Py)Unicore to access a supercomputer, how to implement authorisation flows with cryptographic tokens, or how to read and write data to and from Drive. Another important use of Lab is for deployment of community applications: when a developer creates a service for the EBRAINS community, it is often helpful to document usage with a few code examples. The developer may create a dedicated documentation collab for their service, put example data into the corresponding Drive, set up the library and the required environment in the Lab notebook, and write exemplary output back into the Drive for users to examine. A documentation collab then may be quickly turned into an educational collab by adding users with read access, which allows them to experiment with the code and the documentation from their own JupyterLab instances. To protect sensitive data, users can run EBRAINS services on data stored in their own private collabs, with their isolated Lab computing spaces.

EBRAINS is a platform for community-driven development and therefore invites contributors to program applications and make them available to others on the EBRAINS platform. 'Apps' can be registered and managed with the Community Apps Catalogue to make them visible and conveniently usable. Registration of an App involves curation by a team of specialists that controls the quality of the code, e.g., by checking that applications are well-documented, open source, shared on public repositories, versioned, actively maintained and do not contain malicious code. Once in the catalogue, users can easily add an App to their collabs by clicking on the collab's "Create" button and selecting from the available Community Apps. A versatile Application Programming Interface (API) makes it possible for developers to interconnect Apps with other EBRAINS services such as Drive, Lab or Wiki. Furthermore, developers can create authentication and authorisation flows for their Apps, which makes it easy to control and delegate access to shared resources or confidential data. By adding OpenID connect clients (openid.net) to the application code, Apps can request access tokens for different services, thereby enabling single-sign-on and authorized interaction between services. Read and write operations on Drive can be performed from community apps via Seafile's RESTful (Representational state transfer, a software architecture for interactive applications that use web services) HTTP and Python APIs. Likewise, XWiki comes with a RESTful HTTP API, which allows applications to create, fetch or modify Collaboratory content. EBRAINS uses Google Analytics and Matomo to enable content creators to track traffic between different sites and services. Developers can obtain fine-grained metrics on individual sessions, pages, durations, clicks, geographic origins as well as specific events or actions (e.g., triggering a download, subscribing to a newsletter, etc.). Technical documentation for developers can be found under the URL wiki.ebrains.eu/bin/view/Collabs/the-collaboratory/Technical%20documentation.

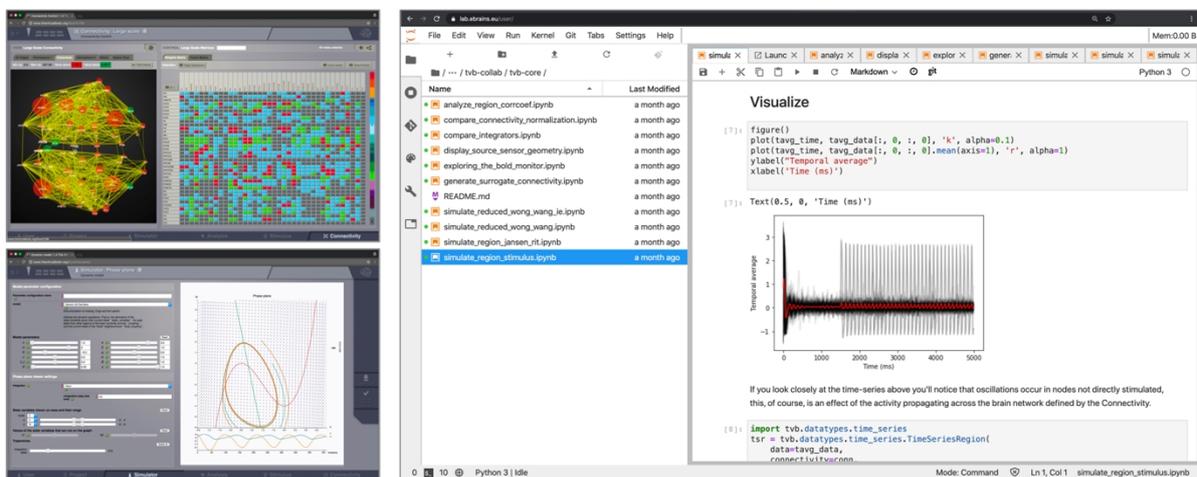
Supplementary Methods: Brain simulation with TVB

TVB is a simulator (Supplementary Figure 1) for brain network models (BNMs) that simulate neural population models that are coupled by structural connectivity, which is an aggregated representation of the brain's white matter axon fiber bundle network. In addition to being

implemented as Python library and deployed as EBRAINS cloud service, TVB is deployed in compiled standalone versions for Linux, Windows and Mac OS as well as execution-ready container images that are freely available for download.

A BNM is a computer model for simulating brain activity based on systems of differential equations that are coupled by a reconstruction of a brain's structural connectome (SC), the white-matter axon fiber bundle network that interconnects brain areas (Breakspear, 2017). Neural populations are simulated by neural mass or neural field models, describing the main modes of activity of coupled neuron networks (Deco et al., 2008). Brain network models simulate neural activity like membrane voltage fluctuations, synaptic current flow, or spike-firing, which is used to predict signals like functional MRI (fMRI) or electroencephalograms (EEG). Predicted signals can be compared with empirical measurements to evaluate and optimize the model or to analyse the underlying model mechanisms (Ritter et al., 2013; Schirner et al., 2018).

Two SC metrics are particularly important for BNMs: coupling weights and coupling delays. Coupling weights quantify how strong brain regions interact, and coupling delays quantify how much time elapses to transmit neuronal signals. Noninvasively, these two metrics are estimated using whole-brain tractography on diffusion-weighted MRI. After the pathways of brain-wide fiber tracts have been reconstructed, they are aggregated according to an atlas parcellation that divides the brain into different regions. The workflow to reconstruct SC is implemented in the cloud service TVB Image Processing Pipeline, which would therefore be the first step in an end-to-end brain simulation workflow. Alternatively, users can use the demo SC shipped with TVB package or find SCs that were computed by other researchers with EBRAINS' KnowledgeGraph (see Supplementary Methods: Data integration and TVB-ready data).



Supplementary Figure 1. TVB simulation services can be accessed via web application GUI (left panel) or Jupyter notebooks (right panel). Both interfaces can be used to configure, run, and analyze brain simulations.

TVB distinguishes between region-level and surface-level simulations. In region-level models the brain is coarsely parcellated into a set of areas, ranging from a few dozen up to several hundred that interact exclusively via SC. In contrast, with 'surface simulations', population models are placed at each vertex of a cortical surface triangulation (the surface of the cortical gray matter can be approximated by a mesh of triangles), which are often on the order of ten thousand or hundred thousand vertices for one cortex. With such surface simulations the large-scale regions of a BNM are

divided into smaller chunks, which allows to simulate the BNM at a higher resolution. In addition to their interaction via long-range SC, the population models at each vertex of a surface simulation interact according to local connectivity kernels that specify the interaction between nearby populations depending on their distance.

The minimal amount of data necessary to perform a region-level simulation is SC. In addition to SC, TVB can import several other data formats and use them for simulation, e.g., surface triangulations for surface simulations. TVB offers a choice of population models that capture different neural population dynamics. For example, the equations of the Generic 2D Oscillator approximate the geometry of the nullclines of simple planar neuron models, which captures typical neuronal population dynamics like multistability, the co-existence of oscillatory and non-oscillatory states and plausible time series dynamics over multiple time scales. Depending on parameters, the Generic 2D Oscillator displays stable fixed points, limit cycles, or a bistable configuration with coexistence of stable limit cycles and fixed points. Other examples of population models in TVB include Wilson & Cowan; Wong, Wang & Deco; Zetterberg, Jansen & Rit; and Stefanescu & Jirsa 3D (Sanz-Leon et al., 2015).

The output of population models are state variable time series that represent ongoing neural population activity like the moment-to-moment mean firing rate or mean membrane potential of an ensemble of neurons. To compare simulated raw neural activity with electroencephalography (EEG) or magnetoencephalography (MEG), forward models and inverted forward models are often used. Forward models are used to approximate electromagnetic signals measurable near the scalp from raw neural activity while inverted forward models are used to approximate signals measurable in the vicinity of neural tissue on the basis of measured EEG or MEG (Hallez et al., 2007; Mosher et al., 1999). Likewise, there exist forward models, in TVB called 'monitors', to simulate Blood-oxygen-level-dependent imaging (BOLD) functional magnetic resonance imaging (fMRI) time series. The BOLD monitor, for example, relates the level of neural activity to oxygen consumption--higher neural activity leads to higher oxygen consumption, which is modelled in TVB by convolution of the neural time series with a hemodynamic response function, e.g., a mixture of gamma functions as implemented in SPM (Friston et al., 1995).

In a typical TVB workflow simulated brain activity is compared with empirical brain activity to tune free model parameters, like the magnitude of long-range coupling, which is necessary because dwMRI tractography only yields networks with relative connection weights. To tune the magnitude of long-range coupling simulated FC is often compared with empirical FC and the parameter is adapted accordingly. Upon model fitting, a typical next step is to analyze whether the optimized parameters reveal group differences, which in turn may point to differences in underlying neural processes. This approach was used in several studies to differentiate between different clinical and non-clinical populations (Aerts et al., 2020, 2018; Costa-Klein et al., 2020; Jirsa et al., 2017; Proix et al., 2017; Zimmermann et al., 2018).

Supplementary Methods: TVB Image Processing Pipeline

This chapter provides details on the steps of the TVB Image Processing Pipeline ('pipeline' in the following), an overview over the structure of output data, and an exemplary input data set. Users control the pipeline via the TVB web GUI (Table 1) to select input MRI data for upload, to configure

the processing steps, and to download the results. To configure processing steps, users can select among a choice of neuroimaging containers, respectively BIDS Apps (see Supplementary Note: BIDS Apps), like fmriprep (Esteban et al., 2019), the Human Connectome Project pipeline (Glasser et al., 2013), and Mrtrix3_connectome (Smith and Connelly, 2019). In addition, users can supply parameters to control the internal execution of steps in these containers. In the following we provide an overview over the steps for diffusion MRI tractography with Mrtrix3_connectome, functional MRI preprocessing with fmriprep and BNM construction with tvb-pipeline-converter.

Pipeline processing steps

Diffusion MRI tractography

MRtrix3 is a powerful and actively developed toolbox for diffusion modelling and tractography (Tournier et al., 2019). Notably, it includes methods to increase the accuracy of tractography and reducing false positives by removing tracks that are anatomically implausible, called Anatomically-Constrained Tractography (Smith et al., 2012). In order to estimate region-to-region coupling strengths, the MRtrix3 programs SIFT, respectively SIFT2, re-weight each track in the reconstructed tractogram such that streamline densities become proportional to the cross-sectional area connecting each pair of brain regions (Smith et al., 2015). The workflow of Mrtrix3_connectome (Smith and Connelly, 2019) is controlled by a Python script that can be modified to match specifics of the input data or the research question. In the following we describe the default setup of the workflow as it is currently implemented. As input, the workflow requires T1-weighted and diffusion-weighted MRI data, as well as at least one phase-reversed dwMRI of the B0 field for susceptibility distortion correction. Outputs are whole-brain tractograms and SC. In between, the following processing steps are performed. First, dwMRI is denoised by removing noise-only principal components (Veraart et al., 2016) using *dwidenoise*. Next, Gibbs ringing artifacts (Kellner et al., 2016) are removed using *mrdegibbs*, and distortions are corrected with *dwipreproc*: eddy current-induced distortion correction and motion correction is performed using FSL *eddy*, and (optionally) susceptibility-induced distortion correction is performed using FSL *topup*. In the next step, B1 bias field inhomogeneities are corrected using *dwibiascorrect* and a brain mask for DWI is created using *dwi2mask* and *maskfilter*. After computing fractional anisotropy maps using *dwi2tensor*, *dwi2response* is used to estimate the response functions for spherical deconvolution. To obtain fiber orientation distributions, spherical deconvolution is performed with *dwi2fod*. For inter-modal registration, the workflow extracts the brain with *ROBEX* and performs a bias field correction on the T1 image in its original space using ANTS *N4BiasFieldCorrection*. Contrast-matched images for inter-modal registration between DWIs and T1 are generated using *mrhistmatch* and the T1w MRI is registered to DWI using *mrregister*. The MRtrix program *5ttgen* in conjunction with FSL or FreeSurfer is then used to segment tissues into cortical grey matter, sub-cortical grey matter, white matter and cerebrospinal fluid. Next, cortical gray matter parcellations are obtained. For the atlases 'desikan', 'destrieux', and 'hcpmmp1', the parcellation information is taken from FreeSurfer's *recon-all* output. For the atlases 'aal', 'aal2', 'craddock200', 'craddock400', and 'perry512', a non-linear registration to the MNI152_T1_2mm template is performed with optionally ANTS or FSL, and the registration result is used to transform the atlas information into individual subject space. Based on this nonlinear registration any atlas defined on the MNI152 template can be used to parcellate the subject's brain, e.g., cytoarchitectonic information from Julich-Brain. Next, the central step in this workflow is performed: whole-brain fibre-tractography using *tckgen* with ACT to control for anatomical

plausibility of constructed tracks using information from the tissue-segmented image (Smith et al., 2012). As additional plausibility criteria, tracks are truncated and re-tracked if a poor structural termination is encountered, respectively cropped when they hit the gray-matter-white-matter interface and when they exceed a length of 250 mm. Seed points are determined dynamically using the SIFT model (Smith et al., 2015). If the number of tracks to be generated is not manually specified, it is set to $500 \cdot N \cdot (N-1)$, where N is the number of brain regions in the parcellation. After a whole-brain tractogram has been generated, SIFT2 (Smith et al., 2015) is used next to filter tractography results to determine streamline weights. SIFT2 optimises per-streamline cross-section multipliers to match a whole-brain tractogram to voxel-wise fibre densities. As last step of the MRtrix workflow, *tck2connectome* is used to aggregate the whole brain tractogram into a region-by-region connectome matrix using the specified region parcellation. In addition to outputting the sum of streamline weights for each region pair, also the mean streamline lengths between each region is output. Optionally, the workflow also outputs track density images, which are useful for visualising and evaluating the results of tractography.

Functional MRI processing

To process fMRI data *fmriprep* can be selected, which is a preprocessing pipeline for fMRI data that is relatively robust to variation in scan acquisition protocols, sequence parameters, or presence of fieldmaps for artifact correction (Esteban et al., 2020, 2019). A characteristic feature of *fmriprep* is its "glass box" philosophy, according to which reports for visual verification are produced after important processing steps. This is an important feature as many steps of MRI preprocessing pipelines are susceptible to errors and verification is therefore generally recommended after major steps. The workflow combines the following software: FreeSurfer, FSL, AFNI, ANTS, BIDS validator and ICA-AROMA. Auxiliary tools are: Python (miniconda), pandoc, SVGO, neurodebian and git. As input the workflow requires data to be in BIDS format, and it must include at least one T1w structural image and one fMRI series. Output contains preprocessed fMRI data, as well as noise components estimated with independent component analysis, which are used to remove associated variance from the fMRI data during the *tvb-pipeline-converter* workflow. The following processing steps are performed: T1w volumes are corrected for intensity non-uniformity using *N4BiasFieldCorrection* and skull-stripped using *antsBrainExtraction.sh*. Brain surfaces are reconstructed using FreeSurfer's *recon-all*, and the temporary brain mask is refined to reconcile ANTs-derived and FreeSurfer-derived segmentations of the cortical gray-matter of Mindboggle. Spatial normalization to the ICBM 152 Nonlinear Asymmetrical template version 2009c is performed through nonlinear registration with the *antsRegistration* tool, using brain-extracted versions of both T1w volume and template. Brain tissue segmentation of cerebrospinal fluid (CSF), white-matter (WM) and gray-matter (GM) is performed on the brain-extracted T1w using FSL *FAST*. Functional data is slice time corrected using *3dTshift* from AFNI and motion corrected using *mcfliirt* from FSL. Distortion correction is performed using an implementation of the TOPUP technique using *3dQwarp*. This is followed by co-registration to the corresponding T1w using boundary-based registration with nine degrees of freedom, using *bbregister* from FreeSurfer. Motion correcting transformations, field distortion correcting warp, BOLD-to-T1w transformation and T1w-to-template (MNI) warp are concatenated and applied in a single step using *antsApplyTransforms* using Lanczos interpolation. Physiological noise regressors are extracted applying CompCor. Principal components are estimated for the two CompCor variants: temporal (tCompCor) and anatomical (aCompCor). A mask to exclude signal with cortical origin is obtained by eroding the brain mask, ensuring it only contained

subcortical structures. Six tCompCor components are then calculated including only the top 5% variable voxels within that subcortical mask. For aCompCor, six components are calculated within the intersection of the subcortical mask and the union of CSF and WM masks calculated in T1w space, after their projection to the native space of each functional run. Frame-wise displacement is calculated for each functional run using the implementation of Nipype. ICA-based Automatic Removal Of Motion Artifacts (*AROMA*) was used to generate aggressive noise regressors as well as to create a variant of data that is non-aggressively denoised (see section `thevirtualbrain/tvb-pipeline-converter` for details on aggressive vs. non-aggressive cleaning). For details, please refer to fmriprep's documentation (<https://fmriprep.readthedocs.io/>).

Brain network model construction

The container `tvb_converter` (Table 1) takes the outputs of `Mrtrix3_connectome` and `fmriprep` as input and provides as output TVB-ready connectomes in BIDS format (Supplementary Table 1) and in native TVB-ready format (Supplementary Table 2), which can be directly used to run BNM simulations in TVB. Outputs include SC, FC, EEG/MEG projection matrices, and brain surface triangulations. As first step, the workflow performs a "non-aggressive" cleaning of motion artefacts in the output fMRI data that was processed with `fmriprep`. Following its glass box philosophy, `fmriprep` involves the researcher in critical decisions about the denoising strategy, as it can have considerable effects on the ensuing analyses. To this end, `fmriprep` provides both aggressively and non-aggressively denoised fMRI. The output of the `fmriprep` workflow already contains ICA-AROMA denoised 4D NIFTI files mapped to MNI space. Based on these results the `tvb-pipeline-converter` performs non-aggressive denoising following the approach described in Pruim et al. (2015) using the independent components (ICs) together with their classification obtained in the `fmriprep` processing. ICA-AROMA classifies ICs into movement-related and movement-unrelated ICs. It does not require manual training of the classifier--the classifier is already trained using four theoretically motivated spatial and temporal features: high-frequency content, correlation with realignment parameters, edge fraction, and CSF fraction of each IC (Pruim et al., 2015). With aggressive denoising the entire temporal waveform of movement-related ICs is regressed from the data, analogous to the commonly done regression of nuisance parameter time courses (like mean global signal, mean tissue class signal, frame-wise displacement, motion parameters) from the data. Critically, all variance associated with such "nuisance" time courses is removed, including shared variance with actual brain signals. This is problematic, because global fMRI time courses often share a lot of variance with neural activity: even if two signals originate from different locations in the brain, they may still be highly correlated. Therefore, regressing out global signals likely removes signals of interest (Liu et al., 2017; Murphy and Fox, 2017). As an alternative, the non-aggressive approach based on ICA-AROMA is more conservative by first performing a regression on the full set of IC time series, including both signal and noise ICs. Then, to clean the data, only the motion-related regressors are subtracted from the data, which specifically removes variance associated with motion-related-ICs that is not shared by the remaining ICs. Problematically, with this non-aggressive approach motion-related dynamics that were not identified as such are specifically retained in the data. Conversely, as mentioned, the drawback of aggressive denoising is that it may remove shared variance between signal and noise. To clean fMRI data non-aggressively the workflow uses `fsf_regfilt` with the provided AROMA ICs and a list of (motion-related) ICs to reject. After this step, further nuisance regression may be performed (if it fits the study design) as well as removal of linear trends and high-pass filtering. It is, however,

important that during such a second regression step variables correlated with motion are not used as regressors, as this may re-introduce motion-related waveform components into the time series.

After denoising, tvb-pipeline-converter computes region-average fMRI time courses by resampling the brain parcellation to fMRI resolution and averaging over all voxel time series in each region. The converter also outputs cortical surfaces, which are used for detailed surface simulations and for EEG or MEG source modelling, e.g. in order to extract region-average EEG or MEG source activity, which can be used to constrain model dynamics (Schirner et al., 2018). The converter merges the left and right hemisphere cortical surface triangulations reconstructed by FreeSurfer and generates a mapping between each vertex and the large-scale regions of the brain atlas. Each vertex of the cortical white matter and pial surface triangulations are associated with the region-label of the region at that location. This step is carried out with HCP Connectome Workbench for atlases that are defined in volumes (e.g., "aal", "aal2", "craddock200", "craddock400", "perry512", atlas names as used in MRtrix3_connectome workflow) and outputs a GIFTI label file for each hemisphere. For atlases defined on surfaces (e.g., "desikan", "destrieux", "hcpmmp1"), the 'annot' files from FreeSurfer are used to obtain the region mapping. The following atlases are currently natively supported, for other atlases the workflow needs to be adapted: "aal", "aal2", "craddock200", "craddock400", "desikan", "destrieux", "hcpmmp1", "perry512".

Next, the MNE toolbox (Gramfort et al., 2013) is used to compute forward and inverse models for electromagnetic source imaging. First, surfaces are decimated to 30,000 triangles and the region-mapping is obtained by nearest neighbor interpolation in the original high-resolution surface. For the head model, BEM surfaces using the FreeSurfer watershed algorithm are constructed. To make the workflow automatic, standard EEG montage locations are projected onto the individual head surfaces. The outputs of this step are a projection or lead-field matrix (LFM), EEG sensor location coordinates, and a mapping between surface vertices and large-scale regions. The LFM has the dimensions $M \times N$, with M being the number of vertices on the cortical surface and N the number of EEG sensors. Next, surfaces are exported: the downsampled pial surface from FreeSurfer (used to define source space) as well as the BEM surfaces of the inside and the outside of the skull and the scalp. Finally, the converter outputs the SC weights and distances matrices as well as region centroids, average orientations (average over all vertex-normals), surface areas, and two files that indicate to which hemisphere each region belongs and whether it is a cortical or a subcortical region. All output files are plaintext ASCII files that are zipped for uploading to TVB. As last step, the converter generates metadata for the BIDS output (see Supplementary Methods: Data integration and TVB-ready data). Supplementary Table 1 summarizes the folder and filename structure in BIDS format and Supplementary Table 2 summarizes folder and filename structure for outputs in TVB format, which are all stored in the output file "TVB_output.zip" that can be imported into TVB using the "Upload Connectivity ZIP" functionality of TVB.

Path and filename		Description
sub-<id>/		
	anat/	
	sub-<id>_desc-centroid_morph.tsv	table with region names and 3D location of centres
	sub-<id>_space-individual_dparc.label.gii	Region mapping, assign each surface vertex to a region in the connectome

	sub-<id> space-individual dial.surf.gii	FreeSurfer: cortical surface, both hemispheres
	sub-<id> space-individual innerskull.surf.gii	FreeSurfer: inner skull surface
	sub-<id> space-individual outerskull.surf.gii	FreeSurfer: outer skull surface
	sub-<id> space-individual scalp.surf.gii	FreeSurfer: scalp and face surface
connectivity/		
	sub-<id>_desc-distance_conndata-network_connectivity.tsv	structural connectivity: connection distances (mm)
	sub-<id>_desc-weight_conndata-network_connectivity.tsv	structural connectivity: connection weights
	sub-<id> conndata-network connectivity.json	structural connectivity: metadata
eeg/		
	sub-<id> desc-eeg proi.tsv	EEG lead field matrix
	sub-<id> task-simulation electrodes.tsv	EEG 3d coordinates electrode positions
func/		
	sub-<id>_task-rest_atlas-desikan_timeseries.tsv	functional MRI time series for parcellation (cortical and subcortical structures)
meg/		
	sub-<id>_desc-meg_proj.tsv	MEG lead field matrix
	sub-<id>_task-simulation_space-anat_sensors.tsv	MEG 3d coordinates sensor locations and normals

Supplementary Table 1. Naming schema and description of TVB Processing Pipeline output files in BIDS format.

Level	Mandatory filename part	Description	Format	Output tvb-pipeline-converter
Minimum large-scale	weight	SC coupling weights	N x N	sub-<id>_Connectome/ weights.txt
	tract	SC track lengths	N x N	sub-<id>_Connectome/ tract_lengths.txt
	centres	Centers of brain regions	N x 4; column 1: region label; columns 2 – 4: 3D coordinates	sub-<id>_Connectome/ centres.txt
Extended large-scale	area	Surface areas (mm ²)	N x 1	sub-<id>_Connectome/area.txt
	cortical	cortical or subcortical	N x 1; 1: cortical; 0: subcortical	sub-<id>_Connectome/cortical.txt
	orientation	orientation of the average normal vector	N x 3; 3D coordinates	sub-<id>_Connectome/ orientation.txt
	hemisphere	left or right	N x 1; 1: right; 0: left	sub-<id>_Connectome/hemisphere.txt
Surfaces <surf> = {"Cortex", "inner_skull_surface", "outer_skull_surface", "outer_skin_surface"}	vertices	surface triangulation	N x 3; 3d coordinates	sub-<id>_<surf>/ vertices.txt,
	triangles	surface triangulation	N x 3; vertex indices	sub-<id>_<surf>/ triangles.txt
	normals	surface triangulation	N x 3; 3d coordinates	sub-<id>_<surf>/ normals.txt
Region-mapping	N/A	associates surface vertices with large-scale regions	M x 1; region label	sub-<id>_region_mapping.txt
Sensors: EEG, SEEG, MEG	N/A	sensor locations	K x 4 or K x 7; column 1: channel name; columns 2-4: 3d coordinates; columns 5-7: normals	sub-<id>_EEG_Locations.txt
Projection matrix	N/A	mapping between regions/vertices and sensors	K x N or K x M	sub-<id>_EEGProjection.mat

Supplementary Table 2. Naming schema and description of TVB Processing Pipeline output files in TVB format. N is the number of brain regions, M the number of surface vertices, K the number of sensors.

Exemplary input data

In this section we describe how to prepare MRI data for processing with the TVB Image Processing Pipeline. To construct structural connectivity (SC) and functional connectivity (FC), the pipeline requires the following MRI data:

- diffusion-weighted MRI (dwMRI), for SC;
- functional MRI (fMRI), for FC;
- anatomical T1-weighted MRI, for parcellation and surface reconstruction;
- field inhomogeneity maps and pairs of phase-encoding reversed acquisitions to correct susceptibility distortions.

In addition to diffusion-weighted and functional MRI data, computing SC and FC requires at least one high-resolution T1-weighted structural MRI for parcellating the subject's brain and, optionally, to reconstruct brain surfaces. If the input data was not corrected for susceptibility distortions, then additionally phase-reversed images of the B0 field in dwMRI data and fieldmaps for fMRI are required. To correct for susceptibility distortions, dwMRI and fMRI processing workflows often use so-called fieldmaps. When an object is placed in a homogeneous magnetic field, like a head in an MRI scanner, it will disrupt that field so it will no longer be homogeneous. This leads to signal loss from T2*-dephasing and spatial signal mismapping that manifests in considerable displacements of brain tissues. Fieldmaps quantify field inhomogeneities in order to correct the difference between the homogeneous field and the actual field. There are different approaches for field-mapping, e.g., phase-difference B0 mapping (typically consisting of one phase difference image and one magnitude image), direct B0 mapping sequences, or fieldmaps that are based on acquiring at least two images with varying phase-encoding directions. In BIDS, fieldmaps are stored in the folder `fmap/` and linked with the specific scans they were acquired for by filling the `IntendedFor` field in the corresponding JSON file. The value may contain one or more filenames with paths relative to the subject subfolder, e.g.

```
{
  "IntendedFor": ["ses-pre/func/sub-01_ses-pre_task-motor_run-1_bold.nii.gz",
                 "ses-post/func/sub-01_ses-post_task-motor_run-1_bold.nii.gz"]
}
```

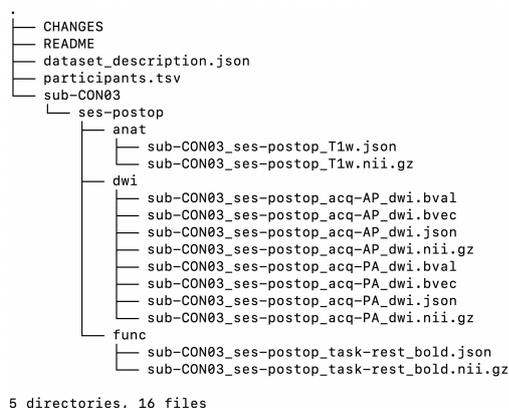
In the `Mrtrix3_connectome` workflow susceptibility distortions are corrected with the FSL tool `topup`, which is an important prerequisite for the Anatomically Constrained Tractography (ACT) algorithm (Smith et al., 2012). ACT reduces false positives by removing tracks that are biologically implausible, e.g., because they terminate in a ventricle. Importantly, the JSON sidecar files must contain the BIDS fields `"SliceEncodingDirection"` and `"SliceTiming"` to correctly distinguish the main dwMRI scans from phase-reversed B0 (or gradient) scans for distortion correction. Since the phase-reversed images are not fieldmaps, but regular dwMRI acquisitions, they are not stored in the folder `fmap/` but in the same folder as the main acquisition. Compliant with BIDS, the `'acq-XX'` key-value pair of the filename denotes the phase-encoding direction.

As exemplary input data set for the TVB image processing pipeline we select the data set "TVB time series and connectomes for personalized brain modeling in brain tumor patients" (DOI10.25493/1ECN-6SM), which can be obtained from the EBRAINS KnowledgeGraph

(search.kg.ebrains.eu). This dataset was already used for simulation with TVB and results were published in two studies (Aerts et al., 2020, 2018). When the data set entry is opened in KnowledgeGraph, users see meta information like DOI, License, Abstract and a link to the raw data files, which are hosted at openneuro.org. The imaging data set contains multi-shell high angular resolution dwMRI, b0 dwMRI with reversed phase encoding directions, resting-state fMRI and T1w MRI from 11 healthy controls and 25 tumor patients prior to a tumor resection operation. The second part of the data set contains post-operative data from 19 of the 25 tumor patients as well as data from 10 of the 11 healthy control subjects. Here we use subject 'sub-CON03' from the post-operative data set. To this end we download the full data set by clicking on the "Download" button and following the provided instructions. Next, we delete the imaging data from all other subjects in order to make data transfer between local file systems, the EBRAINS platform and the supercomputers faster. That is, all folders 'sub-CONXX' and 'sub-PATXX' are deleted except for the folder 'sub-CON03'. To make the metadata consistent, we need to adapt the file *participants.tsv*, where we remove all rows except the one for our selected subject. When working with BIDS data sets it is often useful to check their validity using the BIDS validator, for example, by using its deployment as the Docker container bids/validator (<https://hub.docker.com/r/bids/validator>), which can be invoked like this (assuming the main folder of the data set has the path /path/to/demodata):

```
docker run -ti --rm -v /path/to/demodata:/data:ro bids/validator /data
```

For our demo data set the BIDS validator reported zero errors and one warning that indicated that *participants.tsv* contains custom values, which we ignored as this is irrelevant to pipeline processing. The final directory tree is shown in Supplementary Figure 2. The data set is now ready to be processed with the TVB Image Processing Pipeline via the TVB web GUI.



Supplementary Figure 2. Directory tree of the exemplary BIDS MRI data set.

Supplementary Note: BIDS Apps

The "Brain Imaging Data Structure" (BIDS) is an emerging standard for organizing and annotating neuroimaging datasets (Gorgolewski et al., 2017, 2016) that is currently being adopted by an increasing number of projects. BIDS entails a schema to arrange data and metadata within a data set folder. Naming conventions specify the formation of filenames as a series of key-value pairs and a file type suffix. Data dictionaries, implemented as TSV and JSON files, store key-value pairs that identify important features of a data set. BIDS Apps are container images that implement neuroimaging pipelines that take BIDS formatted datasets as input. All BIDS Apps have the same core

set of command line arguments, making usage straightforward across platforms. Containerization of all required dependencies makes it possible that BIDS Apps do not depend on any software outside of the image other than the container engine. Validation software automatically check compliance of a given dataset with the BIDS standard (e.g., BIDS validator: <https://hub.docker.com/r/bids/validator>). In order to create BIDS data sets we refer to the BIDS specification (doi.org/10.5281/zenodo.3686061) and recommend usage of automatic tools based on dcm2niix (<https://github.com/rordenlab/dcm2niix>), the BIDS validator App (<https://hub.docker.com/r/bids/validator>), and the steps outlined in the original BIDS publication (Gorgolewski et al., 2016), namely

- Convert DICOM files to NIFTI
- Create folder structure, rename, and copy NIFTI files
- Add remaining data
- Add missing metadata
- Validate the data set with a BIDS validator.

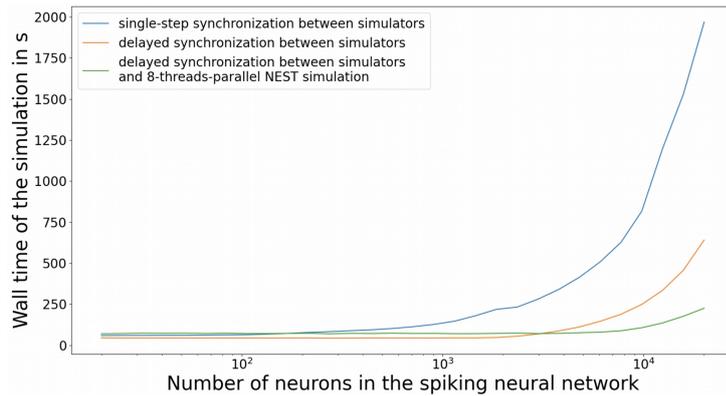
A number of (semi-)automatic tools have been developed that take DICOM as input and generate BIDS data sets based on the DICOM-to-NIFTI converter *dcm2niix* (<https://github.com/rordenlab/dcm2niix>). To build and run container images a platform like Docker (<https://www.docker.com/>) is required. While it is typically straightforward to deploy and run Docker on personal computers and cloud services, however, the Docker Engine is rarely available on HPCs, due to security risks. For HPC usage there exist alternatives like Sarus, Shifter or Singularity, which do not require root access for container execution and are thus safer to use on multi-tenant systems. In our tests, the three BIDS Apps worked without modification with Sarus and Shifter.

Supplementary Methods: Multiscale Co-Simulation

Multiscale Co-Simulation is implemented in the form of two toolboxes: the TVB-Multiscale toolbox and the Parallel CoSimulation toolbox (Table 1). Both are based on common concepts and architectures, but they are independently developed to focus on different goals: TVB-Multiscale focusses on rapid prototyping of scientific use cases while Parallel CoSimulation focusses on the optimization of co-simulation performance. The idea behind multiscale co-simulation is to study multiscale neural interaction. Population model state variables are coupled with single neuron state variables or parameters to exchange inputs between the two scales. For example, TVB state variables that simulate ongoing population firing are used to inject spikes into a NEST spiking network by sampling spike times from a probability distribution in dependence of the instantaneous firing rate of the population model. Vice versa, the mean activity of a NEST neuron network is used to inform inputs to a TVB population model. For example, averaging over spike trains of a NEST network can be used to update instantaneous population firing rates of TVB variables. Coupling can be unidirectional, to study effects of large-scale inputs on small-scale spiking-network activity, or bidirectional, to study how both scales mutually interact. The bidirectional case enables to "substitute" large-scale population models by small-scale spiking networks to simulate one or more specific populations of a brain network model on a finer scale. Large-scale inputs to the substituted populations will then be forwarded to the small-scale network, while small-scale activity will be averaged, or transformed with a custom-defined transformation function, and forwarded as input to the connected large-scale nodes. Like in the case of large-scale-only brain network models, it is

possible to subsequently input simulated neural activity into forward models to simulate typical neuroimaging signals like fMRI or EEG. The TVB to NEST interface is based on the creation of TVB "proxy" nodes within the spiking network model. Proxy nodes are NEST stimulation devices used to inject spikes or currents into NEST neurons. The interface allows to generate currents or spike trains with a desired first order (mean firing rate), or second (correlations) or higher order statistics such that the statistics of the produced spikes or currents follow the corresponding statistics of the large-scale population activity. Proxy nodes can be coupled to NEST spiking networks with user defined connection weights and delays, in order to simulate large-scale features of coupling on the small scale. To compute inputs from NEST to TVB, NEST recording devices are used to aggregate the activity of spiking neuron populations. Multiscale Co-Simulation gives the user flexibility for custom configuration of co-simulations regarding the network structures, the state variables used for coupling, the transformation functions and the devices for computing and applying inputs, the allocation of computational resources, and the storage of results.

The Parallel CoSimulation toolbox is currently under development to optimize the performance of co-simulation as well as for integration with the TVB-Multiscale toolbox. The idea for optimization is to reduce the number of costly communication operations between NEST and TVB, because inputs often do not need to be exchanged in every single time step of the model integration. Rather, it is often the case that the model does not require instantaneous interactions, which depends on the axonal transmission delays in the network. Consequently, the Parallel CoSimulation toolbox therefore employs a strategy where each simulator integrates their respective model equations independently and in parallel for several time steps, until again an exchange of inputs becomes unavoidable. Put differently, costly synchronization operations to exchange inputs often do not need to be performed in every single time step but can also be delayed until they become critical for model integration. The toolbox uses MPI (Message Passing Interface) for communication and is implemented in a modular fashion: independent modules for simulation and for input transformation enable to allocate different compute nodes to each module allowing to scale co-simulations over multiple compute nodes. In the following we provide a benchmark for the performance of the optimized co-simulation algorithms implemented in the Parallel CoSimulation toolbox. The goal for the optimization was to reduce the costs of communication such that the wall time for co-simulation is close to the time that the slower of the two would need to simulate its part of the network independently. For the benchmark 102 regions were simulated in TVB with the Mean Adaptive Exponential model; two of these regions were simulated in NEST using Adaptive Exponential Integrate-and-Fire spiking networks of varying sizes, 80 % excitatory neurons and a connection probability of 0.05 (Supplementary Figure 3). Simulations are performed on a DELL Precision-7540 (Intel® Xeon(R) E-2286M CPU @ 2.40GHz × 16, 64 GB of RAM) and an Ubuntu operating system. As expected, the implemented optimizations ("delayed synchronization") reduces the wall time for co-simulation. and wall time scales with the number of threads used for integrating NEST (Supplementary Figure 3). Up to a spiking network size of 300 neurons NEST was faster than TVB, but slower for larger spiking networks.



Supplementary Figure 3: Wall time benchmark results with the Parallel CoSimulation toolbox. The x-axis represents the total number of neurons equally distributed between the two regions. The y-axis represents the wall time to simulate one second of biological time. The blue curve represents the case without optimization, where inputs are exchanged at every time step of integration. The orange curve represents the optimized case where inputs are communicated only when necessary, depending on the minimum time delay of the model. For the green curve eight threads were used for simulating NEST instead of only one as in the other simulations.

Supplementary Methods: High-performance implementations

To make BNM simulation with the main TVB software faster the Python compiler Numba is used, which translates a subset of Python and NumPy into high-performance machine code. However, the central loop for integrating the general evolution equation of the BNM (Sanz-Leon et al., 2015) is implemented in pure Python for modularity and generality and therefore constitutes a bottleneck for simulation speed, which is why two dedicated high-performance implementations were created: TVB-HPC and Fast_TV, which we describe in the following.

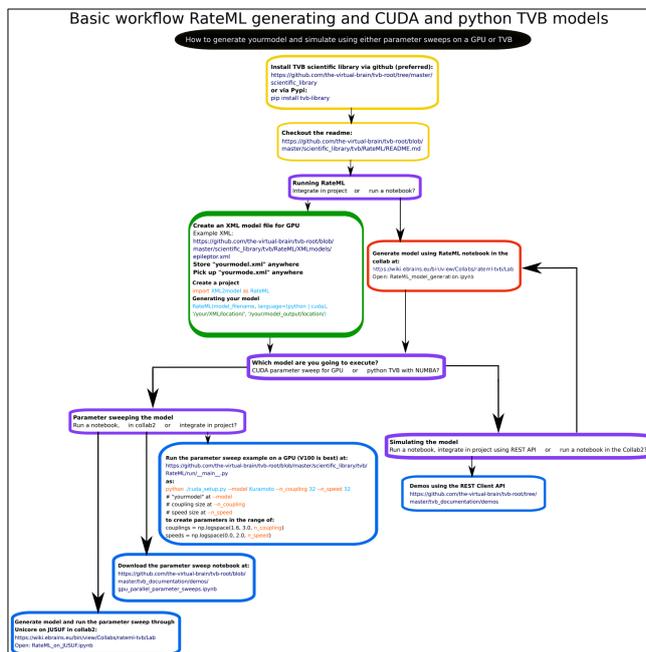
TVB-HPC

For TVB-HPC the domain-specific language (DSL) RateML was developed (Table 1), which allows to specify custom population models without requiring knowledge about how to optimally implement such models: Python code for CPU and CUDA code for GPU is automatically produced from RateML model specifications. The resulting Python code uses Numba vectorization (Lam et al., 2015) to integrate the population models. The resulting CUDA code uses the highly parallel architecture of modern GPUs by spawning one thread for every simulated parameter set, allowing to simulate multiple models in parallel. RateML is based on the domain-independent language 'LEMS' (Vella et al., 2014), which supports the declarative description of model components in a concise XML representation. The PyLEMS expression parser is used to check and parse mathematical expressions (Vella et al., 2014). Exemplary RateML ports of the TVB population models Epileptor, Kuramoto, 2D Oscillator, Montbrio and ReducedWongWang are provided with the main TVB software package. During the automatic code generation, the RateML model equations are mapped to *generalized universal function* (gufuncs) using Numba's *gufunc* decorator, which compiles a pure Python function directly into machine code that can be used to operate on NumPy arrays. An example of a Numba generated *gufunc* function is displayed in Supplementary Listing 1. In this example the *gufunc_numba_dfun_Epileptor* accepts two n and m sized float64 arrays and 19 scalars as input and returns a n sized float64 array. The benefit of writing NumPy gufuncs with Numba's decorators, is that it automatically uses parallel operations such as reduction, accumulation and broadcasting to efficiently implement an algorithm.

```
@guvectorize([(float64[:,], float64[:,], (float64 * 19), float64[:,])],
             '(n),(m)' + ',()' * 19 + '->(n)', nopython=True)
def _numba_dfun_EpileptorT(vw, coupling, a, b, c, d, r, s, x0, Iext, slope,
                           Iext2, tau, aa, bb, Kvf, Kf, Ks, tt, modification, local_coupling, dx):
    c_pop1 = coupling[0]
    c_pop2 = coupling[1]
    c_pop3 = coupling[2]
    c_pop4 = coupling[3]
    ... # calculate derivatives
    return dx
```

Supplementary Listing 1. Example of a gufunc header for the epileptor model, where the variable vw holds the input and dx the computed output derivatives.

RateML supports the specification of population model equations, coupling functions and noise functions. The implementation of TVB-HPC for automatic CUDA code generation differs from the Python variant, due to architectural constraints of the main TVB implementation. For CUDA, the constructs *Parameter* and *DerivedParameter*, for performing parallelized parameter space exploration on GPU, as well as the constructs *coupling* and *noise*, for specifying custom coupling and noise functions, are available, but not supported for Python code generation. Supplementary Figure 4 shows a workflow diagram that introduces usage of RateML for the two different output languages on local hardware and in the EBRAINS Collaboratory. Using RateML involves creating a Python object of the RateML class with the arguments model filename, language, XML file location and model output location. The model filename is the name of the XML file and will then also be used as name for the Python model class, respectively the CUDA kernel name. The TVB-HPC collab (Table 1) contains a ready to be executed Jupyter notebook that demonstrates the entire TVB-HPC workflow.



Supplementary Figure 4. RateML workflow for generating and simulating Python and CUDA code either on own hardware or in a ready to be executed notebook available in the EBRAINS Collaboratory (Table 1).

Component type "derivatives"

	DSL construct	Python	CUDA
File header		<Lems description = "XML for ReducedWongWang">	
ComponentType		<ComponentType name = "derivatives">	
Parameter		N/A	<Parameter name="global_speed"

Derived parameters	N/A	<pre> dimension="float"/> <DerivedParameter name="rec_speed_dt" value="1.0f/globalspeed"/> <DerivedParameter name="nsig" value="42"/> </pre>
Constants		<pre> <Constant name="gamma_E" dimension="lo=-3.0, hi=-1.0, step=0.1" value="-2.0" description="Excitatory population kinetic param"/> <Constant name="tau_E" value="100.0" description="[ms]. Excit. Pop. NMDA decay time c"/> <Constant name="I_0" value="0.382" description="[nA]. Effective external input"/> <Constant name="w_I" value="0.7" description="Inhibi. pop. ex. input scaling weight"/> <Constant name="G" value="2.0" description="Global coupling scaling"/> <Constant name="J_NMDA" value="0.15"/> <Exposure name="V" dimension="sin(V)"/> <Dynamics> </pre>
Exposures		
Dynamics	State variables	<pre> <StateVariable name="V" dimension="0.0, 1.0" exposure="0.0, np.inf"/> </pre>
	Derived variables	<pre> <DerivedVariable name="imintau_E" value="-1.0*tau_E"/> <DerivedVariable name="w_I__I_0" value="w_I * I_0"/> <DerivedVariable name="G_J_NMDA" value="G*J_NMDA"/> <DerivedVariable name="w_plus__J_NMDA" value="w_plus * J_NMDA"/> <DerivedVariable name="tmp_H_E" value="a_E * (w_E__I_0 + w_plus__J_NMDA * V + c_pop1 - JI*W) - b_E"/> </pre>
	Conditional derived variables	<pre> <ConditionalDerivedVariable name="ydot0"> <Case condition="x &lt; 0.0" value="x"> <Case condition="else" value="y"> </ConditionalDerivedVariable> </pre>
	Time derivatives	<pre> <TimeDerivative variable="dV" value="(imintau_E*V)+(tmp_H_E*(1-V)*gamma_E)"/> </Dynamics> </Lems> </pre>
Dynamics		
File header		

Supplementary Table 3. Different DSL constructs for Python versus CUDA for the derivatives component type. Note that the ConditionalDerivedVariable construct is for demonstration purposes and does not exist in the ReducedWongWang model (Deco et al., 2014).

The component type "derivatives" supports the specification of population model equations in the form of time derivatives. The expression parser embedded in the LEMS library allows the following function expressions for the value attributes of the constructs: 'exp', 'log', 'sqrt', 'sin', 'cos', 'tan', 'sinh', 'cosh', 'tanh', 'abs' and 'ceil'. The following operands can be used to specify the mathematical expressions: '+', '-', '*', '/', '^', and '~'. The power symbol in the expression fields needs to be entered within curly brackets, e.g.: {x^2}. Comparison operators must be specified with escape sequences, e.g. '\<(=);' for the "less than or equal to" and '\>(=);' for the "greater than or equal to" operator. See the original publication introducing LEMS for more documentation (Cannon et al., 2014). Supplementary Table 3 compares how DSL constructs in RateML differ for Python versus CUDA for the *derivatives* component type, using the ReducedWongWang population model as an example. The sequence of constructs in this example shows the order in which they need to appear in the XML file. The component type also supports constructs that define initial conditions and additional dynamical equations not captured by the main population model. The *Parameter* construct defines parameters to be used in a parameter sweep; the *name* keyword specifies the name, and the *dimension* keyword specifies the type of the parameter. This sets the variable, which is a single parameter for a CUDA kernel, to receive a value from the parameter array, resulting in the line:

```
const float global_speed =(params[(size * (i_par)) + id]).
```

Here, *i_par* represent the order in which the parameters have been entered in the XML file, *params* represents the parameter array, *size* represents the total number of threads and *id* is the current thread. An example on how to implement the model in Python using the PyCuda library is available in the run folder of the main TVB GitHub repository (Table 1). The following lines of code from this example, show how to setup the parameter array:

```
couplings = np.logspace(1.6, 3.0, 32)
speeds = np.logspace(0.0, 2.0, 32)
params = itertools.product(speeds, couplings)
params = np.array([vals for vals in params], np.float32)
```

This creates a range of 32x32 possible parameters combinations of the couplings and speeds array. Each thread of the CUDA kernel will take a single parameter combination of this array and runs a TVB simulation with this combination. The *DerivedParameter* construct specifies parameters which are computed during run-time and which depend on parameters used in sweeps. This construct will create a *float* with the *const* keyword, which is initialized with the result of the *value* keyword. Special derived parameters are *rec_speed_dt* and *nsig*. The first one is the inverse of coupling speed (no conduction delays if unset) and the second one is a multiplicative parameter to scale noise variance (see sections on coupling and noise components). The *Constant* construct is used to create constants of the type *float*. The *value* field represents the initial value and a description can be entered optionally in the *description* field. For the Python models either a domain must be specified in the *dimension* field, or *'None'*, to exclude it from parameter sweeping. This domain is used to generate GUI slider bars in the TVB Framework. For the CUDA code generation, the *dimension* field can be omitted, because no slider bars are generated for these models. The *Exposures* construct specifies variables that the user wants to monitor and that are returned from the simulation; *name* is the name of the variable to be monitored and the keyword *dimension* defines any mathematical operation performed on the variable. The *StateVariable* construct is used to define the state variables of the model. The *dimension* field can be used to initialize state variables. The *exposure* field sets the state variable's upper and lower boundaries, for both languages. These boundaries ensure that the values of state variables stay within the set boundaries and resets the state if the boundaries are exceeded. The *DerivedVariable* construct can be used to define temporary variables that depend on the state of state variables and which then appear as terms in the time derivative equations. The related mathematical expression would be defined as *name=value*. The *ConditionalDerivedVariable* construct can be used to create if-else constructs. Within the construct a number of *Cases* can be defined to indicate the actions taken for the different conditions, as defined by the value of the *condition* keyword. The *value* field holds an expression to be evaluated and assigned to the variable indicated with the *name* field. Specification of one case is mandatory, two cases will produce an if-else structure and more than two cases will result in 'else-if' statements. The *TimeDerivative* construct combines the defined parameters and variables to construct the actual expression of the model's differential equation. The *variable* field specifies the variable on the left-hand side of the differential equation, while the *value* field specifies the right-hand side expression.

DSL construct	Python	CUDA
Coupling function	<code>class Coupling()</code>	<pre> <ComponentType name="coupling_pop1"> <Parameter name="V_j" dimension='0' /> <DerivedParameter name="c_pop1" value="global_coup * local_coup" /> <Constant name="c_c_a" value="1" /> <Dynamics> <DerivedVariable name="pre" value="sin(V_j-V)" /> <DerivedVariable name="post" value="c_c_a * 2" /> </Dynamics> </ComponentType> </pre>
Noise function	<code>class Noise()</code>	<code><ComponentType name="noise"></code>

Supplementary Table 4: Component types for coupling and noise functionality for Python and CUDA code generation.

Component type "coupling"

The coupling component type (Supplementary Table 4) gives the user the flexibility to define arbitrary coupling functions for the automatic CUDA code generation. To identify this component type, the *name* field must include the term "coupling". For the Python variant, coupling functions need to be encoded individually for each population as *c_pop* variables, due to architectural constraints of the main TVB implementation. The constructs *Parameter*, *DerivedParameter* and *DerivedVariable* have a similar keyword naming scheme as in the "derivatives" component type. It is possible to specify multiple coupling functions for a set of populations. The *Parameter* constructs specify the *name* of variables used to compute coupling and the *dimension* field indicates the used state variable. In our example (Supplementary Table 4), the first defined state variable is selected, as indexing starts from zero; the corresponding line 4 of Supplementary Listing 2 shows how a shift of '0' encodes the selection of the first state variable. The *DerivedParameter* construct defines the *name* of the variable to store the coupling input and the *value* field contains a function that transforms the sums of all inputs to a specific node. This variable, which has the name *c_pop1*, can again be used as input to compute pre- and postsynaptic dynamics. Line 20 of Supplementary Listing 2 shows the variable and the expression applied to it. The expression holds *global_coup*, which is defined as a parameter and *local_coup* which is defined as a constant outside of the coupling construct. The *DerivedVariable* construct can be used to enter custom "pre"- and "post"-synaptic coupling functions. This pre- and postsynaptic coupling functions are then multiplied with each other and again multiplied by the weights of the node and finally assigned to *c_pop1*, which is shown on line 16 of Supplementary Listing 2. As is mentioned in the *derivatives* section, if the derivatives component type does not have a derived parameter named "rec_speed_dt", the coupling function will not have the delay aspect and thus does not have a temporal component. Line 12 of Supplementary Listing 2 shows the fetching of the delays and the usage of "rec_speed_dt". If "rec_speed_dt" is not defined "dij" will be set to 0. Other variables or constants defined inside or outside of this component type can also be used to populate this function.

```

1 // Loop over nodes
2 for (int i_node = 0; i_node < n_node; i_node++){
3     c_pop1 = 0.0f;
4     V = state((t) % nh, i_node + 0 * n_node);
5     unsigned int i_n = i_node * n_node;
6     // Loop over nodes
7     for (unsigned int j_node = 0; j_node < n_node; j_node++){
8         // Get the weight of the coupling between node i and node j
9         float wij = weights[i_n + j_node];
10        if (wij == 0.0) continue;
11        // Get the delay between node i and node j
12        unsigned int dij = lengths[i_n + j_node] * rec_speed_dt;
13        // Get the state of node j which is delayed by dij
14        float V_j = state(((t - dij + nh) % nh), j_node + 0 * n_node);
15        // Sum using coupling function (constant * weight * pre * post)
16        c_pop1 += wij * c_c_a * 2 * sin(V_j - V);
17    }
18 }
19 // Export c_pop1 outside loop and process
20 c_pop1 *= global_coupling;
21 // Do derivative calculation

```

Supplementary Listing 2: Generated CUDA code for coupling

Component type "noise"

To specify noise terms, a component type with the *name* "noise" can be defined, which is only applicable for the CUDA variant. The CUDA code generation makes use of the Curand library to add a random value to the calculated derivatives, which is comparable to Gaussian noise (`curand_normal`). As is mentioned in the **derivatives** section, if the derivatives component type has a derived parameter with the *name* "nsig" a noise amplification of *value* will be used to amplify the noise. Thus specifying:

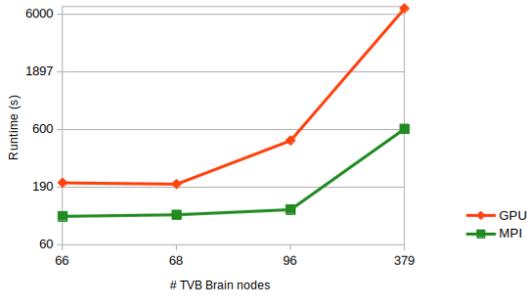
```
<ComponentType name="noise" />
```

will add this line to the model, in which *nsig* is the noise amplifier *V* is the current state and *dV* is the numerical solution step of the derivatives:

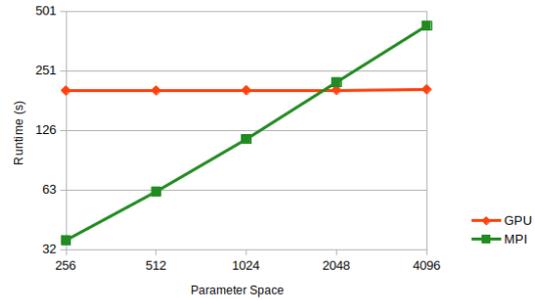
```
V += nsig * curand_normal(&crndst) + dV;
```

Benchmark Python and CUDA: ReducedWongWang parameter exploration

TVB-HPC makes efficient use of the large number of parallel cores in GPUs for high-dimensional parameter space exploration. Supplementary Figures 5 and 6 show performance results for Python and CUDA model code generated from a ReducedWongWang model (Deco et al., 2014) RateML specification. The results in Supplementary Figure 5 shows the runtimes for simulating 1024 parameter combinations with the RateML generated models for an increasing numbers of BNM nodes. For a 68-node BNM we explored parameter scaling behavior, shown in Supplementary Figure 6. These setups explore a range of parameters for the coupling and speed of the TVB connectivity. The Python models are distributed through MPI over 128 cores of an AMD EPYC 7742 with 2×64 cores running at 2.25 GHz. The GPU used is an NVIDIA V100 GPU with 16 GB HBM2e. MPI was used to disperse the workload on the cores, once the simulation started no information was communicated between the cores. Both setups were executed on JUSUF supercomputer located at the Jülich Supercomputing Centre. Occupancy analysis showed that when the thread block dimension is 32x32, a CUDA programming abstraction indicating thread count dispersion for the GPUs multiprocessors, the GPU performance is optimal with regards to execution runtime.



Supplementary Figure 5: Walltime results when scaling the number of TVB nodes for GPU and Python TVB models for 1024 parameter combinations and 30 s of simulated time.



Supplementary Figure 6: Walltime results when scaling the number of parameters for GPU and Python TVB models for a 68-nodes BNM and 30 s of simulated time

Fast_TV B

Fast_TV B was developed in C and makes extensive use of optimization techniques to increase the speed of brain network model simulations with the Dynamic Mean Field model (Deco et al., 2014). For example, it uses Single Instruction, Multiple Data instructions where the same operation is concurrently applied to multiple values contained in one large register. In the main integration loop no function calls are made to avoid possible overhead and instead all necessary code is directly written into the main integration loop. Inner loops are unrolled, jammed and bound to avoid "end of loop" tests and branch penalties. Intermediary results are re-used in other parts of the algorithm and not computed again. For example, some terms and expressions occur multiple times in the equations or they resolve to a constant and therefore only need to be computed once and not repeatedly in each time step. To reduce memory-related performance bottlenecks and cache failures, data to compute coupling inputs is organized in an efficient ring-buffer that stores related data in close locality. SC is stored in a sparse layout that scales linearly with the number of connections (and not quadratically as when stored as array), which enables to hold large brain network models efficiently in memory. The computation of coupling inputs also requires that the program steps through distant memory locations (in order to fetch time-delayed state variables), which is why pointer tables that directly link to the required addresses are used instead of array indexing tables, which saves one dereferencing operation for each memory access. Input and output operations happen only before or after the main simulation to avoid waiting time for devices. The Fast_TV B EBRAINS collab (Table 1) explains how to run the image container on supercomputers from a Jupyter notebook using *Fast_TV B* command line arguments. Having pulled the image (with e.g., `docker pull thevirtualbrain/fast_tv b`), the container can be run by invoking the command

```
docker run --rm --mount type=bind,source=/path/to/output,target=/output --
mount type=bind,source=/path/to/input,target=/input thevirtualbrain/fast_tv b
/start_simulation.sh <parameter_file> <subject_id> <num_threads>
```

and analogous commands for Docker alternatives like Singularity, Sarus or Shifter. The three command line arguments specify the following:

- `<parameter_file>` specifies the name of the parameter file,
- `<subject_id>` specifies the ID of the subject, used as prefix for SC filenames,
- `<num_threads>` sets the number of threads to be run during parallel execution (set to 1 to disable multithreading).

As input, *Fast_TVB* needs a parameter file and two SC files, one for connection weights and one for distances. All three files must be plaintext ASCII and stored in the directory `input/`. Matrix columns are separated by whitespaces, and rows by newline characters. The unit of distances is *mm*, whereas coupling weights are dimensionless. To write outputs, *Fast_TVB* expects a folder called `output/`. Input and output folders must exist on the host computer, they must be bind-mounted to a folder in the container as shown above, and the container client must have read and write access. SC files in the input folder must have the suffixes "`_SC_weights.txt`" for the coupling weights file, and "`_SC_distances.txt`" for the distances file:

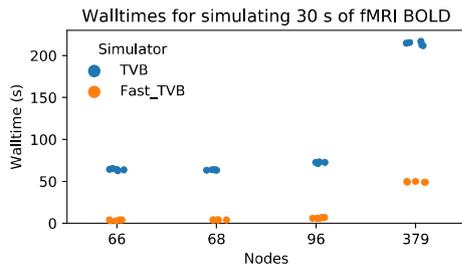
- connection weights file: `<sub_id>_SC_weights.txt`
- connection distances file: `<sub_id>_SC_distances.txt`

The subject identifier `<sub_id>` should be an arbitrary short alphanumeric string, e.g. "sub002", and must be provided as an argument to the program call. The parameter file contains the parameters as a list of numbers separated by whitespaces. The expected sequence of parameters, an explanation, and examples of the parameters are shown in Supplementary Table 5. Please note that for the parameter file correct formatting of floating point versus integer numbers is mandatory: integer numbers must be formatted without a radix point, while floating point numbers must always have a radix point (Supplementary Table 5). For a detailed description of parameters please consult Deco et al. (2014).

Parameter	Description	Number format	Example
nodes	number of nodes in brain network model	Integer	379
G	global coupling scaling factor	Float	1.0
J_NMDA	strength of excitatory (NMDA) synapses	Float	0.15
w_plus	strength of local excitatory recurrence	Float	1.4
Ji	strength of local inhibitory (GABA) synapses	Float	1.0
sigma	noise strength	Float	0.01
time_steps	length of the simulation (ms)	Integer	10000
BOLD_TR	TR of simulated fMRI BOLD signal (ms)	Integer	720
global_trans_v	transmission velocity of large-scale network (m/s)	Float	12.5
rand_num_seed	seed to initialize random number generator	Integer	42

Supplementary Table 5. Parameters of ReducedWongWang as implemented in *Fast_TVB*.

For efficient dense parameter space explorations, it is beneficial to use the code in a serial manner, i.e., without multithreading. Simulations are more efficient when only a single thread is created, due to absence of interthread communication overhead, but faster for multiple threads. Instead, multiple instances of the program can be spawned to compute several parameters sets in parallel. To create the Docker image from scratch please refer to code and instructions in the source code repository (Table 1). *Fast_TVB* was used in several publications for brain simulation (Aerts et al., 2020, 2018; Costa-Klein et al., 2020; Schirner et al., 2018; Shen et al., 2015; Zimmermann et al., 2018). Supplementary Figure 7 compares runtimes of single-threaded *Fast_TVB* with TVB version 1.5.8-8852 on a MacBook Pro (15-inch, 2017, 3.1 GHz 6-Core Intel Core i7, 16 GB 2133 MHz LPDDR3) for different network sizes.



Supplementary Figure 7. Walltimes for TVB and Fast_TV B for simulating 30 s of fMRI BOLD activity for different parcellation sizes.

Supplementary Methods: TVB atlas and data adapters

Atlas and data adapters are under development to connect TVB with the EBRAINS Brain Atlas, KnowledgeGraph and Human Intracerebral EEG services for data exchange via RESTful APIs. Atlas adapters will enable to localize and extract empirical data from multimodal brain atlases to inform BNM parameterization. For example, setting coupling parameters heterogeneously for each region according to its ratio of T1- to T2-weighted MRI, a proxy of Myelin content (Glasser and van Essen, 2011), leads to a better fit with empirical fMRI and it also predicts empirically observed hierarchical organization of spectral power (Demirtaş et al., 2019). Similarly, Stefanovski et al. (2019) used information from PET data to heterogeneously set local population model parameters in order to simulate patient-specific amyloid beta burdens in Alzheimer's, which produced a characteristic slowing of EEG. In addition to supplying inputs, adapters are also needed for analyzing simulation outputs, which often requires several additional software packages. For example, TVB has interfaces with Python, MATLAB, the Brain Connectivity Toolbox, or the Allen Brain Atlas (see TVB's documentation, Table 1).

The Human Brain Atlas is EBRAINS' central reference space for integrating different maps of structure, function and connectivity, based on the Julich-Brain Cytoarchitectonic Atlas (Amunts et al., 2020). Julich-Brain is a 3D atlas of the human brain based on histological staining of 20- μ m sections of 23 post-mortem brains, providing a highly detailed anatomical reference space that resolves large cell bodies in individual cortical layers. Upon digitization, a 3D reconstruction of the histological sections was computed for each individual and registered with the MNI-Colin27 template and the ICBM 2009c Nonlinear Asymmetric template. Due to this alignment with standard brain templates, the atlas features can be mapped to individual brain anatomy using nonlinear registration and spatial warping. A growing selection of multimodal data features will then be usable to inform large-scale BNMs, including transmitter receptor densities (Palomero-Gallagher and Zilles, 2019), cell distributions, and physiological recordings. Initial work on the atlas adapters, an overview of which can be inspected in a dedicated collab (wiki.ebrains.eu/bin/view/Collabs/atlas-tvb-mapping), is currently focusing on providing a full-brain coverage parcellation derived from the Julich-Brain probabilistic cytoarchitectonic maps as well as creating SC using this parcellation and diffusion tractography data from the 1000BRAINS study (Caspers and Schreiber, 2020). Upon completion, density measurements for 16 receptors will be provided for each brain region, and high-resolution tractography maps (full brain post-mortem 200 μ m isotropic diffusion MRI and 60 μ m isotropic 3D Polarised Light Imaging) will increase the reliability of SC (Axer et al., 2016; Beaujoin et al., 2018). Further characterizations of microstructural properties will be derived from HBP's BigBrain Atlas

(Amunts et al., 2013), which contains cell body stains at 20 μm isotropic resolution. In addition, post-mortem polarized light imaging with 1 μm in-plane resolution will provide characterizations of tangential connectivity throughout the cortical mantle. Lastly, it is also planned to link intracranial electrophysiology recordings of different cohorts with the respective Julich-Brain regions, which opens up opportunities to set model parameters based on functional data, for example, estimates of effective connectivity and transmission delays from stimulation experiments (Trebaut et al., 2018).

The idea of data adapters is to connect TVB with EBRAINS data repositories, with the goal that services can effectively and securely exchange data via KnowledgeGraph as central database and graph ontology, in order to compare simulation results with empirical data. Connecting different repositories with TVB comes with the need to account for the heterogeneity of data formats and data base interfaces. For example, TVB shall connect with the Human Intracerebral EEG Database and Analysis Platform, which is currently developed in EBRAINS. To use this type of data it is, for example, important that the geometry and location of sensors is taken into account, which requires interfaces between the involved reference spaces and data formats. To conveniently set up adapters for the large variety of applicable neuroscience repositories it is planned to develop a meta-language where such interfaces can be specified with XML or JSON syntax.

Supplementary Methods: Data integration and TVB-ready data

openMINDS metadata annotation

The KnowledgeGraph is the central service in EBRAINS to store, organize, share, and find data. To increase findability and re-usability, integration of data into the KnowledgeGraph requires that the data is properly documented, organized, and annotated with metadata that specifies key features of the data that allow their identification and interpretation, e.g., data set identifiers, full-text descriptions, licenses, embargos, owners, contributors, involved specimen, subject identifiers, methods, datatypes, measurements, activities, repositories, versions, etc. The KnowledgeGraph uses the openMINDS metadata models (Table 1), which are defined by lightweight JSON templates that are serialized in the KnowledgeGraph using JSON-LD. The schemas of the openMINDS metadata model serve as basic architectural building blocks underlying the KnowledgeGraph graph database and are linked with existing neuroscience ontologies and terminologies. The openMINDS core metadata-model is flexible enough to capture the intricacies of non-standardized research data while being strict enough to enable structured database querying. To support users in creating openMINDS metadata we provide a simple openMINDS JSON-LD writer that semi-automatically creates the required file and folder structure with respective contents (Table 1).

BIDS metadata annotation

For compliance with the BIDS standard (bids-specification.readthedocs.io), derivative data sets must contain the metadata JSON file *dataset_description.json*. This file must exist in the top-level folder of every BIDS data set and contains a description of the data including Name, Authors, License, Acknowledgments, Funding, DOI, etc. The TVB Processing Pipeline first propagates all key-value pairs from the *dataset_description.json* file of the input data, and then, following the 'BIDS Derivatives' standard, adds key-value pairs as shown in Supplementary Listing 3.

```
"PipelineDescription":{
```

```

    "Name": "TVB Image Processing Pipeline",
    "Version": "1.0",
    "CodeURL": [
      "https://github.com/BrainModes/tvb-pipeline-sc",
      "https://github.com/BrainModes/fmriprep",
      "https://github.com/BrainModes/tvb-pipeline-converter"
    ]
  },
  "SourceDatasetsMetadata": [
    {
      ...
    },
    ...
  ]
]

```

Supplementary Listing 3. Key-value pairs added to the dataset_description.json metadata file of the TVB Image Processing Pipeline output.

The keys belonging to "PipelineDescription" describe the pipeline and are filled with above-mentioned default values. The values belonging to the key "SourceDatasetsMetadata" collect all metadata schemas from the source data sets.

TVB-ready data

KnowledgeGraph provides exemplary TVB-ready connectivity data in BIDS format from tumor patients and matched control participants. The data set contains region-average fMRI time series, FC, and SC from 31 brain tumor patients before and after surgery, and 11 healthy controls from tumor patients and healthy controls. The data have been reported in previous studies (Aerts et al., 2020, 2018). Both studies were approved by the Ethics Committee at Ghent University Hospital. All participants received detailed study information and gave written informed consent before study enrolment. The dataset contains MRI derivatives from patients who were diagnosed with either a glioma, developing from glial cells, or a meningioma, developing in the meninges, as well as healthy controls. Patients were recruited from Ghent University Hospital (Belgium) between May 2015 and October 2017 on the day before each patient's tumor surgery. Patients were eligible if they (1) were at least 18 years old, (2) had a supratentorial meningioma (WHO grade I or II) or glioma (WHO grade II or III) brain tumor, (3) were able to complete neuropsychological testing, and (4) were medically approved to undergo MRI investigation. Partners were also asked to participate in the study to constitute a group of control subjects that suffer from emotional distress comparable to that of the patients. Data from 11 glioma patients (mean age 47.5 y, SD = 11.3; 4 females), 14 meningioma patients (mean age 60.4 y, SD = 12.3; 11 females), and 11 healthy partners (mean age 58.6 y, SD = 10.3; 4 females) was collected. From all participants, three types of MRI scans were obtained using a Siemens 3T Magnetom Trio MRI scanner: T1-MPRAGE anatomic images, resting-state functional echo-planar imaging data, a multishell high angular resolution diffusion-weighted MRI scan, and two DWI $b = 0$ s/mm² images were collected with reversed phase-encoding blips for the purpose of correcting susceptibility-induced distortions. Further information on the dataset, preprocessing and analysis can be found in the original journal articles using this dataset (Aerts et al., 2020, 2018) and in the dataset publication on EBRAINS KnowledgeGraph (Aerts et al., 2019).

Supplementary Methods: Advanced use cases and training

Bayesian Virtual Epileptic Patient

The Bayesian Virtual Epileptic Patient tutorials (Table 1) showcase how Bayesian inference is used to compute posterior probability distributions for parameters of TVB's Epileptor population model

(Hashemi et al., 2020; Jirsa et al., 2017, 2014; Proix et al., 2017). About one-third of the 70 million Epilepsy patients worldwide suffer from drug-resistant Epilepsy: patients experience uncontrolled seizures despite pharmacotherapy and in many cases brain surgery is the last resort (Kwan and Brodie, 2000; Tang et al., 2017). A major problem for Epilepsy surgery is that seizures can spread to brain regions without clear relationship to the onset zone, which is why modelling efforts are now considering large-scale brain network interaction effects. Mathematical models of seizure initiation and recruitment have advanced our understanding of the dynamical mechanisms underlying seizures. For instance, taxonomy analysis of seizure dynamics has shown that a system of five linked state variables, called 'the Epileptor', is sufficient to describe onset, time course and offset of ictal-like discharges and their recurrence (Jirsa et al., 2014). A detailed analysis over 2000 focal-onset seizures from multiple centers (Saggio et al., 2020) has shown that the Epileptor model is able to realistically simulate the most dominant dynamics of onset, progression and offset in seizure-like events. However, predicting seizure propagation is challenging because inversion of BNM equations suffers from the curse of dimensionality, uncertainty about parameter ranges (which often cannot be measured) and variability of observations. The BVEP uses Bayesian inference to compute posterior probability distributions for parameter values in order to study the spread of Epileptic seizures in patient-specific models. The approach works by applying Bayes' theorem on prior distributions obtained from observed data and model simulations while taking into account the likelihood for these observations. In this way the approach integrates empirical information as priors, such as a patient's SC, or lesions detected in MRI. The goal of the BVEP is to estimate the excitability parameter of the Epileptor population model for every brain region to yield a map of epileptogenicity for informing clinical decisions. The excitability parameter controls whether the Epileptor shows epileptogenic behavior or not and the estimated values are used to classify brain regions into three categories: epileptogenic zones (EZ), which can autonomously trigger seizures; propagation zones (PZ), which do not trigger seizures autonomously but may be recruited during the seizure evolution; and healthy zones (HZ), where no seizures occur. The target of this inference is to obtain the posterior distributions of the excitability parameter for every brain region, as well as initial conditions, global coupling, and noise. Priors for the excitability parameter are either uninformative (e.g., a uniform distribution over the entire range for EZ, PZ and HZ), or they can express clinical hypotheses (e.g., by centering a distribution over the range for one category), or they can be used for including empirical data (e.g., when the region was recorded to show seizure activity, the parameter value range for HZ can be excluded from the prior). Resulting model evidence can be used for comparison of different clinical hypotheses and to inform diagnosis and therapy. Simulation results are compared with empirical measurements (e.g., EEG from implanted electrodes) and model selection with cross-validation metrics are computed to evaluate clinical hypotheses and to assess the model's ability to predict new data. As exact posterior inference in such high-dimensional models is intractable, approximate methods are used, e.g., Markov Chain Monte Carlo (MCMC), which enables to generate correlated samples that converge to a (potentially complex) target distribution. Gradient-based MCMC algorithms like Hamilton Monte Carlo provide efficient convergence to high-dimensional target distributions, but their performance is highly sensitive to hyperparameters, which often need to be re-tuned to arrive at the desired target distribution, which is solved in BVEP by using No-U-Turn Samplers for adaptive self-tuning of hyperparameters (Hoffman and Gelman, 2014). Alternatively, Automatic Differentiation Variational Inference (ADVI) is used to approximate the posterior by first positing a family of densities and then finding a member of that family that is closest to the target distribution as measured by Kullback-Leibler divergence (Kucukelbir et al., 2017). BVEP is implemented in the open-source probabilistic programming language tools Stan (mc-stan.org) and PyMC3 (docs.pymc.io), which use automatic differentiation to compute gradients of specified model density functions for NUTS and ADVI. The workflow was tested with ground-truth synthetic data for two patients with symptomatic and asymptomatic seizures where it was possible to accurately and efficiently infer the spatial map of epileptogenicity

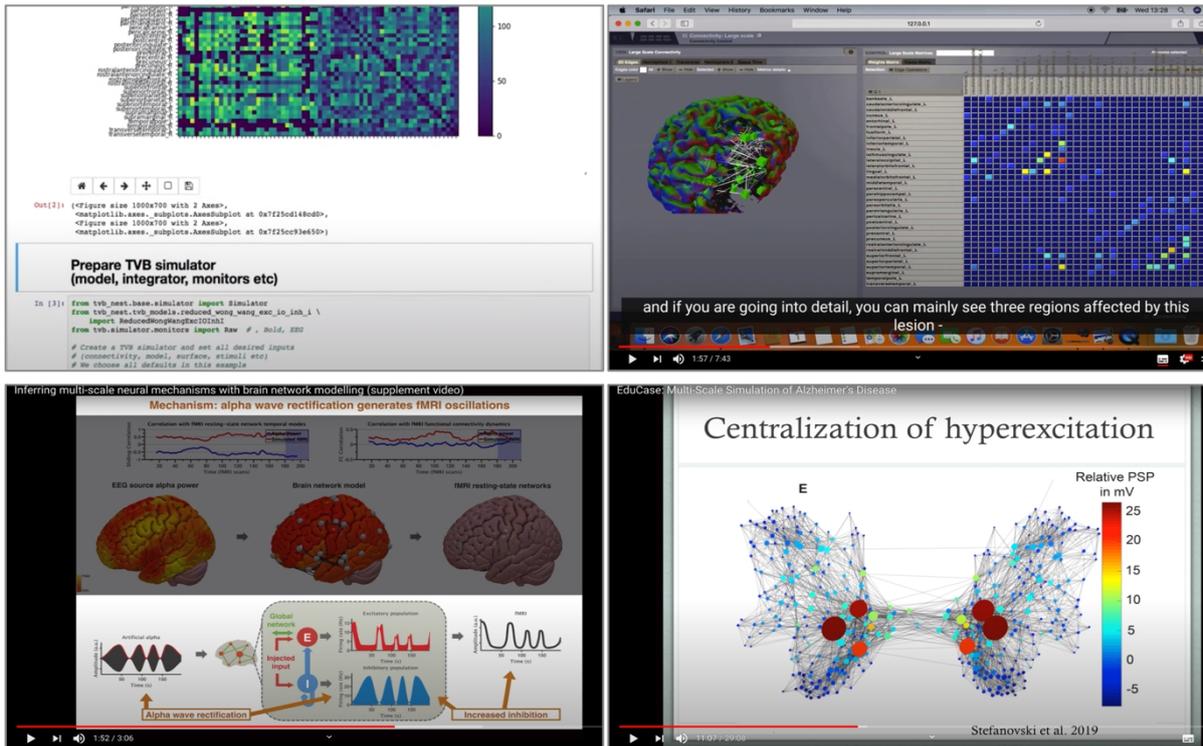
for all brain regions (Hashemi et al., 2020). The currently running EPINOV clinical trial (epinov.com) investigates informing clinical decisions with virtual patient studies to improve surgery outcome.

TVB Mouse Brains

The TVB Mouse Brain (Melozzi et al., 2019, 2017) tutorials (Table 1) show how tracer-based mouse SCs at different resolutions can be constructed from the Allen Mouse Brain Connectivity Atlas (Oh et al., 2014). The "Mouse Stroke" tutorials show how to introduce structural alterations related to stroke into SC and to associate these with corresponding changes in FC and population synchronization (Allegra Mascaro et al., 2020). To model stroke and subsequent rehabilitation, a pre-defined fraction of incoming connections is removed, e.g., 0 % corresponds to no stroke, while 100 % would correspond to a devastating stroke that destroyed all incoming connections to the region. Similarly, recovery from stroke is modelled by compensatory rewiring (Nudo, 2013) where the connection strengths of the non-damaged connections are increased relative to healthy connectivity. Different stroke connectomes are produced in this manner and the resulting FC is then fitted and compared with empirical data. By testing different re-wiring scenarios, the relative importance of compensatory rewiring in different connections for re-establishing healthy FC is studied. For these brain network models the Kuramoto model (Kuramoto, 1984) was used, which is a phenomenological model for emergent group dynamics of weakly coupled oscillators that we used previously to link SC with oscillatory dynamics in different modalities (Petkoski et al., 2018; Petkoski and Jirsa, 2019).

INCF training space

The INCF (International Neuroinformatics Coordination Facility) training space holds a dedicated collection for The Virtual Brain where didactic use cases are available with video tutorials, notebooks, and example data sets (Table 1, Supplementary Figure 8). INCF's TVB EduPack module gives a thorough introduction into work with TVB in general and EBRAINS cloud services in particular, helping the user to get quickly started with TVB and to reproduce several of the publications made with TVB. Tutorials consist of short lecture videos, scripting tutorials with Jupyter notebooks and code. Use cases demonstrate, for example, how to use TVB via the Collaboratory, run multi-scale co-simulations with NEST, process imaging data to construct personalized virtual brains of healthy individuals and patients, or analyze simulation results.



Supplementary Figure 8. In the INCF training space users find video tutorials, notebooks, and example data sets.

TVB Made Easy is a series of short lectures that introduce work where TVB was used in a clinical context. The training modules exemplify how TVB can help to predict recovery after stroke and how individual epileptic seizures are simulated. Brief lectures describe the methods and results, as well as how researchers can replicate the principal ideas of the articles directly with the TVB GUI and explore how artificial lesions alter brain dynamics, or how seizures spread through the brain network. An advanced application of TVB is the specification of disease models like it was done in the publication by Stefanovski et al (2019). In the INCF training space an extensive tutorial is provided that walks the user through the approach of the paper, in which the concentration of the protein amyloid beta, measured with positron emission tomography, was added as a prior for brain network model parameterization, leading to a better understanding of the hyperexcitability phenomenon in Alzheimer's disease.

Supplementary Methods: Data protection in the TVB on EBRAINS cloud

Biomedical research is currently facing challenges because many methods lack technical infrastructure to ensure the protection of personal data. Problematically, biomedical data cannot be easily anonymized or pseudonymized such that personally identifiable information are removed, and potential re-identification is excluded (Mostert et al., 2016). Person-related information is the explicit target in biomedical research and for personalized therapy. The information that characterizes the biomedical state of a person may also be used to (re-)identify a person. Precisely those features that create a deep understanding of health and illness of individuals are often highly unique. High-dimensional neuroimaging data, and especially data related to personalized brain simulation, like structural connectomes, functional connectomes, brain surfaces, MRI images, fMRI

time series, PET loadings, simulated brain activity, parameter estimation results, etc. are specifically acquired with the goal of deep phenotyping the data subjects. Typical data anonymization or pseudonymization strategies like removing metadata, releasing only subsets of data or even deleting facial features from MRI data are insufficient to protect the identity of the person, as it was shown that re-identification is almost perfect with as few as 40 out of 64,620 functional connections (Byrge and Kennedy, 2018) or 15 demographic attributes (Rocher et al., 2019). Currently, several large-scale initiatives, like the UK Biobank, are collecting thousands of features from millions of people. Health records of substantial parts of the world's population are digitized and exchanged between different computing systems. In the TVB on EBRAINS cloud personal data is exchanged over the internet between the user and the different cloud components. Moreover, the personal data is stored and processed on shared supercomputers. Transmitting personal data over non-secure channels like the internet as well as storing and processing personal data with shared resources poses the risk that the data is subject to unauthorized access. If privacy is breached, the individuals from which these data were collected face a number of potential harms. Here, we first provide a general outline of risks involved in cloud computing and highlight key aspects of the currently implemented security mechanisms. We also draft usage agreement terms that shall enable using TVB on EBRAINS cloud services to process personal data in compliance with General Data Protection Regulation (GDPR) as well as national laws. GDPR and similar national laws are legal frameworks that protect personal information by imposing restrictions to storing, sharing, and processing of personal data, which is here primarily neuroimaging data, derivatives thereof, neurosimulation results and associated metadata. In the following we describe cloud computing risks and protection measures that can help prevent unauthorized access and leakage of personal data.

Cloud computing risks

In the following we list the major classes of risks involved in cloud computing with respect to GDPR based on recommendations for data security published by the European Union Agency for Network and Information Security (Catteddu et al., 2012; Rijmen et al., 2013). *Logical leakage* includes all scenarios where data was exposed in transit. For example, an unauthorized copy of files was made because a communication channel was attacked. When data is transferred between different sites there is the possibility that the transfer can be intercepted. Sending data over open and insecure networks, like the Internet, poses security risks like code execution exploits, person-in-the-middle-attacks, DNS poisoning, redirect vulnerabilities, sniffing, spoofing, side channel and replay attacks. Data can be protected against this type of attack by state-of-the-art authenticated encryption, which is a form of encryption that simultaneously assures the confidentiality and authenticity of data. Authenticate-encrypted data can be considered protected if only the cipher text was exposed, but the decryption key remains secret. During *break-in on a live device* an attacker gains access to a device while it is working on sensitive data. The device is active during the attack, but the attacker has no physical access. For example, an attacker gets access to a device while a user is logged in. Or, a system has been hacked, e.g., by exploiting backdoors or application vulnerabilities. Or a virus with surveillance capabilities is active on the device. Since the cloud interface is remotely accessible and mediates access to larger sets of resources, similar attack scenarios like above ensue. In addition, isolation failure in shared environments can lead to situations where one tenant gets access to another tenant's resources or data, e.g., guest-hopping attacks, SQL injection attacks, or side-channel attacks. Furthermore, malicious insiders of the cloud provider, e.g., current, or former employees, can cause various kinds of damage, e.g., by implanting malicious code in cloud services.

Authenticated encryption can only protect against this type of attack if the data and the encrypting keys are not present in unencrypted form during the attack. *If data resides in unencrypted form in the RAM, it is not protected against this attack. Data can only be considered secure if one can guarantee that the data was not opened for the entire duration of the attack.* If cryptographic keys are present in RAM, then they are also to be considered as leaked, as well as all data that was encrypted using these keys. The amount of data leaked during this type of attack can be reduced by keeping as many keys as possible inside secure hardware and using a proper key architecture. In addition, firewalls and sandboxes may help to prevent leakage of data that is unencrypted in RAM by blocking network traffic and inter-process communication. This prevents that worms, trojan horses, surveillance viruses or other malicious code intercept sensible data while unencrypted in main memory. For that, firewalls and sandboxes should be configured such that all traffic and inter-process communication that is not explicitly permitted is blocked by default. In case of *full user impersonation* an attacker gets access to secret information that allows to impersonate a user. For example, a user loses passwords, access tokens or cryptographic keys. The amount of data leaked during this type of attack can be reduced by keeping as many keys as possible inside secure hardware and using a proper key architecture. Keys that have to reside in RAM during their usage shall only encrypt small amounts of data. Furthermore, it should not be possible to derive data encryption keys from access control keys or vice versa. Lastly, *hardware leakage* means that hardware containing sensitive data was brought into possession of an attacker. For example, portable computers, hard drives, USB sticks or any device that stores sensitive data were lost. This allows applying physical attacks that bypass security measures at the logical level. The problem persists even if the data was deleted, because simple deletion does not guarantee that the data is properly destroyed and ultimately removed from the storage or backup media. If data is not properly destroyed, the data could be recovered and accessed at a later time. Data is better secured against this type of data breach if it is protected by authenticated encryption and if all unencrypted copies of the data that have ever been on this hardware have been deleted securely, such that they cannot be recovered. The same protection must be applied to used cryptographic keys. Top-level cryptographic keys should not be stored on the device at all but should rather be input manually or read from a removable token. In addition to data files, also swap files and memory dumps (hibernation) are potential causes of data breach and such files must therefore also remain encrypted to protect from leakage.

Protective measures

In the following we describe how the TVB on EBRAINS image processing workflow is protected by the data security mechanisms access control, encryption, and sandboxing. EBRAINS access control uses passwords and cryptographic keys to prevent unauthorized access, to provide secure delegated access for connecting different cloud services, as well as to provide single sign-on. Keycloak is used for identity and access management, i.e., user registration, management, and permission control. The OpenID Connect protocol is used for authentication--confirming the identity of a user--based on OAuth 2.0 specifications for delegating and conveying authorization decisions. OAuth 2.0 is a widely adopted standard for access delegation and user authorization flows without the need for sharing credentials, which is realized by issuing cryptographic tokens that provide ongoing access to protected resources on behalf of a user. With this framework TVB services can limit the scope of computing resources accessible by a user: access to services is only provided through a GUI (thevirtualbrain.apps.hbp.eu) that only allows to execute a preconfigured set of programs and only

within a sandbox, respectively container, that does not provide direct access to the underlying supercomputing account. Encryption is a fundamental tool for data privacy, ensuring that data becomes unintelligible without decryption key. Therefore, the workflow is implemented such that sensitive data is encrypted at all times by default with the only exception when it is actively being used as input for a data processing routine. Only for the time window during which the data is actively being processed it may exist in unencrypted form, but then only within file system locations that are not accessible by other users. TVB on EBRAINS workflows (Figure 2) use public key cryptography in a way that the secret keys used for input data upload and results download are created ad-hoc at the beginning of the workflow and remain the entire time in a secure location at the site where the data will be finally used. That is, the decryption key for the uploaded input data is generated and held only on the supercomputer where the processing is being performed, with an automatic procedure, rendering the key inaccessible by any human user or system administrator. That is, the secret private key for decryption is only held in the main memory of a sandboxed process and never written out. Shortly before the upload starts the key pair is generated on the supercomputer, and only the public key is forwarded to the user's computer to encrypt the input data. Each uploaded data set gets its own dedicated key that is automatically created shortly before the encryption. After the encrypted data were uploaded to the supercomputer, they are decrypted into a sandboxed temporary file system. Likewise, during the entire processing intermediate results are only written into such a temporary filesystem that is invisible from the host. The outputs of the workflow are encrypted with a public key that was generated on the data controller's computer and all input and intermediate data of the workflow are deleted, the sandbox destroyed, and the encrypted results returned to the data controller. The decryption key for returning processing results is generated and exists only in an area that is under control by the user who initiated the processing (either within the EBRAINS cloud, or locally on the user's computer). Upon termination of the workflow, all keys and data are fully deleted. Data that is stored within a user's private space of the TVB web GUI stays encrypted while they are stored on EBRAINS servers and are only decrypted when the logged user opens the respective project in the project panel of the TVB GUI. When a decrypted project is not used anymore (i.e., there is no operation running on it, the project is not in the synchronization queue and the project is not used by another logged user) the decrypted data is deleted.

Shared responsibility & compliance

In addition to the technical measures described above, EBRAINS addresses data security through organizational measures and structures that plan, implement and review data protection and data sharing (ebrains.eu/service/share-data). Among these is a data governance working group (humanbrainproject.eu/en/social-ethical-reflective/about/data-governance/dwggw) that oversees and coordinates data-related policies across the infrastructure including access policies and their technical implementation. EBRAINS has a data protection officer who works to ensure that GDPR requirements are clear and that data protection-related obligations are understood and appropriately addressed across the organizations involved. These organizational measures help to review and update the data protection risks and the technical measures aimed to address them. TVB on EBRAINS data protection follows a "data protection by design and by default" strategy as mandated by GDPR Article 25. Data protection by design requires the implementation of state of the art technical and organizational measures, such as pseudonymization and encryption, and adherence to the data protection principles and the safeguarding of data subject rights at the earliest stages of

workflow design in a way that ensures privacy at all stages of the processing. Data protection by default implies that personal data is processed with the highest privacy protection so that by default personal data are not made accessible to an indefinite number of persons. Article 24 GDPR states that the data controller has the responsibility to implement appropriate technical and organizational measures to ensure and to be able to demonstrate that processing is performed in accordance with GDPR, which involves data protection by design and default (Article 25). Article 25 specifically requires data protection by default in a way that only the minimum amount of personal data is processed, the extent of the processing is minimal, the shortest period of storage is chosen, and the possibilities for accessing the personal data are minimized. Following these legal requirements technical and organizational measures were put in place to enable the data controller to exert continuing control over the personal data while in the EBRAINS cloud and to prevent access by others. A user can at any point stop the processing and remove the data from the EBRAINS cloud. Furthermore, the user is the only person that is in control over the data while in the cloud: security measures prevent access to another user's data if not explicitly shared. Accordingly, a user of TVB on EBRAINS services acts as a data controller under GDPR as it determines, alone or jointly with others, the purposes and means of processing as defined in Article 4(7) GDPR. In contrast, EBRAINS only acts as a data processor that processes data on behalf and under the authority of the data controller(s) (Articles 28 and 29, GDPR). As defined by Article 4(8) GDPR, a 'processor' is a natural or legal person, public authority or agency or other body which processes personal data on behalf of the controller. When processing is to be carried out on behalf of a controller, compliance with the GDPR, including the security of the personal data, should be enforced both by the controller and the processor, but it remains the responsibility of each controller to demonstrate compliance with the GDPR, particularly in relation to the necessary technical and organizational requirements and the existence of a lawful basis for processing (Article 5, GDPR). In order to use TVB cloud services, a data controller must therefore agree to terms of service that clarify its personal responsibility with respect to GDPR. These terms of service transparently clarify the nature of security precautions, contact persons, personal responsibilities of the user as data controller, monitoring, logging, and passing of information to third parties. Data controllers must confirm that they understand that active measures have been made to protect sensitive data, but nevertheless vulnerabilities may exist and a remaining risk for data protection incidents cannot be excluded as shared networks and computing systems are inherently risky. In addition to these measures the GDPR requires the controller to demonstrate a legal basis for processing as set out in Article 6(1)(a) for processing to be lawful. In the case of the processing of 'special categories' of personal data, such as data concerning health, there must also be a legal basis for processing as set out in Article 9(2) GDPR. It is often the case in scientific research that in line with Article 9(2)(a), data subjects give their explicit consent to the processing of their personal data for one or more specified purposes. Though, depending on the circumstances, another legal basis may also be appropriate such as Article 9(2)(j) on processing for the purposes of scientific research, in which case national law must be considered and the safeguards set out in Article 89(1) GDPR must be applied. In particular, the data controller is responsible that the data that is uploaded to the TVB on EBRAINS cloud must be strictly limited to those needed for the purpose of the specific processing operation (minimization principle, GDPR Article 5(1)(c)). Furthermore, data controllers are required to pseudonymize data as far as it does not compromise the research goal and to strip data sets from metadata that could be used for (re-)identification like names, birth dates, behavioral scores, etc. The Usage Agreement clarifies that the data is temporarily stored and processed on servers where different or additional areas of

jurisdiction may apply. In some countries, for example, cryptographic methods may not be used without approval. It is made clear at which locations (countries and regions) the data is stored, and that national and international laws therefore may apply to these data. Users acknowledge that, while extensive precautions have been taken, TVB on EBRAINS provides the services "as is" without claiming or guaranteeing correctness, accuracy, reliability, completeness, fitness, or usefulness for any purpose, reason, under any circumstance. Moreover, the user agrees that TVB on EBRAINS cloud providers, administrators or other users have no liability to any person or entity with respect to loss or damage caused directly or indirectly by the services provided by TVB on EBRAINS. Furthermore, the Usage Agreement guarantees no involvement of subcontractors that are unable to guarantee that personal data will be processed under the conditions indicated in the Usage Agreement.

Supplementary Discussion

TVB on EBRAINS is an ecosystem of cloud services for end-to-end brain simulation that were developed to lower the barriers to entry of complex brain simulation workflows, to enhance their reproducibility, and to enable legal compliance and data protection in supercomputing workflows. Tools for creating, simulating, and analyzing brain network models can be conveniently run from a web browser. The implementation allows flexible scaling of backend supercomputing resources to meet the high computational demands for the construction and simulation of brain network models. Several cloud services can also be downloaded and operated on local hardware as standalone software. Dependencies are conveniently packaged into platform-independent container images that can be pulled from Docker Hub and run using the Docker platform or more security-oriented alternatives like Singularity, Shifter and Sarus. All codes are open source and freely available for download from GitHub (Table 1). Most of the software and data components were peer-reviewed and results published in academic journals (see Table 2). The data management solution DataLad (datalad.org) was added to the TVB Image Processing Pipeline to explicitly track all the used data, software and execution steps, allowing to share and reproduce workflows and workflow results in an actionable manner. The different services and workflows are modular and can be flexibly combined to address different research questions. In order to support standardization efforts in the neuroimaging community and to increase interoperability, TVB adopted the BIDS format for inputs and outputs. MRI data in BIDS format can be directly used by the image processing pipeline and its outputs can be read by BIDS Apps (Supplementary Note: BIDS Apps). Comprehensive documentation in the form of manuals, tutorials, lectures, Jupyter notebooks, demo code, demo data, workshops, videos, use cases, mailing lists and support contacts provide efficient and didactic dissemination of knowledge and support. The EBRAINS high-level support team helps with questions about services (ebrains.eu/support). Continuous integration and automated testing is set up to monitor that cloud services are operational; technical and administrative staff provide support and repair broken components (e.g., due to broken dependencies).

The EBRAINS cloud is a community platform that supports collaborative and reproducible research in multiple ways. Storages maintain file version histories with the option for rollback. Change logs track the changes from version to version with time stamps and author names, which enables rigorous versioning and documentation of collaboratively created work. Individual contributions are transparently and openly recorded, which helps to resolve conflicts, to establish authorships, and responsibilities. Files can be securely shared via link to private storages that have explicit access and

permission control, instead of sending confidential data via insecure and intransparent channels. Multiple authors can concurrently work on the same document, instead of creating new document branches every time a document is shared with a co-author, which often creates inconsistencies in the different branches that are tedious to resolve and merge. In the spirit of a "release early, release often" philosophy team members can instantly see results as they are being created and can give critical feedback early on, which increases the speed of research and encourages solid documentation. Taken together, EBRAINS services enable to map and organize complex projects into a persistent and replicable structure at a central and secure place, which makes it easier to pick up projects at a later time. The flexibility of the platform and its focus on community-driven research enable rapid adoption of advances in brain simulation and connectomics, as well as correction of errors. Its modular design simplifies new developments as specialists from different subfields are not required to work into the details of other modules in order to contribute. An important feature of TVB on EBRAINS is that it makes high-performance computing resources available to researchers without supercomputing resources. Thereby, resource-intensive computations, e.g., dense parameter space explorations, become available to the broader community. Currently, TVB on EBRAINS provides 800,000 core hours per year to EBRAINS users, through the Swiss National Supercomputing Centre CSCS, as a part of the Fenix infrastructure (fenix-ri.eu), on a first come, first served basis, without need for individual application.

Privacy protection is an ethical and legal prerequisite to continue working with personal data. Technical and organizational measures are required not only to protect from malicious parties, but also to protect cloud users from making errors, e.g., forgetting to delete a data set in an exposed storage after a set period, unintended backups, accidentally sending it to the wrong recipient, or failure to set appropriate access permissions are just a few examples. TVB on EBRAINS developments are working to ensure highest standards for the protection of personal data, while at the same time providing flexibility for users to deploy custom software to enable state-of-the-art research. On the one hand, private data must be protected from unintended disclosure; on the other hand, researchers must be able to process such data with adjustable and novel procedures. Problematically, pseudonymization is often not possible, because the personal features required for personalized medicine are often highly unique and can be used to identify a person (Byrge and Kennedy, 2018; Gymrek et al., 2013; Rocher et al., 2019). We took an important step to meet these opposing requirements by setting up data processing and brain simulation workflows that enable secure processing of personal data in shared environments by independent or joint data controllers. Ongoing developments of TVB on EBRAINS involve the incorporation of automated testing frameworks for evaluating the plausibility of data and processing results, e.g. regarding the data quality of pipeline inputs, as well as the plausibility of intermediary and final outputs. Of interest are also automatized parameter optimization routines, and workflows that are able to process data from other species or patient data that involve morphological changes like stroke or tumor patients. Most importantly, to bring and maintain high-standard cloud services, efforts are directed towards the continuous integration of improved community standards and best practices.

The TVB on EBRAINS ecosystem is open-source and can be transferred to other Cloud environments within the European Open Science Cloud or beyond. Thus, it serves as a reference architecture for the processing and simulation of neuroscience data in the cloud (Figure 1). Lowering the barriers to use complex neuroimaging and brain simulation workflows within a scalable cloud architecture

fosters the processing of large cohort data sets within large international consortia, which is a prerequisite for the creation of generalizable results, efficient clinical translation, and effective personalized medicine.

Supplementary References

- Aerts, H., Schirner, M., Dhollander, T., Jeurissen, B., Achten, E., Van Roost, D., Ritter, P., Marinazzo, D., 2020. Modeling brain dynamics after tumor resection using The Virtual Brain. *Neuroimage*. <https://doi.org/10.1016/j.neuroimage.2020.116738>
- Aerts, H., Schirner, M., Jeurissen, B., Dhollander, T., Van Roost, D., Achten, E., Ritter, P., Marinazzo, D., 2019. TVB time series and connectomes for personalized brain modeling in brain tumor patients [Data set]. <https://doi.org/10.25493/1ECN-6SM>
- Aerts, H., Schirner, M., Jeurissen, B., Van Roost, D., Achten, E., Ritter, P., Marinazzo, D., 2018. Modeling brain dynamics in brain tumor patients using the virtual brain. *eNeuro*. <https://doi.org/10.1523/ENEURO.0083-18.2018>
- Amunts, K., Lepage, C., Borgeat, L., Mohlberg, H., Dickscheid, T., Rousseau, M.É., Bludau, S., Bazin, P.L., Lewis, L.B., Oros-Peusquens, A.M., Shah, N.J., Lippert, T., Zilles, K., Evans, A.C., 2013. BigBrain: An ultrahigh-resolution 3D human brain model. *Science* (80-.). <https://doi.org/10.1126/science.1235381>
- Amunts, K., Mohlberg, H., Bludau, S., Zilles, K., 2020. Julich-Brain: A 3D probabilistic atlas of the human brain's cytoarchitecture. *Science* (80-.).
- Axer, M., Strohmer, S., Gräßel, D., Bücker, O., Dohmen, M., Reckfort, J., Zilles, K., Amunts, K., 2016. Estimating fiber orientation distribution functions in 3D-Polarized Light Imaging. *Front. Neuroanat*. <https://doi.org/10.3389/fnana.2016.00040>
- Beaujoin, J., Palomero-Gallagher, N., Boumezbeur, F., Axer, M., Bernard, J., Poupon, F., Schmitz, D., Mangin, J.F., Poupon, C., 2018. Post-mortem inference of the human hippocampal connectivity and microstructure using ultra-high field diffusion MRI at 11.7 T. *Brain Struct. Funct.* <https://doi.org/10.1007/s00429-018-1617-1>
- Byrge, L., Kennedy, D.P., 2018. High-accuracy individual identification using a “thin slice” of the functional connectome. *Netw. Neurosci*. https://doi.org/10.1162/netn_a_00068
- Cannon, R.C., Gleeson, P., Crook, S., Ganapathy, G., Marin, B., Piasini, E., Silver, R.A., 2014. LEMS: A language for expressing complex biological models in concise and hierarchical form and its use in underpinning NeuroML 2. *Front. Neuroinform*. <https://doi.org/10.3389/fninf.2014.00079>
- Caspers, S., Schreiber, J., 2020. 1000BRAINS study, connectivity data. <https://doi.org/10.25493/61QA-KP8>
- Catteddu, D., Hogben, G., Haeberlen, T., Dupré, L., 2012. Cloud Computing. Benefits, risks and recommendations for information security.
- Costa-Klein, P., Ettinger, U., Schirner, M., Ritter, P., Falkai, P., Koutsouleris, N., Kambeitz, J., 2020. Investigating the Effect of the Neuregulin-1 Genotype on Brain Function Using Brain Network

Simulations. *Biol. Psychiatry*. <https://doi.org/10.1016/j.biopsych.2020.02.118>

- Deco, G., Jirsa, V.K., Robinson, P.A., Breakspear, M., Friston, K., 2008. The dynamic brain: from spiking neurons to neural masses and cortical fields. *PLoS Comput Biol* 4, e1000092.
- Deco, G., Ponce-Alvarez, A., Hagmann, P., Romani, G.L., Mantini, D., Corbetta, M., 2014. How Local Excitation–Inhibition Ratio Impacts the Whole Brain Dynamics. *J. Neurosci.* 34, 7886–7898.
- Demirtaş, M., Burt, J.B., Helmer, M., Ji, J.L., Adkinson, B.D., Glasser, M.F., Van Essen, D.C., Sotiropoulos, S.N., Anticevic, A., Murray, J.D., 2019. Hierarchical Heterogeneity across Human Cortex Shapes Large-Scale Neural Dynamics. *Neuron*. <https://doi.org/10.1016/j.neuron.2019.01.017>
- Esteban, O., Ciric, R., Finc, K., Blair, R.W., Markiewicz, C.J., Moodie, C.A., Kent, J.D., Goncalves, M., DuPre, E., Gomez, D.E.P., Ye, Z., Salo, T., Valabregue, R., Amlien, I.K., Liem, F., Jacoby, N., Stojić, H., Cieslak, M., Urchs, S., Halchenko, Y.O., Ghosh, S.S., De La Vega, A., Yarkoni, T., Wright, J., Thompson, W.H., Poldrack, R.A., Gorgolewski, K.J., 2020. Analysis of task-based functional MRI data preprocessed with fMRIPrep. *Nat. Protoc.* <https://doi.org/10.1038/s41596-020-0327-3>
- Esteban, O., Markiewicz, C.J., Blair, R.W., Moodie, C.A., Isik, A.I., Erramuzpe, A., Kent, J.D., Goncalves, M., DuPre, E., Snyder, M., Oya, H., Ghosh, S.S., Wright, J., Durnez, J., Poldrack, R.A., Gorgolewski, K.J., 2019. fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nat. Methods*. <https://doi.org/10.1038/s41592-018-0235-4>
- Friston, K.J., Holmes, A.P., Worsley, J.-P., Frith, C.D., Frackowiak, R.S., 1995. Statistical parametric maps in functional imaging: a general linear approach. *Hum. Brain Mapp.*
- Glasser, M.F., Sotiropoulos, S.N., Wilson, J.A., Coalson, T.S., Fischl, B., Andersson, J.L., Xu, J., Jbabdi, S., Webster, M., Polimeni, J.R., Van Essen, D.C., Jenkinson, M., 2013. The minimal preprocessing pipelines for the Human Connectome Project. *Neuroimage*. <https://doi.org/10.1016/j.neuroimage.2013.04.127>
- Glasser, M.F., van Essen, D.C., 2011. Mapping human cortical areas in vivo based on myelin content as revealed by T1- and T2-weighted MRI. *J. Neurosci.* <https://doi.org/10.1523/JNEUROSCI.2180-11.2011>
- Gorgolewski, K.J., Alfaro-Almagro, F., Auer, T., Bellec, P., Capotă, M., Chakravarty, M.M., Churchill, N.W., Cohen, A.L., Craddock, R.C., Devenyi, G.A., Eklund, A., Esteban, O., Flandin, G., Ghosh, S.S., Guntupalli, J.S., Jenkinson, M., Keshavan, A., Kiar, G., Liem, F., Raamana, P.R., Raffelt, D., Steele, C.J., Quirion, P.O., Smith, R., Strother, S.C., Varoquaux, G., Wang, Y., Yarkoni, T., Poldrack, R.A., 2017. BIDS apps: Improving ease of use, accessibility, and reproducibility of neuroimaging data analysis methods. *PLoS Comput. Biol.* <https://doi.org/10.1371/journal.pcbi.1005209>
- Gorgolewski, K.J., Auer, T., Calhoun, V.D., Craddock, R.C., Das, S., Duff, E.P., Flandin, G., Ghosh, S.S., Glatard, T., Halchenko, Y.O., Handwerker, D.A., Hanke, M., Keator, D., Li, X., Michael, Z., Maumet, C., Nichols, B.N., Nichols, T.E., Pellman, J., Poline, J.B., Rokem, A., Schaefer, G., Sochat, V., Triplett, W., Turner, J.A., Varoquaux, G., Poldrack, R.A., 2016. The brain imaging data structure, a format for organizing and describing outputs of neuroimaging experiments. *Sci. Data*. <https://doi.org/10.1038/sdata.2016.44>
- Gramfort, A., Luessi, M., Larson, E., Engemann, D.A., Strohmeier, D., Brodbeck, C., Goj, R., Jas, M., Brooks, T., Parkkonen, L., Hämäläinen, M., 2013. MEG and EEG data analysis with MNE-Python. *Front. Neurosci.* <https://doi.org/10.3389/fnins.2013.00267>

- Gymrek, M., McGuire, A.L., Golan, D., Halperin, E., Erlich, Y., 2013. Identifying personal genomes by surname inference. *Science* (80-.). <https://doi.org/10.1126/science.1229566>
- Hallez, H., Vanrumste, B., Grech, R., Muscat, J., De Clercq, W., Vergult, A., D'Asseler, Y., Camilleri, K.P., Fabri, S.G., Van Huffel, S., Lemahieu, I., 2007. Review on solving the forward problem in EEG source analysis. *J. Neuroeng. Rehabil.* <https://doi.org/10.1186/1743-0003-4-46>
- Hashemi, M., Vattikonda, A.N., Sip, V., Guye, M., Bartolomei, F., Woodman, M.M., Jirsa, V.K., 2020. The Bayesian Virtual Epileptic Patient: A probabilistic framework designed to infer the spatial map of epileptogenicity in a personalized large-scale brain model of epilepsy spread. *Neuroimage*. <https://doi.org/10.1016/j.neuroimage.2020.116839>
- Hoffman, M.D., Gelman, A., 2014. The no-U-turn sampler: Adaptively setting path lengths in Hamiltonian Monte Carlo. *J. Mach. Learn. Res.*
- Jirsa, V.K., Proix, T., Perdakis, D., Woodman, M.M., Wang, H., Bernard, C., Bénar, C., Chauvel, P., Bartolomei, F., Guye, M., Gonzalez-Martinez, J., 2017. The Virtual Epileptic Patient: Individualized whole-brain models of epilepsy spread. *Neuroimage*. <https://doi.org/10.1016/j.neuroimage.2016.04.049>
- Jirsa, V.K., Stacey, W.C., Quilichini, P.P., Ivanov, A.I., Bernard, C., 2014. On the nature of seizure dynamics. *Brain* 137, 2210–2230. <https://doi.org/10.1093/brain/awu133>
- Kellner, E., Dhital, B., Kiselev, V.G., Reiser, M., 2016. Gibbs-ringing artifact removal based on local subvoxel-shifts. *Magn. Reson. Med.* <https://doi.org/10.1002/mrm.26054>
- Kucukelbir, A., Blei, D.M., Gelman, A., Ranganath, R., Tran, D., 2017. Automatic Differentiation Variational Inference. *J. Mach. Learn. Res.*
- Kuramoto, Y., 1984. *Chemical Oscillations, Waves, and Turbulence*, Springer Series in Synergetics. Springer Berlin Heidelberg, Heidelberg. <https://doi.org/10.1007/978-3-642-69689-3>
- Kwan, P., Brodie, M.J., 2000. Early identification of refractory epilepsy. *N. Engl. J. Med.* <https://doi.org/10.1056/NEJM200002033420503>
- Lam, S.K., Pitrou, A., Seibert, S., 2015. Numba: a LLVM-based Python JIT compiler. *Proc. Second Work. LLVM Compil. Infrastruct. HPC - LLVM '15.* <https://doi.org/10.1145/2833157.2833162>
- Liu, T.T., Nalci, A., Falahpour, M., 2017. The global signal in fMRI: Nuisance or Information? *Neuroimage*. <https://doi.org/10.1016/j.neuroimage.2017.02.036>
- Mosher, J.C., Leahy, R.M., Lewis, P.S., 1999. EEG and MEG: Forward solutions for inverse methods. *IEEE Trans. Biomed. Eng.* <https://doi.org/10.1109/10.748978>
- Mostert, M., Bredenoord, A.L., Biesart, M.C.I.H., Van Delden, J.J.M., 2016. Big Data in medical research and EU data protection law: Challenges to the consent or anonymise approach. *Eur. J. Hum. Genet.* <https://doi.org/10.1038/ejhg.2015.239>
- Murphy, K., Fox, M.D., 2017. Towards a consensus regarding global signal regression for resting state functional connectivity MRI. *Neuroimage*. <https://doi.org/10.1016/j.neuroimage.2016.11.052>
- Nudo, R.J., 2013. Recovery after brain injury: mechanisms and principles. *Front. Hum. Neurosci.* 7, 1–14. <https://doi.org/10.3389/fnhum.2013.00887>
- Palomero-Gallagher, N., Zilles, K., 2019. Cortical layers: Cyto-, myelo-, receptor- and synaptic

- architecture in human cortical areas. *Neuroimage*.
<https://doi.org/10.1016/j.neuroimage.2017.08.035>
- Petkoski, S., Jirsa, V.K., 2019. Transmission time delays organize the brain network synchronization. *Philos. Trans. R. Soc. A Math. Phys. Eng. Sci.* 377, 20180132.
<https://doi.org/10.1098/rsta.2018.0132>
- Petkoski, S., Palva, J.M., Jirsa, V.K., 2018. Phase-lags in large scale brain synchronization: Methodological considerations and in-silico analysis. *PLoS Comput. Biol.* 14, 1–30.
<https://doi.org/10.1371/journal.pcbi.1006160>
- Proix, T., Bartolomei, F., Guye, M., Jirsa, V.K., 2017. Individual brain structure and modelling predict seizure propagation. *Brain*. <https://doi.org/10.1093/brain/awx004>
- Pruim, R.H.R., Mennes, M., van Rooij, D., Llera, A., Buitelaar, J.K., Beckmann, C.F., 2015. ICA-AROMA: A robust ICA-based strategy for removing motion artifacts from fMRI data. *Neuroimage*.
<https://doi.org/10.1016/j.neuroimage.2015.02.064>
- Rijmen, V., De Cock, D., Smart, N.P., 2013. Recommended cryptographic measures – Securing personal data.
- Ritter, P., Schirner, M., McIntosh, A.R., Jirsa, V.K., 2013. The virtual brain integrates computational modeling and multimodal neuroimaging. *Brain Connect.* 3, 121–145.
<https://doi.org/10.1089/brain.2012.0120>
- Rocher, L., Hendrickx, J.M., de Montjoye, Y.A., 2019. Estimating the success of re-identifications in incomplete datasets using generative models. *Nat. Commun.* <https://doi.org/10.1038/s41467-019-10933-3>
- Saggio, M.L., Crisp, D., Scott, J.M., Karoly, P., Kuhlmann, L., Nakatani, M., Murai, T., Dümpelmann, M., Schulze-Bonhage, A., Ikeda, A., Cook, M., Gliske, S. V., Lin, J., Bernard, C., Jirsa, V.K., Stacey, W.C., 2020. A taxonomy of seizure dynamotypes. *Elife* 9, 1–56.
<https://doi.org/10.7554/eLife.55632>
- Sanz-Leon, P., Knock, S.A., Spiegler, A., Jirsa, V.K., 2015. Mathematical framework for large-scale brain network modeling in The Virtual Brain. *Neuroimage* 111, 385–430.
- Schirner, M., McIntosh, A.R., Jirsa, V.K., Deco, G., Ritter, P., 2018. Inferring multi-scale neural mechanisms with brain network modelling. *Elife*. <https://doi.org/10.7554/eLife.28927>
- Shen, K., Hutchison, R.M., Bezgin, G., Everling, S., McIntosh, A.R., 2015. Network structure shapes spontaneous functional connectivity dynamics. *J. Neurosci.* 35, 5579–5588.
- Smith, R., Connelly, A., 2019. MRtrix3_connectome: A BIDS Application for quantitative structural connectome construction., in: *OHBM*. p. W610.
- Smith, R., Tournier, J.D., Calamante, F., Connelly, A., 2015. SIFT2: Enabling dense quantitative assessment of brain white matter connectivity using streamlines tractography. *Neuroimage*.
<https://doi.org/10.1016/j.neuroimage.2015.06.092>
- Smith, R., Tournier, J.D., Calamante, F., Connelly, A., 2012. Anatomically-constrained tractography: Improved diffusion MRI streamlines tractography through effective use of anatomical information. *Neuroimage*. <https://doi.org/10.1016/j.neuroimage.2012.06.005>
- Stefanovski, L., Triebkorn, P., Spiegler, A., Diaz-Cortes, M.A., Solodkin, A., Jirsa, V.K., McIntosh, A.R.,

- Ritter, P., 2019. Linking Molecular Pathways and Large-Scale Computational Modeling to Assess Candidate Disease Mechanisms and Pharmacodynamics in Alzheimer's Disease. *Front. Comput. Neurosci.* <https://doi.org/10.3389/fncom.2019.00054>
- Tang, F., Hartz, A.M.S., Bauer, B., 2017. Drug-resistant epilepsy: Multiple hypotheses, few answers. *Front. Neurol.* <https://doi.org/10.3389/fneur.2017.00301>
- Tournier, J.D., Smith, R., Raffelt, D., Tabbara, R., Dhollander, T., Pietsch, M., Christiaens, D., Jeurissen, B., Yeh, C.H., Connelly, A., 2019. MRtrix3: A fast, flexible and open software framework for medical image processing and visualisation. *Neuroimage.* <https://doi.org/10.1016/j.neuroimage.2019.116137>
- Trebaul, L., Deman, P., Tuyisenge, V., Jedynak, M., Hugues, E., Rudrauf, D., Bhattacharjee, M., Tadel, F., Chanteloup-Foret, B., Saubat, C., Reyes Mejia, G.C., Adam, C., Nica, A., Pail, M., Dubeau, F., Rheims, S., Trébuchon, A., Wang, H., Liu, S., Blauwblomme, T., Garcés, M., De Palma, L., Valentin, A., Metsähonkala, E.L., Petrescu, A.M., Landré, E., Szurhaj, W., Hirsch, E., Valton, L., Rocamora, R., Schulze-Bonhage, A., Mindruta, I., Francione, S., Maillard, L., Taussig, D., Kahane, P., David, O., 2018. Probabilistic functional tractography of the human cortex revisited. *Neuroimage.* <https://doi.org/10.1016/j.neuroimage.2018.07.039>
- Vella, M., Cannon, R.C., Crook, S., Davison, A.P., Ganapathy, G., Robinson, H.P.C., Angus Silver, R., Gleeson, P., 2014. libNeuroML and PyLEMS: Using Python to combine procedural and declarative modeling approaches in computational neuroscience. *Front. Neuroinform.* <https://doi.org/10.3389/fninf.2014.00038>
- Veraart, J., Novikov, D.S., Christiaens, D., Ades-aron, B., Sijbers, J., Fieremans, E., 2016. Denoising of diffusion MRI using random matrix theory. *Neuroimage.* <https://doi.org/10.1016/j.neuroimage.2016.08.016>
- Zimmermann, J., Perry, A., Breakspear, M., Schirner, M., Sachdev, P., Wen, W., Kochan, N.A., Mapstone, M., Ritter, P., McIntosh, A.R., Solodkin, A., 2018. Differentiation of Alzheimer's disease based on local and global parameters in personalized Virtual Brain models. *NeuroImage Clin.* <https://doi.org/10.1016/j.nicl.2018.04.017>