Development of AI methods for deep learning of the MRI signature of the cortex cytoarchitecture with diffusion MRI

Anas Bachiri

université
PARIS-SACLAY

Development of AI methods for deep learning of the MRI signature of the cortex cytoarchitecture with diffusion MRI

Développement de méthodes d'IA pour l'apprentissage profond de la signature en IRM de diffusion de la cytoarchitecture du cortex cérébral

Thèse de doctorat de l'université Paris-Saclay

École doctorale n°575 : Electrical, Optical, Biophysics and Engineering (EOBE)
Spécialité de doctorat : Physique et imagerie médicale
Graduate School : Science de l'ingénierie et des systèmes
Référent : Faculté des sciences d'Orsay

Thèse préparée au sein de l'unité de recherche BAOBAB (Université Paris-Saclay, CEA, CNRS), sous la direction de **Cyril POUPON**, directeur de recherche, et le co-encadrement de **Ivy USZYNSKI**, ingénieure chercheur.

Thèse soutenue à Paris-Saclay, le 20 Septembre 2024, par

Anas BACHIRI

Composition du jury

Membres du jury avec voix délibérative

Tim Dyrby Professeur, Technical University of Denmark	Président et Rapporteur
Christophe Destrieux Professeur, Université de Tours	Rapporteur et Examinateur
Carole Frindel Professeur associé, INSA-Lyon	Examinatrice
Frédéric Andersson Ingénieur chercheur, INSERM Tours	Examinateur

THESE DE DOCTORAT
NNT : 2024UPAST105

université PARIS-SACLAY
ÉCOLE DOCTORALE
Physique et ingénierie : Electrons, Photons, Sciences du vivant (EOBE)

Titre : Développement de méthodes d'IA pour l'apprentissage profond de la signature en IRM de diffusion de la cytoarchitecture du cortex cérébral
Mots clés : IRM de diffusion, Simulations de Monte Carlo, Fantômes numériques, microstructure de la matière grise, Calcul de haute performance

Résumé : L'IRM de diffusion est un outil précieux pour étudier la microstructure du cerveau de manière non invasive. Comme l'IRM de diffusion est sensible au processus de diffusion, il permet d'obtenir indirectement des informations sur le milieu de diffusion, offrant ainsi une approche intéressante pour caractériser les tissus cérébraux. Au cours des deux dernières décennies, un important corpus de recherche au sein de la communauté de l'IRM de diffusion a visé à décoder la microstructure : estimer les caractéristiques liées aux cellules, telles que le diamètre moyen et la densité des cellules, à partir des signaux de l'IRM de diffusion. Les méthodes scientifiques proposées peuvent être classées en deux catégories : les modèles analytiques et les modèles computationnels. La première approche repose sur des modèles analytiques de diffusion restreinte dans des géométries simples pour estimer les paramètres de la microstructure. Malgré leur simplicité, les modèles analytiques reposent sur des hypothèses trop simplistes concernant la microstructure du cerveau, comme la modélisation des axones en tant que cylindres parfaits ou la forme du soma en tant que sphères. En revanche, l'approche computationnelle offre une alternative plus flexible qui permet de modéliser des microstructures complexes au-delà des simples cylindres et sphères. Toutefois, cette flexibilité s'accompagne d'un coût de calcul élevé. Ce travail s'inscrit dans l'approche computationnelle et vise à améliorer les méthodes de génération de microstructures synthétiques de matière grise et à aborder la question de l'extensibilité de l'approche computationnelle. Ce travail s'appuie sur des neurones réalistes reconstruits par microscopie électronique et sur de nouvelles représentations géométriques pour générer des échantillons de microstructures virtuelles de matière grise réalistes. Un nouveau schéma de simulation du processus d'IRM de diffusion est présenté, suivi d'une étude se basant sur la réalisation d'une campagne de simulations à grande échelle.

Title : Development of AI methods for deep learning of the MRI signature of the cortex cytoarchitecture with diffusion MRI
Keywords : Diffusion MRI, Monte Carlo Simulations, Numerical phantoms, The gray matter microstructure, High-performance computing

Abstract : Diffusion MRI provides an invaluable tool to study the brain's microstructure non-invasively. Since diffusion MRI is sensitized to the diffusion process, it allows to indirectly gain information about the diffusion medium, therefore offering an interesting approach to characterizing brain tissues. In the last two decades, an important body of research within the diffusion MRI community aimed at decoding microstructure : estimating cell-related features such as average cell diameters and density from diffusion MRI signals. The proposed scientific methods can be broadly separated into two classes : analytical and computational models. The first class of methods relies on analytical models of restricted diffusion in simple geometries to estimate microstructural parameters. Despite their simplicity, analytical models rely on oversimplistic hypotheses about brain microstructure, such modeling axons as perfect cylinders, or soma shape as spheres. In contrast, the computational approach offers a more flexible alternative that allows to model complex microstructures beyond simple cylinders and spheres. However, this flexibility comes at an expensive computational cost. This work fall within the computational approach and aims at improving on gray matter synthetic microstructure generation methods and addressing the scalability of the computational approach. This work leverages realistic neurons reconstructed with electron microscopy and novel geometric representations to generate realistic gray matter virtual microstructure substrates. It proposes a new scheme to efficiently simulate the dMRI process and perform large-scale simulations.

NUMERICS

This project has received funding from the European Union's Horizon 2020 research and innovation program under grant agreement No 800945 -- NUMERICS -- H2020-MSCA-COFUND-2017

Acknowledgements

I would like to express my deepest gratitude to my supervisor, Cyril Poupon, whose unwavering support, insightful guidance, and encouragement have been instrumental in the completion of this PhD work. Your dedication to supervision, your patience in explaining complex concepts, and your expertise in assisting with various aspects of this research have been invaluable. Your mastery of code design and building for the long term are things among many that I truly admire. Your insistence on rigorous scientific methods and your support in pursuing curiosity-driven ideas have profoundly shaped my approach to research and significantly contributed to the quality of this work.

I am also indebted to my co-supervisor, Ivy Uszynski, for your continuous support during this thesis. Your insights and guidance throughout this thesis were invaluable to the developments made. I acknowledge and admire your sense of urgency in providing actionable insights to your students and for generously allocating time to provide instant technical and scientific support. I would also like to thank you for providing important feedback throughout my thesis and for helping to shape my scientific communication.

I would like to thank Prof. Tim Dyrby and Prof. Christophe Destrieux for accepting to be reviewers for my Ph.D thesis. I also thank Prof. Caroline Frindel and Dr. Frédéric Andersson for accepting to be examinators of my thesis.

I am deeply appreciative of my colleagues, whose support and camaraderie have made this journey more enjoyable and fulfilling. It has been a pleasure working with a team with diverse backgrounds working on different projects. I want to thank Alexis, with whom I collaborated and had interesting scientific discussions. Simon for his constant support. Bastien and Mykyta for sharing their detailed knowledge of the human brain's anatomy, and Anais for sharing her knowledge of MRI sequences and her support. Also, big thanks to Raissa, Alexandros, Maelig, Elea, Bosco, Fadwa, Jean-Charles, and Aurelien. Your willingness to engage in stimulating discussions, share ideas, and provide feedback has been invaluable. The collaborative spirit and intellectual exchange within our group have been a constant source of inspiration and have significantly contributed to the development of this research. I have also had the privilege of participating in the weekly AI reading sessions which allowed me to keep in touch with the latest research and the new ideas proposed. For this, I am grateful to Benoit, Corentin, Clément, Pierre, and all the people who have participated in maintaining this initiative.

I would like to thank the Numerics PhD program which helped to fund my thesis. I am also immensely grateful for the opportunities that facilitated attending international conferences and workshops. These experiences have not only broadened my knowledge but also allowed me to connect with other experts in the field, enhancing the depth and scope of my research. Furthermore, I would like to extend my thanks to the AIDAS initiative. Your organization of scientific meetings and provision of funding for young researchers have been pivotal in advancing my research.

In addition, I would like to extend my heartfelt gratitude to my family, who provided continuous support during all these years. I would like to thank you for your understanding throughout this challenging journey. Your encouragement, patience, and love have been my anchor during difficult times, providing me with the strength and resilience needed to persevere. Your belief in me and constant support have been invaluable, and I am deeply thankful for your presence in my life.

During this thesis, I also had the opportunity to participate in the EzyMRI NerdFest, a hardware hackathon to build a low-field MRI system. I would like to thank Huang Shaoying for organizing this hackathon which played an important role in learning more about the hardware aspects of the MRI. This experience also taught me about the progress and efforts in making MRI hardware more accessible.

Finally, to everyone who has contributed to this journey, directly or indirectly, your support and encouragement have been a source of motivation and inspiration. Thank you.

Gif-sur-Yvette, 2024

Anas BACHIRI

Contents

Acronyms

dMRI Diffusion Magnetic Resonance Imaging
CSN Central Nervous System
PNS Peripheral Nervous System
WM White Matter
GM Gray Matter
CSF CerebroSpinal Fluid
MEDUSA Microstructure Environment Designer with Unified Sphere Atoms
CONFIG Contextual Fiber Growth
FEM Finite Element Method
CHARMED Composied Hindered and Restricted Models of Diffusion
SANDI Soma And Neurite Density Imaging
NODDI Neurite Orientation Dispersion and Density Imaging
FOV Field Of View
VOI Volume Of Interest
MC Monte Carlo
FID Free Induction Decay
RF Radio Frequency
DMS Diffusion Microscopist Simulator
RMS Realistic Microstructure Simulator
DSI Diffusion Spectrum Imaging
DTI Diffusion Tensor Imaging
DKI Diffusion Kurtosis Imaging
ADC Apparent Diffusion Coefficient
ODF Orientation Distribution Function
PDF Probability Distribution Function
PGSE Pulsed Gradient Spin Echo
OGSE Oscillating Gradient Spin Echo
CPU Central Processing Unit
GPU Graphical Processing Unit

Résumé

L'imagerie par résonnance magnétique (IRM) est un outil puissant qui permet d'investiguer le cerveau non-invasivement. Cette caractéristique de l'IRM rend son utilisation très attractive au sein de la communauté médicale et scientifique et ouvre la possibilité d'approfondire la compréhension du cerveau humain et d'étudier les maladies qui l'impactent. En particulier, La modalité de l'IRM de diffusion (IRMd) est d'intérêt croissant. Cette dernière permet d'obtenir des signaux à partir du phénomène de diffusion des molécules d'eau dans le cerveau. Comme la diffusion se fait dans des tissus cérébraux, elle est contrainte aux géométries complexes des cellules nerveuses à l'échelle micrométrique. Ansi, le signal de l'IRM de diffusion encode indirectement des informations sur ces contraintes géométriques complexes issues du milieu cellulaire du cerveau connu sous le nom de la microstructure cérébrale. Cette sensibilité au milieu de diffusion rend l'IRMd indispensable pour la neuroimagerie, voir même une pratique courante pour le diagnostique d'incidents comme l'accident vasculaire cérébral ischémique qui se caractérise par l'interruption de la circulation sanguine dans une artère cérébrale.
Cette thèse s'inscrit dans un continuum visant à estimer des paramètres caractéristiques de la microstructure du cerveau humain à partir des signaux d'IRM de diffusion. Cette problématique, connu sous le nom du décodage de la microstructure, a fait l'objet d'une enquête approfondie durant ces quinze dernières années par la communauté de l'IRM de diffusion. Les études établies précédemment ont permis le développement de plusieurs méthodes visant à décoder la microstructure cérébrale. Ces méthodes s'inscrivent globalement dans deux types d'approches: une approche analytique et une computationnelle. La première approche repose sur le développement de formules analytiques du signal d'IRMd en fonction de paramètres microstructuraux comme le diamètre moyen des axones ou la diffusivité du milieu intracellulaire. En revanche, la deuxième approche dite computationnelle repose sur des méthodes numériques pour étudier la microstructure. Cette approche utilise des méthodes de simulations et d'apprentissage profond pour estimer les paramètres d'intérêts liés à la microstructure cérébrale. Bien que le premier type d'approche analytique permet d'établir des formules analytiques explicites, ces méthodes reposent sur des hypothèses simplistes comme la sphéricité des corps cellulaires ou des formes cylindriques pures pour les axones. Par conséquent, ces hypothèses ne prennent pas en compte la complexité de la microstructure impactant le signal d'IRMd. En outre, certaines hypothèses des modèles analytiques s'appliquent dans certains régimes de diffusion prédéfinis ne permettant pas de capturer la totalité de l'information provenant de régimes complémentaires. Tenant compte de ces éléments, l'approche computationnelle offre une alternative plus flexible. Les méthodes développées sous le volet computationnel visent à synthétiser des échantillons numériques de la microstructure cérébrale et d'obtenir des signaux d'IRMd qui leurs correspondent. Malgré la fléxibilité de cette approche, cette denière est loin d'être aboutie. En effet, plusieurs défis restent à soulever quand

il s'agit d'adopter l'approche computationnelle:

- **Scalabilité des méthodes computationnelles.** L'approche computationnelle repose sur des méthodes numériques complexes demandant de massives ressources de calcul. D'une part la génération de jumeaux numériques de la microstructure cérébrale et d'autre part l'extraction du signal d'IRMd à partir des échantillons synthétisés. L'important coût de calcul des méthodes computationnelles précédemment développés empêche l'adoption massive de ce type d'approche. Plus concrètement, la taille de mémoire des jumeaux numériques de la microstructure, le temps nécessaire pour la génération un échantillon, ainsi que le temps d'extraction des signaux d'IRMd. Ces trois points forment un obstacle important entravant l'exploitation complète des avantages de l'approche computationelle.

- **Ecart entre la simulation et la réalité.** Un des aspects importants de l'adoption de jumeaux numériques est de minimiser l'écart entre l'objet physique en question et sa version numérique. Dans le cas de méthodes de génération d'échantillons numériques de microstructure cérébrale, cet écart se traduit par la différence entre les échantillons virtuels et celles présentes dans le cerveau humain. Cette différence peut être mesurée en adoptant plusieurs métriques caractéristiques comme la taille moyenne des cellules, la fraction volumique de différents types de cellules dans un échantillon, etc... Par conséquent, la réduction de cet écart nécessite des bases de données de haute résolution de la microstructure comme source d'information pour confronter avec les échantillons de microstructure virtuelle.

- **Complexité de la matière grise.** L'étude de la microstructure dans la communauté d'IRMd a été globalement séparée en deux parties l'une traitant la matière blanche et l'autre la matière grise. Les efforts précédents ont été principalement orienté vers la microstructure de la matière blanche. Par la suite, peu de travaux ont abordé la modélisation de la matière grise caractérisée par des structures plus complexes. Notamment, la coexistance de cellules de plusieurs types et de formes variéees constitue un obstacle important pour la modélisation des microstructures de la matière grise.

Cette thèse s'inscrit dans le cadre global de l'approche computationnelle et a pour objectif d'adresser les défis présentés ci-dessus. Pour ce faire, plusieurs méthodes ont été développés au cours de cette thèse dans chacun des trois volets spécifiques à l'approche computationnelle.
Ci-dessous, un exposé détaillé des méthodes développés ainsi qu'une discussion des contributions et leur impact sera présentée:

- **La méthode de décomposition SGEAD.** La méthode SGEAD est un acronyme pour Skeleton-Guided Ellipsoid-Atom Decomposition (SGEAD). Cette dernière est une méthode computationnelle qui permet de convertir des maillages de géométries de cellules cérébrales en un ensemble d'ellipsoïdes approximant la surface de ces maillages. Le développement de cette méthode a été motivé par les points suivants:

  1. L'efficacité en mémoire des ellipsoïdes pour représenter la surface des objets 3D.
  2. La capacité des ellipsoïdes à fournir une bonne approximation des surfaces des maillages de géométries tridimensionnelles.

3. L'efficacité computationnelle pour la vérification de l'appartenance d'un point à une ellipsoïde, ce qui constitue un ingrédient clé pour la simulation Monte Carlo du processus d'IRMd.

Pour démontrer l'impact de cette première contribution, la méthode SGEAD a été évaluée en l'appliquant sur le jeu de donnée H01 (https://h01-release.storage.googleapis.com/landing.html). Ce dernier constitue l'un des jeux de données les plus précis caractérisant les cellules provenant d'un échantillon de cortex humain avec la microscopie électronique. Plus précisément, le jeu de données H01 forme une base de données de maillages de cellules cérébrales, neurones, cellules gliales, et autres avec une résolution à l'échelle nanométrique. Cette base de donnée est aussi volumineuse et est hébergée sur les serveurs de stockage de Google.

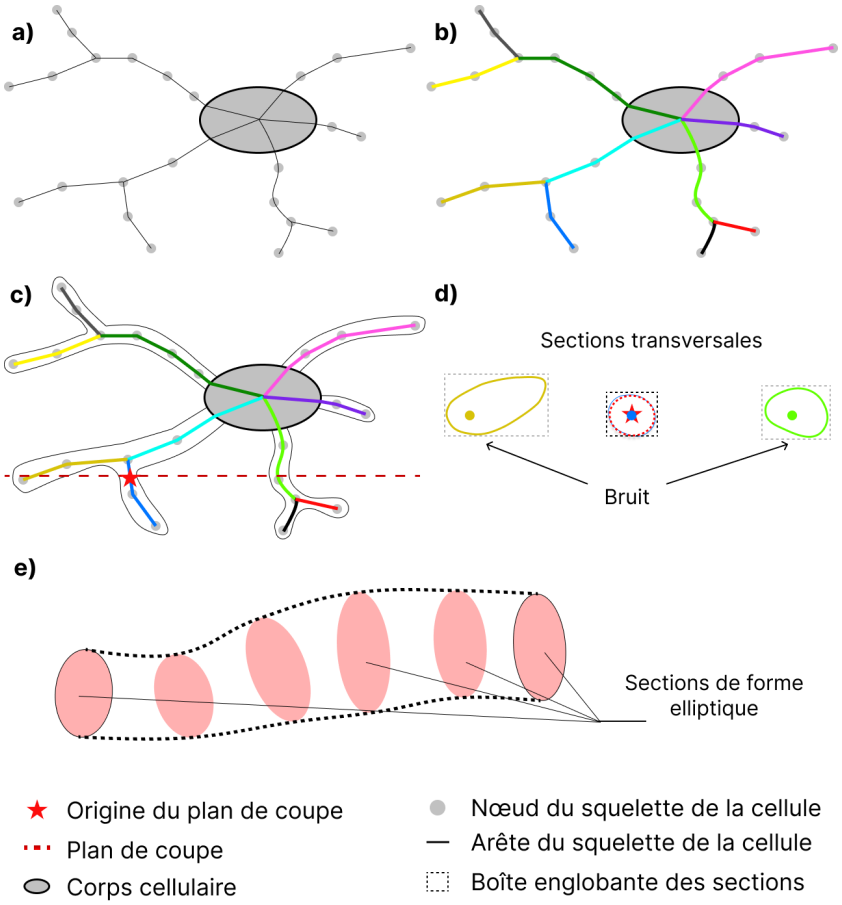

Figure 1: *Illustration résumant les différentes étapes de la méthode SGEAD. a) Le squelette du neurone à décomposer. b) Ségmentation du squelette en simple fragments continus représentés par différentes couleurs. c) Illustration d'un plan de coupe en trait pointillé rouge pour extraire la section transversale de la cellule correspondant au fragment du squelette en bleu foncé. d) Exemple de sections transversales obtenues dont seulement une située au centre correspond à la section d'intérêt. e) Illustration d'un fragment du neurone reconstruit à l'aide des sections elliptiques obtenues durant l'étape précédente.*

La Figure.(1) illustre les différentes étapes de la méthode SGEAD. Cette dernière a été évaluée sur un ensemble de 842 cellules de différents types. Les résultats de cette campagne de décomposition démontrent la capacité des ellipsoïdes à compresser des les maillages tridimen-

sionnelles de neurones, ainsi, fournissant une alternative plus efficace en mémoire permettant une meilleure scalabilité des méthodes computationnelles.

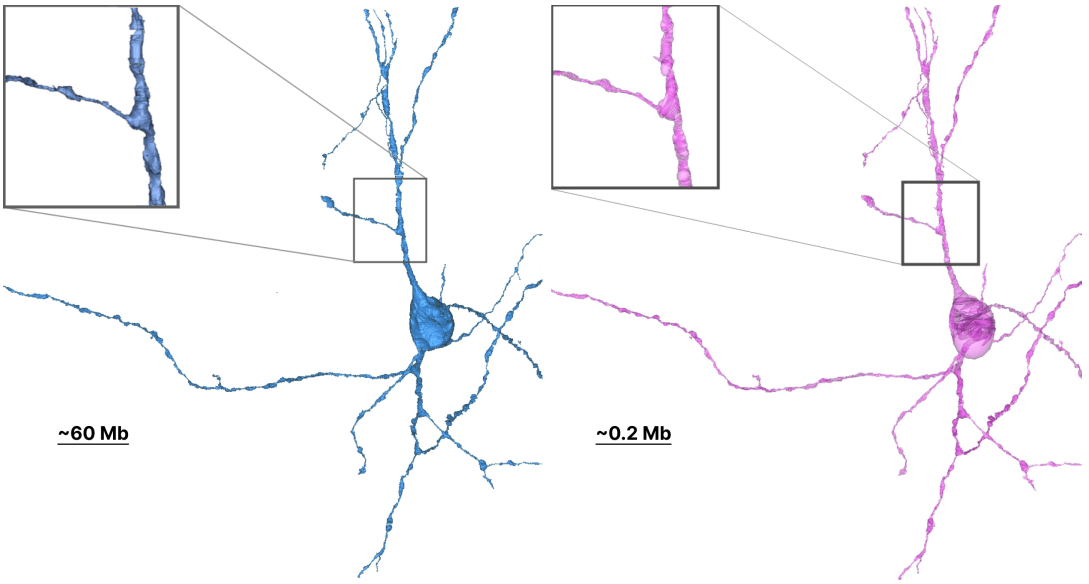

Figure 2: *Illustration d'un premier résultat de la décomposition d'un maillage d'une cellule de type interneurone en un ensemble d'ellipsoïdes en utilisant la méthode SGEAD. Le panel a) représente le maillage de l'interneurone provenant de la base de données H01 identifié par ID 5043316825 pesant environs 60 Mégaoctets. Le panel b) représente la version en ellipsoïdes obtenue avec la méthode SGEAD.*

La réduction de la taille en mémoire des cellules comme illustré dans la Figure.(2) permet de réduire la barrière de mémoire imposée par l'utilisation de maillage. Cette réduction de mémoire permet potentiellement la possibilité d'avoir des scènes virtuelles de microstructure avec densité de cellules supérieur à celle possible avec des maillages ou pour une densité cellulaire donnée, générer des scènes de microstructure plus larges.

- **Génération de scènes virtuelles de microstructure de la matière grise.** Cette seconde partie des contributions a pour but d'avancer les méthodes de génération de jumeaux numériques d'échantillons de cerveaux d'une taille allant de quelques dizaines à cent micromètres. La génération de telles microstructures dans la cadre de cette thèse est ciblée sur la matière grise. Pour permettre la création de microstructures synthétiques, une approche hybride a été adoptée. Cette approche consiste à générer des échantillons de microstructure en combinant des cellules digitales reconstruites du cerveau humain avec des méthodes de génération purement synthétiques. La première contribution de la méthode SGEAD a permis l'obtention d'une première source de cellules réalistes du cerveau humain. Par la suite, une méthode a été développée pour reconstruire des neurones du cerveau humain provenant d'une deuxième base de donnée nommée NEUROMORPHO https://neuromorpho.org/. Cette base de donnée contient des cellules cérébrales de plusieurs espèces et est enregistrée sous un format spécifique ".SWC". La méthode développée consiste à reconstruire des neurones représentés par un ensemble de sphères. La développement de cette méthode est motivé par plusieurs points: la diversification de sources de neurons pour la génération d'échantillons synthétiques de la microstructure de la matière grise, la représentation en un

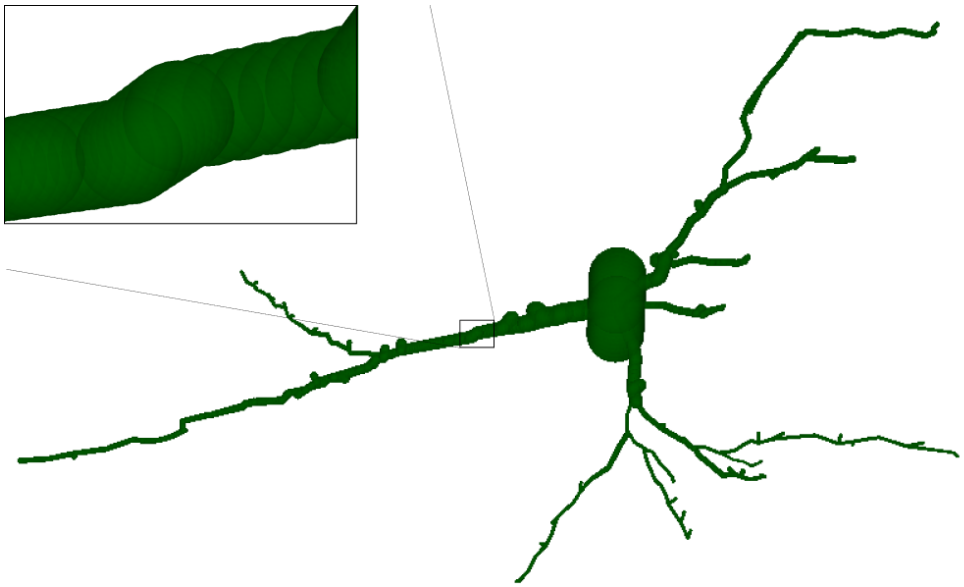

Figure 3: *Illustration de la reconstruction en sphères d'un neurone pyramidale issu du jeu de donnée Neuromorpho.*

ensemble de sphères qui est convenable pour simuler le processus de diffusion, et finalement, la compatibilité avec la toolkit MEDUSA [Ginsburger *et al.*, 2019] introduite pour génrérer des échantillons synthétiques de la microstructure de la matière blanche.

La Figure.(3) montre un exemple d'une cellule pyramidale reconstruite à partir du fichier ".SWC" provenant de la base de données de NEUROMORPHO dont la méthode est mise en open-access sur la dépôt Git suivant: `https://framagit.org/cpoupon/gkg/-/tree/master/python/simulation`.

En plus de la reconstruction des cellules, plusieurs contributions ont été faites au code source du toolkit MEDUSA en C++: développements pour que MEDUSA supporte des géométries à base d'ellipsoïdes, l'intégration de cellules reconstruites issues des jeux de données H01 et NEUROMORPHO, et finalement, le développement d'une méthode qui permet la génération de microstructures synthétiques suivant l'approche hybride. Ces intégrations ont permis la génération de premiers échantillons de microstructure.

Quelques résultats de ces développement sont illustrés dans les Figures (4) et (5). La première Figure(4) montre un échantillon de microstructure généré à partir de cellules issues du jeu de données H01 obtenues par la méthode de décomposition SGEAD. Cet échantillon présente un mixte de deux types de cellules: des neurones pyramidales en bleu clair et des interneurones en rose. La deuxième Figure.(4) représente quatre divers échantillons génrérés par l'outil MEDUSA. Le premier échantillon a) représente un cas classique d'échantillons générés par le même outil [Ginsburger *et al.*, 2019]. Les échantillons b) et c) sont hybrides et représentent un mélange d'axones et de neurones pyramidales issus des reconstructions des cellules de la base de donnée NEUROMORPHO. La différence entre les échantillons b) et c) est dans la densité de neurones. Finalement, le dernier échantillon d) représente une population de neurones pyramidales provenant de NEUROMORPHO.

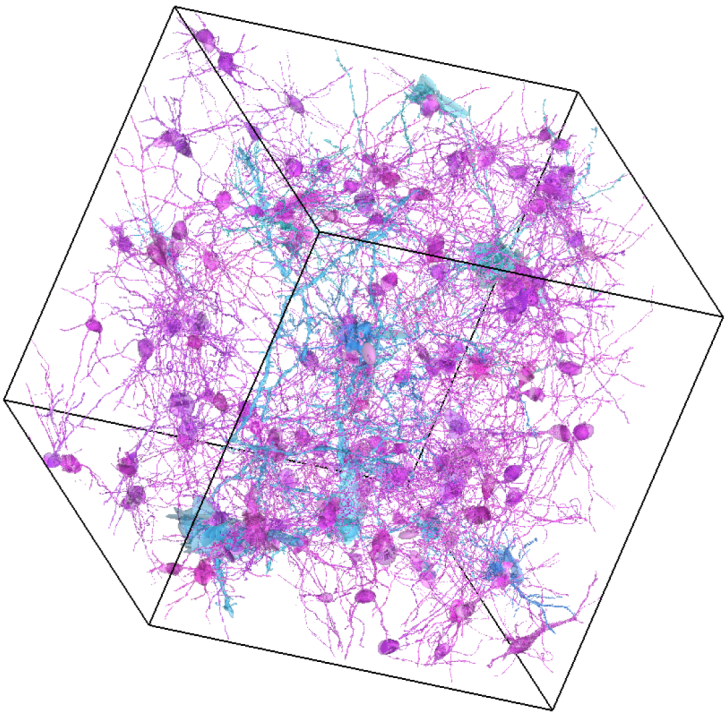

Figure 4: *Preuve de concept de la génération d'un échantillon de microstructure synthétique combinant des interneurones et cellules pyramidales provenant de la base de donnée H01.*

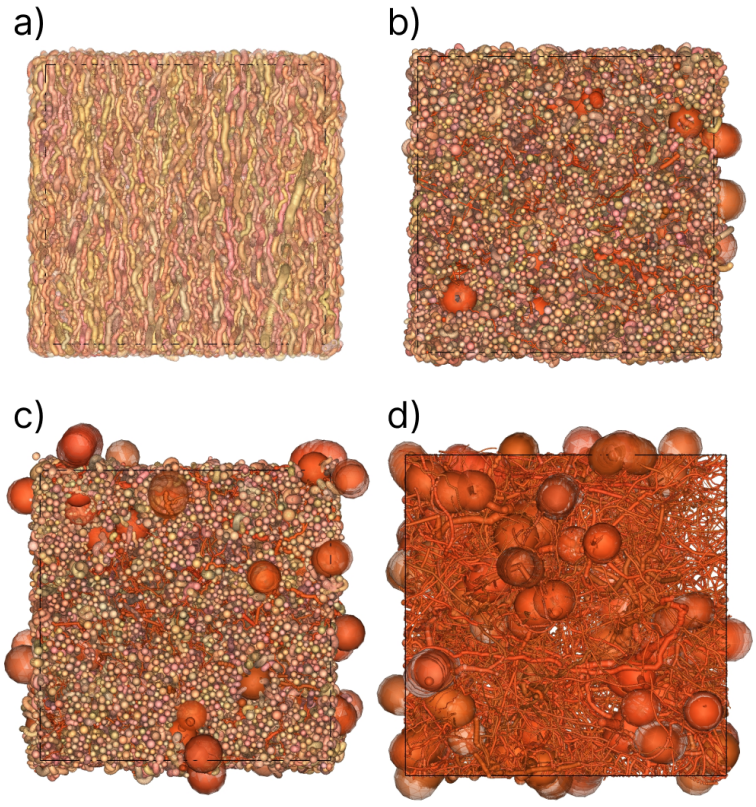

Figure 5: *Illustration de quatre échantillons de microstructure cérébrale synthétique obtenus après l'extension du toolkit MEDUSA pour pouvoir intégrer des cellules issus de bases de données différentes. Les deux échantillons b) et c) sont hybrides et mixent des neurones pyramidales provenant de la basse de données NEUROMORPHO avec des axones synthétiques.*

- **Extension et validation du simulateur Monte Carlo d'IMRd MEDUSA.** Après avoir généré quelques premiers échantillons synthétiques de la microstructure du cortex cérébral, il est important de pouvoir extraire les signaux d'IRMd qui leurs correspondent. Par conséquent, des développements ultérieurs dans l'outil MEDUSA ont été introduits pour permettre la simulation du processus de diffusion sur des représentations géométriques diverses. Plus précisément, cette partie du travail a consisté en la généralisation de l'outil MEDUSA pour pouvoir faire des simulations Monte Carlo sur des microstructures représentées par des sphères, ou ellipsoïdes, ou les deux à la fois. Une autre contribution faite visait à généraliser le type d'interaction entre les particules et les membranes cellulaire de la microstructure. Cette contribution se résume en une méthode qui permet de gérer la collision élastique entre les particules qui diffusent dans la microstructure virtuelle représentée par des sphères ou ellipsoïdes. Finalement, une autre généralisation a été implémentée dans MEDUSA pour pouvoir attribuer des valeurs variables à certaines propriétés du tissu cellulaire comme une diffusivité et perméabilité variables selon le type de cellule. Ces propriétés de tissus sont prédéfinis par l'utilisateur dans un fichier de paramètres ayant une forme très similaire au type de fichier JSON.

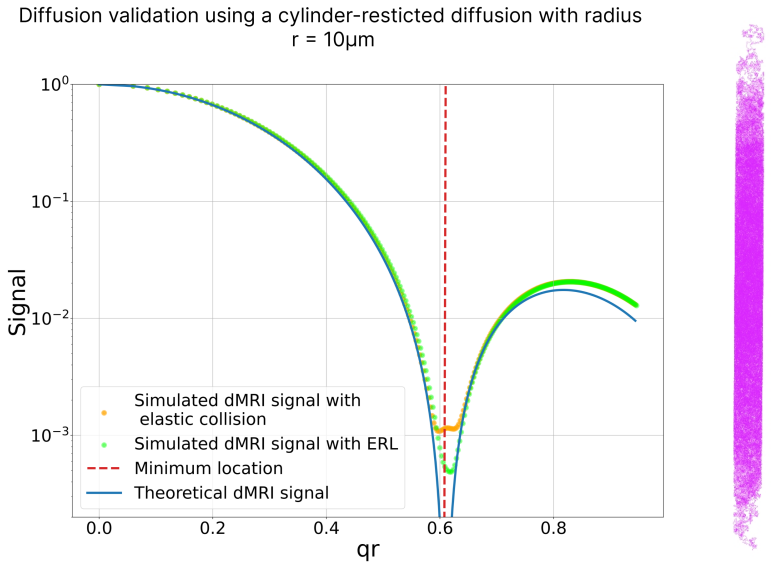

Figure 6: *Illustration de la courbe de validation du simulateur MEDUSA dans le cas de diffusion dans un cylindre représenté par un ensemble de sphères. La comparaison des signaux simulés utilise deux types d'interactions entre les particules et les membranes: collision élastique représentée vert et sans collisions (ERL) en jaune. La comparaison est faite avec la courbe bleue obtenue à partir de la forme analytique du signal d'IRMd d'un cylindre de rayon de 10µm. La deuxième figure à droite illustre les trajectoires des particules simulées contraintes à la géométrie du cylindre.*

Bien que l'ensemble de ces contributions ont permis d'obtenir les signaux d'IRMd pour des échantillons diverses, il n'est pas clair si les signaux obtenus sont équivalents à ceux provenant des mêmes géométries représentés par des maillages. Classiquement, les simulateurs classiques d'IRMd dans la littérature se basent principalement sur des maillages et en second lieu sur des géométries voxélisées. Par conséquent, un travail de validation a été fait pour démontrer la validité des signaux d'IRMd obtenus par l'outil MEDUSA qui repose sur

des représentations géométriques atomiques à base de sphères et ellipsoïdes. Cette validation est faite sur deux niveaux, un premier qui est analytique et qui repose sur la comparaison du signal simulé d'IRMd issu d'un cylindre avec la formule analytiquement dérivée. Le deuxième niveau est empirique et repose sur la comparaison des signaux d'IRMd avec un simulateur établi dans la communauté scientifique.
Une partie des résultats de la validation du simulateur d'IRMd MEDUSA est illustrée dans la Figure.(6). Cette dernière compare deux signaux simulés d'IRMd avec la courbe analytique. Chacun des deux signaux simulés représente un type d'interaction entre les particules et les membranes cellulaires. Le premier type d'interaction est la collision élastique, développée durant cette thèse, et le deuxième, nommé ERL pour Equal-step Random Leap, consiste à rejeter les pas de déplacements pour lesquels une collision est détectée et ré-échantillonner des nouveaux pas.
La deuxième partie des résultats concernant la validation s'est basée sur la comparaison des résultats issues du simulateur MEDUSA avec le simulateur Disimpy [Kerkelä *et al.*, 2020] publié précédemment. La comparaison s'est faite en simulant le processus de diffusion avec les deux simulateur en utilisant une géométrie et des séquences IRM communes. La géométrie utilisée consiste en un ensemble de sphères sans intersections avec un rayon moyen de 8μm.

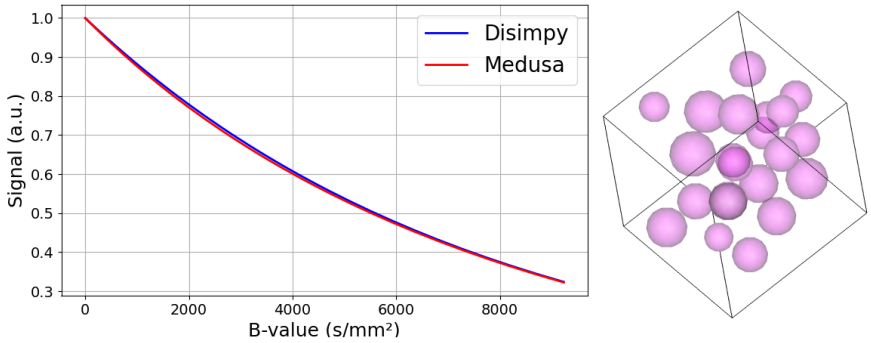

Figure 7: *Illustration des résultats de comparaison des signaux des simulateur MEDUSA et Disimpy issus du processus de diffusion intracellulaire. La figure à droite représente la géométrie utilisée pour obtenir les signaux.*

Les résultats de comparaison dans la Figure.(7) démontrent la similarité des résultats obtenus avec deux simulateurs différents. Ainsi, la combinaison de la validation analytique et empirique permet de démontrer la validité des signaux d'IRMd obtenus en utilisant le simulateur MEDUSA qui repose sur des nouvelles représentations géométriques en sphères et ellipsoïdes.
Finalement, une dernière expérience a été menée pour étudier l'effet de de l'erreur l'approximation d'une géométrie par un ensemble de sphères sur le signal d'IRMd. Comme les échantillons de microstructure sont représentés par un ensemble de sphères, cette expérience a pour objectif de vérifier si une telle approximation de géométrie par des sphères impliquait un biais dans les signaux d'IRMd simulés. Pour quantifier ce biais, une métrique nommée l'*overlapping ratio* O_r a été introduite pour mesurer la qualité d'une fibre de forme cylindrique représentée par des sphères. En second lieu, l'erreur du signal IRMd pour des fibres de différentes

qualités a été mesurée pour des cylindres de trois tailles différentes. L'erreur du signal est mesurée par l'écart de deux signaux provenant d'une fibre idéale de qualité supérieure et d'une autre fibre avec une qualité spécifiée par une valeur de l'*overlapping ratio*.

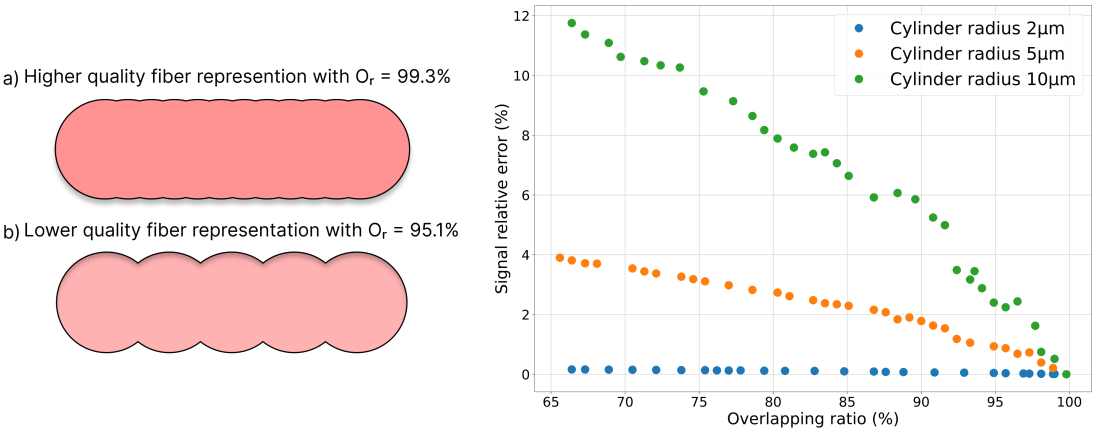

Figure 8: *Illustration des résultats de l'expérience.*

Les résultats de cette expérience dans la Figure.(8) montrent que l'effet de la représentation d'une fibre par des sphères est plus apparent pour des fibres avec un large diamètre. Aussi, cette expérience a permis de définir un seuil minimal de la métrique O_r de 96% pour lequel l'erreur induite par l'approximation des géométries par des sphères reste négligeable pour des séquences d'IRM avec des paramètres semblables aux imageurs cliniques ou pré-cliniques disponibles.

- **Large campagne de simulations au TGCC.** Finalement, cette dernière partie a consisté en la réalisation d'une campagne de simulations en utilisant l'infrastructure du TGCC: Très Grand Centre de Calcul du CEA. Cette contribution vise d'abord a démontré la scalabiltié du simulateur MEDUSA pour générer des échantillons virtuelles de la microstructure de la matière grise et de pouvoir réaliser des simulations Monte Carlo pour extraire les signaux d'IRMd. La campagne de simulations vise aussi à faire une première étude exploratoire des effets de certains paramètres de la microstructure sur le signal d'IRMd. Pour réaliser cette campagne de simulations, l'outil de virtualisation Docker a été adopté pour permettre le déploiement de MEDUSA aux super-calculateurs du TGCC. En outre, une pipeline en Python de récolte et traitement de données a été développée durant cette thèse pour permettre l'extraction des données de la campagne au TGCC.
  Cette campagne a été réalisée en deux parties: une première dédiée uniquement à la génération des échantillons de microstructure et une deuxième partie consacrée à la simulation Monte Carlo pour l'extraction des signaux d'IRMd. La première partie de la campagne a consisté en la création de quatre types de microstructures: un premier abrégé GM contenant uniquement une population de neurones pyramidales, le deuxième type abrégé WM_GM est hybride et contient une mixture d'axones synthétiques et de neurones pyramidales reconstruits à partir de la base de données de NEUROMORPHO. Le troisième type WM consiste en une population d'axones orientée selon l'axe z. Finalement, le dernier type d'échantillon nommé WM_2Pops représente deux populations d'axones perpendiculaires. Un ensemble

de plus 45 000 échantillons de microstructure ont été générés combinant les quatre différents types.

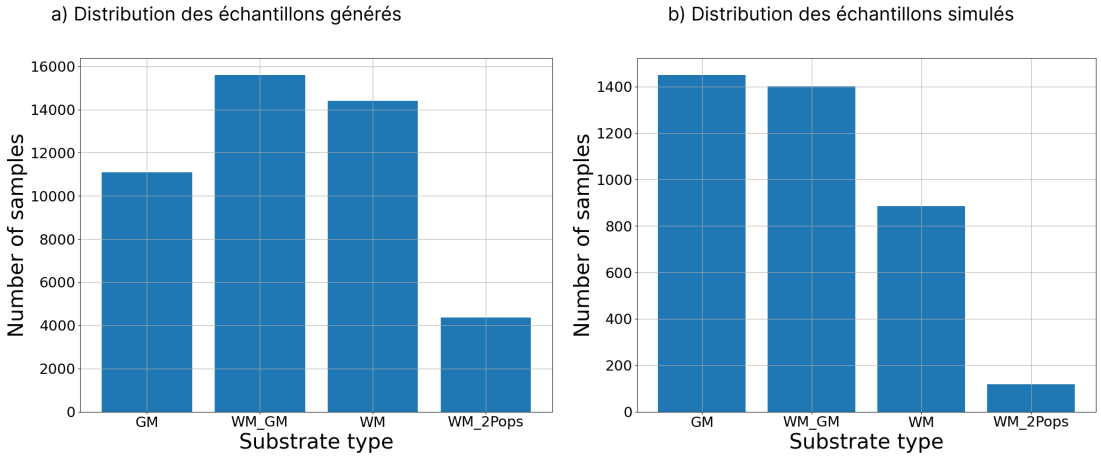

Figure 9: *Illustration des résultats de l'expérience.*

La distribution des échantillons générés et simulés est présentée dans la Figure.(9). Pour chacun des trois types suivants: GM, WM_GM, et WM il y a plus de 10 000 échantillons générés par type. Une fraction des échantillons générés ont été simulés: utilisés pour extraire leurs signaux d'IRMd correspondants. Un total de 3856 échantillons ont été simulés et leur distribution est illustrée dans le panel b) de la Figure.(9). Les signaux obtenus à partir des échantillons générés ont permis d'établir des premiers résultats préliminaires sur l'effet du diamètre moyen des corps cellulaire et la densité de neurones sur les signaux d'IRMd.

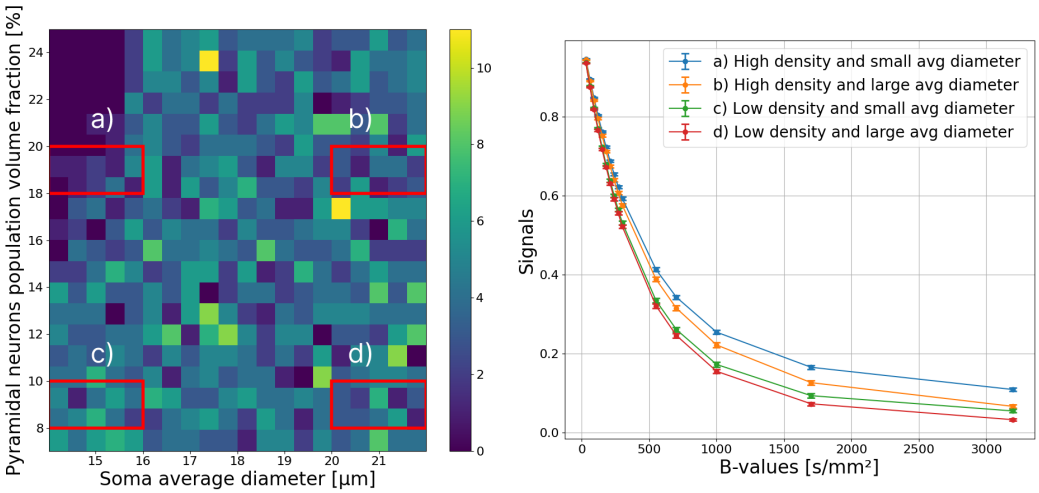

Figure 10: *Les effets du diamètres et fraction volumique des neurones sur les signaux d'IRMd.*

Un premier résultat d'exploration est présenté dans la Figure.(10) montrant la distribution des paramètres de microstructure des échantillons du type GM. Quatre zones cadrées en rouge ont été sélectionnées pour la comparaison des signaux d'IRMd qui leurs correspondent. Ces premiers résultats montrent une différence à l'échelle du signal d'IRMd pour les différentes valeurs des paramètres sélectionnés de la microstructure.

Un deuxième résultat préliminaire conduite vise à étudier l'effet de la densité de soma sur

le signal d'IRMd. Pour ce faire, des échantillons du type WM_GM ont été considérés. Ces échantillons ont été séparés en deux groupes avec des paramètres de microstructure similaire mais avec des densités de neurones différentes.

- Groupe a) Comporte des échantillons avec une densité volumique d'axones dans l'intervalle [0.35, 0.42] et un diamètre moyen dans l'intervalle [1.0, 3.0] μm. Ces même échantillons comportent des neurones pyramidaux avec un diamètre moyen du corps cellulaire dans l'intervalle [17, 18] μm et une densité volumique dans l'intervalle [0.2, 0.25].
- Groupe b) comportes aussi des échantillons du type WM_GM avec les mêmes caractéristiques que le groupe q) mais avec une densité volumique de neurones plus faible dont les valeurs sont comprises dans l'intervalle suivant [0.07, 0.11].

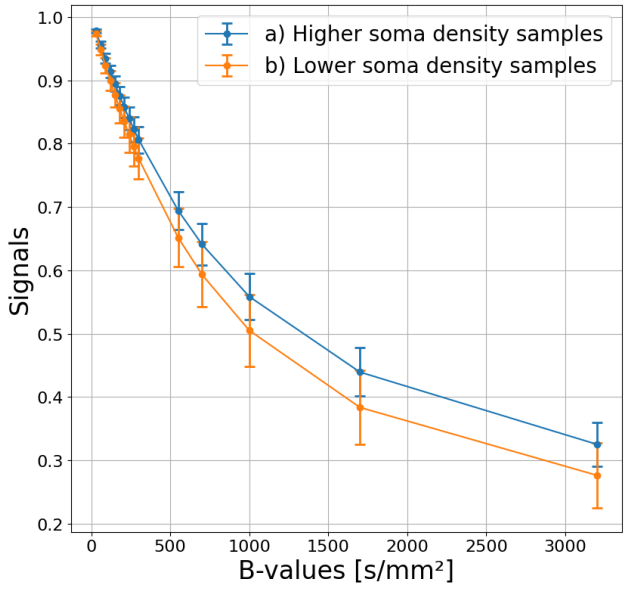

Figure 11: *Résultats de comparaison des signaux d'IRMd entre deux groupes d'échantillons virtuels de microstruture a) et b) caractérisés par des densités de corps cellulaires différentes.*

En conclusion, les travaux menés durant cette thèse ont permis l'avancement des méthodes de génération d'échantillons virtuels de la matière grise tout en étant efficace en mémoire. Cette approche hybride consistant d'intégrer des cellules issues de plusieurs bases de données permet d'améliorer le réalisme des échantillons virtuels générés, ainsi, contribuant à la réduction de l'écart entre la simulation et la réalité provenant de l'histologie. Néaumoins, il reste encore d'autres composantes à intégrer dans le future comme les vaissaux sanguins qui, durant cette thèse, n'étaient pas pris en considération. En outre, la méthode SGEAD a démontré sa capacité à convertir des cellules larges en des cellules légères en mémoire, facilitant ainsi une future comparaison des résultats des signaux issus de la simulation avec ceux obtenus des imageurs disponibles.
Finalement, différents projets à venir sont envisagés dans le cadre de l'approche computationnelle pour décoder la microstructure. Ces projets comprennent de réaliser un réseau de neurones pour estimer les paramètres de la microstructure à partir des données obtenues lors de la campagne de simulations.

Introduction

Context and motivation

The human brain plays a paramount role in almost every aspect of our lives. It is involved in sensorimotor functions as well as in higher-level processes such as planning, reasoning, and learning. The human brain is also characterized by its capacity to change and adapt by reorganizing its structure, a property known as the plasticity of the brain.

To study the brain non-invasively, medical imaging techniques have provided a powerful set of tools to reconstruct images of the brain. Particularly, Magnetic Resonance Imaging (MRI) techniques allow to study and understand various aspects of the human brain: its connectivity, its structures, and functional regions. In addition to its prominent use in research, MRI is also an indispensable clinical tool for neurologists. It is used for various diagnoses including brain strokes, cancerous tumors, multiple sclerosis, etc...

In the context of this thesis, the focus is on the particular MRI modality named diffusion MRI, which relies on the physical diffusion phenomenon, i.e the displacement of water molecules inside the brain, to acquire signals characterizing the diffusion properties of the brain tissues. These tissues form a complex diffusion medium within which the water molecules diffuse and encounter different cellular barriers. Thus, the sensitivity of diffusion MRI to the diffusion process indirectly provides information about the diffusion medium. This sensitivity is used in the context of this thesis to characterize the brain's microstructure, i.e the mesoscopic organization of brain cells, with a focus on the gray matter. This characterization is known by microstructure imaging and it aims at inferring cellular properties such as average cell diameters and densities.

To study the gray matter microstructure, a computational approach relying on an end-to-end simulation process is used in this work. The simulation process is divided into two parts: the first one is dedicated to generating synthetic virtual geometries modeling gray matter microstructures, and the second one is dedicated to simulating the diffusion process to acquire diffusion MRI signals.

Keywords

Diffusion MRI, Inverse problems, Simulations, 3D geometry, Modeling, High-performance computing

Thesis organisation

This thesis is organized into two sections with three chapters within each section. The first section provides the state-of-the-art in the field of diffusion MRI and microstructure imaging, and the second one presents the contributions made in the course of this thesis. The chapters are organized as follows:

Section I: State-of-the-art.

- **Chapter 1.** It presents a general overview of the human brain anatomy at both macro and microscopic scales.
- **Chapter 2.** It presents the fundamentals of MRI physics, the principles of diffusion MRI and finally lists some widely used MRI sequences.
- **Chapter 3.** This chapter provides an overview of the state-of-the-art synthetic microstructure generation methods and diffusion MRI simulators.

Section II: Contributions

- **Chapter 4.** This chapter presents contributions made in the generation of virtual gray matter microstructures and facilitating the use of realistic cells reconstructed using electron microscopy.
- **Chapter 5.** Presents contributions in diffusion MRI simulations consisting in the presentation of a new scheme to perform diffusion MRI simulations and validation of the MEDUSA dMRI simulators.
- **Chapter 6.** This chapter provides the results of a large-scale simulation campaign performed using the high-performance computing platform in France named Très Grand Centre de Calcul du CEA (TGCC).

Finally, the last chapter presented at the end is dedicated to providing a general conclusion and future perspectives in this research area.

Chapter 1

Human brain anatomy

The human brain is probably one of the most complex objects for scientists to study and understand. To picture this complexity, one can think of the brain as a large networked system with over 80 billion nodes [Azevedo *et al.*, 2009] where each node represents a neuron cell body. Grouped together, these nodes share over 100 trillion connections[1]. To have a visual of the scale of the brain network, we can consider the example of the World Wide Web (WWW). In fact, the average number of neurons in the human brain is more than twice as large as the total number of indexed web pages on the Google search engine, which is estimated at around 32 billion web pages[2]. Despite the challenging complexity of the brain, it is worth making an effort to explore and study it. The human brain possesses remarkable abilities that allow humans to perform different tasks, navigate various environments, and adapt to changes.
Almost every aspect of the human brain is interesting to investigate: The vision system for example, and how humans are able to process a large amount of visual data. The auditory system and the processing of different sounds. Speech and language are also one of the major topics of study of the brain. Motor skills and coordination: it is impressive how easy it is for humans to maintain a certain balance when moving and to smoothly manipulate different objects. These same abilities can be very difficult to achieve in the context of robotics systems [Breazeal & Scassellati, 2000]. Reasoning and cognition: the human brain's ability to learn abstract notions, understand concepts, and derive logical conclusions from first principles. Finally, another interesting aspect of the brain is its energy efficiency.
This chapter will lay the foundations for understanding the human brain anatomy in two steps: first, a macroscopic scale description of the human brain, then a detailed description of the microscopic anatomy. As this thesis is related to the microstructure of the cerebral cortex, more focus will be put on the microscopic part of the brain, especially the gray matter.

1.1 Macroscopic anatomy

1.1.1 Global anatomy

The human central nervous system (CNS) consists of the brain (*encephalon*) and the spinal cord (*medulla spinalis*) [Nieuwenhuys *et al.*, 2007]. The peripheral nervous system (PNS) is composed of nerves, ganglia, and plexuses.

[1] https://hms.harvard.edu/news/new-field-neuroscience-aims-map-connections-brain
[2] Accessed on Feb 2024: https://www.worldwidewebsize.com/

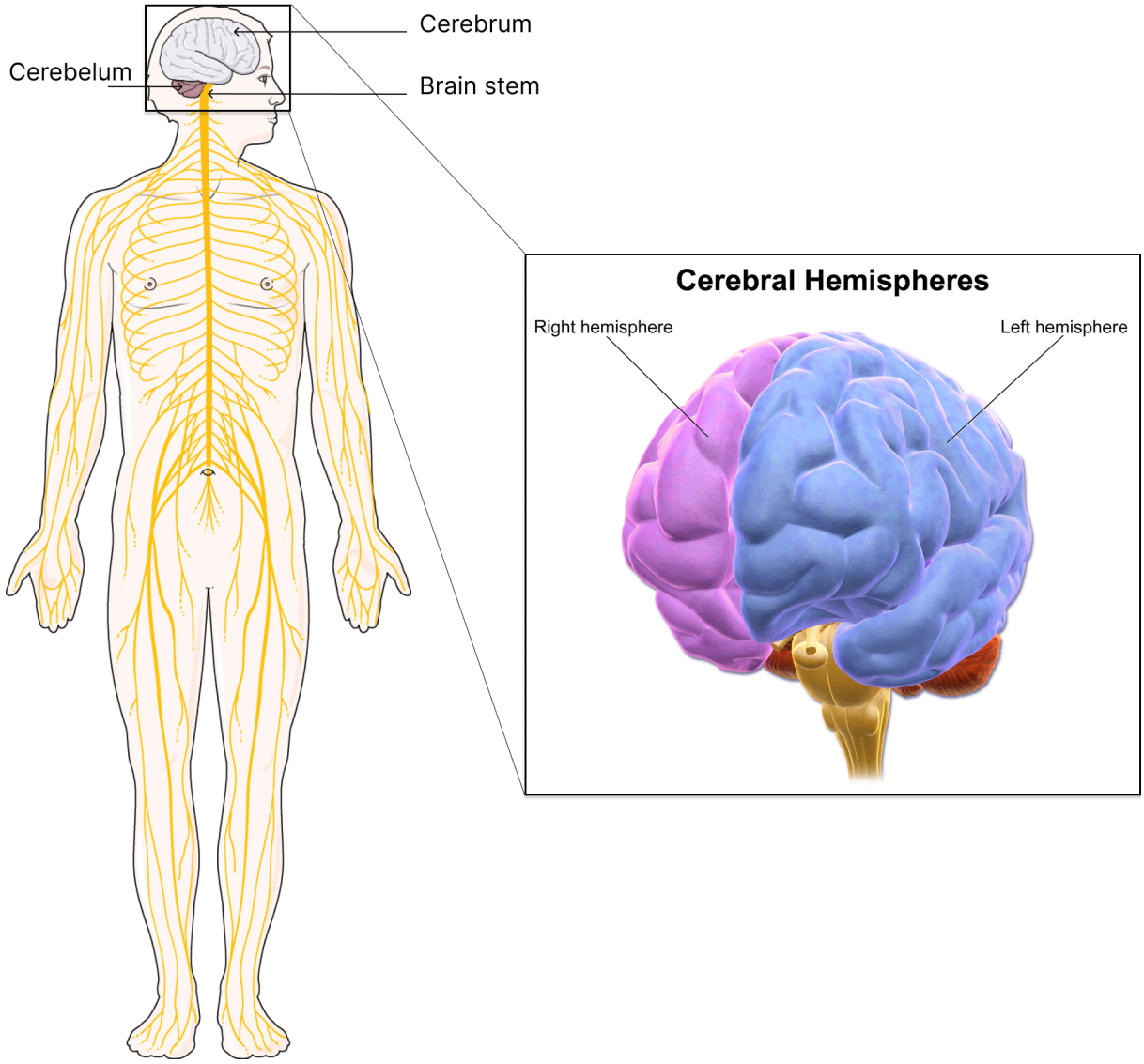

Figure 1.1: *Illustration of the human nervous system (adapted from https://smart.servier.com/).*

The brain is encased by the skull which provides a protective bone membrane. The space between the skull and the brain is occupied by three membranes called the meninges. These membranes are comprised of the following [Kayalioglu, 2009] [Nieuwenhuys *et al.*, 2007] [Yebga Hot, 2021] [Ginsburger, 2019]:

- **Pia mater**: It is a thin and loose membrane that is in direct contact with the brain. It closely covers the surface of the brain.
- **Arachnoid mater**: Named for its spiderweb-like appearance, it is a thin, transparent membrane covering the brain at a variable distance. The space between the pia and the arachnoid contains the cerebrospinal fluid (CSF).
- **Dura mater**: Also known as the pachymeninx, located at the outermost meningeal layer, the dura mater is a thick and strong layer. It is composed of fibroblasts and extracellular collagen.

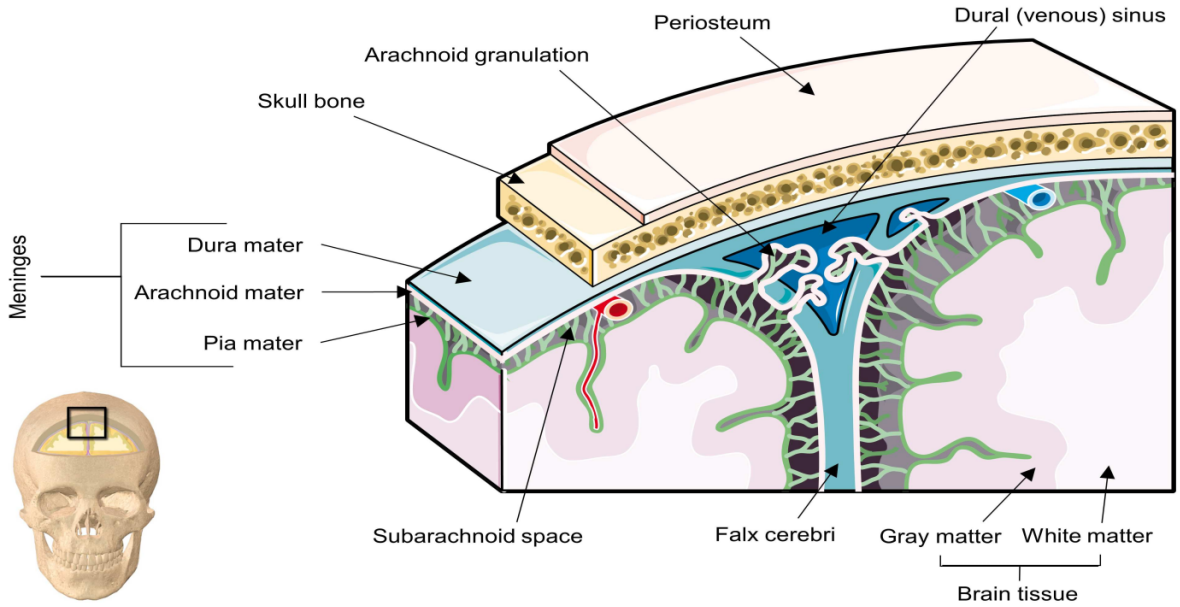

Figure 1.2: *Scheme of the meninges and their compositions (adapted from [Yebga Hot, 2021]).*

The CSF between the pia and arachnoid maters plays different important roles. First, it provides an additional protection layer against brain and spinal cord shocks. It also serves different functions, such as cleaning the waste produced by the brain, transporting important nutrients for different brain tissues, and reducing the effective weight of the brain [Wright *et al.*, 2012]. In fact, the human adult brain weights between 1.2 and 1.5 kilograms, but when submerged in the CSF, its total effective weight is reduced to around 50 grams.

Eighty percent of the CSF is produced in the lateral, third, and fourth ventricles by the choroid plexus (CP) (see Fig.(1.3)) with a smaller volume being produced by the ventricular ependyma, arachnoidal membrane, and brain tissue itself [Wright *et al.*, 2012]. Around 500 mL of CSF is generated each day, and at any time in normal adult brains, the volume of CSF is between 125 and 150 ml, of which approximately 20% is contained within the ventricles. The CSF can also be drained from the subarachnoidal space into the venous circulation. The scheme in Fig.(1.3) depicts the ventricular system of the human brain.

The brain's composition. On a global scale, the human brain can be subdivided into three components: the cerebrum, the cerebellum, and the brainstem.

- **The cerebrum.** In the uppermost region of the CNS, the cerebrum is the largest part of the human brain. It is also named the telencephalon. The cerebrum is mainly composed of two cerebral hemispheres surrounding and covering the diencephalon and the midbrain. Each hemisphere is made of different gyri or convolutions separated by fissures named sulci. The outer layer of the hemispheres is named the cortex (the gray matter) and the inner layer contains the white matter.

- **The cerebellum.** It is part of the hindbrain (rhombencephalon) and is located behind the medulla oblongata and pons. It comprised three parts: a median, two laterals as hemispheres, and the cerebellum worm. It is linked to the medulla oblongata, pons, and cerebral peduncles with the cerebellar peduncles. It is a terminal organ connected to voluntary and

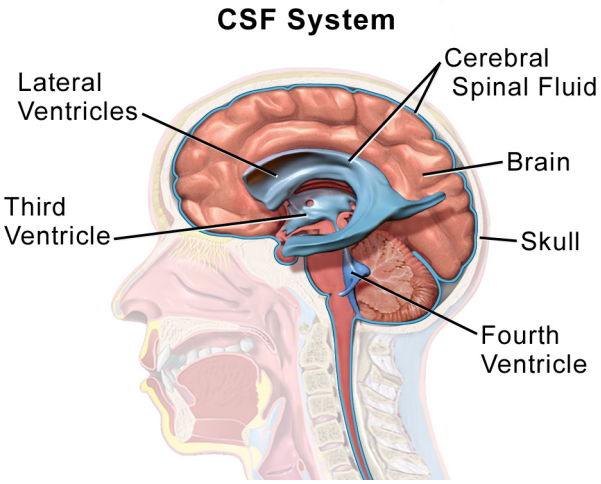

Figure 1.3: *Scheme of the human brain ventricular system responsible for CSF (adapted from [Wikiversity, 2020]).*

involuntary motor pathways. Thus, it controls equilibrium, posture tone, and all automatic movements [Yebga Hot, 2021].

- **The brainstem.** It is the structure that connects the cerebrum of the brain to the spinal cord and cerebellum. It is composed of three sections in descending order: the midbrain, pons, and medulla oblongata. It is responsible for many vital functions of life, such as breathing, consciousness, blood pressure, heart rate, and sleep. The brainstem contains many critical collections of white and gray matter [Basinger & Hogg, 2019].

1.1.2 Brain lobes

The outer layer of the brain hemispheres, i.e. the cortex, can be subdivided into four different regions called lobes: the frontal lobe, the parietal lobe, the occipital lobe, and finally, the temporal lobe. This section presents a brief overview of the different lobes of the cortex and their main functions [Jawabri & Sharma, 2023]:

- **The frontal lobe.** It is the largest lobe, located in front of the cerebral hemispheres, and has significant functions for our body. These functions include prospective memory, which is a type of memory that involves remembering the plans made, from a simple daily plan to lifelong plans. Speech and language are also important functional networks hosted by the frontal lobe. They have a dedicated region in the posterior inferior frontal gyrus named Broca's area. The frontal lobe also encompasses personality. In fact, when the prefrontal cortex suffers damage, personality changes can occur. In addition, other functions of the frontal lobe include decision-making, reasoning, learning, creativity, and movement control.

- **The parietal lobe.** It is located posterior to the frontal lobe and superior to the temporal lobe and is classified into two functional regions. The anterior parietal and the superior parietal lobes. The anterior parietal lobe comprises the primary sensory cortex. It receives most sensory inputs from the thalamus, and it is also responsible for interpreting simple

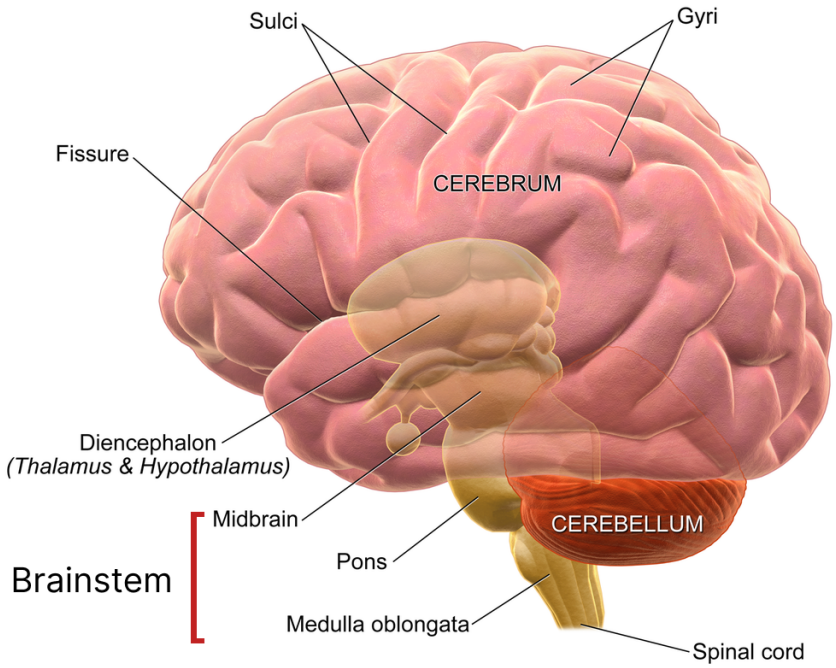

Figure 1.4: *Scheme of the global brain structures (adapted from [Wikiversity, 2020]).*

somatosensory signals such as touch, position, vibration, pressure, pain, and temperature. The inferior parietal lobe has the secondary somatosensory cortex, which receives the somatosensory inputs from the thalamus and the contralateral. The anterior and inferior parietal lobes integrate those inputs with other major modalities, such as visual and auditory inputs, forming higher-order complex functions like learning, language, sensorimotor planning, and the ability to differentiate between objects given their size, shape, and other features.

- **The temporal lobe.** It lies posterior to the frontal lobe and inferior to the parietal lobe. It contains two surfaces, the lateral and the medial surfaces. The lateral surface contains the superior and lateral temporal sulci. The parietal lobe has different functions, such as sound recognition, semantic retrieval, and semantic memory which is responsible for remembering thoughts or objectives that are common knowledge.

- **The occipital lobe.** It is the smallest lobe in the cerebrum cortex. It is located in the most posterior region of the brain, posterior to the parietal and temporal lobes. The role of this lobe is visual processing and interpretation. It receives, processes, and interprets visual information, which is then transferred to other regions of the brain.

The illustrations in Fig.(1.5, 1.6) show the four different lobes of the brain in addition to the motor and sensory functional regions of the cerebral cortex.

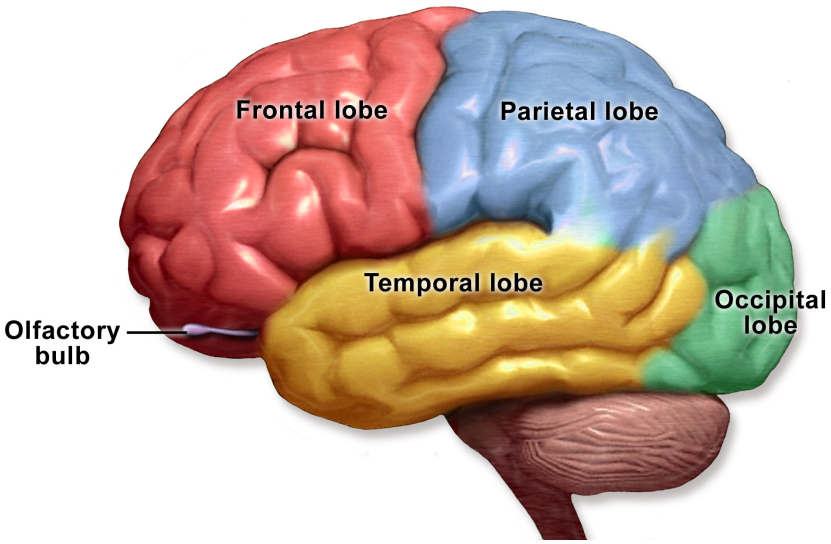

Figure 1.5: *Illustration of the brain lobes (adapted from [Wikiversity, 2020]).*

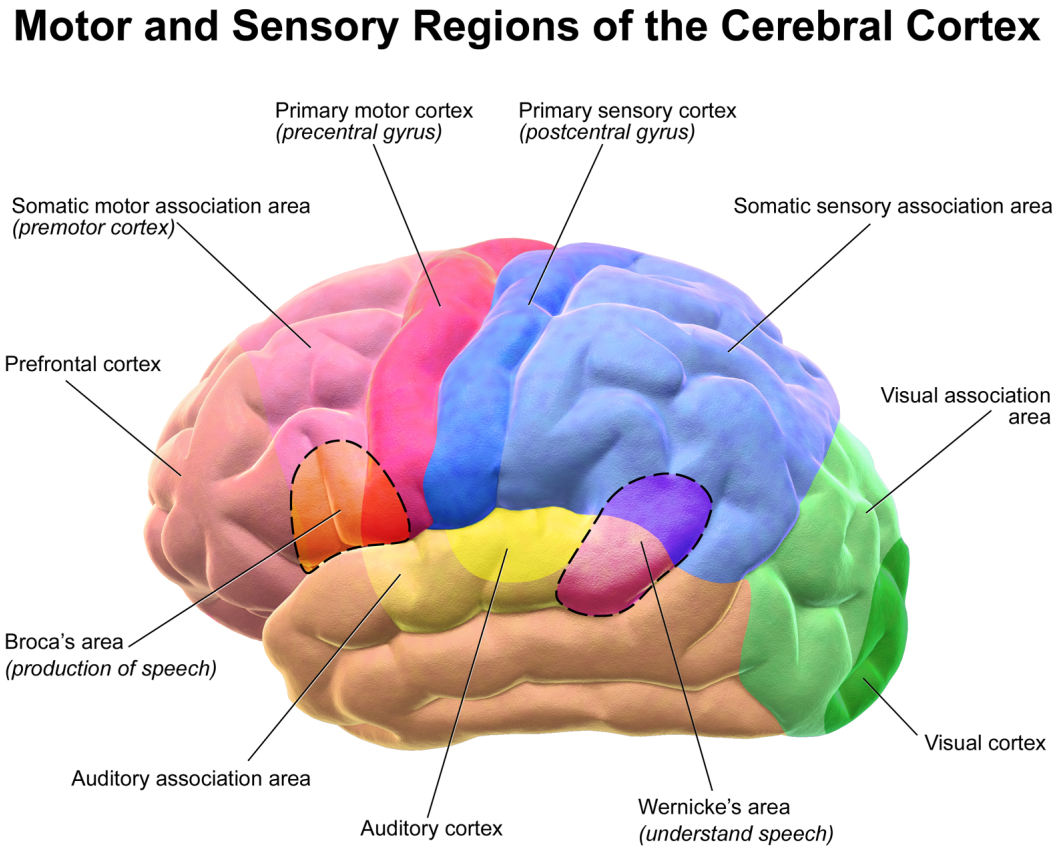

Figure 1.6: *Scheme of the global brain structures (adapted from [Wikiversity, 2020]).*

1.1.3 Gray matter

When looking at a dissected section of the human brain, as in Fig(1.7), it is possible to distinguish two different regions. First, a set of regions that are characterized by a slightly dark color and located in different parts of the brain, including a variable thickness layer located at the brain's outer surface. This set of regions forms what is known as the gray matter. The second region has a lighter color and is known as white matter. The color difference between the two regions is attributed to their cellular composition. The gray matter contains a large number of neuronal cell bodies, while the white matter contains the axonal prolongations of neurons, most of them being surrounded by a lipid myelin sheath responsible of its white color. The gray and white matter microstructures will be further detailed in the next section. The gray matter exists in different regions of the brain:

The cortex. It is the outermost layer of the cerebrum with a surface area ranging from around 1700 to 2300cm^2 [Tramo *et al.*, 1995]. The cortex can be subdivided into two regions: the neocortex and the allocortex. The neocortex means the new cortex, referring to its late appearance in the evolutionary history. It is present in all mammalians with varying sizes. The neocortex occupies a large proportion of the human cerebral cortex and has a laminar structure with six layers. The other region, the allocortex, occupies only a small proportion of the cerebral cortex compared to the neocortex. It is characterized by three or four cortical layers and includes the olfactory cortex, the hippocampal cortex, and finally, the periallocortex, which is a transitional zone between the neocortex and the allocortex.

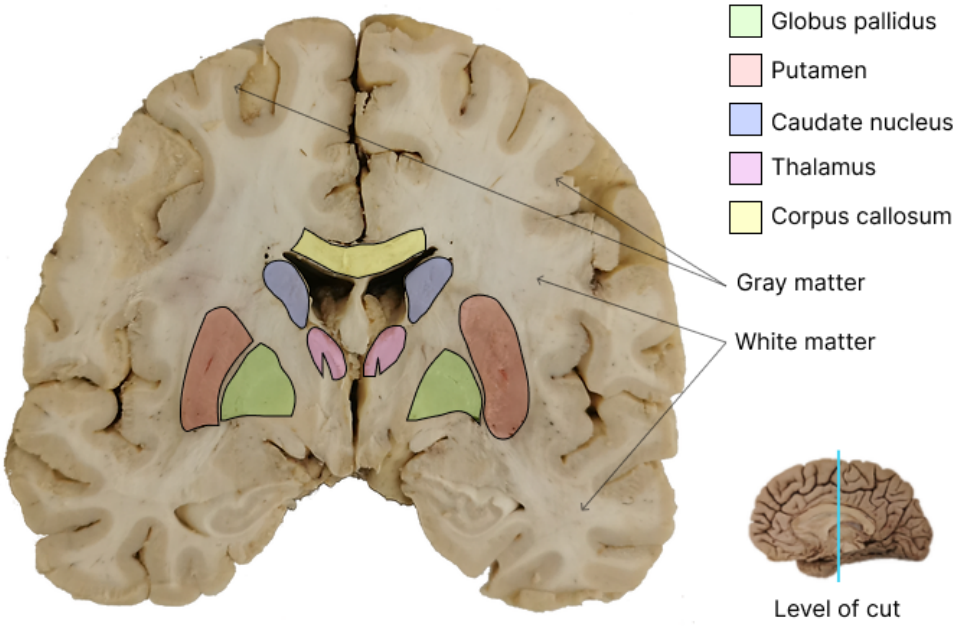

Figure 1.7: *Coronal section of the human brain illustrating the contrast between white and gray matter (adapted from [tau, 2024]). The legend highlights four deep gray matter structures as well as the corpus callosum, a white matter region connecting the two hemispheres of the brain.*

The basal ganglia. As seen in Fig.(1.7), the gray matter is not only located at the outer surface of the cerebral cortex. It can also be found in the cerebrum's deeper regions. The basal ganglia

refer to nuclei within these deep regions; they are engaged in different functions including motor control, emotions, behavior, and executive functions. At a high level, the basal ganglia nuclei can be subdivided into input and output nuclei. The input nuclei receive information from different sources, while output nuclei send information to the thalamus [Lanciego *et al.*, 2012]. The input nuclei include the caudate nucleus, the putamen, and the accumbens nucleus. The output nuclei include the globus palidus and the substancia nigra pars recticulata.

The diencephalon. Located between the corpus callosum and the cerebral cortex, the diencephalon is the central portion of the brain. It contains four different structures: the thalamus, the hypothalamus, the epithalamus, and finally, the subthalamus or the ventral thalamus. First, the thalamus forms a pair of ovoid-shaped structures consisting of gray matter and also fibers. The thalamus is an intermediary stage bridging the brain and body. It receives, filters, and sends sensory signals to its associated cortical areas. The epithalamus is a small region of the diencephalon, and it includes the pineal gland, the habenular nuclei, and the stria medullaris thalami. The hypothalamus contains different nuclei and plays an important role in homeostasis, i.e., the regulation of different body properties such as temperature, pressure, etc. Finally, the subthalamus occupies a small region of the diencephalon, and it is located between the thalamus and the hypothalamus.

1.1.4 White matter

Forming around 40% of the human's CNS, the white matter is another important part of the brain. It is characterized by a lighter color than gray matter, and its main functions consist in connecting different regions of the brain and transporting information. White matter is mostly made of nerve fibers named axons being prolongations of neuronal cells, and of glial cells. Some of the axons are enveloped with a lipid-rich layer called myelin, which provides the characteristic color of white matter. Myelin sheath plays an important role in insulating axons. Consequently, brain signals are able to travel faster through the axons. Thus, myelin damage can affect signal transport, which in turn affects motor and sensory functions. White matter also plays a key role in learning [Long & Corfas, 2014]. The same work mentions the following: « To learn is to myelinate ». In fact, the process of learning involves changes in white matter. More precisely, when learning to perform a new task, the creation of new myelin sheaths provides more insulation and increases the traveling speed of brain signals. Thus, the task can be done faster in a more "automatic" fashion. The white matter axons tend to cluster into bundles of fibers, forming fiber tracts. There are three types of white matter fibers: projection, association, and commissural fibers.

Projection fibers. They connect the cerebral cortex with nuclei of subcortical regions, brainstem, and spinal cord. The projection fibers can be divided into efferent and afferent fibers. This subdivision is based on the direction of the axonal projections. *Efferent* fibers are axonal projections that transmit information from the cerebral cortex. They include motor tracts. In contrast, *afferent* fibers transmit information to the cerebral cortex, and they include somatosensory tracts. They start from peripheral nerves and pass through the spinal cord to reach different regions in the cortex.

Association fibers. Connect one cortical area to another within the same hemisphere. They can be classified according to their length into two types: long and short association fibers. The short association fibers connect adjacent gyri. They are also known as U-fibers given their shape. The long fibers connect gyri that are relatively far from each other. They can be clustered into different bundles [Mandonnet *et al.*, 2018], mainly the cingulum, the external capsule, the inferior fronto-occipital fascicle, the inferior longitudinal fascicle, the superior longitudinal fascicle or arcuate fascicle, and finally, the uncinate fascicle. The subdivision of the association fibers is done based on multiple factors, including the fiber endpoint locations. Such fibers can be investigated using diffusion MRI methods such as tractgraphy. An illustration in Fig.(1.9) provides a sketch of the important association fibers and their location.

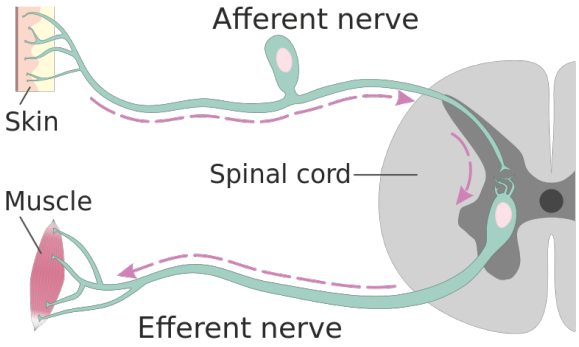

Figure 1.8: *Illustration of afferent and efferent nerves in the spinal cord (adapted from [Wikipedia contributors, 2024a]).*

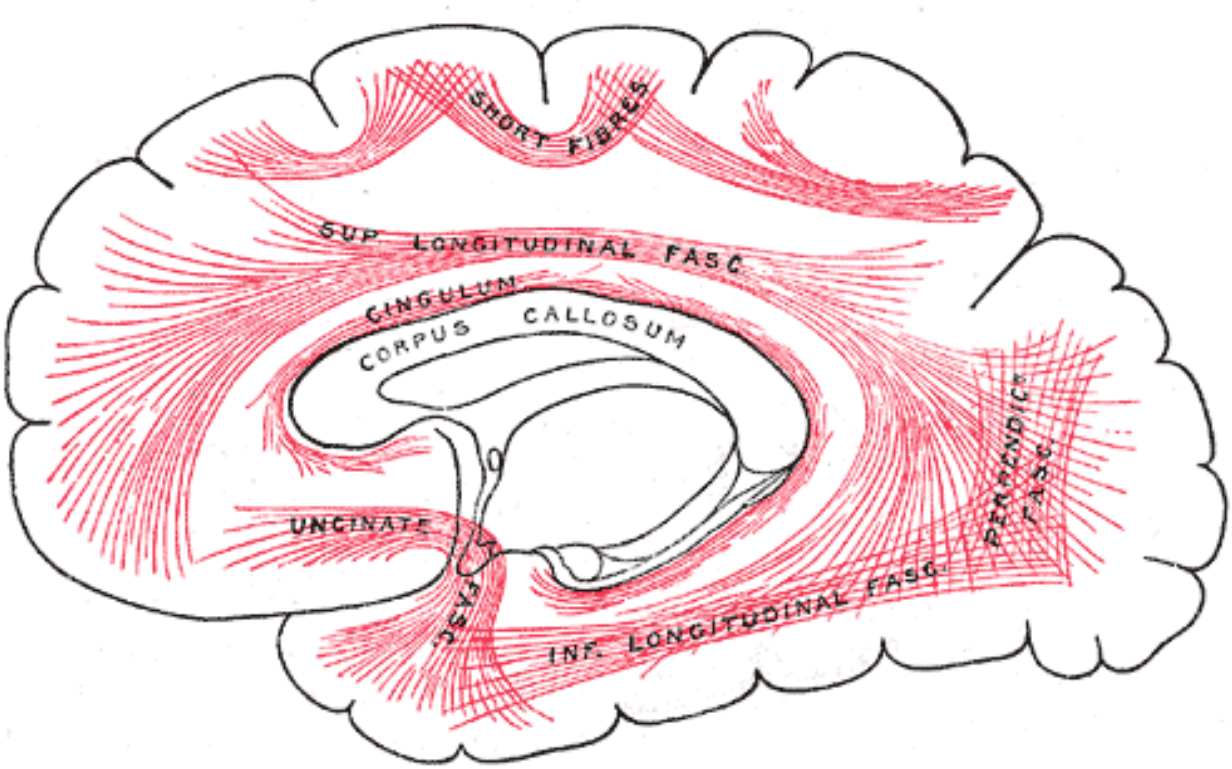

Figure 1.9: *Illustration of the main association fibers in the human brain (Image taken from [Henry Gray (1825–1861), 2024]).*

Commissural fibers. In contrast to association fibers connecting regions within the same hemisphere, commissural fibers provide connections between the two hemispheres. They can be subdivided into three different types of commissures: the corpus callosum, the anterior commissure, and finally, the posterior commissure. The corpus callosum, as labeled in Fig.(1.7,1.9), is the largest commissure [Webb, 2017]. It connects similar regions in the two hemispheres and conveys information between them. The corpus callosum can be further subdivided into different groups of fibers: the genu, literally meaning a "knee", the body, the isthmus, and finally, the splenium (see Fig.(1.10)).

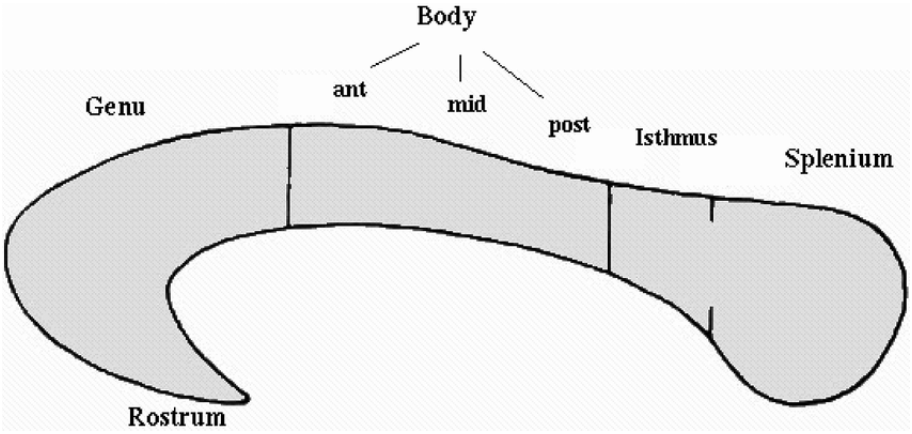

Figure 1.10: *Illustration of the different components of the corpus callosum (Image taken from [Di Paola* et al.*, 2010]).*

The anterior commissure has around 7 million fibers and connects olfactory and amygdaloid regions [Swaab *et al.*, 2012]. Finally, the posterior commissure connects occipital lobe regions, and its compression can lead to vision-related complications such as Parinaud's syndrome [Caskey *et al.*, 2023].

1.1.5 Brain vascular system

Despite its small relative body mass ratio, the brain consumes a significant amount of energy of 20% [Padamsey & Rochefort, 2023] of the total body consumption. The nutrients and the oxygen are carried to the brain via the vascular system. The blood inflow to the brain goes through two pairs of large vessels traversing the neck: the internal carotid and the vertebral arteries. Through the cavernous sinus, the internal carotid artery enters the skull and loops through the cavernous sinus. The internal carotid mainly supplies blood to the cerebrum.
The vertebral arteries merge to form the basilar artery. Branches of the basilar and vertebral arteries provide blood to the cerebellum and the brainstem via the circle of Willis which is composed of three pairs of main arteries: anterior, middle, and posterior cerebral arteries (see Fig.(1.11)). These arteries branch out into multiple smaller arteries that spread over the surface of the brain in the connective tissue derived from the pia mater and the arachnoid mater (see Fig.(1.2)). Thus, enabling multiple brain tissues to be supplied with the essential elements and nutrients to maintain their functions.

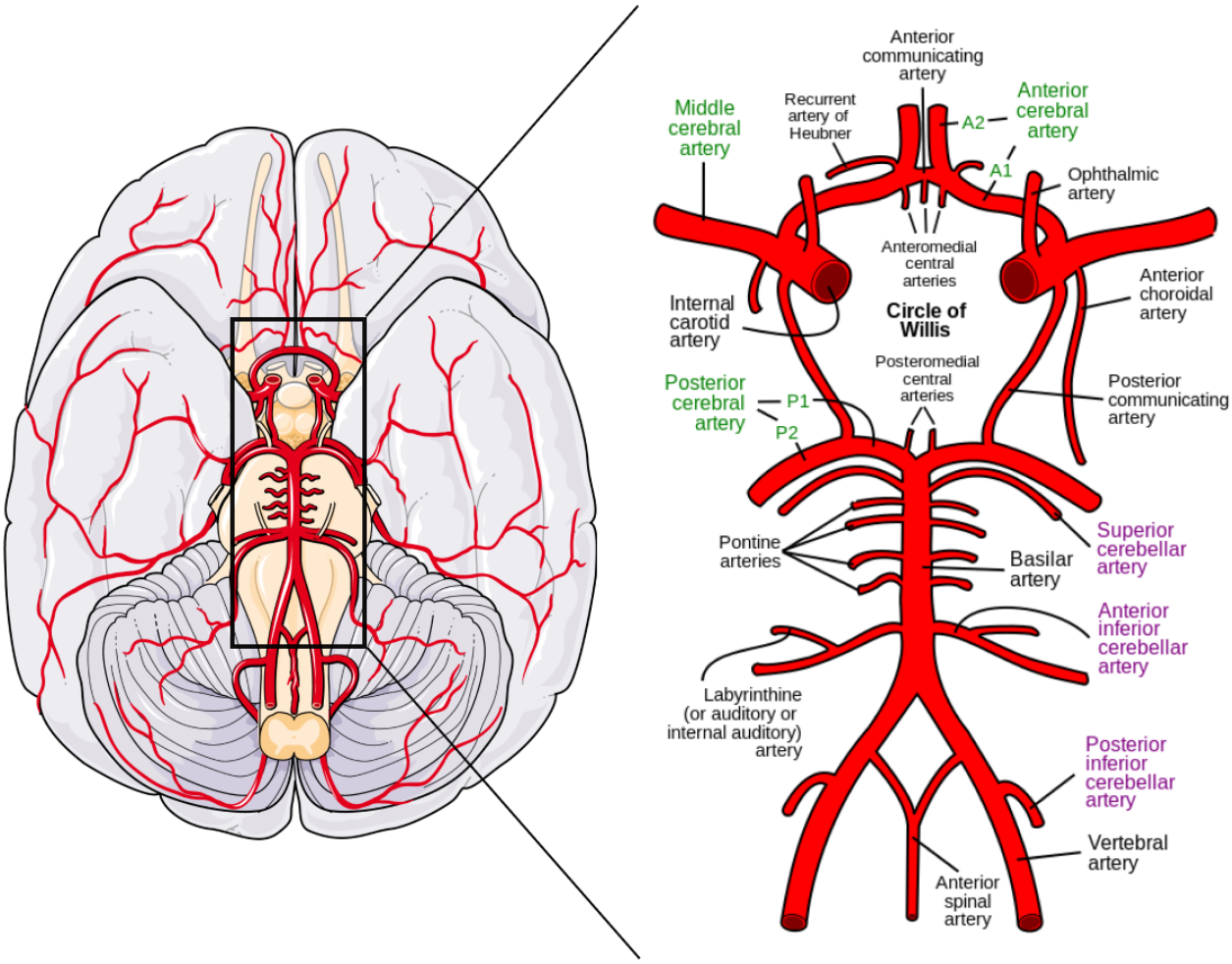

Figure 1.11: *A macroscopic schematic of the brain vascular system (adapted from [Wikipedia contributors, 2024c, Servier Medical Art, 2024]).*

1.2 Micro and Mesoscopic anatomy

Although the microscopic and macroscopic views of the human brain were separated in this work, these two are tightly related. In fact, one helps to inform the other to build a complete picture of the human brain. For instance, regions with similar cellular properties can be clustered together. In addition, macroscopic brain tissues can be further subdivided using cellular features. For instance, the subdivision of the cortex into the neocortex and allocortex is based on microscopical cellular features. This section presents the complex microscopic components of the human brain: the main cells present in the brain.

1.2.1 Neurons

Previously, we mentioned the network analogy of the brain, i.e., the human brain can be viewed as a large networked system. This picture will become clear in this section which presents the most important cell in the brain.

The ***neuron*** is the elementary unit of the brain and the nervous system. Just as matter is made of atoms, live beings are made up of cells. These cells are specialized and differ in shape, size,

and function. Neurons are at the core of information processing in the brain. They are able to communicate through electrical impulses traveling along their membranes.

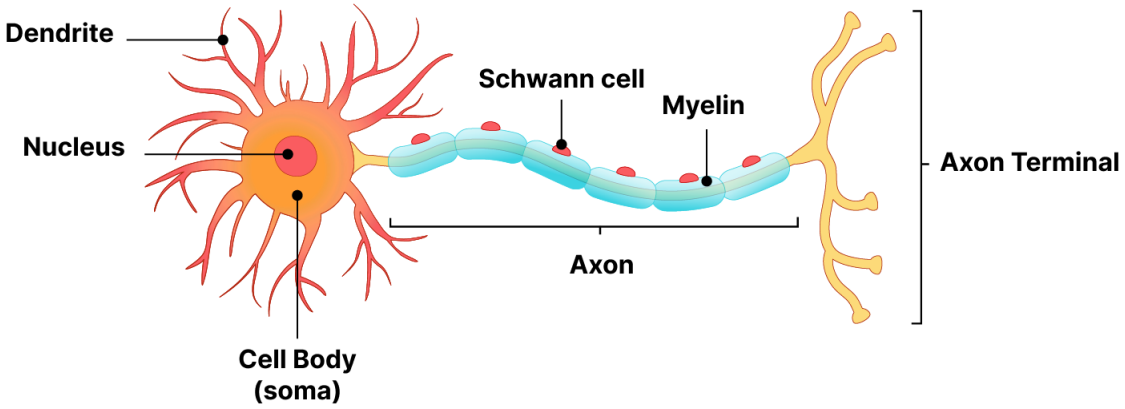

Figure 1.12: *Illustration of the neuron and its main constituents (Image adapted from [mha, 2024]).*

The neuron is made of different components: the soma, the axon, the dendrites, and the axon terminals (see Fig.(1.12)).

The soma. Also called the cell body, it is the main part of the neuron. It contains the main elements found in a regular cell, i.e., a nucleus, mitochondria for energy production, cytoplasm, and other elements. The soma has an ovoid-like shape with an average effective diameter varying between 3 and 40 µm [Palombo *et al.*, 2021].

The dendrites. They form an extension of the soma with a "tree-like" structure. They form the communication endpoints between neurons. They receive and process signals from other neurons and transmit them to the soma. Dendrites form thin branches with increasing thickness towards the soma (see Fig.(1.12)). Both the soma and dendrites are highly concentrated in the cortex, consequently leading to a darker apparent color for the gray matter.

The axons. They play a similar role to connections in a networked system. They are nerve fibers extending from the soma as illustrated in Fig.(1.12). The axons are vectors for nerve impulse transmission: they conduct action potentials, i.e., nerve impulse signals, from one endpoint to the other. Axons can exist in different forms: myelinated, partly myelinated, and completely unmyelinated. Myelinated axons are simply axons that are covered with a fatty substance named myelin. It is a lipid-rich sheath that wraps the axon multiple times. Myelin plays an important role in nerve impulse propagation's speed and efficiency. It is organized as a set of sheaths covering the axon with regular gaps: small regions that are not covered with myelin. These regions are called Ranvier nodes and are regularly separated by a distance of around 1mm [Allman, 1999]. Thus, when action potentials are traveling a myelinated axon, they "jump" from one node to its neighbor. This process is known as saltatory conduction. The signal conduction speed in axons depends on different factors [Waxman, 1980]. First, the radius of an axon, the conduction speed in large axons is greater than in small ones. Second, the thickness of the myelin sheath, the thicker

it is, the faster the signal's velocity. Also, the conduction speed in myelinated axons seems to be a linear function of the axon's diameter, while for unmyelinated axons, the conduction profile as a function of the diameter exhibits a parabolic curve. Finally, the internode distance, i.e., the distance separating Ranvier nodes, also affects the propagation velocity of action potentials. An important metric used to describe axons is the g-ratio or g-factor, defined as $g = \frac{d}{D}$ as in Fig.(1.13) where d is the diameter of the unmyelinated axon and D is the diameter of the same axon including the myelin sheath.

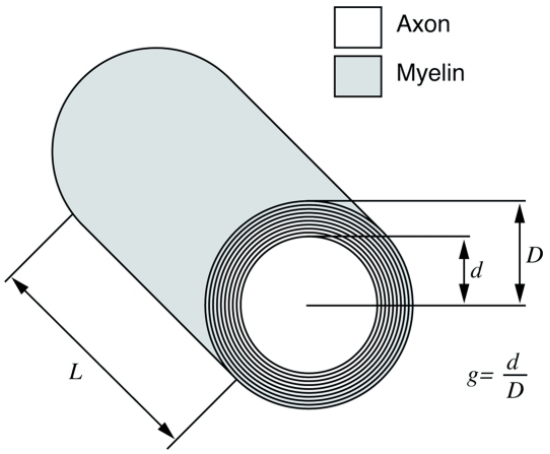

Figure 1.13: *Schematic illustration of a segment of an axon with the g-ratio metric (Image taken from [Paus & Toro, 2009]).*

An optimal conduction speed for action potentials is reached for a g-ratio of about 77% [Chomiak & Hu, 2009]. Axons in the human brain have different diameters ranging from very small ones of around 0.16µm to large ones reaching 9µm [Liewald *et al.*, 2014]. A thorough investigation of the axonal diameter distribution in the human brain was done in the same work. The axonal distribution in the uncinate fascicle is depicted in Fig.(1.14). The depicted diameter distributions seem to be unimodal with a high frequency of axon with a diameter of around 0.5µm. The empirical diameter distribution can be approximated by different analytical formulas [Sepehrband *et al.*, 2016]. For instance, the generalized extreme value (GEV) distribution provides the best fit. There are also other distributions that provide decent approximations to the empirical radii, such as the log-normal and the Gamma distributions.

Although schematized by a perfect cylinder in Fig.(1.13), the actual cross sections of axons display a slightly different shape than a perfect circle. In fact, some works investigated white matter cross-sections through electron microscopy micrographs. The pictures depicted in Fig. (1.15) reveal interesting features of white matter. For relatively large axons, it is indeed possible to distinguish the multiple layers of myelin wrapping the axon.

Axon terminal. This is the last component of the neuron. it is also the endpoint reached by action potentials. The axon terminal contains different branches as depicted in Fig.(1.12). This allows the neuron to communicate with different parties. A synapse is located at the end of each branch such that when an action potential reaches the synapse, chemical neurotransmitters are released. Once released, the neurotransmitters bind to the ion channels of a close cell, i.e., a dendrite of another neuron, consequently generating a propagating action potential in the dendrite. The axon terminal connections are not restricted only to soma and dendrites, they can

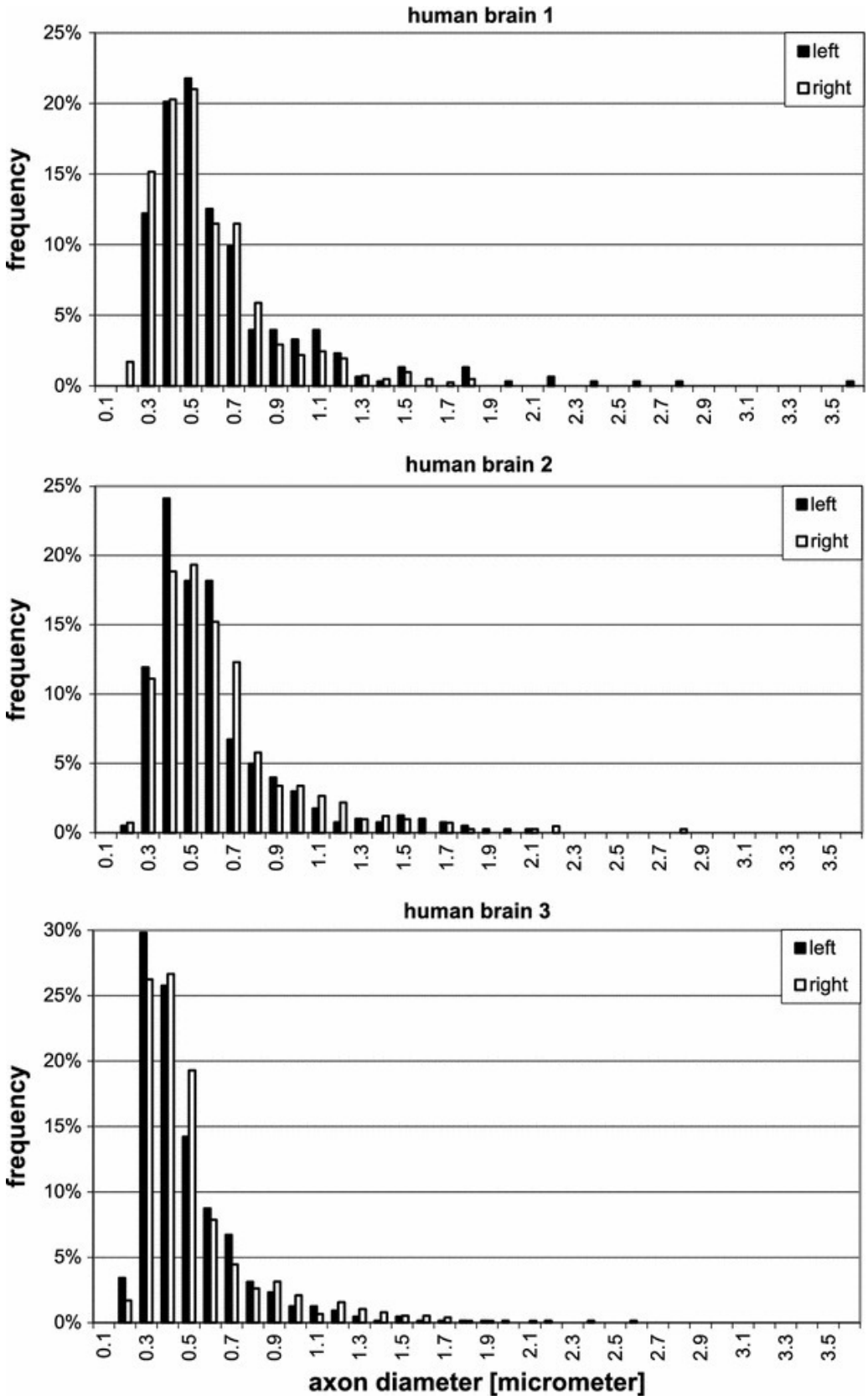

Figure 1.14: *Diameter distribution of axons in the uncinate fascicle from three different human brain samples (Image taken from [Liewald et al., 2014]).*

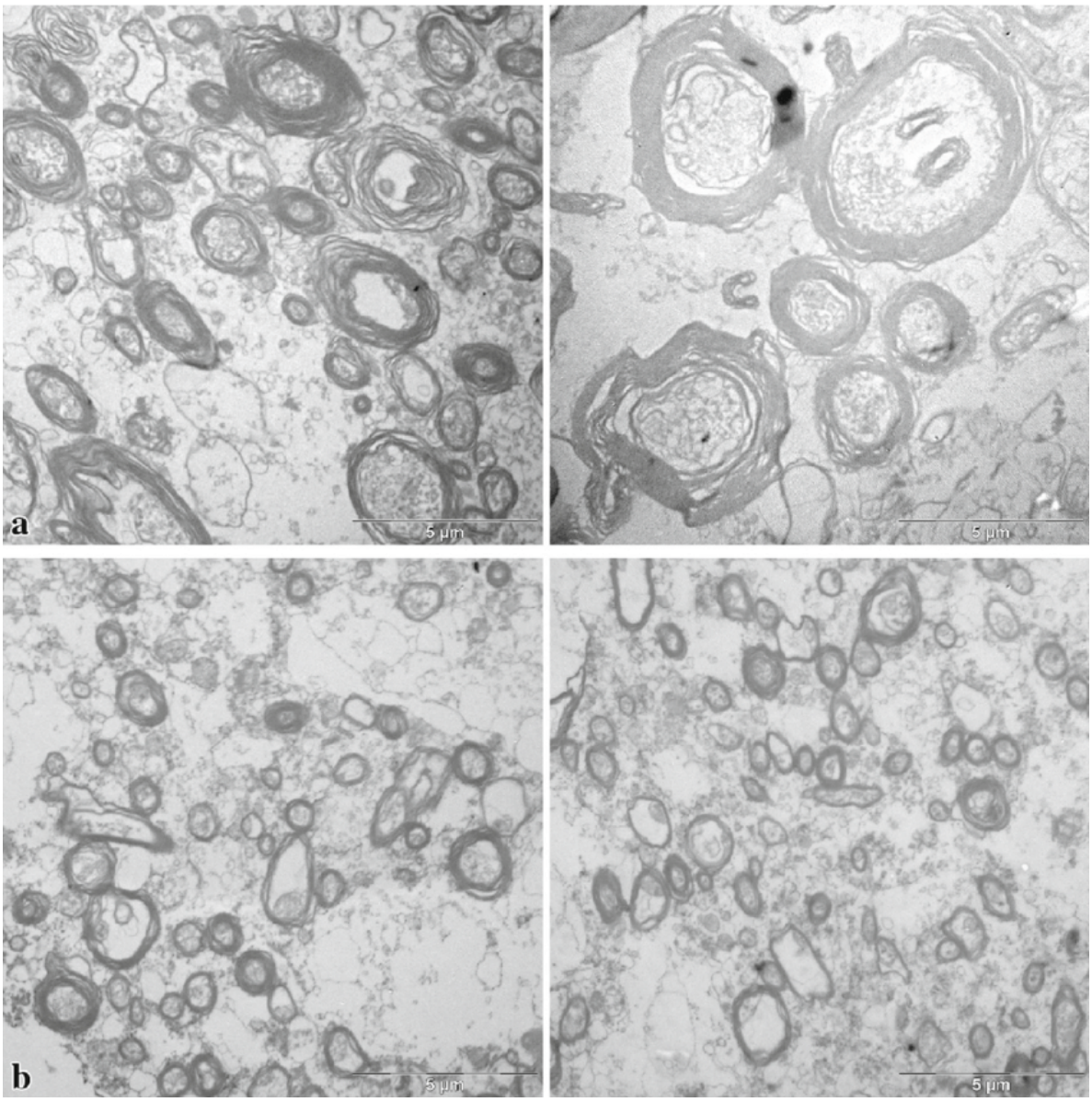

Figure 1.15: *Illustration of electron micrographs from two different regions in the human brain white matter. Panel a) is from the superior longitudinal fascicle, and panel b) is from the uncinate/inferior occipitofrontal fascicle (Image taken from [Liewald et al., 2014]).*

also bind into axons forming axo-axonic synapses [Cover & Mathur, 2020]. This forms the base of communication between neurons.

As mentioned previously, the human brain contains a large amount of neurons. Thus, an important question arises: how to make sense of all this complexity? In other terms, is it possible to simplify further and subdivide this large volume of cells into subsets that share common features? The task of clustering neurons is very challenging and still an active field of research [Yuste *et al.*, 2020]. There are different criteria by which neurons can be classified [Zeng & Sanes, 2017]. First is ***morphology***, which is one of the most used criteria. It allows the grouping of neurons by their geometrical features and shape. The second criterion is ***physiology***, which consists of grouping neurons by their electrophysiological responses to a square pulse. Thus, neurons that display similar excitation patterns when exposed to a square stimulus tend to be grouped together. A

third criterion is the ***molecular signature***, known also as transcriptomics. It relies on RNA-sequencing data of individual cells. This allows building a gene expression profile which is used to cluster neurons. Finally, a last criterion that is less frequently used is ***connectivity***. Although these classification criteria are widely used, they are not the only way to group neurons. In fact, neurons can also be grouped by neuron organization at the mesoscopic scale; this is known as cytoarchitecture. They can be grouped by function, or according to the neuronal circuit to which they belongs.

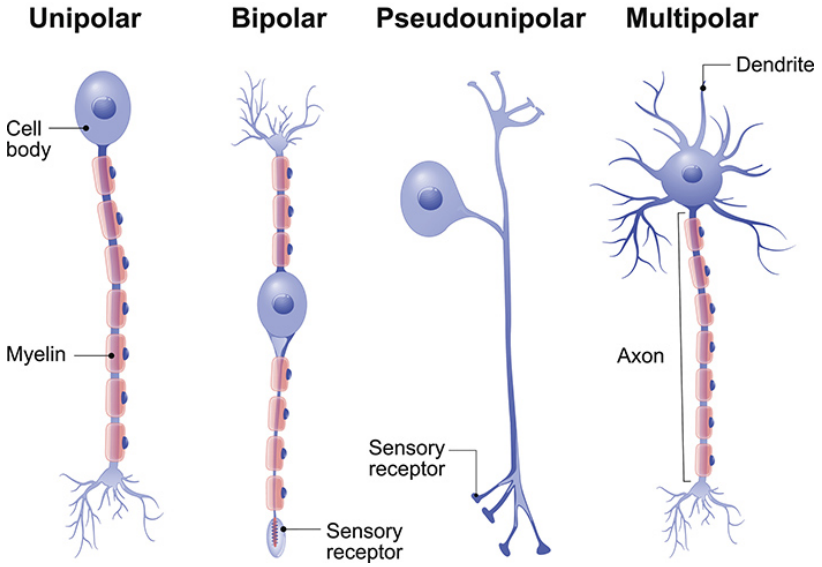

Figure 1.16: *A rough morphological subdivision of neurons. (Image taken from [typ, n.d.]).*

At a high level, neurons can be classified into two groups, excitatory, i.e., glutamatergic, and inhibitory, i.e., GABAergic neurons. Neurons within each group are further classified into sub-types. Excitatory neurons include different types such as star pyramidal cells, spiny stellate cells, and other types of neurons. In contrast, inhibitory neurons include different sub-types such as basket, chandelier, long-projecting, neurogliaform, and single bouquet cells. In addition, morphologically, neurons can be roughly subdivided into four different types: unipolar, bipolar, pseudo-unipolar, and finally, multipolar (see Fig.(1.16)). Finally, some neuron types have been named according to their soma shape such as pyramidal neurons, and according to the person who discovered them such as Purkinje or Martinotti cells.

1.2.2 Glial cells

The name glial cells originates from the Greek word glia, which means glue. There are about as many of them as neurons in the human brain. Although still a debated subject, previous investigations reported a glial cells to neurons ratio of 10:1 while more recent studies report a ratio around 1:1 [von Bartheld *et al.*, 2016]. Glial cells play important roles in the brain and can be subdivided into four different groups: astrocytes, oligodendrocytes, microglia, and oligodendrocytes progenitor cells.

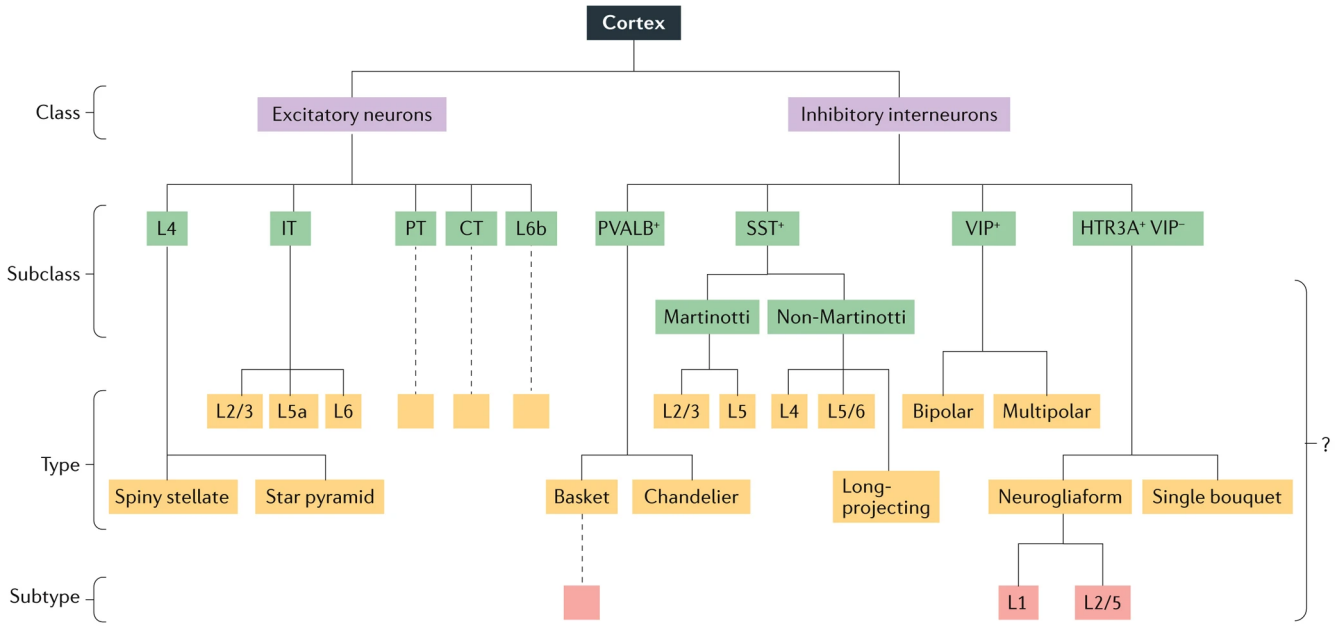

Figure 1.17: *Illustration of a hierarchical classification of neurons in the human cerebral cortex. The question mark indicates that the status of the cortical cell groups indicated may be either subclasses, types, or subtypes. (Image adapted from [Zeng & Sanes, 2017]).*

Astrocytes. Of all the glial cell types, astrocytes are the most frequent type of glial cells present in the adult human brain. Named "Astro" after its star-like shape, the astrocytes play important roles in the brain. First, they maintain water ion homeostasis in the brain, for example, maintaining controlled levels of K^+ concentration in the extracellular space and controlling its acidity, i.e., the concentration of H^+. Second, the astrocyte processes surround a large proportion of blood vessels and help control blood flow. Finally, they also help form and maintain the brain-blood barrier (BBB) [Lundgaard *et al.*, 2014].
Astrocytes can be subdivided into multiple classes [Vasile *et al.*, 2017] summarized as illustrated in Fig.(1.18). There are interlaminar astrocytes, protoplasmic astrocytes, varicose projection astrocytes, and finally, fibrous astrocytes. Each class is predominantly present in some cortical layers, for example, interlaminar astrocytes exist in layer I of the cortex, while varicose projection astrocytes are in layers V and VI.

Oligodendrocytes. Literally meaning few trees, i.e., cells with a small amount of processes. Oligodendrocytes are located in both white and gray matter and play important roles in the human brain. They are involved in the creation of myelin in the CNS and in providing metabolic support to axons.

Microglia. Accounting for about 5% to 15% total brain cells, microglia are the main neuroimmune cells in the CNS [Zhang *et al.*, 2023]. They play important roles in both healthy and pathological brain tissues. For instance, microglia are involved in synaptic pruning: a process of selective removal of synaptic connections. They also play an important surveillance role. In

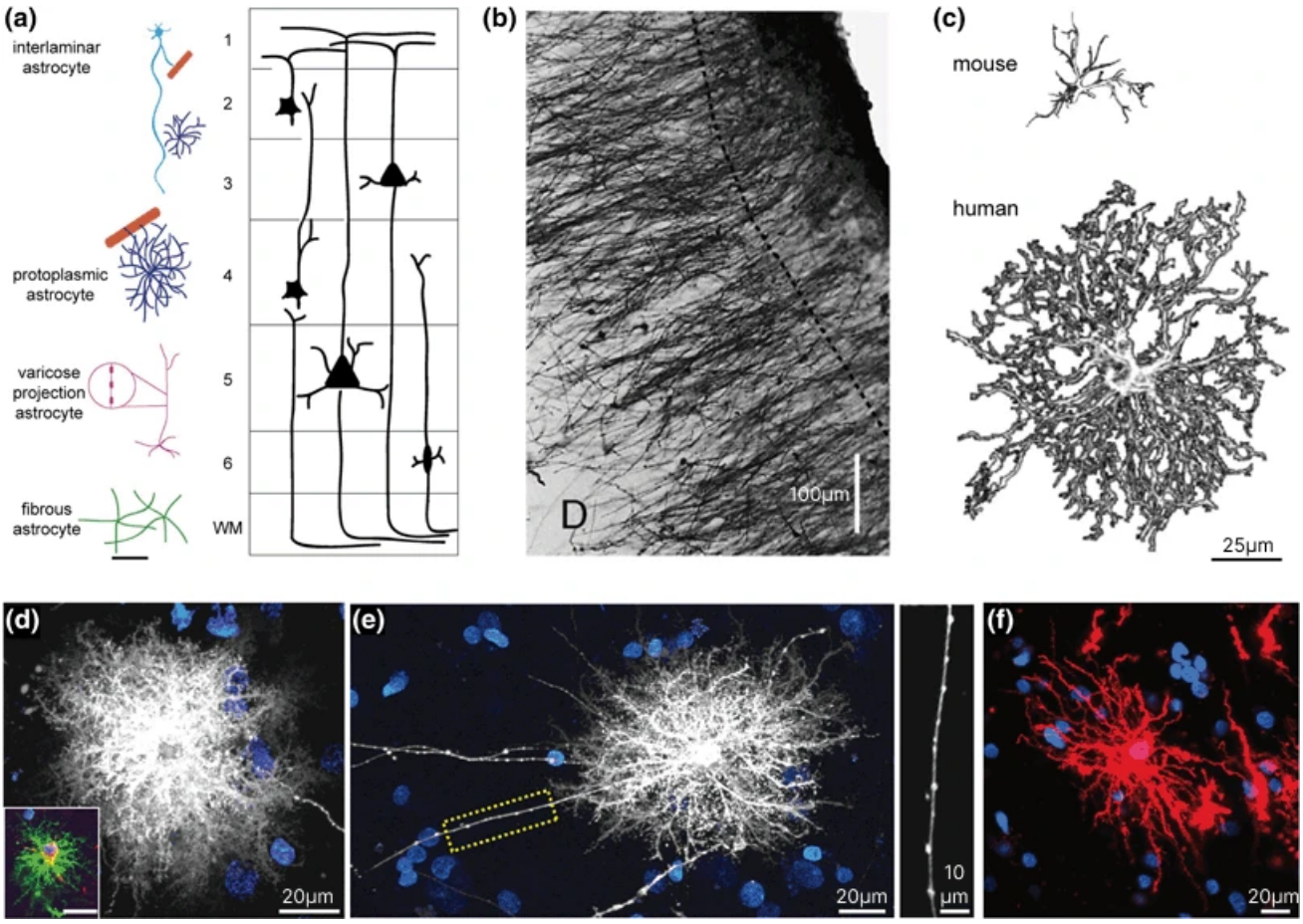

Figure 1.18: *Illustration of different classes of astrocytes in the human brain. a) Location of the different types of astrocytes within different cortical layers. b) Interlaminar astrocytes in the human occipital cortex. c) Representation of mouse and human cortical protoplasmic astrocytes. d) Human protoplasmic astrocyte. e) Varicose projection astrocyte with its process in. f) Human fibrous astrocyte. (Image adapted from [Vasile et al., 2017]).*

fact, microglia are connected to different synapses which enables to survey them and act accordingly [Wu *et al.*, 2015]. Another important property of microglia is their repopulation capacity [Jäkel & Dimou, 2017]. Ablation studies, i.e., studies that investigate the role of a cell population by removing it and observing the tissue changes, have revealed that it takes around one week for a destroyed population of microglia to regenerate. Finally, microglia are also involved in pathological brain tissues, they can change their state from "resting" to "activated". When activated, microglia can migrate, proliferate, and digest damaged tissue in a process known as phagocytosis.

Olygodendrocyte progenitor cells (OPCs). They also form a subgroup of the glial cells. OPCs are also known by different names such as oligodendrocyte precursor cells, or NG-2 glia which is an abbreviation of neural glial antigen 2. The OPCs play an important role in creating oligodendrocytes which myelinate axons. OPCs and some of their functions are still an active area of research [Jäkel & Dimou, 2017].

1.2.3 Cortex cytoarchitecture and myeloarchitecture

Up until now, both macroscopic and microscopic descriptions of the brain were presented. However, covering the brain at both scales is not enough to understand it completely. An important missing piece of information is how the brain cells, i.e., neurons, organize themselves spatially. The investigation of the brain's cellular organization is known as cytoarchitectonics, and when the focus is on the cerebral cortex, this is known as the cortex cytoarchitecture. The study of cortex cytoarchitecture requires the observation of the brain at the mesoscopic scale: an intermediary scale between the micro and macroscopic scales. For this purpose, the use of optical microscopy to image brain tissue samples provides insightful information on the cortex cytoarchitecture.
The observation of the brain at the mesoscopic scale reveals the interesting laminar structure of the cerebral cortex. Such a structure is organized into six different layers as illustrated in Fig.(1.19). These numbered layers are such that the first layer is located at the outermost part of the cerebral cortex and the last one is directly interfacing with the white matter. The differentiating factor between the cortical layers is their cellular composition: the combination of different cell types, their cell body shapes, and their corresponding densities.
The different cortical layers are listed below [Fatterpekar *et al.*, 2002]:

- **Layer I.** Also called the molecular layer, it mainly contains axons, axon terminals, and dendrites from neurons in deeper layers. The axons in the first layer tend to cluster horizontally (parallel to the cortical surface).
- **The external granular layer (layer II).** Located just under layer I, layer II contains tightly packed pyramidal-shaped cells and interneurons. Their axons project to other layers.
- **The external pyramidal layer (layer III).** It contains mostly pyramidal cells of various sizes with other scattered non-pyramidal cells. The differentiation between layer II and layer III can be difficult in some histological sections (see Fig.(1.19)).
- **The internal granular layer (layer IV).** In many cases, it is the narrowest cortical layer. It contains tightly packed spiny and aspiny stellate cells as well as a variety of granule cells. It is wide in primary sensor areas.

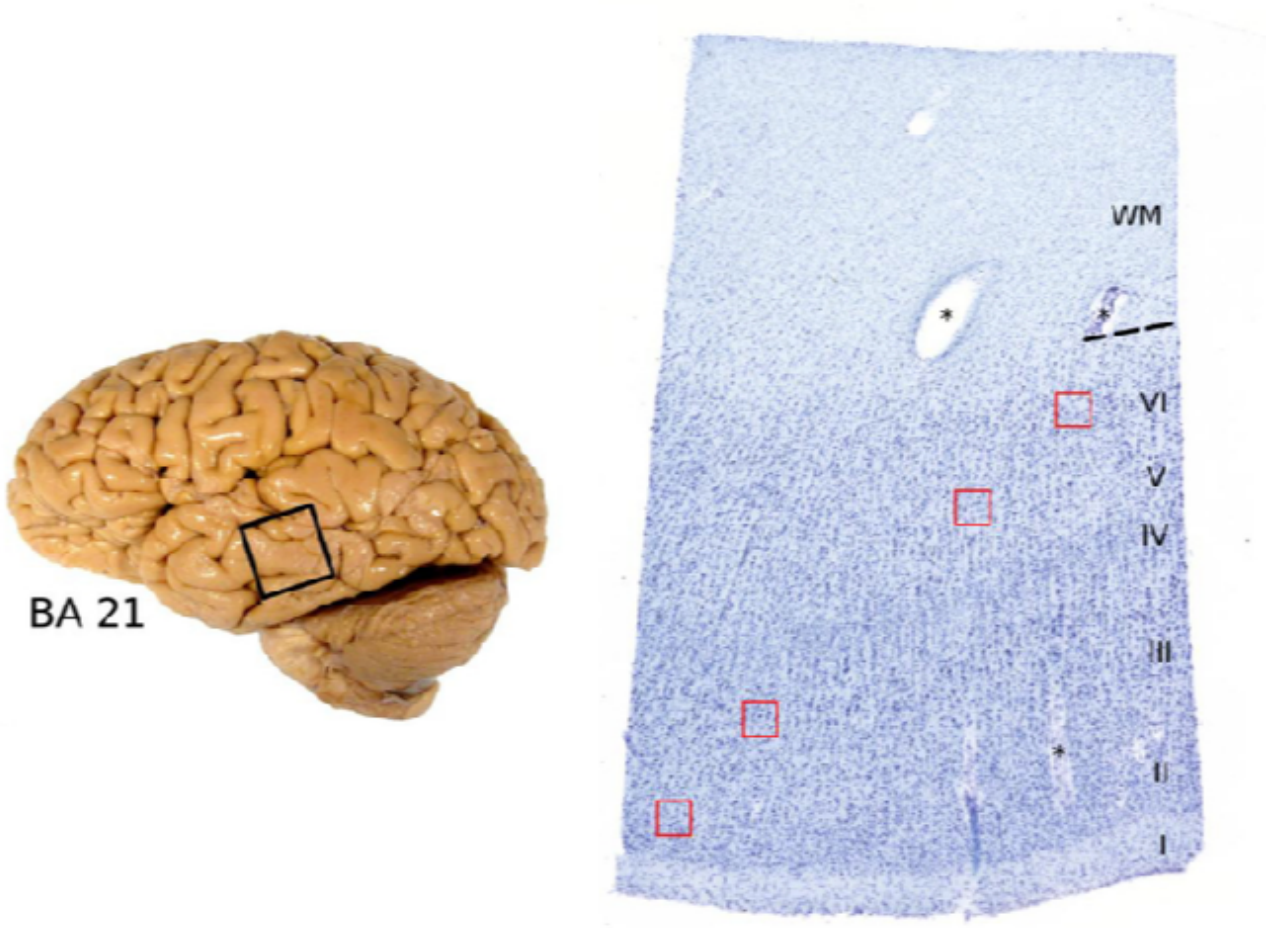

Figure 1.19: *Illustration of a digitized Nissl stained histology section in the right panel. The histology section is obtained from a brain sample in the central temporal region shown in the left panel (Image adapted from [Magnain et al., 2019]).*

- **The internal pyramidal cell layer (layer V).** Contains the large pyramidal cells, typically larger than those found in layer III, it contains as well as other scattered non-pyramidal cells. The large neurons in this layer project major axonal pathways to other cortical regions and subcortical structures.

- **The multiform layer (layer VI).** Also known as the Fusiform or the Pleomorphic layer, it is relatively thin and contains tightly packed neurons with heterogeneous shapes.

All the cortical layers as well as the myeloarchitecture, i.e., the organization of axons in the brain at the mesoscopic scale, are summarized in Fig.(1.20).
The investigation of the cortex cytoarchitecture reveals the interesting laminar structure of the cerebral cortex. While this type of cellular organization is done in the depth dimension, another organization is found exploring the cortical surface. Actually, when observing the cellular organization at the cortical surface, it is possible to distinguish different areas. One of the founding works in this direction was Brodmann's cytoarchitectonics parcellation of the cortex. In his work [Brodmann, 1909], Brodmann distinguished 44 different areas of the brain that group cells sharing similar geometrical features such as cell shape, size, and density.

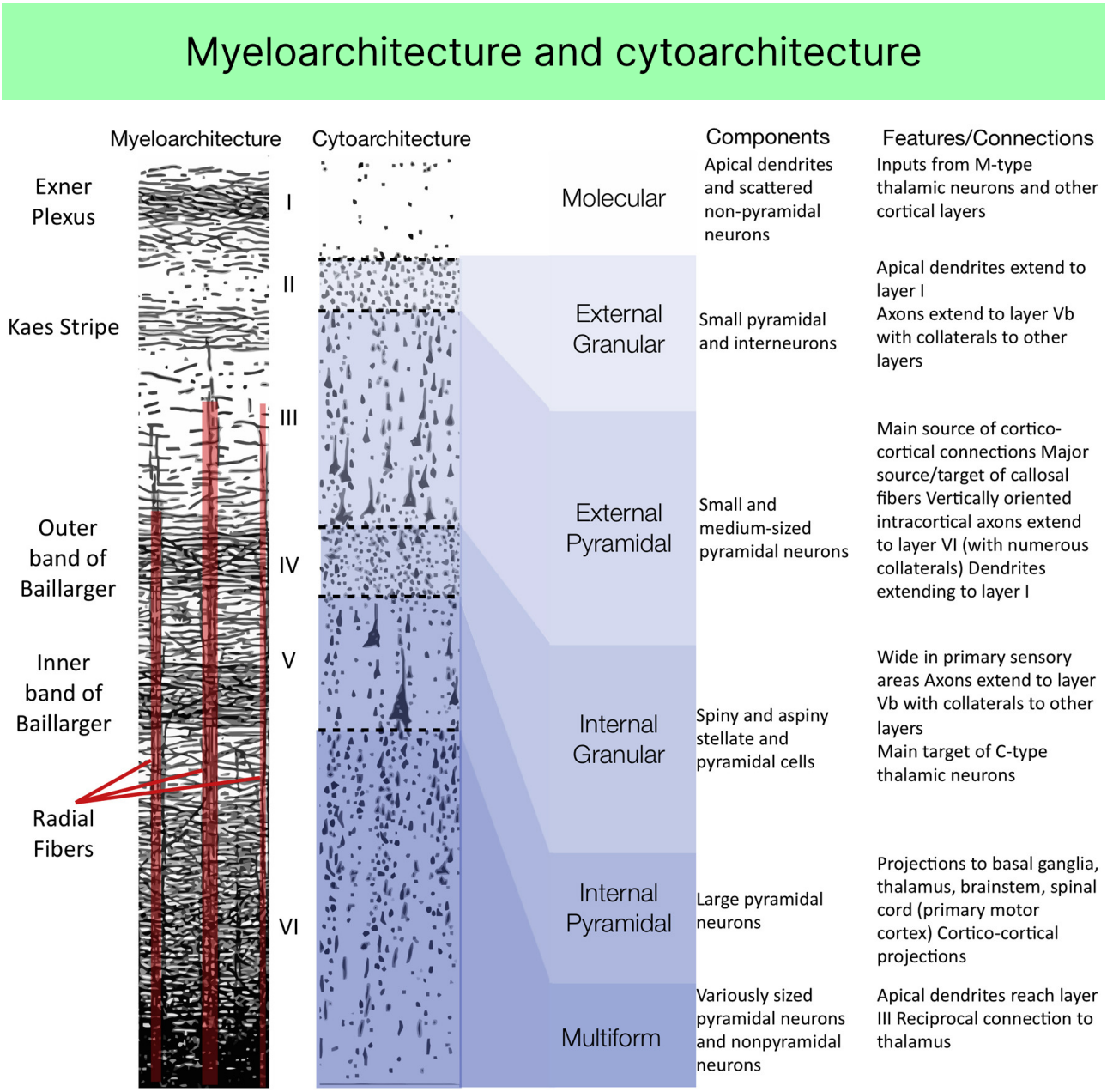

Figure 1.20: *Illustration of the myeloarchitecture and the cytoarchitecture of the cerebral cortex inferred from the Nissl staining method. (Image adapted from [Nazeri et al., 2020]).*

The same illustration depicted in Fig.(1.21) shows interesting aspects of the myeloarchitecture changing with different cortical layers. The deeper layer VI comprise long radial fibers extending to higher cortical layers. In addition, layers V, IV, III, II, and I contain lateral fibers that are perpendicular to the radial fibers. The lateral fibers are characterized by varying densities across the cortical layers. The myeloarchitecture was also thoroughly investigated in the studies of Vogt and Vogt [Vogt, 1910].

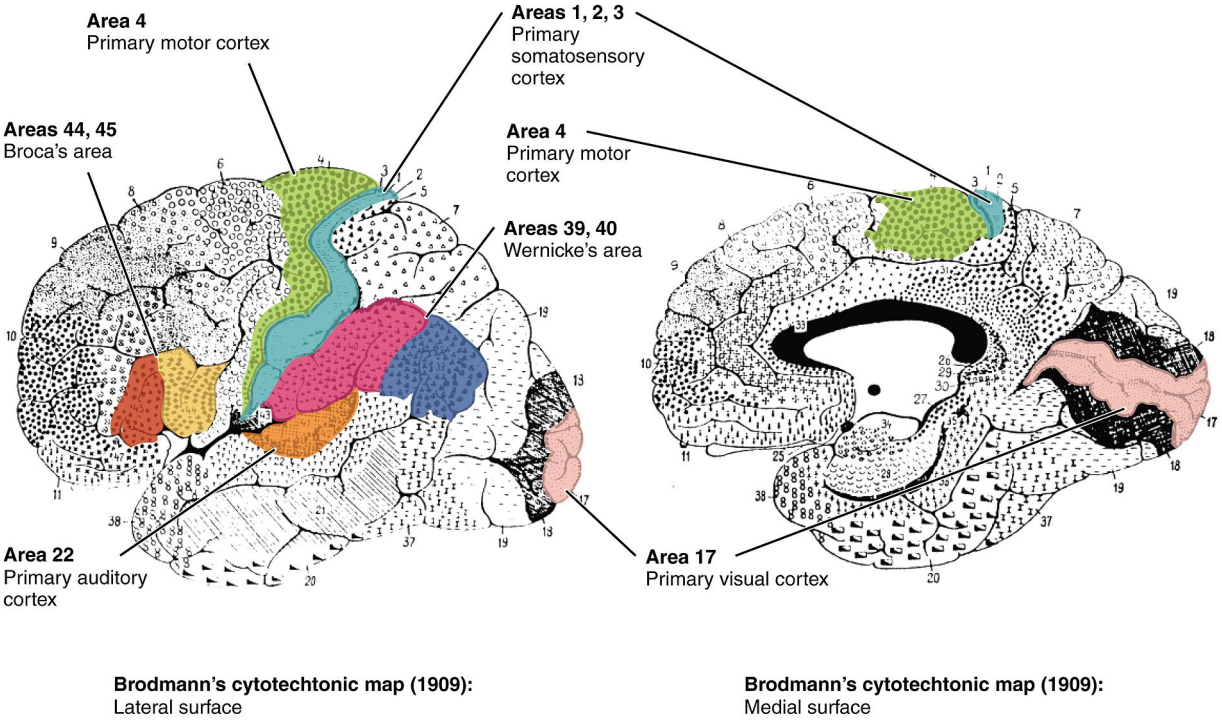

Figure 1.21: *Illustration of the Brodmann cytoarchitectonics map of the human brain with different areas and their function. (Image taken from [Wikipedia contributors, 2024b]).*

1.3 Conclusion

This chapter covered the essentials of the brain's anatomy from the macroscale to the microscopic scale. The diversity and complexity of neurons were briefly presented. This point will become more apparent in the next sections as this topic is essential in the course of this thesis. The next part is dedicated to the second pillar of the thesis which is the principles of MRI and the diffusion MRI modality.

Chapter 2

MRI Principles

MRI essentials are presented in this chapter from the first principles in physics. Different essential notions are presented such as the phenomenon of Nuclear Magnetic Resonance (NMR), free induction decay (FID), and image reconstruction. The second part of this chapter is dedicated to the modality relevant to the developments of this thesis, it covers the principles of the diffusion MRI modality and some of the common MRI sequences used in medical imaging.

2.1 Magnetic Resonance physics

Nuclear magnetic resonance (NMR) is a phenomenon that occurs when the nuclei of atoms with non-zero spins are immersed in two magnetic fields: one that is static and sufficiently strong and a second, non-static, oscillating magnetic field that is perpendicular to the static one. The NMR was first reported in the work of Rabi [Rabi *et al.*, 1938] and then in the works of Bloch [Bloch *et al.*, 1946] and Purcell [Purcell *et al.*, 1946]. All these works were awarded Nobel prizes, one for Isaac Rabi in 1994 and another awarded jointly to Felix Bloch and Edward Mills Purcell in 1952[1].

2.1.1 Spins

Spins. To dive deep into the NMR, it is necessary to present spins and their interaction with magnetic fields. In quantum physics, a spin is a quantum-mechanical property that represents the intrinsic angular momentum. Some atoms have no intrinsic spin, i.e., carbon ^{12}C, while other nuclei do possess a spin, i.e., hydrogen ^{1}H, fluorine ^{19}F, carbon ^{13}C, and other elements. Generally, when both numbers of protons and neutrons are even, the underlying atom nuclei have no spin. Nuclei made of an odd number of nucleons have a $\frac{1}{2}$ Spin. Finally, when both numbers of protons and neutrons are odd, the spin value is 1. A spin $\mathbf{S}$ is characterized with an angular momentum $\boldsymbol{\mu}$ since the nucleus can be classically viewed as a rotating charged particle.

$$\boldsymbol{\mu} = \gamma \mathbf{S}, \tag{2.1}$$

where γ in equation Eq.(2.1) is the gyromagnetic ratio, it is expressed in rad.s^{-1}.T^{-1}, and its value differs from one nucleus to another.

[1] https://www.nobelprize.org/prizes/physics/1952/summary/

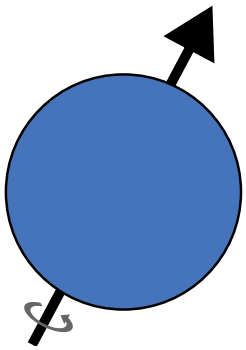

Figure 2.1: *Illustration of the spin of a nucleus as a spinning particle.*

When exposed to an external static magnetic field $\mathbf{B_0}$, spins tend to align themselves with the static field and follow a precession movement about the field $\mathbf{B_0}$ with a frequency known as the Larmor frequency f_0.

$$f_0 = \frac{\gamma}{2\pi} B_0 \tag{2.2}$$

The precession movement is the result of the interaction of the magnetic moment $\boldsymbol{\mu}$ with the static field $\mathbf{B_0}$.

$$\frac{d\mathbf{S}}{dt} = \boldsymbol{\mu} \times \mathbf{B_0} \tag{2.3}$$

The equation Eq.(2.3) can be understood classically as the change in "angular momentum", i.e., the left-term. And the right term is equal to the torque.
Describing single spins and their interactions with the magnetic field is useful from a theoretical standpoint. However, in practice, one needs to relate to measurable quantities. When considering a piece of material or biological tissue sample of volume V, it is practical to regard the magnetization vector $\mathbf{M}$, which results from the contribution of all the magnetic moments within the sample.

$$\mathbf{M} = \frac{1}{V} \sum_i \boldsymbol{\mu_i} \tag{2.4}$$

The illustration in Fig.(2.2) shows a sample with different spins both in the absence and presence of an external static magnetic field $\mathbf{B_0}$. As illustrated, in the absence of a static field, spins tend to take random orientations such that the net resulting magnetization averages zero. In contrast, when a static field $\mathbf{B_0}$ is present, spins tend to align themselves following parallel or anti-parallel directions depending on the state of the spin, i.e., spin-up or spin-down. The co-existence of both spin states in different proportions results in a non-zero net magnetization $\mathbf{M}$. In practice, this net magnetization is difficult to measure as it is orders of magnitude lower than the static $\mathbf{B_0}$ field. In order to measure it, a "trick" has to be used to flip the spins in an orientation perpendicular to $\mathbf{B_0}$. This is done by introducing a second magnetic field $\mathbf{B_1}$ perpendicular to $\mathbf{B_0}$ and that is oscillating at the Larmor frequency f_0 introduced in Eq.(2.2), thus resulting in the NMR phenomenon.
When $\mathbf{B_1}$ is applied, the net magnetization $\mathbf{M}$ starts heading to the transverse plane of the $\mathbf{B_0}$ field while maintaining the precession movement. The combination of these two motions gives rise to the nutation motion illustrated in Fig.(2.3).

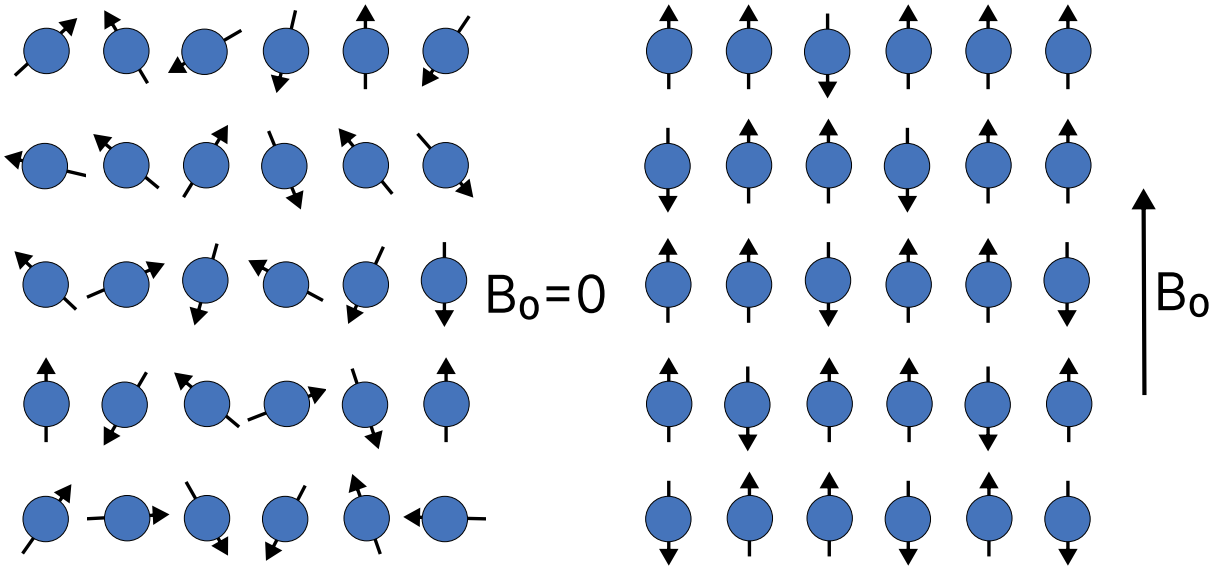

Figure 2.2: *Illustration of different spins in two cases. The left one shows the random orientation of spins in the absence of an external magnetic field* $\mathbf{B_0}$*. The right one shows an average alignment of the spins with* $\mathbf{B_0}$*.*

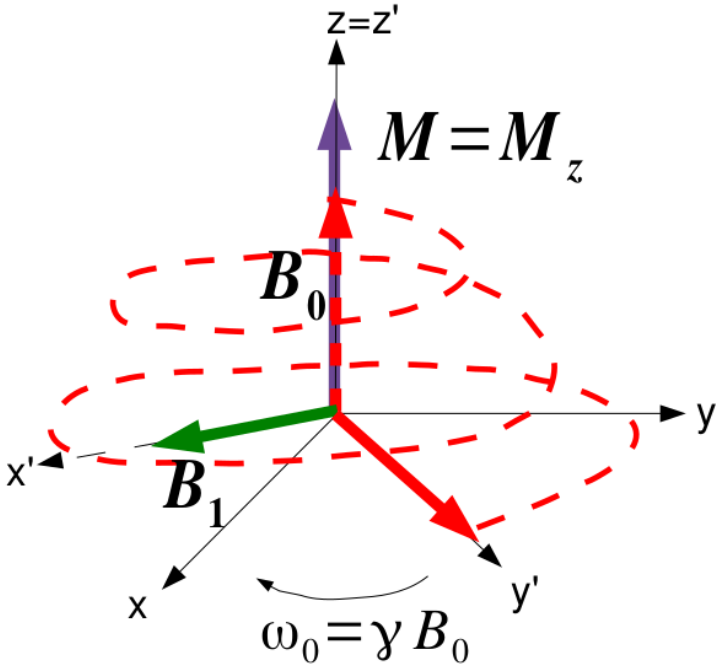

Figure 2.3: *Illustration of the nutation of the magnetization* $\boldsymbol{M}$ *in the presence of the oscillating field* $\mathbf{B_1}$ *at the Larmor angular frequency* ω_0 *(Courtesy of Cyril Poupon).*

2.1.2 Bloch equations and relaxation

To unlock the full dynamics of the magnetization in the presence of the $\mathbf{B_0}$ and $\mathbf{B_1}$ fields, we rely on the phenomenological Bloch-equations [Bloch, 1946], which characterize the temporal dynamics of the magnetization in the presence of an external magnetic field.

$$\begin{cases} \frac{dM_x}{dt} = \gamma(\mathbf{M(t)} \times \mathbf{B(t)})_x - \frac{M_x(t)}{T_2} \\ \frac{dM_y}{dt} = \gamma(\mathbf{M(t)} \times \mathbf{B(t)})_y - \frac{M_y(t)}{T_2} \\ \frac{dM_z}{dt} = \gamma(\mathbf{M(t)} \times \mathbf{B(t)})_z - \frac{M_z(t)-M_0}{T_1}, \end{cases} \tag{2.5}$$

with M_0 being the steady state nuclear magnetization and T_1 and T_2 being, respectively, time scale characteristics of the recovery and vanishing of the longitudinal and transverse magnetization components. Solving the Bloch equation in Eq.(2.5) allows the retrieval of the temporal magnetization in the **excitation** phase, i.e., both $\mathbf{B_0}$ and $\mathbf{B_1}$ are applied, and in the **relaxation** phase, i.e., when the $\mathbf{B_1}$ field is no longer applied. For this purpose, the magnetization $\mathbf{M}$ is decomposed into transverse and longitudinal components:

$$\begin{cases} \mathbf{M} = \mathbf{M_L} + \mathbf{M_T} \\ \mathbf{M_L} = M_z\hat{\mathbf{z}} \\ \mathbf{M_T} = M_x\hat{\mathbf{x}} + M_y\hat{\mathbf{y}} \end{cases} \tag{2.6}$$

Also, the magnetic fields $\mathbf{B_0}$ and $\mathbf{B_1}$ are given by:

$$\begin{cases} \mathbf{B_0} = B_0\hat{\mathbf{z}} \\ \mathbf{B_1} = B_1(cos\omega t\hat{\mathbf{x}} - sin\omega t\hat{\mathbf{y}}) \end{cases} \tag{2.7}$$

The magnetization in Eq.(2.6) and the magnetic fields in Eq.(2.7) are plugged in the phenomenological Bloch equations in Eq.(2.5). When the $\mathbf{B_1}$ field is no longer applied and the magnetization is in the transverse plane, the longitudinal magnetization changes in time according to:

$$M_L = M_0\left(1 - e^{-\frac{t}{T_1}}\right) \tag{2.8}$$

The longitudinal magnetization in Eq.(2.8) recovers exponentially with a time scale T_1, which is the longitudinal relaxation time. It characterizes the time required for the longitudinal magnetization to recover 63% of its original value. Solving the Bloch equations also gives the relaxation for the transverse magnetization:

$$\begin{cases} M_x = M_0\sin(\omega t)\, e^{-\frac{t}{T_2}} \\ M_y = M_0\cos(\omega t)\, e^{-\frac{t}{T_2}} \end{cases} \tag{2.9}$$

Thus, the relaxation process combines two mechanisms: **longitudinal**, also called **spin-lattice relaxation**, characterized by the relaxation time T_1. The second is **transverse**, also called **spin-spin relaxation**, and it is characterized by the relaxation time T_2.

2.1.3 The spin-echo sequence

The previous section presented the magnetization relaxation process and its characteristic relaxation times. In practice, due to magnetic field inhomogeneities, the transverse magnetization relaxes faster than T_2 and is therefore characterized by a different time scale T_2^* such that:

$$\frac{1}{T_2^*} = \frac{1}{T_2} + \frac{1}{T_2^i}, \tag{2.10}$$

where T_2^i represents the delay due to local field inhomogeneities.
The observed NMR decaying signal with a T_2^* relaxation time and an angular frequency ω_0 corresponding to the Larmor frequency f_0 is called the Free Induction Decay (FID). It is the building foundation for acquiring MR images. Hahn made an important contribution to the field by introducing the phenomenon of spin echoes [Hahn, 1950]. He described an experiment where an NMR sample is subjected to two sequential short RF pulses. In such a setting, there is a constructive interference of moments which results in a measured signal referred to as the spin-echo. The principle of the spin echo is illustrated in Fig.(2.5). First, an NMR sample immersed in a static magnetic field B_0 is perturbed by a short magnetic 90° RF pulse. This flips the spins to the transverse plane relative to the static magnetic field. These spins start dephasing and after a certain time duration τ, the NMR sample is perturbed by a second short magnetic 180° RF pulse. The second pulse is referred to as the refocusing pulse, it makes the spin "re-phase" back to reach a constructive interference after the same time duration of τ during which the spin echo signal reaches its peak. The sequence of the two RF pulses described above is referred as **the spin echo sequence**. Using the principle of the spin echo, Hahn was able to provide accurate measurements of the relaxation times T_1 and T_2. This was performed by repeating the spin echo sequence using different values of the time duration τ which enabled the disentanglement of the relaxation times T_2^* and T_2 with and without local field inhomogeneities.

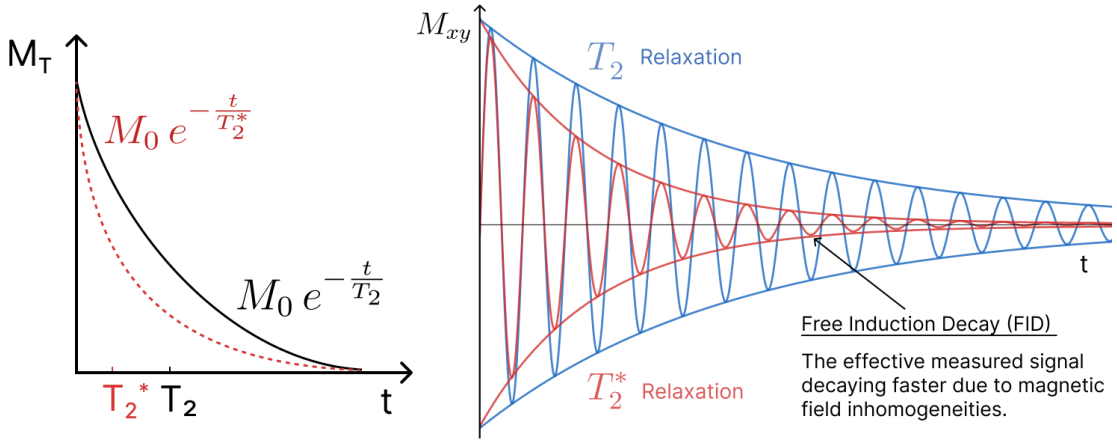

Figure 2.4: *Illustration of the transverse magnetization relaxation, i.e., spin-spin relaxation and the reduced time constant T_2^* due to local magnetic field inhomogeneities.*

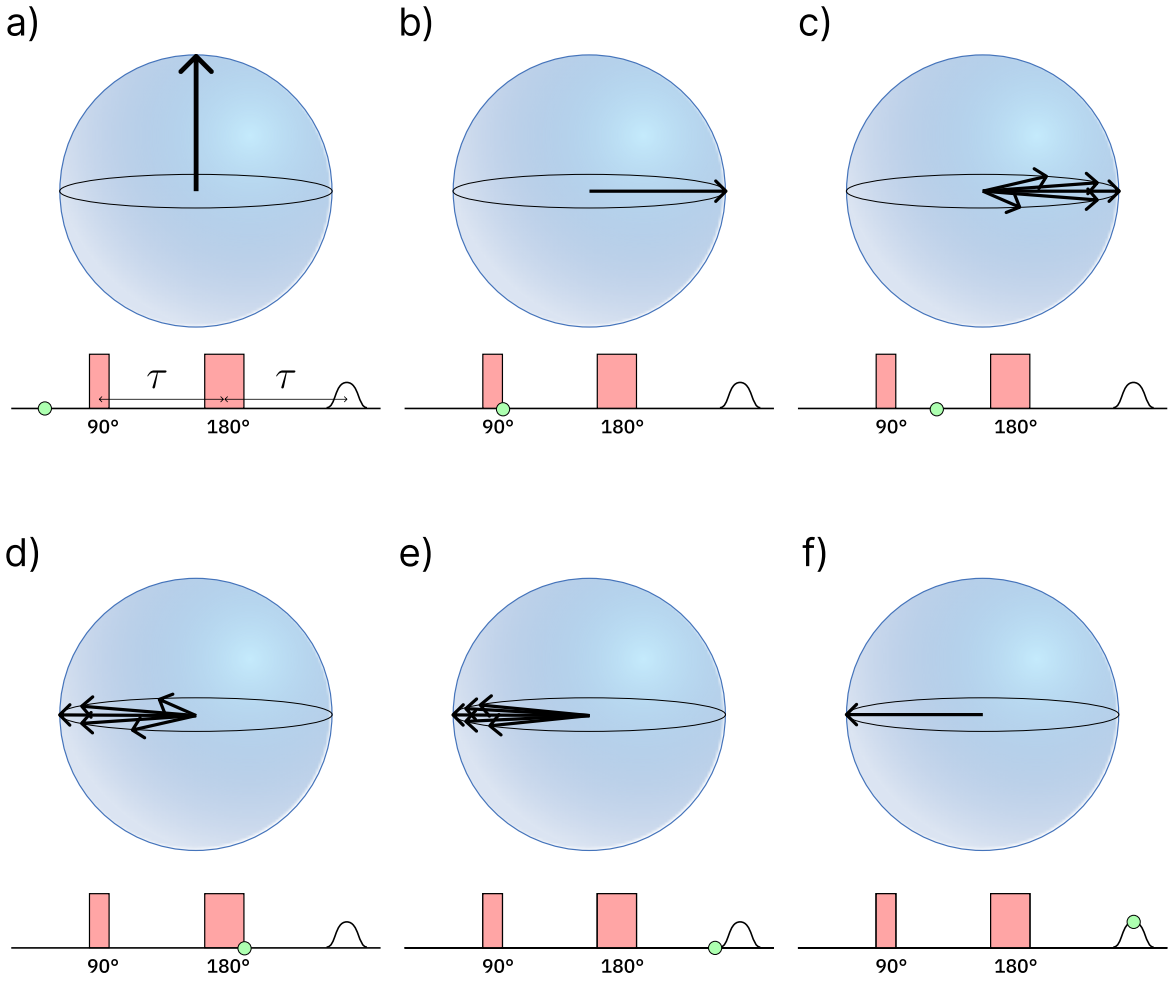

Figure 2.5: *Illustration of the echo signal and the spin echo sequence with the current time step represented with a green circle. a) The average spin moment aligned with the static field B_0. b) The average spin moment just after a short $\frac{\pi}{2}$ RF pulse. c) Spins dephasing after the application of the first pulse. d) Spins after the application of the refocusing pulse after a time duration of τ. e) Spins just before the echo-signal. f) Constructive interference of spins at the peak of the echo signal.*

2.1.4 Image reconstruction

This section briefly covers the formation process of the MRI image from the FID signals. As seen previously, the FID signal results from the whole NMR volume. Therefore, to construct an image, other mechanisms have to be introduced to acquire portions of the sample leading to the complete MR image.

Gradient Slice Selection (GSS). This is the first step towards making an MR image. It consists of applying a linearly varying magnetic field in space: a magnetic field gradient. The introduction of such a gradient allows the spatial encoding of the Larmor frequency such that regions that are excited with a given frequency ω_i form a planar geometry with a given thickness. Regions with lower magnetic fields can be excited with lower Larmor frequencies and vice versa. More precisely, for a given gradient G_z along the Z-axis, the Larmor frequency is given by: $\omega_z = \gamma(B_0 + G_z\Delta z)$. The principle of the Gradient Slice Selection is illustrated in Fig.(2.6). The selected planar section has a certain thickness which depends on the bandwidth of the B_1 RF pulse and also on the gradient magnitude G_z.

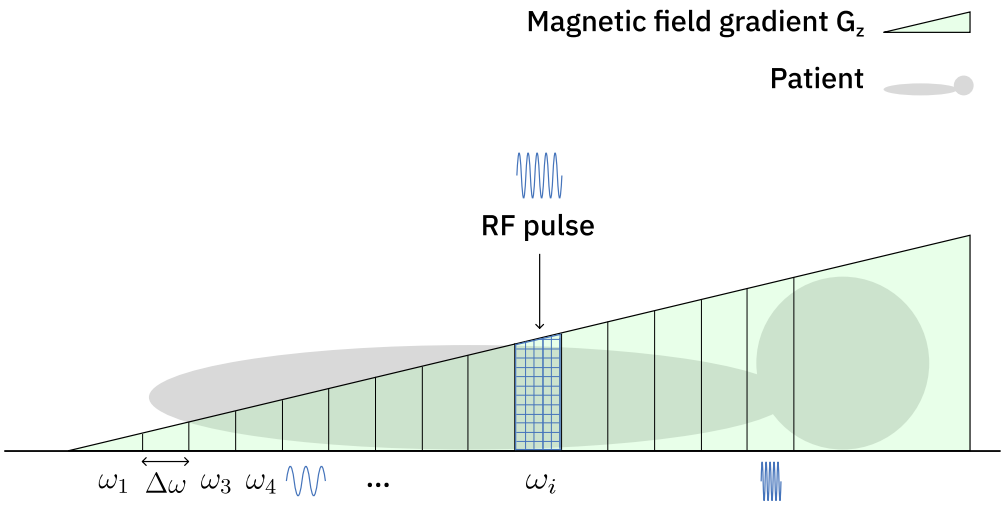

Figure 2.6: *Illustration of the principle of gradient slice selection with an RF pulse of a Larmor frequency ω_i exciting a specific planar section with thickness related to $\Delta\omega$.*

Frequency encoding. Getting the FID signal from a planar section is not enough to construct an MR image. Given the planar slice selected using GSS, the frequency encoding consists of applying a second gradient G_ω along the x-axis. This gradient causes spins to precess following different frequencies along the same axis as the gradient G_ω. This frequency encoding scheme allows the decomposition of the planar slice into columns such that spins within each column have the same precession frequency.

Phase encoding. This encoding is applied before the frequency encoding and after the GSS. It allows to attribute a phase value to each spin using a phase encoding gradient G_ϕ. In contrast to frequency encoding, phase encoding decomposes the planar slice into lines such that spins in each line have the same phase.

The combination of the phase and frequency encodings allows to have a couple of frequency and phase which is the data acquired by an MR imaging system. This data in the case of 2D imaging, is stored in a matrix called the K-space to which an inverse Fourier transform is applied to finally reconstruct the MR image (More information here https://larsonlab.github.io/MRI-education-resources/Spatial%20Encoding.html).

2.2 Diffusion MRI

Diffusion MRI (dMRI) has become increasingly important over the last decades. This modality provides a powerful tool to probe brain tissue in a noninvasive manner and is clinically used for diagnosing brain stroke. This modality is also intensively used in research to understand and characterize other neurodegenerative diseases. In the course of this thesis, the focus will be on studying dMRI data and researching the possibility of inferring brain-related microstructure information, i.e. estimating the average diameter of neurons and neuron density.
As the brain is made up mostly of water, at a finite temperature, these water protons follow random trajectories constrained by the geometrical barriers they encounter. When there are no barriers, and the particles move in free space, the motion is said to be Brownian.

2.2.1 The diffusion process and Brownian motion

The diffusion process. The diffusion process was classically described by Fick in his work [Fick, 1855]. Based on experimental observations, a concentration gradient is related to the flux according to the following equations:

$$\begin{cases} \mathbf{J} = -D\nabla n(\mathbf{r}, t), \\ \\ \frac{\partial n}{\partial t} = D\nabla^2 n \end{cases} \tag{2.11}$$

These relations in Eq.(2.11) are known as the Fick first and second laws. They describe the diffusion flux $\mathbf{J}$ in $\mathrm{mol\,m^{-2}\,s^{-1}}$ as being proportional to the concentration gradient $\nabla n(\mathbf{r}, t)$ in $\mathrm{mol\,m^{-4}}$, and the temporal change rate of the concentration n being proportional to the term $\nabla^2 n$ which is analogous to the heat equation.

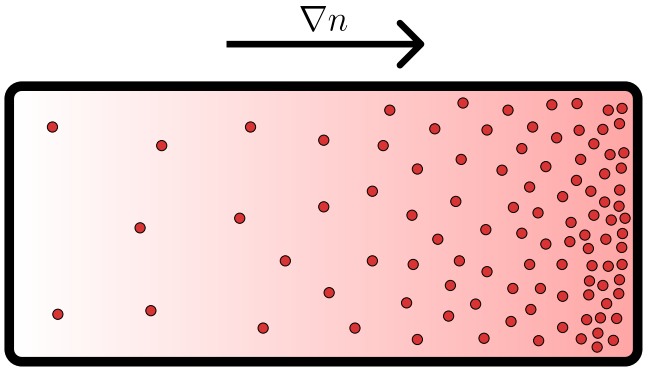

Figure 2.7: *Illustration of the diffusion related to a concentration gradient* ∇n.

The proportionality relation is characterized by a constant D, which is the diffusion coefficient characteristic of the medium, also named the mass diffusivity. A high diffusivity value means a faster recovery to the "equilibrium" state with no gradient concentration, i.e. $\nabla n = \mathbf{0}$.

Brownian motion. Named after the botanist Robert Brown, who observed and studied pollen grains suspended in water. The pollen grains seemed to move continuously and follow random trajectories. The Brownian motion is the result of the collisions of the pollen grains with a large number of water molecules. Einstein further studied this phenomenon and provided a probabilistic model to describe the Brownian motion [Einstein, 1905]. In the same study, a relation between the diffusion coefficient D in $\mathrm{m}^2\,\mathrm{s}^{-1}$ and the temperature. This is known as the Einstein relation:

$$D = \mu k_B T \tag{2.12}$$

With μ being the mobility in $\mathrm{m\,N^{-1}\,s^{-1}}$, k_B is the Boltzmann constant in $\mathrm{J\,K^{-1}}$, and T is the temperature in K.

The probabilistic model describes the Brownian motion as a stochastic process, and it characterizes it using the following equation:

$$\frac{\partial}{\partial t} P(\mathbf{r}|\mathbf{r}_0, t) = D\nabla^2 P(\mathbf{r}|\mathbf{r}_0, t) \tag{2.13}$$

The term $P(\mathbf{r}|\mathbf{r}_0, t)$ in the partial differential equation in Eq.(2.13) is the probability for a particle in 3D space at an initial position $\mathbf{r}_0$ to transition to the position $\mathbf{r}$ after a time duration t. The same probability is called **the diffusion propagator**. When at time $t = 0$, the particle is initialized at $\mathbf{r}_0$, i.e., $P(\mathbf{r}|\mathbf{r}_0, t = 0) = \delta(\mathbf{r} - \mathbf{r}_0)$ where $\delta(\mathbf{x})$ is the Kronecker delta function, the diffusion equation in Eq.(2.13) admits the following solution:

$$P(\mathbf{r}|\mathbf{r}_0, t) = (4\pi D t)^{3/2} \exp\left(-\frac{\|\mathbf{r} - \mathbf{r}_0\|^2}{4Dt}\right) \tag{2.14}$$

The obtained solution is a multivariate normal distribution centered at the initial position $\mathbf{r}_0$ with variance proportional to elapsed time t and to the diffusion coefficient D. Untill now, the diffusion coefficient was linked to physical quantities such as the temperature T and the Boltzmann constant k_B. However, it is possible to obtain the diffusion coefficient D from the Mean Squared Displacement (MSD) defined as:

$$\begin{aligned} \mathrm{MSD} &= \frac{1}{N}\sum_{i=0}^{N} \left\| \mathbf{r}^{(i)}(t) - \mathbf{r}^{(i)}(0) \right\|^2 \\ &= \langle (\mathbf{r} - \mathbf{r}_0)^2 \rangle, \end{aligned} \tag{2.15}$$

where $\mathbf{r}^{(i)}(t)$ is the position of the i-th particle at time t. The MSD is basically a measure of the spread of the position of a particle after the time duration t relative to its initial position $\mathbf{r}(t = 0) = \mathbf{r}_0$. Given that the coordinates are independent, by expanding the MSD Eq.(2.15), it is possible to link the MSD to the variance of the probability distribution function (PDF)

characterizing the Brownian motion Eq.(2.14) through the following relation:

$$\text{MSD} = 2nDt, \tag{2.16}$$

with D being the diffusivity and n the dimension of the space in which the Brownian motion occurs. Thus, the equations Eq.(2.12) and Eq.(2.16) show that it is possible to know the diffusivity coefficient D either from physically measurable quantities or from trajectory data of the particles undergoing Brownian motion. For example, in 3D space, the MSD is $\text{MSD} = 6Dt$.
Inversely, given a certain diffusivity D, it is possible to "simulate" Brownian motion by sampling from the PDF in equation Eq.(2.14). The illustration in Fig.(2.8) depicts a simulated Brownian trajectory in the 2D free space.

The diffusion coefficient D presented was assumed to be a scalar quantity. However, this

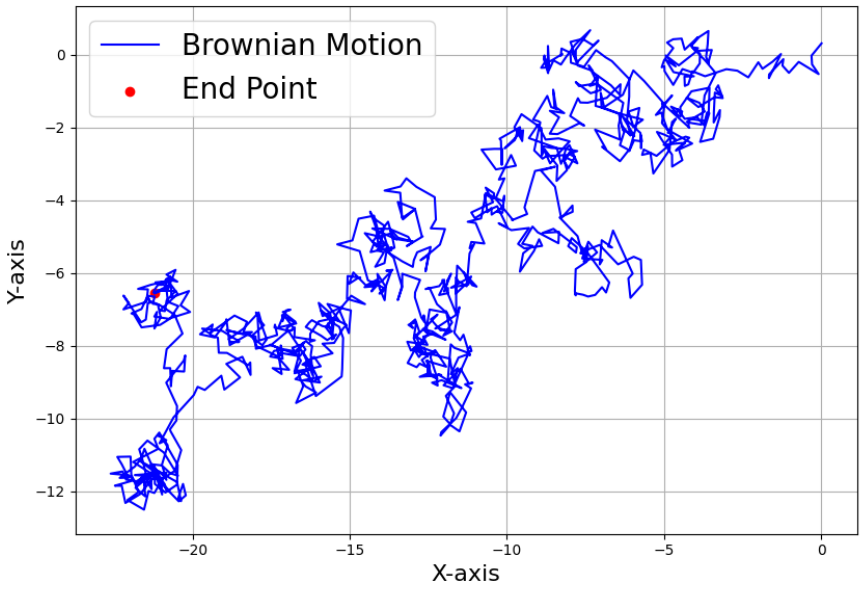

Figure 2.8: *Illustration of a simulated Brownian motion in 2D free space.*

hypothesis does not hold everywhere: it only holds for isotropic media. In contrast, for anisotropic media, i.e., media where the diffusion coefficient following a certain direction can be different when following another one, the diffusivity coefficient D is no longer a scalar parameter. Instead, it is represented by a tensor $\underline{\underline{\mathbf{D}}}$. The diffusion tensor $\underline{\underline{\mathbf{D}}}$ is a 3x3 positive-definite, symmetric matrix:

$$\underline{\underline{\mathbf{D}}} = \begin{pmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{yx} & D_{yy} & D_{yz} \\ D_{zx} & D_{zy} & D_{zz} \end{pmatrix} \tag{2.17}$$

The diffusion tensor in equation Eq.(2.17) can be interpreted geometrically as an ellipsoid with principal axes corresponding to the eigenvectors of the tensor $\underline{\underline{\mathbf{D}}}$ and with radii related to its eigenvalues. In the case of anisotropic diffusion, the diffusion equation Eq.(2.13) is no longer

valid. Instead, a more general version that accounts for anisotropy is presented below:

$$\frac{\partial}{\partial t}P(\mathbf{r}|\mathbf{r}_0,t) = \nabla.[\underline{\underline{\mathbf{D}}}\nabla P(\mathbf{r}|\mathbf{r}_0,t)] \tag{2.18}$$

In the particular case of $D_{xx} = D_{yy} = D_{zz}$ and $D_{ij} = 0$ for $i \neq j$, the case of isotropic diffusion is retrieved and the equation Eq.(2.18) becomes identical to Eq.(2.13). Provided the same initial conditions: $P(\mathbf{r}|\mathbf{r}_0, t = 0) = \delta(\mathbf{r} - \mathbf{r}_0)$, the more general diffusion equation admits the following solution:

$$P(\mathbf{r}|\mathbf{r}_0,t) = (4\pi|\underline{\underline{\mathbf{D}}}|t)^{3/2}\exp\left(-\frac{(\mathbf{r}-\mathbf{r}_0)^{\mathsf{T}}\underline{\underline{\mathbf{D}}}^{-1}(\mathbf{r}-\mathbf{r}_0)}{4t}\right) \tag{2.19}$$

The propagator expression in equation Eq.(2.19) generalizes the formula in Eq.(2.14) with the diffusion tensor $\underline{\underline{\mathbf{D}}}$ analogous to the covariance matrix in a multivariate normal distribution.

Fractional anisotropy. To characterize the degree of anisotropy within a medium, the fractional anisotropy (FA) is a widely used measure within the diffusion MRI community. It is defined as:

$$\begin{aligned} \text{FA} &= \sqrt{\frac{3}{2}}\frac{\sqrt{(\lambda_1-\hat{\lambda})^2+(\lambda_2-\hat{\lambda})^2+(\lambda_3-\hat{\lambda})^2}}{\sqrt{\lambda_1^2+\lambda_2^2+\lambda_3^2}} \\ \text{ADC} &= \hat{\lambda} = \frac{\lambda_1+\lambda_2+\lambda_3}{3} \end{aligned} \tag{2.20}$$

With λ_1, λ_2, and λ_3 being the eigenvalues of the diffusion tensor $\underline{\underline{\mathbf{D}}}$, and $\hat{\lambda}$ the arithmetic average of the eigenvalues, the quantity $\hat{\lambda}$ is called the apparent diffusion coefficient (ADC). The FA values are between 0 and 1, where the minimal value of zero is attained in the case of an isotropic medium, i.e., $\lambda_1 = \lambda_2 = \lambda_3$. The maximal value of one is attained in the extreme case where the diffusion is restricted and happens only in one direction. Three different cases of FA values are depicted in Fig.(2.9).

2.2.2 From the diffusion process to diffusion MRI signal

This section will discuss how MRI is sensitized to the diffusion process through the phase shift. One of the important contributions in this area was the work of Carr and Purcell [Carr & Purcell, 1954]. This work presented a study of the effects of diffusion on free precession in NMR experiments. The main contribution was the use of a combination of two RF pulses: one at 90° and the other at 180°. In addition, the effect of diffusion on the change in the rate of nuclear precession was studied, and a method for measuring water's diffusivity constant D was provided.

Bloch-Torrey equations. This is another important contribution to the field of diffusion MRI. In his study [Torrey, 1956], Torrey generalized the phenomenological Bloch equations Eq.(2.5) by accounting for the diffusion process through which the magnetization is transferred. The new form of the phenomenological Bloch equations that account for diffusion is provided by:

$$\frac{\partial \mathbf{M}(t)}{\partial t} = \gamma\mathbf{M}(t)\times\mathbf{B}(t) - \frac{(M_z - M_0)\hat{\mathbf{z}}}{T_1} - \frac{\mathbf{M_T}(t)}{T_2} + \nabla.\left(\underline{\underline{\mathbf{D}}}\nabla\mathbf{M}\right) \tag{2.21}$$

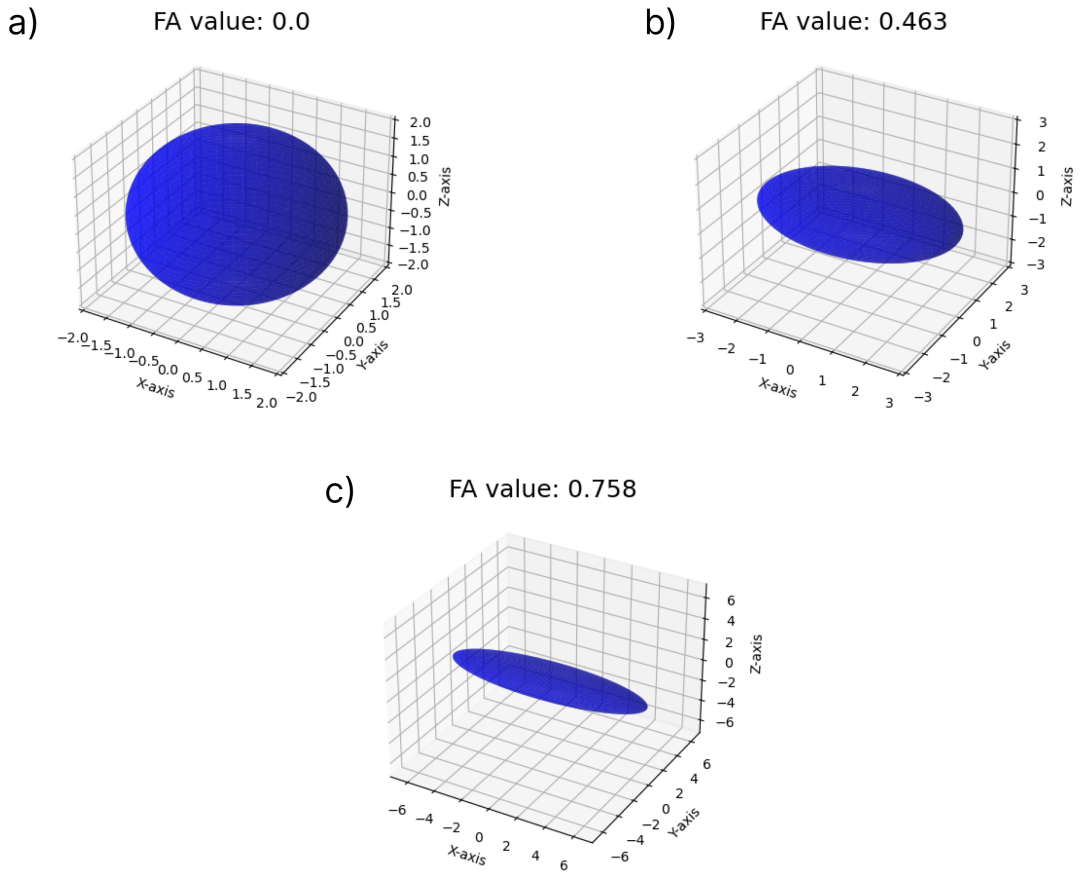

Figure 2.9: *Illustration of the fractional anisotropy with three different cases: the first case a) characterizes an isotropic medium. The second b) characterizes an anisotropic medium with moderate anisotropy. Finally, the last case c) represents a medium with high anisotropy.*

With M_0 being the equilibrium magnetization, T_1 and T_2 are the spin-lattice and spin-spin relaxation times, respectively. The introduction of this equation allowed to have a more general mathematical framework to understand the effect of diffusion on magnetization. In the case of isotropic free diffusion, the Bloch-Torrey equation for the transverse magnetization in the rotating frame admits a solution of the following form[Jones, 2010]:

$$\mathbf{M}_+(\mathbf{r},t) = E(t)\exp\left(-i\gamma\mathbf{r}.\int_0^t \mathbf{G}(t')dt'\right)\exp\left(-\frac{t}{T_2}\right) \tag{2.22}$$

The expression of the term $E(t)$ is obtained by plugging the solution form in the Bloch-Torrey equation for the transverse magnetization in the rotating frame. By using the echo-condition, i.e., $\int_0^t \mathbf{G}(t')dt' = 0$, the $E(t)$ takes the following form [Jones, 2010]:

$$E(t=TE) = \exp\left(-D\gamma^2\int_0^{TE}\left[\int_0^{t'}\mathbf{g}^*(t'')dt''\right]^2 dt'\right)\exp\left(i\gamma\mathbf{v}.\int_0^{TE}\left[\int_0^{t'}\mathbf{g}^*(t'')dt''\right]\right) \tag{2.23}$$

In addition to the T_2 relaxation, the diffusion term in equation Eq.(2.23) also attenuates the echo signal. Thus, measuring accurately the attenuation of the echo signal makes it possible to infer the diffusion coefficient D of the experiment sample. The diffusion-related relaxation term next to the diffusivity D is called the b-value or the b-factor:

$$b = \gamma^2\int_0^{TE}\left(\int_0^t G(t')dt'\right)^2 dt \tag{2.24}$$

The b-value term Eq.(2.24) controls the strength of the diffusion effect. The higher it is, the stronger the attenuation of the echo signal.
Consider a single spin immersed in a magnetic field gradient located in a position $\mathbf{r}$. The cumulative phase shift given the gradient $\mathbf{G}(t)$ is given by[Bammer, n.d.]:

$$\phi(t) = \gamma B_0 t + \gamma\int_0^t \mathbf{G}(t').\mathbf{r}(t')dt' \tag{2.25}$$

The first term in Eq.(2.25) represents the phase-shift related to the constant magnetic field $\mathbf{B_0}$. The second term is related to the gradient $\mathbf{G}(t)$. Thus, by designing specific gradients, it is possible to obtain interesting phase shift expressions that allow the labeling of the spin's position in terms of small differences in the Larmor frequency. Since the spin's precession frequency is determined by the local magnetic field in Eq.(2.2), when the spin changes its position, its precession frequency will change and thus will acquire a phase shift. These are the basis for measuring the diffusion MRI signal.

Stejskal-Tanner's Pulsed Gradient Spin Echo sequence. Another important contribution to the field of MRI was driven by the work of Stejskal and Tanner [Stejskal & Tanner, 1965] in which the now-standard Pulsed Gradient Spin-Echo (PGSE) sequence was introduced. The PGSE sequence illustrated in Fig.(2.10) shows a succession of two identical diffusion gradients preceded by two RF pulses, the first one at 90° and the second one at 180°. The gradients are characterized by the following parameters: The gradient magnitude $||\mathbf{G}|| = G$, the gradient duration δ, and the

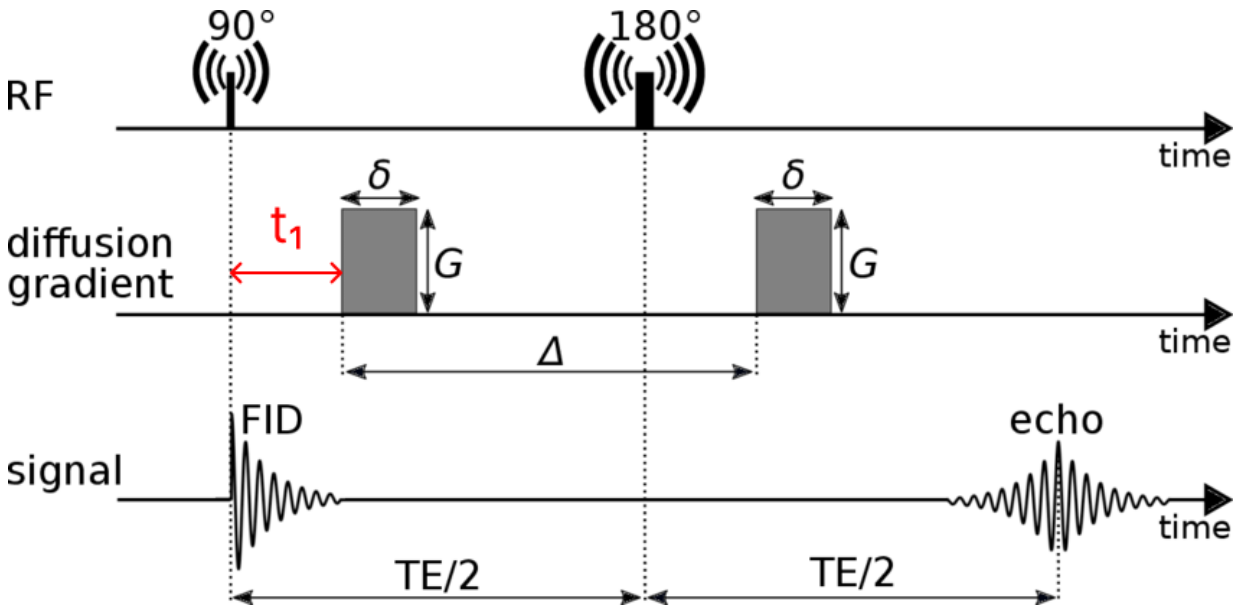

Figure 2.10: *Illustration of the PGSE sequence with the parameters Δ, δ, and the gradient magnitude G (adapted from [Fick, 2017]).*

time separation between the two gradients Δ. Let t_1 be the time duration between the first RF pulse at 90° and the first diffusion gradient, given the form of the PGSE sequence, the resulting phase shift is [Bammer, n.d.]:

$$\phi(\text{TE}) = \gamma \int_{t_1}^{t_1+\delta} \mathbf{G}(t').\mathbf{r}(t')dt' - \gamma \int_{t_1+\Delta}^{t_1+\delta+\Delta} \mathbf{G}(t').\mathbf{r}(t')dt' \tag{2.26}$$

The obtained phase shift in equation Eq.(2.26) is the phase shift resulting from plugging the PGSE sequence in the cumulative phase shift in Eq.(2.25). Since the gradient terms **G** in the PGSE sequence are identical, a non-vanishing phase shift results from the displacement of the considered spin. Inversely, when the spin's position remains the same, i.e., no spin motion, the two terms cancel each other, and the resulting phase shift is null. The figure Fig.(2.11) illustrates the diffusion-related echo signal attenuation.

In the case of a PGSE sequence, the b-value in Eq.(2.24) can be evaluated using the temporal profile of the PGSE sequence. The evaluation leads to the following expression:

$$b = \gamma^2 G^2 \delta^2 \left(\Delta - \frac{\delta}{3}\right) \tag{2.27}$$

Although the PGSE sequence is assumed to have perfectly rectangular gradients, in practice, physical limitations of the gradient coils imply that the maximal gradient value G can only be attained in a time duration ξ called the ramp time: the time necessary to go from zero to G. Thus, the equation Eq.(2.27) does not hold when the ramp time is no longer negligible. The quantity $\tau = \left(\Delta - \frac{\delta}{3}\right)$ is called the diffusion time.

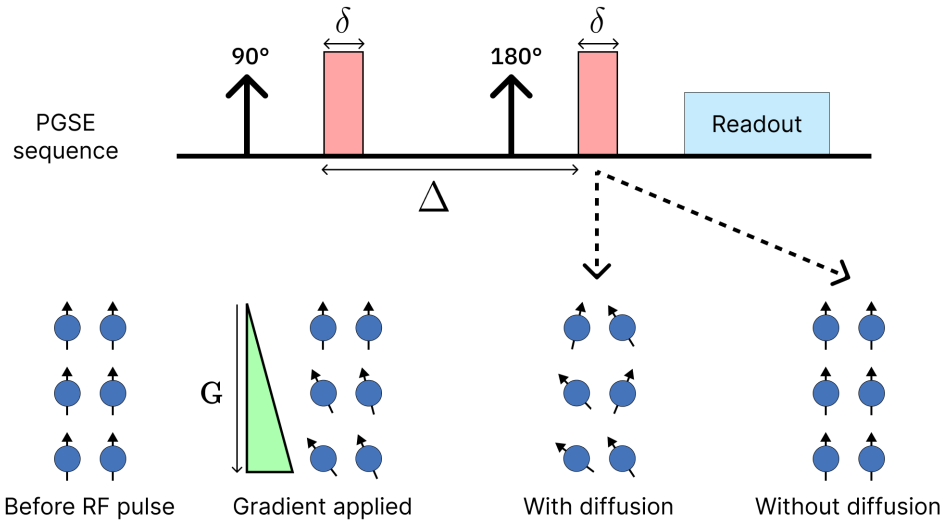

Figure 2.11: *Illustration of the echo signal attenuation due to the diffusion process for a PGSE sequence.*

2.2.3 Restricted diffusion

The considered solution Eqs.(2.22, 2.23) of Bloch-Torrey equations allows a basic understanding of the effect of free isotropic diffusion in NMR experiments. However, in the context of medical imaging, the biological tissues in which the diffusion occurs form a complex geometrical medium. Thus, the diffusion in such media is no longer isotropic nor free: It is highly restricted and anisotropic. The important distinction between free and restricted diffusion can be expressed by the form of the diffusion propagator $P(\mathbf{r}|\mathbf{r}_0, t)$. In the free space, this propagator is given by the multivariate distribution in Eq.(2.14) in free space, however, this form is no longer valid for restricted diffusion.

Stejskal's work [Stejskal, 1965] provides a modified version of the Bloch-Torrey equation generalized to the anisotropic case. The same work presents a solution for two cases of restricted diffusion using the **short pulse gradient (SPG) approximation**, i.e. magnetic field gradient applied in short and intense pulses: $\delta \ll \Delta$. In this case, the gradient pulses are modeled as two δ-Kronecker functions separated by a time interval of Δ.

In the SPG approximation, a link between the diffusion propagator and the attenuation of the echo signal can be derived. By considering the Bloch-Torrey equation, the SPG approximation, and a magnetization solution in the form of $M_T = \mathrm{e}^{-R_2 t - i<\omega>t}\psi(\mathbf{r}, t)$, according to [Stejskal, 1965], the diffusion-weighted signal in a voxel is given by the following expression [Novikov *et al.*, 2019]:

$$\frac{S(\mathbf{q}, \Delta)}{S(\mathbf{q}, t)|_{\mathbf{q}=\mathbf{0}}} = \int\int \frac{d\mathbf{r}_0 d\mathbf{r}}{V} P(\mathbf{r}|\mathbf{r}_0, \Delta) \exp\left(i\mathbf{q}.(\mathbf{r} - \mathbf{r}_0)\right) \tag{2.28}$$

The vector $\mathbf{q}$ is called the wave vector, and it is defined as:

$$\mathbf{q} = \gamma \left(\int_0^t \mathbf{G}(t')dt' \right) \tag{2.29}$$

The evaluation of the $\mathbf{q}$ using a PGSE sequence leads to the following expression:

$$\mathbf{q} = \frac{\gamma}{2\pi}\delta\mathbf{G} \tag{2.30}$$

The expression Eq.(2.28) of the signal obtained in the case of the short pulse gradient approximation reveals an interesting relation between the diffusion propagator and the measured signal. The signal is the spatial Fourier transform of the voxel-averaged propagator, which is given by the following expression [Novikov *et al.*, 2019]:

$$<P(\mathbf{r}+\mathbf{r}_0|t)>_{\mathbf{r}_0} = \int \frac{d\mathbf{r}_0}{V} P(\mathbf{r}+\mathbf{r}_0|\mathbf{r}_0,t) \tag{2.31}$$

The relation between the diffusion-weighted signal attenuation and the voxel-averaged propagator in Eq.(2.28) allows to access to the propagator from the signal attenuation. Therefore, by measuring the signal $S(\mathbf{q},t)$ in the $\mathbf{q}$-space, it is possible to estimate the voxel-averaged propagator: this is the foundation of the $\mathbf{q}$-space sampling.

The Gaussian phase distribution and restricted diffusion. The SPG approximation provides insightful results. However, those are only valid for MRI systems with strong gradient requirements. An alternative approximation presented in this study [Neuman, 1974] relies on the hypothesis that the phase shift at the echo time follows a Gaussian distribution. This hypothesis is known by the Gaussian Phase Distribution (GPD) approximation. The same study also investigated the echo signal in simple restricted geometries such as a sphere, a cylinder, and parallel planes. This work was extended by Callaghan's study [Callaghan, 1994] which presents a closed analytic ansatz for the average propagator and for the echo signal for parallel planes. The following study [Callaghan, 1995] provides a closed-form solution for the echo signal in the case of spherical and cylindrical geometries, and it also revisits the closed formula for the case of parallel planes. To derive an analytic solution for the propagator, an eigenmode expansion of the diffusion propagator $P(\mathbf{r}|\mathbf{r}_0,t)$ is used [Callaghan, 1995]:

$$P(\mathbf{r}|\mathbf{r}_0,t) = \sum_{n=0}^{\infty} \exp(-\lambda_n t)\, u_n(\mathbf{r}) u_n^*(\mathbf{r}_0) \tag{2.32}$$

The eigenfunctions $u_n(\mathbf{r})$ are an orthonormal set of solutions to the Helmholtz equation with their corresponding eigenvalues λ_n. By plugging the propagator expansion Eq.(2.32) in the diffusion equation Eq.(2.13), and by using the proper boundary conditions for the given restricting geometry, an analytic expression of the propagator can be obtained. For the case of cylindrical geometry of radius R, the echo signal in the radial direction is given by [SÖDERMAN & JÖNSSON, 1995]:

$$\frac{S}{S_0} = 8(2\pi qR)^2 \sum_{k=1}^{\infty}\sum_{m=0}^{\infty} \alpha_{km}^2 [J_m'(2\pi qR)]^2 \frac{\exp\left(-\left(\frac{\alpha_{km}}{R}\right)^2 D\Delta\right)}{[\alpha_{km}^2 - (2\pi qR)^2]^2(\alpha_{km}^2 - m^2)} \tag{2.33}$$

The expression Eq.(2.33) is the echo signal for a gradient perpendicular to the cylinder's main orientation. The function J_m' is the derivative of the standard cylindrical Bessel function, α_{km} is the k-th root of the Bessel function's derivative J_m', i.e., $J_m'(\alpha_{km}) = 0$, with the convention $\alpha_{10} = 0$.

2.2.4 q-Space sampling

From the previous section, in the SPG approximation, the averaged propagator can be accessed from the signal attenuation in the q-space. Although this relation is interesting, in practice, it is not feasible to acquire the echo signals for all the possible **q** values. Therefore, sampling the q-space becomes necessary. The PGSE sequence introduced in [Stejskal & Tanner, 1965] can be considered as an example of a single q-space sample. Another work [Le Bihan *et al.*, 1986] proposes a sampling scheme with three samples from the q-space, the s-samples have the same b-value and with orthonormal gradient orientations. The seminal work of Baser [Basser *et al.*, 1994] proposed to sample six q-space samples located on a sphere to estimate the coefficient of the diffusion tensor $\underline{\underline{\mathbf{D}}}$ in Eq.(2.17), this method is known as **the diffusion tensor imaging (DTI)**.
Later, in his work [Wedeen *et al.*, 2005a, Wedeen *et al.*, 2000], Van Weeden introduced the diffusion spectrum imaging (DSI): a method for 3D Cartesian sampling of the q-space that was used to characterize the PDF of a crossing fiber population. Such a sampling scheme can be difficult to deploy in a clinical setting given the time necessary to acquire the signal for all the q-samples.

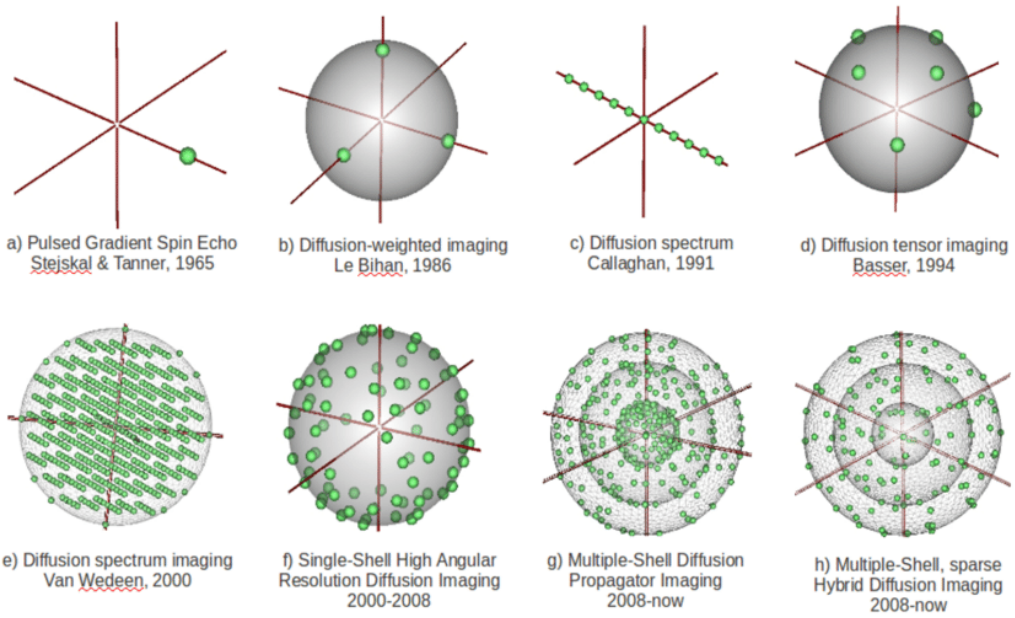

Figure 2.12: *The evolution of the q-space sampling depicting sparer sampling strategies over the years. Such strategies were enabled by innovations in the MRI hardware and signal modeling which allowed to push further the q-space sampling [Descoteaux & Poupon, 2014].*

2.2.5 Beyond PGSE sequences

Although frequently used, the PGSE sequence is not the only available sequence in diffusion MRI. One of its limitations is its difficulty accessing short diffusion times which enables to capture smaller brain microstructural features. Other sequences were developed to overcome some of the limitations of the PGSE sequence. For example, the double diffusion encoding sequences use two pairs of gradients instead of one. This type of sequence can provide additional information relative to the classical PGSE when the diffusion medium is not translation invariant

[Novikov *et al.*, 2019]. It also has the ability to distinguish microscopic anisotropy of a macroscopically isotropic geometry, i.e a geometry with spheres with different sizes averaging a radius of $\bar{r}$ compared to a geometry with spheres with the same radius $\bar{r}$. This ability to discriminate such features is important when investigating brain microstructures.

Another important sequence is the **Oscillating Gradient Spin Echo (OGSE)** sequence. Originally presented in this work [Stepišnik, 1981], the OGSE sequence consists in the application of a gradient that oscillates in time in contrast to the squared pulse in the case of the PGSE sequence. The application of the OGSE sequence allows to overcome the long diffusion times limitations of the PGSE determined by Δ in the SPG approximation. In the case of the OGSE sequence, the diffusion time is related to the oscillating period which allows probing smaller diffusion times. In addition, the OGSE sequence enables to probe the spectral density of molecular diffusion. The OGSE sequence OGSE comes in different variants: sine, cosine, and trapezoidal. These variants refer to the periodic function used to describe the gradient temporal form. In practice, the trapezoidal gradient form in Fig.(2.13) is advantageous as it allows to reach stronger b-values.

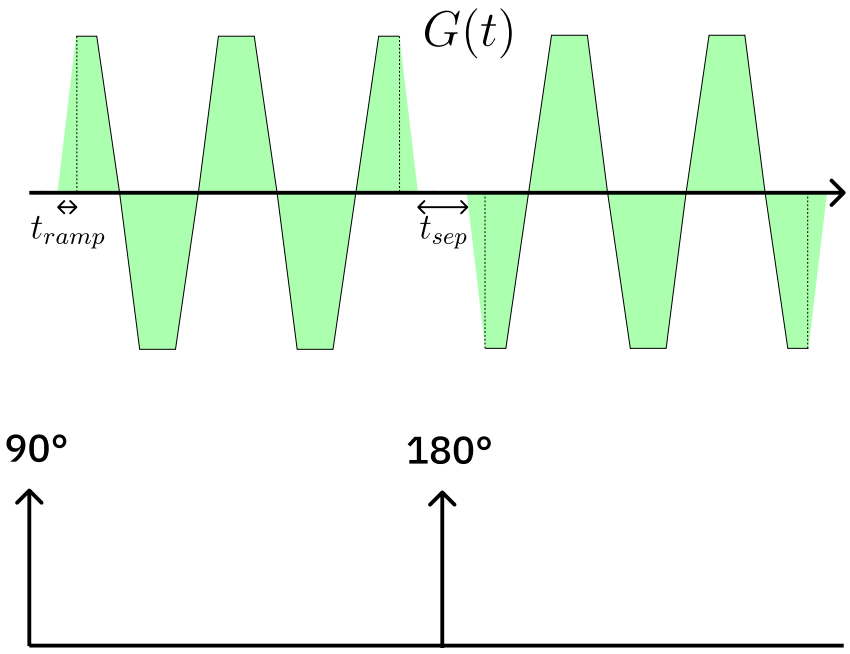

Figure 2.13: *Illustration of OGSE sequence in the case of a trapezoidal cosine sequence.*

B-tensor formalism. It provides a more general framework for representing different gradient waveforms [Szczepankiewicz *et al.*, 2021]. This framework considers more complex gradient waveforms represented as trajectories in the gradient space. While offering great flexibility in waveform design, such sequences are still an active area of research to overcome the hardware and physiological limitations related to the use of complex gradient waveforms.

2.3 Conclusion

This chapter presented the basics of the MRI and the diffusion MRI physics and principles. It outlined the difference between Gaussian and restricted diffusion and set the basis to introduce the various methods used within the dMRI community to study brain microstructure. The next chapter presents works in the diffusion MRI literature that are relevant to this thesis with a focus on the simulation methods.

Chapter 3

Litterature review

This chapter presents a general review of the tools and methods used within the diffusion MRI community to address the microstructure decoding problem. State-of-the-art diffusion MRI simulators and synthetic substrates are presented in this chapter. This chapter distinguishes between two broad tissue microstructure models, analytic-based and simulation-based.

3.1 dMRI signal modeling

This section presents a broad overview of the different analytical models used to decode brain microstructure from diffusion MRI data.
Before presenting analytical models for microstructure, it is important to differentiate between **representations** and **models**.

3.1.1 Signal representations

Also called model-free techniques, representations are mathematical expressions or numerical values that are general and model-independent. In other words, representations are merely formulas for the dMRI signal that do not carry a particular physical meaning.
There are different signal representations adopted within the dMRI community:

- The bi-exponential representation formula of the signal which is given by:

$$S = w_1 e^{-bD_1} + w_2 e^{-bD_2} \tag{3.1}$$

- **The cumulant expansion** is also another widely used signal representation. It stems from probability theory, and its formula is given by:

$$ln\left(\frac{S}{S_0}\right) \simeq -b\sum_{i,j}^{3} D_{ij}g_i g_j + \frac{1}{6}(b\bar{D})^2 \sum_{i,j,j,l=1}^{3} g_i g_j g_k g_l W_{ijkl} - ... \tag{3.2}$$

- **Diffusion tensor imaging (DTI).** It is a special case of the cumulant expression such that only the first term is kept. It captures the diffusion tensor $\underline{\underline{\mathbf{D}}}$ and assumes that the diffusion medium behaves like a Gaussian medium.

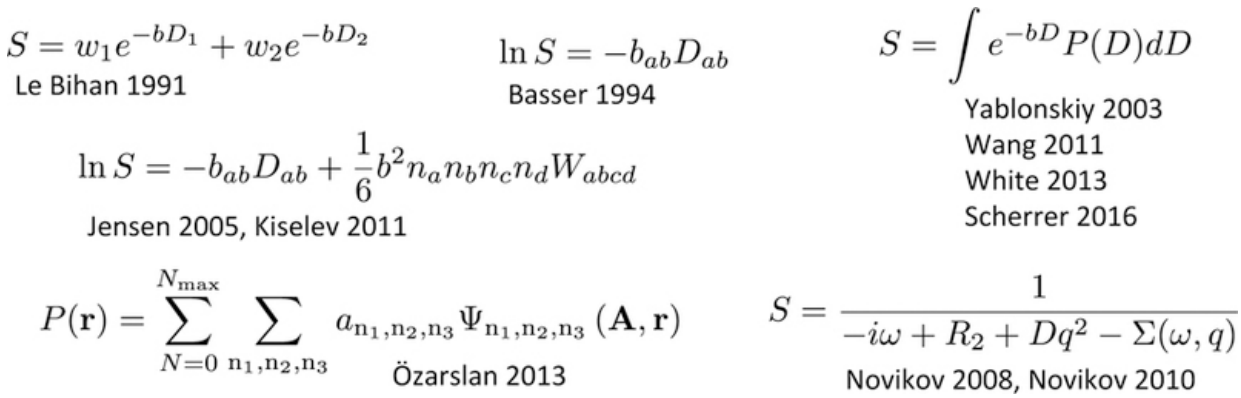

Figure 3.1: *Illustration of some of the most used dMRI signal representations (image taken from [Novikov et al., 2019]).*

- **Diffusion Kurtosis Imaging (DKI).** [Jensen *et al.*, 2005] Can also be considered as a special case of cumulant expansion such that the expansion is done up to the quadratic term.
- **Diffusion Spectrum Imaging (DSI).** It consists of performing a dense 3D Cartesian sampling of the q-space [Wedeen *et al.*, 2000] [Wedeen *et al.*, 2005b]. The acquisition of these samples allows to reconstruct the voxel-averaged diffusion propagator using the Fourier transform.
- **Q-ball imaging (QBI).** This method was developed to improve DTI which can only provide a single direction of diffusion [Tuch, 2004]. Thus, not representing more complex cases of fiber configurations such as fiber crossing. QBI consists of sampling a sphere in the q-space, thus allowing reduced times compared to cartesian q-space sampling (DSI).
- **Mean apparent propagator (MAP).** Decomposes the signal in terms of the eigenfunctions of the simple harmonic oscillator representations, i.e., the SHORE basis [Özarslan *et al.*, 2013]. This decomposition allows the computation of interesting quantities such as the return to origin probability (RTOP), i.e., the probability for a null net displacement, and the return to plane and axis probabilities (RTPP, and RTAP).

An illustration of frequently used dMRI signal representations is provided in Fig.(3.1).

3.1.2 Analytical modeling

Models can be viewed as pictures of the physical reality of the brain [Novikov *et al.*, 2019]. They provide assumptions about the geometrical representation of the brain tissue and hypotheses about the relevant parameters that govern the diffusion MRI signal behavior. In practice, the underlying assumptions of a tissue model are summarized in a set of analytical equations that are fitted to the diffusion data to estimate the parameters of the tissue model. At the voxel scale, Tissue models usually come in the following functional form:

$$\begin{cases} S(\mathbf{b}) = \sum_{i=1}^{N} f_i S_i(\mathbf{b}) \\ \sum_{i=1}^{N} f_i = 1. \end{cases} \tag{3.3}$$

The equation Eq.(3.3) provides a general form of a non-exchanging multi-compartment tissue model. The total signal is a weighted sum of multiple signal components $S_i(\mathbf{b})$, each representing a specific part of the brain tissue. Each component is weighted with a fraction f_i representing the relative signal fraction. Depending on the type of the compartment and the used hypotheses, an explicit formula for S_i is used to model the contribution of the compartment i to the signal attenuation. The explicit formula for S_i comprises parameters of the signal model. There are different elements to be taken into account when analytically modeling dMRI signals:

- **The constituents of the signal models.** What type of geometries to consider? This depends on the brain region to be modeled. For example, some analytical models distinguish gray from white matter.

- **The diffusion regime.** Depending on the diffusion time, equivalently, the diffusion length, the geometry "perceived" by the dMRI process is different. The diffusion process can be considered as an example of coarse-graining [Novikov *et al.*, 2019], which means that fine details of the brain tissue microstructure are homogenized with larger diffusion times. This has important implications for analytical models.

- **The fitting method.** It is also an important point. After developing a model, the main goal is to be able to infer microstructural parameters of interest from dMRI signals. Therefore, it is essential to be able to obtain reasonably good fits for the collected data.

Other elements are also taken into consideration when developing analytical signal models. For instance, different analytical models using a compartmentalized tissue approach assume that the compartments are separated, i.e., there is no water exchange between compartments. Such a hypothesis does not hold in gray matter. To account for exchange, Kärger's contributions in [Kärger *et al.*, 1988] and [Kärger, 1985] have introduced a model for molecular transport in a two-compartments system characterized with two diffusion coefficients.
Different analytical models have been proposed in the last decades within the diffusion MRI community. Each model has its own parameters and hypotheses, and a list of some of these models is provided below:

Stanisz model. Is considered one of the first works proposing an analytical model for dMRI signals. In his work [Stanisz *et al.*, 1997], Stanisz considers a microstructure model consisting of prolate ellipsoids for axons and spheres for glial cells. The model has a set of nine parameters and requires a diverse set of diffusion times and gradient magnitudes to be fitted.

Ball and stick model. One of the simple models presented in this study [Behrens *et al.*, 2003]. It separates the intra and extra-axonal compartments and allows the estimation of parameters such as the local fiber orientation.

CHARMED. It is an acronym for the Composite Hindered And Restricted Model of Diffusion [Assaf *et al.*, 2004]. It is also a two-compartment model that incorporates more microstructural parameters compared to the ball and stick model. In the case of CHARMED, the extra-axonal

space is modeled by a Gaussian diffusion tensor, while the intra-axonal space is modeled by restricted diffusion inside straight impermeable cylindrical geometries. This model allows us to infer white-matter microstructural parameters such as the fiber's orientation, the intra-axonal volume fraction, and principal diffusivities.

AxCaliber. In [Assaf *et al.*, 2008], an additional contribution was made in modeling white-matter microstructure. AxCaliber extends the CHARMED model by using the cylinder-restricted echo signal formula in [Callaghan, 1995] to model intra-axonal signal. More precisely, a population of cylinders with diameters following a Gamma distribution is used. Therefore, this model allows to estimate the axonal diameter distribution from dMRI signals non-invasively.

Active Ax. This model also extends the CHARMED model and targets white matter microstructure [Alexander, 2008]. It aims to overcome some of the limitations of the AxCaliber model by proposing acquisition protocols feasible in human and animal MRI systems. It allows to estimate microstructural parameters such as axon density and diameter.

Neurite orientation dispersion and density imaging (NODDI). This model has three compartments [Zhang *et al.*, 2012]: intra-axonal space, extra-axonal space, and final compartment representing the CSF. NODDI is characterized by a set of seven parameters, two of which describe volume fractions of intra-axonal space and CSF. Four parameters describe the diffusivity coefficients in the compartments, and finally, a last parameter describes the orientation dispersion of the axons.

White matter tract integrity (WMTI). This mode comprises two Gaussian compartments [Fieremans *et al.*, 2013], one for extra-axonal space and the other intra-axonal consists of sticks. This model characterizes better white matter substrates with highly aligned axon bundles. It can easily be fitted using diffusion and kurtosis tensors data.

The standard model (SM). It is one of the important tissue models in the diffusion MRI community [Novikov *et al.*, 2019]. The SM is a two-component tissue model for intra and extra-neurite spaces. The model has the following form:

$$S(\mathbf{b}) = f_{in} S_{in}(\mathbf{b}) + f_{en} S_{en}(\mathbf{b})$$

The intra-neurite compartment is represented as a convolution between the response kernel K obtained from perfectly aligned fibers and the fiber orientation distribution function fODF $S_{in}(\mathbf{b}) = \int_{\hat{n}=1} d\hat{n} P(\hat{n}) K(b, \hat{n}.\hat{g})$. The extra-neurite space is modeled as an isotropic or anisotropic diffusion tensor.

Soma and neurite density imaging (SANDI). [Palombo *et al.*, 2020] Introduced recently, SANDI is another important microstructure model that generalizes the standard model. The particularity of this model is that it accounts explicitly for the soma compartment and has a soma-specific signal term. It relies on some hypotheses, such as the soma having spherical shapes

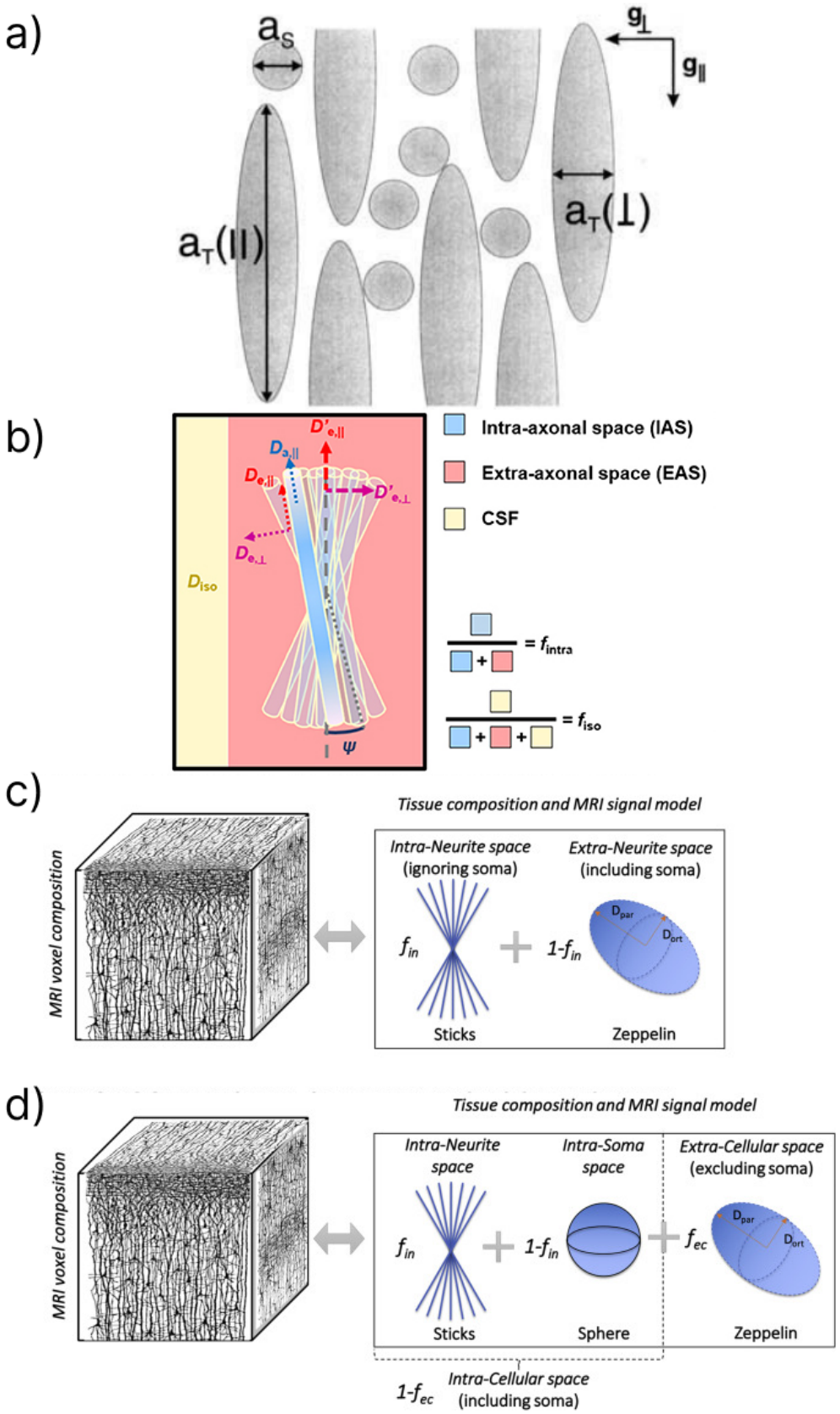

Figure 3.2: *Illustration of some of the dMRI analytical models. a) Tissue model adopted in [Stanisz et al., 1997]. b) Illustration of the NODDI model. c) Schematic of the standard model (SM)[Palombo et al., 2020]. d) Schematic of the SANDI model[Palombo et al., 2020]. (image adapted from the cited references).*

with a given radius r_s and that there is no exchange between soma and neurites. The SANDI model has the following form:

$$\begin{cases} S(\mathbf{b}) = (1-f_{ec})(f_{in}S_{in}(\mathbf{b}, D_{in}) + (1-f_{in})S_{is}(\mathbf{b}, D_{is}, r_s)) + f_{ec}S_{ec}(\mathbf{b}, D_{ec}) \\ S_{in}(\mathbf{b}, D_{in}) \simeq \sqrt{\frac{\pi}{4bD_{in}}} erf\left(\sqrt{bD_{in}}\right) \\ S_{ec}(\mathbf{b}, D_{ec}) \simeq \exp\left(-bD_{ec}\right) \\ S_{is}(\mathbf{b}, D_{is}, r_s) = \text{Sphere theoretical signal (explicit formula in [Palombo } et\ al.\text{, 2020])} \end{cases}$$

SANDI treats the case of the short diffusion time regime in which the water membrane exchange can be neglected.

Neurite exchange imaging (NEXI). [Jelescu *et al.*, 2022] This is another analytical model that was introduced to fill some gaps arguably uncovered by the standard model. This work underlines the importance of covering some important aspects such as the water exchange across the cell membranes, the inclusion of the signal contribution from the soma compartments, and non-Gaussian diffusion along processes.
The dMRI signal models listed above do not form an extensive list of all the analytical signal models. There are other works and analytical models such as Cellular exchange imaging [Gardier *et al.*, 2023] (CEXI), or NODDIDA which is close to NODDI with density imaging [Jelescu *et al.*, 2015].

Despite the variety of dMRI signal models, this analytical approach still has many drawbacks such as the requirement of high amplitude gradients inaccessible in clinical MRI systems and difficulties related to non-linear fitting of the signal models. In some cases, analytical models require an extensive amount of data and do not provide satisfactory fitting results. Also, some of these models suffer from solution degeneracy, which consists in the existence of multiple biologically plausible solutions for the observed diffusion signal data. The existence of such degenerate solutions makes it difficult to discriminate and accurately estimate the parameters of the brain microstructure. Moreover, models can require some of their parameters such as the diffusivity values to be fixed. Consequently, any bias in the numerical values of the fixed parameters will propagate to the fitted microstructure parameters. In addition, the analytical models are only valid for a specific regime of diffusion, i.e., short or long diffusion times, and this does not allow to capture information simultaneously from both regimes. Given these multiple limitations, analytical models are still important to develop a basic understanding of the dMRI signal and to validate/invalidate hypotheses about the brain tissue microstructure.
The next section presents an alternative approach based on simulations which complements the analytical modeling approach and provides more insights about cases not considered in the different analytical models.

3.2 Diffusion MRI simulators: geometry and signal

After the development of an analytical model, one needs to validate it and ensure it provides reasonable estimates for the brain microstructural parameters. For this purpose, relying on simulation tools and methods offers several advantages. Simulations provide an economical way to probe microstructure as validating dMRI on real biological tissue can be very expensive and time-consuming. One has to account for the cost of acquisition of the dMRI signal using an MRI imaging system in addition to the important cost of extracting microstructural features from real tissue which requires advanced microscopy techniques and a lot of labor. This section overviews the simulation approach encompassing virtual geometry and dMRI signal simulations.

3.2.1 Simulating brain tissue membrane geometries

Microstructure geometry modeling is an important part of decoding microstructure properties from diffusion MRI signals. This section starts with a brief introduction to virtual geometry and its different representations and then reviews the methods to synthesize microstructure geometries as well as the available datasets within the dMRI community.

3.2.1.1 Geometric representations

There are different ways to represent a virtual 3D geometry. Each representation has its own advantages and drawbacks. This section lists some of the geometric representations of 3D geometries that are widely used (see Fig.(3.3)).

Polygon mesh. It is the most widely used geometric representation. The polygon mesh represents a 3D surface using a set of polygons. Each polygon consists of a set of points and edges. An edge is a pair of points. The polygon approximates a small portion of the 3D geometry surface, representing a face. Thus, the polygonal mesh is just a collection of faces. There are different types of polygonal meshes, each type is characterized by the type of polygon used to represent a face. For example, quad meshes are a type of polygonal mesh such that each face of a quad mesh is a quadrilateral polygon. Another example is the widely used type of mesh, the triangle mesh. In this case, each face of the mesh is represented by a triangle.

Voxelized geometry. This is another method to represent 3D geometries. In the same manner as pixels represent units of data of an image, voxels represent cubic units for 3D geometries. Thus, voxels form a volumetric representation in contrast to polygon meshes which are surface-based representations. Essentially, a portion of the 3D space is subdivided into a regular grid, i.e. a Cartesian grid where each element of the grid is a cube, such that each grid element has a numerical value. These numerical values could represent occupancy data, i.e. binary data, or continuous numerical values as in the case of computed tomography (CT) scans. For a cubical Cartesian grid of size N, the number of voxels is N^3. Voxelized geometries offer different advantages such as ease of manipulation and simplicity, but come with some limitations such as the boxy-like geometry which reduces the realism of the represented geometry unless a very high resolution voxel unit is used.

A set of elementary geometries. It is not as much used as the voxelized or meshed geometries. This type of representation relies on the use of elementary geometries. For example, representing

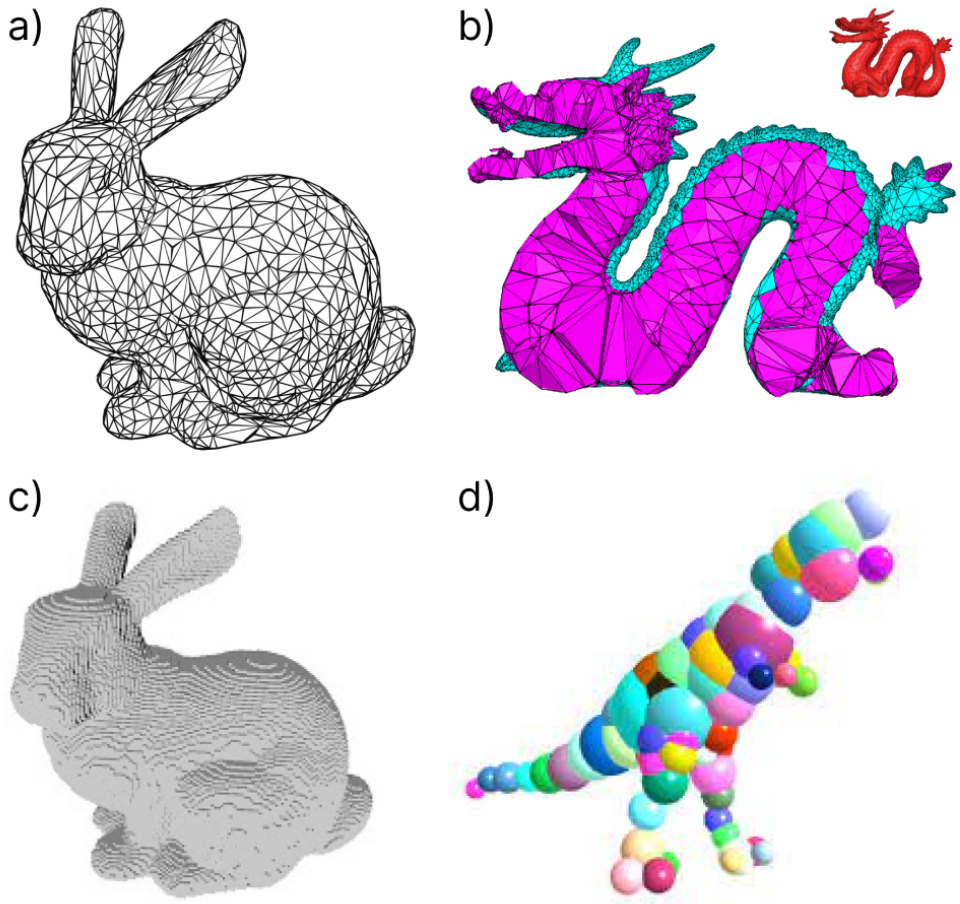

Figure 3.3: *Illustration of some of the geometric representations relevant in the context of this work. a) The Stanford bunny triangle mesh. b) Overview of a section of tetrahedral mesh with the original geometry in red. c) The voxelized version of the Stanford bunny. d) A 3D geometry represented by a set of spheres (Images are taken or adapted from [Rossi et al., 2021, Berlin, 2024, Walsh, 2016, Wang et al., 2006]).*

a 3D geometry as a set of spheres. Such a representation is of major interest in time-intensive applications such as collision detection[Wang *et al.*, 2006]. Such representation can suffer from a reduced representation quality of the original geometry's surface.
The field of computer graphics offers a diverse range of geometric representations. However, this section was limited only to the relevant representations related to this thesis.

3.2.1.2 Synthetic geometries

In this section, we present the tools and frameworks developed within the dMRI community to generate virtual cells or brain tissues. Since there are no closed analytical formulas for the dMRI signal stemming from complex multicellular environments, these tools are important to decode microstructure since they allow the validation of different hypotheses. For example, hypotheses as to how would diffusion MRI signals change in the presence of water exchange across the cellular membranes, or how would the diffusion MRI signals change when realistic brain cells are used compared to packed straight cylinders. These important questions can be tackled using synthetic

microstructure substrates and diffusion MRI simulations.
Synthetic geometries can be roughly grouped into the **white matter** or the **gray matter** virtual geometries.

White matter numerical phantoms. Different methods have been proposed to synthesize white matter models.

- A first set of methods was dedicated to generating simple synthetic substrates such as non-overlapping cylinders of various radii [Balls & Frank, 2009]. These cylinders form axon bundles and can be straight or curved. In addition to the cylinders, ellipsoid geometries were randomly placed in the 3D scene. The work relies on mesh-based representations.

- **Fiberfox** [Neher *et al.*, 2014] is an open-source tool that allows the generation of arbitrary artificial white matter fiber tracts. This work demonstrated the generation of fiber tracts with various configurations: twisting, fanning, kissing, and crossing fibers.

- **Diffusion Microscopist Simulator (DMS).** This tool allows the creation of a wide range of virtual geometries such as beaded axons, bending axons, spheres, ellipsoids, and star-shaped cells[Yeh *et al.*, 2013, Yeh, 2011]. This tool represents an all-in-one dMRI simulation toolkit, as it can be also used to simulate dMRI signals from the virtual substrate meshes.

- The **MEDUSA** toolkit [Ginsburger *et al.*, 2019] proposes a novel method to generate realistic white matter virtual tissues that don't use the mesh-based representation. Instead, it uses a set of highly overlapping spheres to represent fibers. This approach allows fast construction of white matter phantoms containing fibers and glial cells such as astrocytes and oligodendrocytes. In addition, the creation process uses distribution parameters of interesting microstructural features of white matter such as axon average radius, g-ratio values, beading, etc.

- Another method **ConFig** [Callaghan *et al.*, 2020] allows the generation of realistic white matter numerical phantoms. The generation mechanism mimics natural fiber genesis and consists of growing fiber meshes individually. In addition, ConFig also relies on interesting microstructural features such as the density of fibers in a 3D scene, the average fiber radius, and the orientation dispersion of fibers. This work claims to generate fiber tracts reaching over 70% of volume fraction. This approach also relies on a mesh-based representation for the virtual substrates.

- The framework **Cactus** was proposed recently [Villarreal-Haro *et al.*, 2023], it allows the generation of highly realistic white matter microstructure substrates reaching large volume fraction ($\sim$ 95%) and large voxel sizes ($\sim$ 500 μm^3). Cactus relies on mesh-based representations for the virtual substrates. It also uses interesting generative white matter microstructural features such as angular dispersion, diameter distribution, and volume fraction.

Gray matter numerical phantoms. While different methods have addressed the synthesis of white matter substrates, only few frameworks have attempted generating virtual substrates

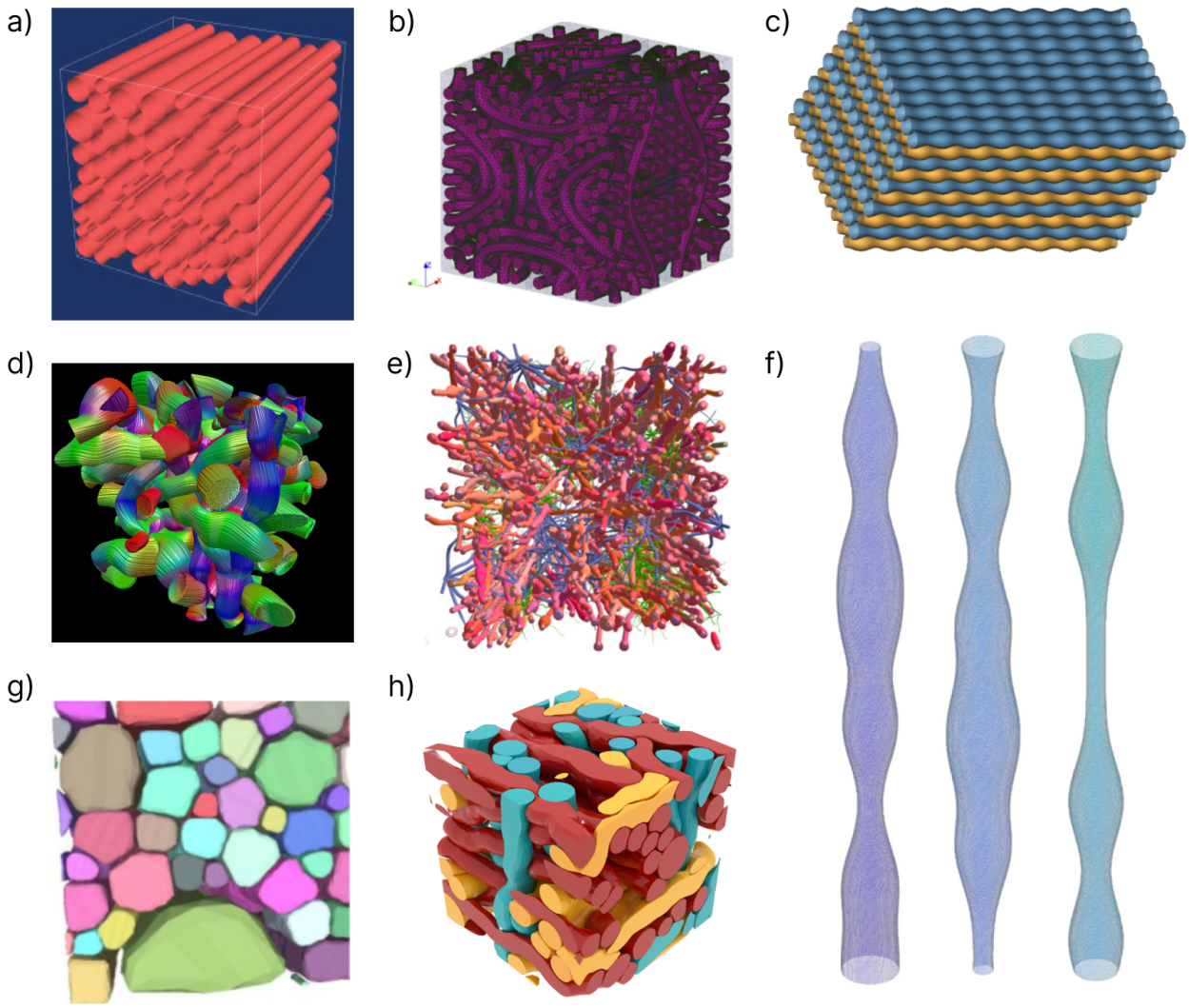

Figure 3.4: *Illustration of different generation methods of white matter substrates. a) White matter representation as simple cylinders. b) Hand-crafted white matter substrates represented as curved cylinders. c) A set of parallel fibers with beading from the DMS toolkit [Yeh, 2011]. d) A sample substrate from the Fiberfox method. [Neher et al., 2014] e) A sample substrate from the Medusa framework [Ginsburger et al., 2019]. f) A set of individual fibers with a varying radius [Lee et al., 2020]. g) A high volume density fiber tract from the CACTUS method [Villarreal-Haro et al., 2023]. h) A sample from the ConFig method [Callaghan et al., 2020].*

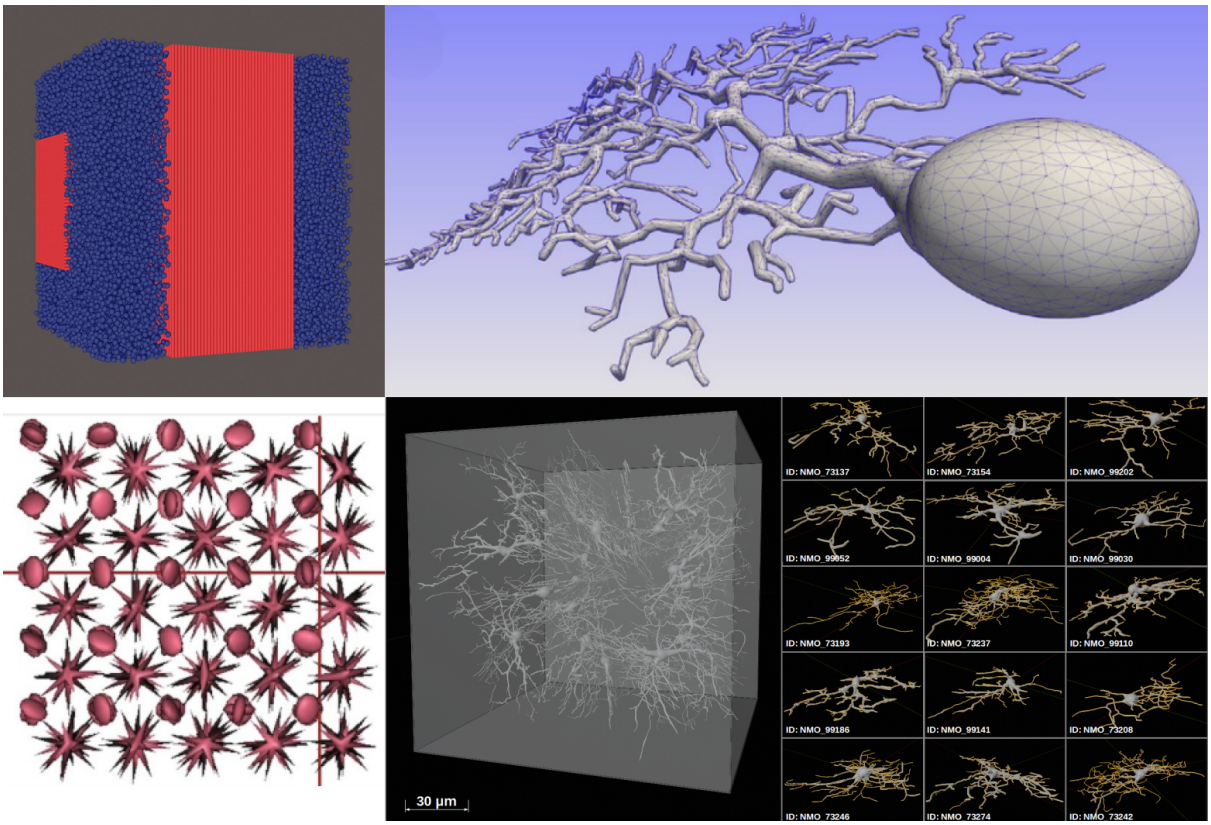

Figure 3.5: *Illustration of different potential generation methods for virtual geometries of the gray matter. The top left panel represents random ellipsoids with fibers [Balls & Frank, 2009]. The bottom left is an illustration of a virtual substrate generated with* ***DMS*** *[Yeh, 2011]. The top right is a synthetic virtual neuron presented in [Palombo* et al.*, 2019]. Finally, the bottom right represents a population of monkey glial cells presented in [Döring* et al.*, 2021]. The images are taken or adapted from the cited references.*

for the gray matter. The gray and white matter microstructures are significantly different from each other. The gray matter, as mentioned previously, contains a collection of neurons. These cells are composed of cell bodies and synapses, and the cell bodies are usually large compared to synapses and have a rounded shape. The synapses are small and split into multiple thin branches, occupying a large part of space. This makes the task of modeling gray matter quite challenging and raises several questions: What parameters should be used to model neurons? How to model a 3D scene of neurons? What geometric representation to use?

The list below provides some proposed work to generate gray matter virtual substrates from simple geometries to more complex ones:

- Already presented in the white matter section, this study [Balls & Frank, 2009] uses simple non-overlapping ellipsoids with arbitrary orientations mixed with parallel cylinders. The work however did not specify what kind of cell the ellipsoids were modeling.

- The framework **DMS** [Yeh, 2011] showcases 3D scenes containing ellipsoids and star-shaped cells. This also marks an important step in modeling gray matter virtual substrates.

- A method to generate realistic virtual neurons was presented in [Palombo *et al.*, 2019]. The generative model comprises different parameters such as the number of projections, i.e.,

the number of branches leaving the soma, the number of successive embranchments, the branch length, and other parameters. The use of specific morphometric statistics along with meshing techniques allows the generation of highly realistic virtual neurons.

- Another interesting approach presented in [Döring *et al.*, 2021] uses monkey glial cells available in open datasets to construct 3D scenes of virtual microstructure. It randomly places the cells and uses Blender [Community, 2018] to remove collisions between different cells.

An overview of some of the methods used to generate virtual gray matter microstructures is illustrated in Fig.(3.5).

3.2.1.3 Microscopy-derived neurons datasets

A direct approach to model gray matter microstructure is to use real cells acquired from real brain samples. For instance, reconstructed models of real neurons found in brain samples, or even fully reconstructed gray matter microstructure with all its components. This section focuses mainly on human-centered datasets; however, reconstructed cells from mice or monkeys are frequently used by the dMRI community. Some of the known datasets are listed below:

- **NEUROMORPHO dataset.** It is a large, curated open dataset of reconstructed neurons and glial cells from different species including mice, rats, and humans [Ascoli *et al.*, 2007]. The cells come in a special format ".SWC" that describes the neuron as a tree with parent and child nodes. Each node has a position, a radius, and a parent except for the cell body -the soma- which has a capsule-like geometry. These data can be used to reconstruct mesh-based representations of the virtual neurons.

- **H01 dataset.** It is, to the best of our knowledge, the largest and most detailed dataset of human brain microstructure [Shapson-Coe *et al.*, 2021]. It contains a nano-metric reconstruction of a small dissected human brain sample from the temporal cortex. The dataset was collected using electron-microscopy techniques with artificial intelligence to segment its various cellular components. The dataset is browsable online at `https://h01-release.storage.googleapis.com/explore.html` where it is possible to visualize detailed brain cell meshes in Neuroglancer, an interactive, user-friendly interface for visualization. The H01 dataset contains different cell types as neurons, interneurons, glial cells, axons, and myelinated axons.

This section covered the available datasets for virtual microstructure. It is noteworthy to mention the few advantages and drawbacks of each dataset. First, NEUROMOPHO provides a very sparse representation of the cells through the ".SWC" format, which offers computational advantages in terms of storage and processing of such geometries, but some drawbacks such as lack of realism. For example, the cell bodies in NEUROMORPHO are capsule-shaped. In contrast, the H01 dataset contains fine details of the brain cells with highly detailed geometries for the cell bodies. However, this comes at the cost of having large meshes, which require more storage volume and more computing resources for any kind of processing.

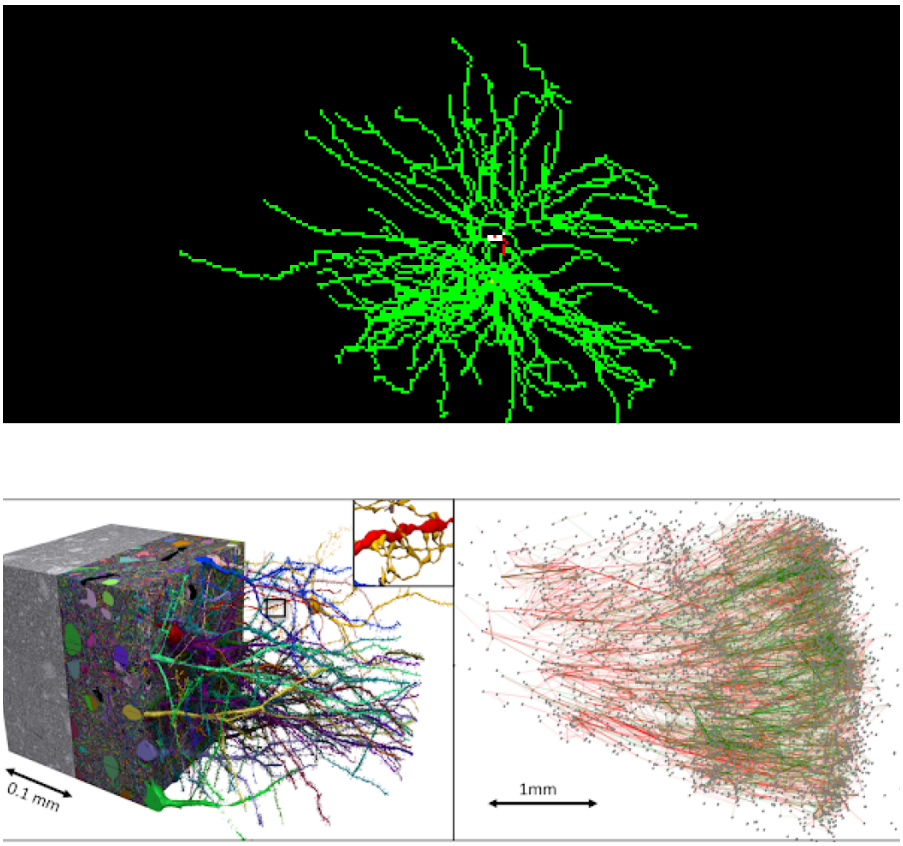

Figure 3.6: *Illustration of some of the available open neurons datasets. The top panel depicts the skeleton of a neuron taken from the NEUROMORPHO dataset (image adapted from https://neuromorpho.org/). The image in the bottom panel shows reconstructed samples at two different scales. The bottom left shows a detailed reconstruction with segmented cells with different colors, and the bottom right shows a reconstruction at a larger scale with excitatory neurons in green and inhibitory neurons in red (Image taken from https://research.google/blog/a-browsable-petascale-reconstruction-of-the-human-cortex/).*

3.2.2 Simulation of the dMRI signal attenuation

Given a virtual substrate, dMRI simulations are an ensemble of methods that, given a set of parameters, are able to compute the dMRI signal attenuation for the given substrate. The set of parameters required by a dMRI simulator is related to the diffusion medium of the substrate, and it also depends on the type of dMRI simulator used.
This section provides an overview of these simulators, their characteristics, and some of their implementation details.

Numerical methods for diffusion MRI simulations. This class of methods relies on numerical methods and schemes to solve the Bloch-Torrey partial differential equations (PDEs). Such approaches operate on the spatial discretization of the virtual geometries with the finite elements method (FEM). The Spindoctor Matlab tool [Li *et al.*, 2019] is a diffusion MRI simulator using numerical methods to synthesize diffusion MRI signals from a given substrate represented

as a tetrahedral mesh. The Spindoctor tool has been shown to outperform classical Monte Carlo simulators such as Camino in terms of run-time in some cases [Fang *et al.*, 2020]. However, one potential limitation of Spindoctor is its increasing run-time at large geometries, i.e., geometries with over one million finite element nodes.

Monte Carlo diffusion MRI simulations. Describe an approach to infer diffusion MRI signal attenuation from the Brownian motion of a large number of particles, i.e. spins, constrained to a given geometry called substrate. The substrate-restricted motion of each particle allows the computation of a phase contribution that is then used to calculate the diffusion MRI signals. There are different parameters required for MC simulations such as the diffusivity, the number of spins, the particle-membrane interaction type, the permeability of the substrate, etc. To explain the principles behind this approach, we recall the expression of the phase shift in Eq.(2.25), which implies a phase shift resulting from the self-diffusion of a spin. The relation between the spin displacement and the corresponding phase shift is inverted in the context of Monte Carlo diffusion MRI simulations. Thus, for a given spin's synthetic random displacement and a gradient $\mathbf{G}$, the phase shift can be computed according to Eq.(2.25). In practice, different spins are generated in a given substrate. Then, a small displacement is generated for each spin according to Eq.(2.16). More precisely, a displacement of $\delta\mathbf{r}$ is generated according to:

$$\delta\mathbf{r} = \sqrt{2nD.\delta t}\ \hat{\mathbf{e}}_r \tag{3.4}$$

Where the scalar term $\sqrt{2nD.\delta t}$ in Eq.(3.4) is the root of the MSD in Eq.(2.16), it is also called the expected mean displacement or the step-length in the context of dMRI simulations. With n being the dimension of the space in which the simulation occurs, in the course of this thesis, the simulations occur in 3D space ($n = 3$), D the diffusivity of the medium in m^2/s, δt the simulation's time step, and finally, $\hat{\mathbf{e}}_r$ a random orientation such that $||\hat{\mathbf{e}}_r|| = 1$. The spin's phase shift at a time $t_i = i\delta t$ is given by:

$$\Delta\phi_i = \gamma\mathbf{G}(\mathbf{r}(t_i), t_i).\mathbf{r}\delta t \tag{3.5}$$

The relation in Eq.(3.5) allows the computation of the phase shift for each spin given its position $\mathbf{r}$ and the gradient $\mathbf{G}(\mathbf{r}(t), t)$. Now that the single phase shift contribution of each spin is calculated, the signal attenuation is computed according to:

$$\frac{S}{S_0} = \frac{1}{N_s}\sum_{n=1}^{N_s} \exp\left(-i\sum_{j}^{N_t}\Delta\phi_j\right) \tag{3.6}$$

The term N_s is the number of spins used in the simulation, N_t is the number of time steps of the simulation, and $\Delta\phi_j$ is the spin's phase shift provided in Eq.(3.5).

A simple sketch of a Monte Carlo diffusion MRI simulation algorithm is provided in Alg.(1, 2) [Rafael-Patino *et al.*, 2020]. Given a number of spins N_s, a number of time-steps N_t, and a number of gradient acquisitions N_G, each spin's position is updated using an *UpdateSpinPosition* method Alg.(2). When the spin doesn't collide with the substrate or with the boundary of the simulation space, the spin's position is incremented with the displacement vector $\delta\mathbf{r}$ defined in Eq.(3.4). Otherwise, the collision is handled with a *handleCollision* method. After updating the spin's position, the total phase shift is calculated using the new spin's position using Eq.(3.5).

```
1:  procedure BasicSimulation
2:      N_s ← Number of spins
3:      N_t ← Number of time-steps
4:      N_G ← Number of acquisitions
5:      for each N_s spins do:
6:          for each N_t time step do:
7:              UpdateSpinPosition(...)
8:          for each N_G acquisition do:
9:              UpdateTotalDephase(...)
10:     for each N_g acquisitions do:
11:         ComputeDWSignal(...)
```

Algorithm 1: Monte Carlo diffusion MRI simulation algorithm sketch [Rafael-Patino *et al.*, 2020].

```
1:  procedure UpdateSpinPosition
2:      Ω ← List of Obstacles
3:      do
4:          for each obstacle in Ω do
5:              checkForCollision(...)
6:              if collision detected then
7:                  handleCollision(...)
8:      while collision is detected
```

Algorithm 2: Complementary Particle Position Update method [Rafael-Patino *et al.*, 2020].

Review of MC dMRI simulators. Numerous scientific studies have proposed new MC simulators along with different use cases involving MC simulations. Diffusion MRI simulators come in different flavors, and they differ from each other by the kind of geometric representation used for the microstructure substrates. For example, some dMRI simulators use mesh objects to model synthetic substrates, while others rely on voxel-based geometries. An overview of these simulators is provided below:

- **Camino** [Hall & Alexander, 2009] is a Java-based software toolkit for diffusion MRI processing and includes a diffusion MRI simulator.
- **Diffusion Microscopist Simulator (DMS)** Implemented in C++, DMS not only synthesizes virtual mesh-based substrates but also provides a tool to simulate dMRI signals using the generated geometries. It uses CPU-based parallel processing to accelerate the run time of simulations.
- In this thesis [Ginsburger, 2019, p. 127], the DMS dMRI simulator was extended and rewritten both in *CUDA* and in C++ using the *Kokkos* framework [Trott *et al.*, 2022] to enable code portability. In the same work, the different versions of the simulator were benchmarked using a white matter substrate generated with MEDUSA on different workstations, i.e., with both CPUs and GPUs.
- This study [Nguyen *et al.*, 2018] presents a Graphical Processing Unit -GPU-based- MC dMRI simulator using voxel-based geometries as input. A benchmark was performed using both a mesh-based and voxel-based substrate. For the mesh-based case, the work relied

on the octree data structure and radius-search algorithm to speed the simulation runtime, i.e. faster collision checks of particles with membranes. As for the voxel-based case, voxel coordinates were used to check if the particle remained inside the virtual substrate. The same work presented a dMRI simulation runtime benchmark showcasing a speedup compared to the Camino simulator. Finally, the same work showed that for a given substrate, the fastest simulation used the voxel-based representation of the substrate with a GPU implementation.

- **MCDC** [Rafael-Patino *et al.*, 2020] is another open-source C++-based diffusion MRI simulator. It relies on mesh-based substrates and allows the use of multiple threads. The same work proposes a range of optimal parameters for diffusion MRI simulation studies. The code is open-sourced in the following Git repository: `https://github.com/jonhrafe/MCDC_Simulator_public`.

- **Disimpy** [Kerkelä *et al.*, 2020] proposes a massively parallel Monte Carlo diffusion simulator implemented in Python using GPU to generate spin trajectories. Disimpy demonstrates a run-time performance gain over Camino as it relies on GPUs, and Disimpy relies on mesh-based geometries. The Disimpy code is available on `https://github.com/kerkelae/disimpy`.

- **Realistic Microstructure Simulator (RMS)** [Lee *et al.*, 2021] proposes a promising approach to simulate diffusion MRI signals from realistic microstructure voxelized substrates. Such substrates can be obtained from the segmentation of cells from microscopy images. RMS is based on CUDA-C++ and allows accelerated computations using GPUs, and in contrast to other simulators, it relies on voxel-based substrates which offer advantages like simplicity of reflections and reduced computational loads, but also come with limitations such as "boxy" geometries and dependence of the step length of the voxel geometry resolution.

The MC approach differs from numerical methods since it allows to simulate dMRI signals for multiple gradients at once.

3.3 Overview of simulation-based studies

This section is dedicated to simulation-based studies, it showcases the importance of simulations in the study of brain microstructure as well as in helping to answer challenging research questions. In [Hall & Alexander, 2009], the number of spins and time steps were investigated for optimal MC dMRI simulations. The work introduced the complexity variable $U = N_s.N_t$, which is the product of the number of spins with the number of time steps. In the case of this study, a complexity value of $U = 10^8$ was shown to provide reasonably accurate simulation results. It was also demonstrated that larger complexity values, i.e $U > 10^{10}$, don't necessarily improve the accuracy of the results. However, one has to keep in mind that these parameters depend also on the complexity of the substrate. Through an extensive set of simulation experiments, the investigation conducted in [Rafael-Patino *et al.*, 2020] confirmed the range of the simulation time steps and the number of particles to be used for reasonably accurate simulation results. The

same study explored the effect of curved axons on the dMRI signal and how it impacted axon radii estimates. It showed an increasing radius estimate bias with the curvature amplitude of the substrates. The same work demonstrated a substrate generation method for white matter axons with a crossing configuration, the method allowed the generation of a mesh-based substrate of 2808 axons totaling around 864k vertices and 1.69M faces in a total duration of about 42 hours of which a large proportion was spent on optimization steps.

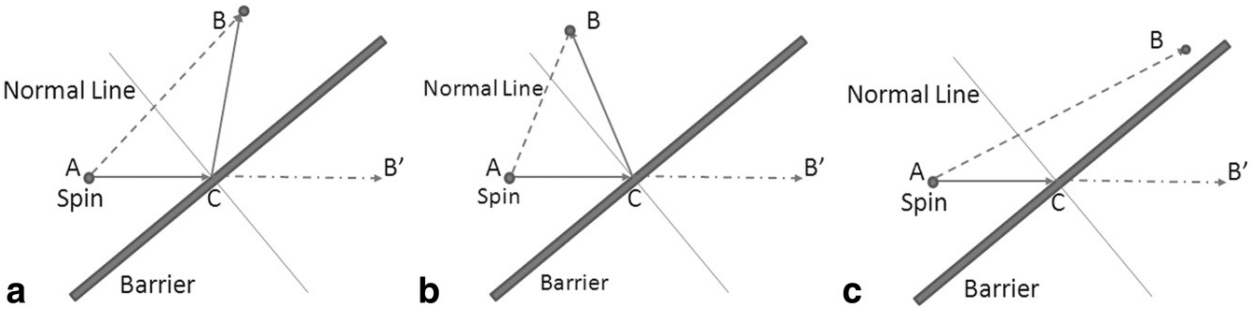

Figure 3.7: *Illustration of different particle-membrane interaction types. a) The conventional elastic boundary reflection (EBR). b) The non-elastic reflection such that the lengths AC and CB add up to give the same length as AB'. c) The equal-step random leap (ERL), re-samples the step length vector when a collision is detected. (The image is taken from [Xing et al., 2013]).*

In [Xing *et al.*, 2013], different particle-membrane interaction types were considered, and their effect on the simulated signals was investigated. The study listed three particle-membrane interaction types: Elastic reflection, non-elastic reflection, and the equal-step random leap (ERL) as illustrated in Fig.(3.7). The elastic reflection is the commonly used particle-membrane interaction type. It consists of reflecting the spin using the normal vector to the surface at the collision point. The non-elastic reflection consists of returning to the original side of the particle with a random displacement. Finally, the equal-step random leap, also called rejection sampling, consists of resampling the step-length vector every time a particle-membrane collision is detected. This process is repeated until the new step does not collide with the substrate's membrane. The work in [Callaghan, 1994] did a benchmark using a cylinder and showed consistent results with the analytical solution when different particle-membrane interaction types are used. The same study mentioned that the time step of the simulation, hence the step-length, should be selected carefully when using the ERL method to ensure accurate results in agreement with the EBR. Note that the use of ERL can be advantageous since it does not require to perform reflection calculations, therefore providing a slightly more efficient way to perform dMRI simulations. It is also important to notice that the range of the step length values for which the simulation are accurate may not be the same when a more complex substrate is used and this study only investigated the case of a cylindrical substrate.

Another important aspect of the dMRI simulations is the handling of boundary conditions. In other words, how the spin behaves when encountering the bounding box of the substrate. There are different ways to handle boundary conditions. The first one is the periodic boundary condition. It consists of spawning a spin that has crossed the boundary from the other side of the same boundary. This results in a simulation environment where the substrate is repeated infinitely. The periodic boundary condition is used in the Camino and Disimpy simulators. An-

other approach relies on reflecting boundary conditions. This consists of reflecting the spin when encountering the substrate's boundary. This implies that the substrate is repeated by its reflective copies. This type of boundary condition can result in an increased fiber orientation dispersion [Fieremans & Lee, 2018]. The reflective boundary condition is also implemented in the Disimpy simulator. Finally, a last approach for boundary conditions consists of using a larger virtual geometry and simulating the diffusion process only in the central sub-volume of the geometry. The signal is only calculated from the trajectories that remained inside the central sub-volume. Therefore, when considering large diffusion times during which an important proportion of the particles leave the central sub-volume, the dMRI results may not be accurate. This method is more adapted for short diffusion times.

The studies mentioned until now focused on specific technical aspects of dMRI simulation frameworks. Other studies that investigated various aspects beyond simulation parameters, such as various microstructure geometries. For instance in [Lee *et al.*, 2020], simulation-based methods were used to study the impact of realistic axonal shapes on the dMRI signal. It developed a theory for the intra-axonal dMRI signal in the presence of beading and undulations. The theory was validated using MC simulation on both synthetic and realistic numerical phantoms of axonal fibers reconstructed from electron microscopy images of a mouse brain. The same study showed that the presence of undulations and beading leads to overestimated values of the effective axonal radius of the axon. Therefore, it highlighted the importance to account for fine geometric details when developing models estimating axon radii.

Another interesting research investigated the resolution limit of axonal diameter that can be detected with realistic MR systems accessible in a clinical setting [Nilsson *et al.*, 2017]. In this work, the resolution limit was defined as the minimal axonal radius for which the dMRI signal can't be distinguished from a signal stemming from a virtual axon with a null diameter, i.e. a stick. This study considered a simple model of axons as straight impermeable cylinders and relied on MC dMRI simulations to investigate the resolution limit and validate the theoretical model. Some of the results of this work are shown in Fig.(3.8) and table (3.1).

$g_{max}[m\mathrm{T/m}]$	$\mathrm{B_0}$[T]	$\mathrm{d}^{\parallel}_{min}$ [μm]	b-val [$ms/\mu m$]
80	3	3.3	20
300	3	1.7	260
60	7	3.5	2.4
80	7	3.0	4.5

Table 3.1: *Numerical values of the resolution limit $d^{\parallel}_{min}$ for parallel axons for typical clinical MRI imaging systems (Adapted from [Nilsson* et al.*, 2017]).*

Another important class of works relied on a computational approach, i.e., an approach that relies on simulations and data-driven methods. In this work [Nedjati-Gilani *et al.*, 2014], machine learning methods were applied to construct a computational model from MC simulation data. This model was used to infer microstructural parameters from dMRI signals. The substrates used in the same work consisted of parallel cylinders of different radii that do not overlap. Simulated dMRI signals from the virtual substrates were then used to train a random forest model to esti-

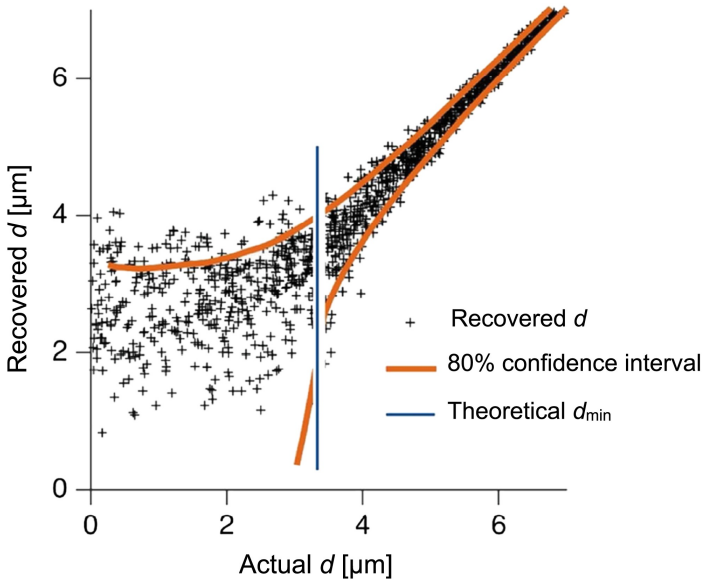

Figure 3.8: *Illustration of the estimated diameter compared to the actual diameter with a vertical line in blue showing the theoretical limit at* $d_{min} = 3.3\mu m$ *for a substrate with parallel cylinder and a PGSE sequence with* $\delta = \Delta = 40ms$ *(Image taken from [Nilsson* et al.*, 2017]).*

mate microstructural parameters such as volume fraction, parallel diffusivity, and residence time. In the same line of methods, this work [Fang *et al.*, 2023] used virtual reconstructed neuron meshes to simulate dMRI intracellular signal using the Spindoctor Matlab toolbox [Li *et al.*, 2019]. The dMRI signals were aggregated to form a large dataset of 1.45 million artificial brain voxels containing tissue microstructure parameters and their corresponding dMRI signals. This dataset was then used to train supervised learning models. More precisely, simple neural networks known as multilayer perceptions (MLPs) were trained to estimate tissue microstructure parameters from dMRI signals.

3.4 Conclusion

This chapter overviewed the different methods to study and estimate microstructure. It distinguished between analytical and simulation-based approaches. In summary, analytical approaches use strong hypotheses on tissue microstructure by considering simple cases for which the analytical expression of the signal is known. In contrast, simulation-based approaches offer a more flexible approach. However, this flexibility comes with a cost associated with the numerical aspect. This cost is translated to computational efficiency challenges in terms of runtime and memory efficiency, and most importantly, at a high level, a critical factor in any simulation-based approach is the simulation-to-reality gap: How directly can one translate a method from simulation-based models to a real-physical method.

Chapter 4

Contributions in virtual substrates

This chapter is dedicated to contributions in the course of this thesis to the modeling of virtual microstructure geometries. It covers different aspects from geometric representations to complete virtual synthetic brain voxels. First, a novel geometrical decomposition method was developed in the frame of this thesis. Then, a second contribution to generating virtual substrates of gray matter is presented. Finally, these results and contributions are discussed in a detailed manner in the context of the presented scientific literature.

4.1 Introduction

The ultimate ground truth for brain microstructure is histology, it provides a detailed description of the cells and their organization in a 3D voxel. Therefore, histology is likely the best way to validate models estimating microstructural parameters for diffusion MRI signals.
In order to learn tissue microstructure parameters, one of the most direct approaches is to have the following steps achieved:

1. Collect a large pool of brain tissue samples
2. Scan the samples to get the dMRI signals
3. Extract microstructure parameters from the samples
4. Learn a relation between the signal and the microstructure parameters

Despite its simplicity, the list of tasks provided above is very challenging. For instance, the first item of collecting a large pool of brain tissue samples can be intricate for different reasons such as the availability of postmortem brain tissue samples, the sample's tissue quality, and most importantly, the potential bias in the collected postmortem brain tissue samples such as age or diseases: are brain tissue of some age range over or under-represented? Are there frequent pathologies present in the tissue samples?
The second challenge is to have a scanning protocol and enough resources and time to scan all the tissue samples.
The third step is perhaps the most challenging as it requires measuring the tissue microstructure from a large set of brain tissue samples. To the best of our knowledge, the measurement of brain tissue microstructure requires expensive equipment and lots of labor. The current techniques for extracting brain tissue microstructure consist of using histology combined with microscopy

imaging techniques. Therefore, slicing brain samples at a large scale and scanning those slices can be very cumbersome.
Given these points, the alternative computational-based approach seems promising as it helps to eliminate some of the mentioned obstacles. For example, instead of using post-mortem brain tissue samples and scanning them with a real MRI imaging system, it is possible to use virtual brain samples and rely on simulations to extract dMRI signals. This offers a tremendous advantage if the used virtual geometry has a known set of tissue microstructural parameters.

4.2 Rethinking the geometric representation

As seen previously, different virtual substrate generation methods use different geometric representations, with the mesh-based representation being the dominant one across different methods.
The mesh-based representation. Despite its wide adoption within the dMRI community, the mesh-based representation has some drawbacks that limit its adoption to generating a large dataset of virtual brain tissue samples. Some of these limitations include memory-related bottlenecks and runtime efficiency.
For the memory-related limitations, a previous PhD thesis [Nguyen, 2014] made a clear statement outlining this obstacle when adopting of the mesh-based representation:

> *"The resolution of dMRI (order of 1mm3) is very large compared to cells features (from sub-micron for the diameter of neurites to tens of micron for the soma). If neurites are described by 3D meshes in a computational domain, one has to deal with an excessive number of elements and vertices to describe the structures accurately because the neurite diameter (order of µm) is much smaller than the neurite length (order of 1mm)"*
>
> (Dang Van Nguyen, 2014)

This quote underlines the memory challenge associated with the adoption of a mesh-based representation. This problem is even bigger for gray matter than axon-only white matter virtual substrates. In fact, cells exhibiting both large and small structures in a voxel tend to significantly increase the number of mesh vertices. Consequently, this leads to increased memory sizes. Such a large memory hinders the path toward a scalable synthesis of a large dataset of virtual brain tissue samples.
In the same direction, other works have outlined the same challenge. For instance, the CONFIG method can be used to synthesize white matter substrates [Callaghan *et al.*, 2020]. The generation process of fibers relied on sampling the space which could require a large amount of memory when large substrates are generated. In addition, the substrates generated are inside a voxel of a 20 μm size. In their paper, the authors mention the memory limits related to generating small fibers in a large voxel which is in agreement with the statement cited above.
Another class of mesh-based white matter phantoms was generated using the CACTUS method [Villarreal-Haro *et al.*, 2023]. This method allowed the generation of highly packed axons in voxels of relatively large sizes, i.e., from 100 to 500 μm. However, the generated white matter phantoms occupied a relatively large memory size of around 5 Gigabytes per sample.

For mesh-based related geometries, there are multiple file formats available within the computer graphics community. For the same geometry, some file formats may take more space than others, and this depends on the information stored in the file. For example, some file formats store only vertices and faces, while others store additional information such as face normals, i.e., the vectors normal to the faces. An illustration of the different formats is provided in Fig.(4.1).

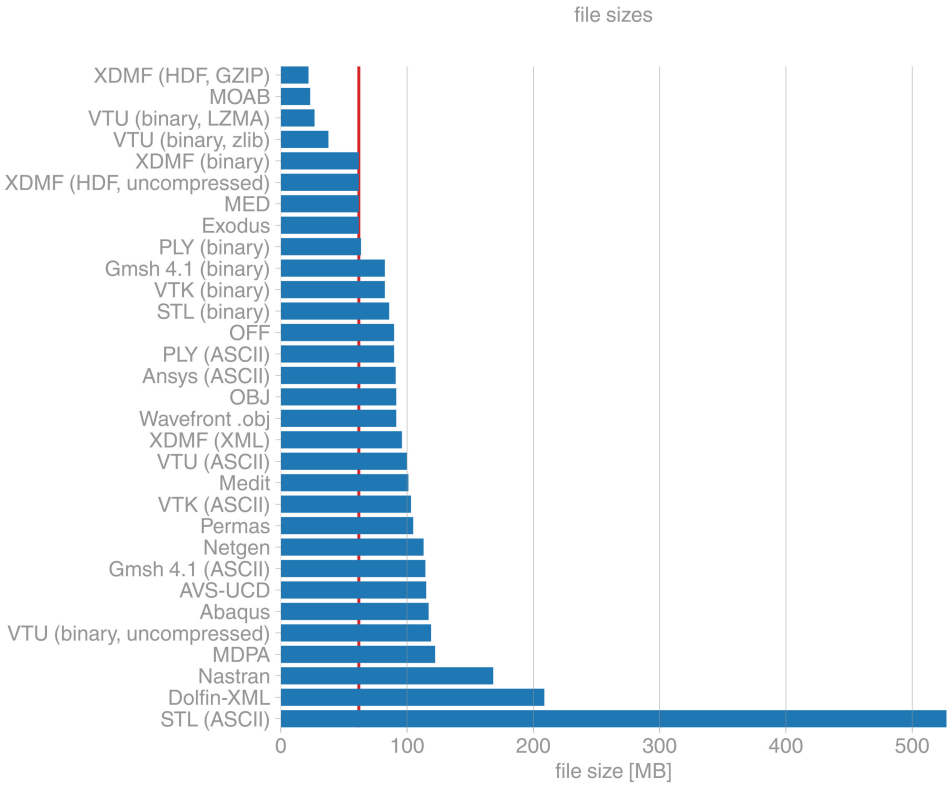

Figure 4.1: *Illustration of the different mesh file formats and their memory size with the red line indicating the file format for a mesh with around 900k vertices and 1.8 million faces (Image taken from the git repository [Schlömer, n.d.]).*

A rough estimation of the memory size of a mesh file can be obtained by simply counting the number of vertices $n_{\mathrm{v}} = |V|$ and faces n_{f}. For a triangle mesh representing a 3D geometry, the faces are represented by a triplet of vertex identifiers- a unique ID for each vertex. Therefore, if the vertex position is represented by three floating numbers and the vertex IDs are represented as a 4-byte unsigned integer. The memory size estimation M_{est} of the mesh in bytes is given by: $M_{est} = 12\,(n_{\mathrm{v}} + n_{\mathrm{f}})$. This relation shows the memory scaling with the number of faces and vertices. This could be a limitation when considering virtual substrates in large voxels or complex virtual geometries with fine details, which is the case of virtual histology. This point becomes more prominent in the next section.

Regarding the computational efficiency, the CONFIG method [Callaghan *et al.*, 2020] stated a generation time of about 6 hours. In contrast, for the Cactus framework [Villarreal-Haro *et al.*, 2023], it was stated that for the largest substrate with a squared voxel of size 500 μm the average generation time per axon was around one minute. Thus, such a large voxel contains around 33k fibers, totaling around 550 hours for the generation time, or, equivalently, around 3 weeks to generate a very large substrate.

The combined memory and computational efficiency challenges limit the scalability of mesh-based representation methods in generating a large collection of virtual brain tissue samples.

The voxel-based representation. This type of geometric representation is more frequent for virtual histology, i.e., voxelized cells stemming from microscopy data. To our knowledge, there are no synthetic methods directly generating voxelized microstructure substrates. However, some methods enable the translation from one geometric representation to another. For example, voxelization algorithms convert a mesh-based to a voxel-based representation. Inversely, the marching cubes method [Lorensen & Cline, 1987] allows the conversion of a voxelized to a mesh-based geometry.

Atom-based geometry. This type of geometric representation was previously proposed within the computer graphics community. For instance, this work [Bradshaw & O'Sullivan, 2004] proposed a method to represent meshes using hierarchies of spheres. The same work motivated the method with increased efficiency for detecting collisions. Although not widely adopted within the dMRI community, only a few substrate simulators rely on this geometric representation. The MEDUSA framework for example [Ginsburger *et al.*, 2019] relies on a set of spheres to represent the virtual microstructure tissues. Such an approach has demonstrated several advantages such as runtime efficiency as the substrate generation time is the order of hundreds of seconds using GPU. The mentioned runtime is for a typical voxel size of around 100 μm^3. Another interesting feature of this framework is its ability to generate virtual substrates with different cell types such as axons and astrocytes. It is also noteworthy to mention that other mesh-based methods rely partly on atom-based representations: this representation is only used in one of the generation steps. For instance, the Config method [Callaghan *et al.*, 2020] uses spheres, or more precisely, metaballs, which are implicit surfaces that can merge together to form smooth surfaces. Such metaballs are processed at a later stage to extract the final substrate's mesh. Another example is the Cactus framework [Villarreal-Haro *et al.*, 2023], which also relies on atom-based representation in the first steps of substrate generation. The geometric unit or "atom" used is the capsule geometry. Fibers in the Cactus framework are represented as capsules meshed after optimization and processing steps. Although presenting many advantages, the sphere-based geometric representation adopted in MEDUSA using a single sphere per cellular cross-section has a limitation related to biophysical realism stated in the original work in [Ginsburger *et al.*, 2019] as:

> *"The assumption of circular axonal cross-sections used in this paper might not stand, as suggested in (Xu et al., 2018; Abdollahzadeh et al., 2017; Lee et al., 2018), where elliptical cross-sections have been considered more plausible."*
>
> (Kevin Ginsburger, 2019)

As stated, the adoption of spheres as a unit of representation implies that the axons have circular cross-sections. However, from histology data, this assumption does not hold as shown in Fig.(4.2).

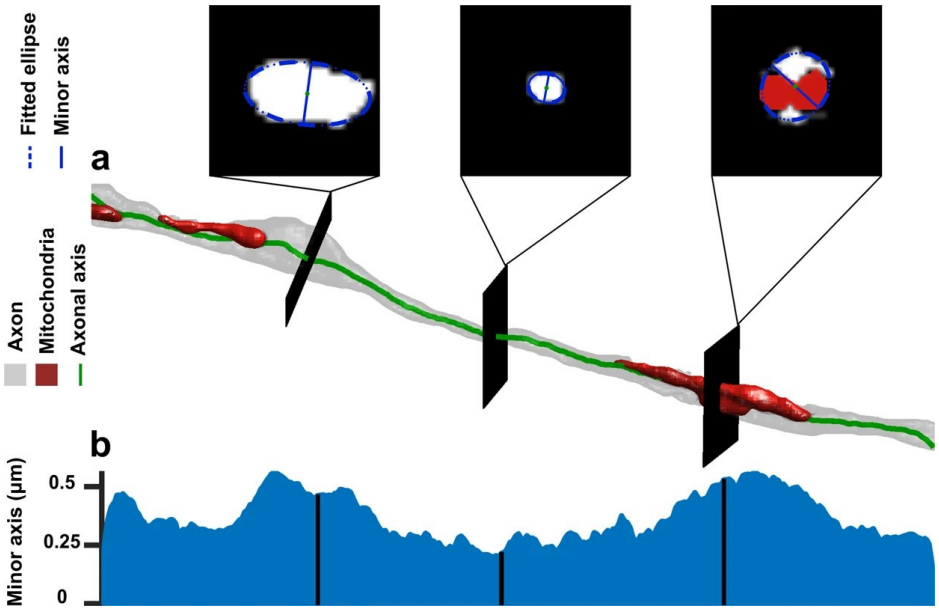

Figure 4.2: *Illustration of a segment of an axon with three different cross sections fitted with ellipses in panel a). The fitted ellipse minor axis value profile along the axonal segment is shown in panel b (Image taken from [Abdollahzadeh et al., 2019]).*

In summary, the mesh-based representation of substrates can be a limiting factor toward the scalability of virtual substrate synthesis and dMRI simulations. In addition, atom-based representations offer an interesting alternative to achieving the scalability of the substrate generation process. The first part of this thesis focused on enhancing the geometric representation by introducing a geometric decomposition method that converts a mesh-based geometry into an atom-represented geometry. Second, a reconstruction method was developed to obtain atom-represented cells from the open dataset NEUROMORPHO. Finally, the MEDUSA framework was extended to ellipsoids in addition to the prior spherical geometrical units.

4.3 The Skeleton-Guided Ellipsoid-atom Decomposition (SGEAD)

In this section, a new geometric method is presented. The development of this method is motivated by the limitation of other geometric representations presented in the previous section. The Skeleton-Guided Ellipsoid-Atom (SGEAD) is a geometric decomposition method that converts a mesh-based into an atom-based representation such that the geometric atom used is an ellipsoid. The use of ellipsoids is motivated by the elliptic cross-sections of axons as well as its memory efficiency when used to describe 3D geometries.

Many methods have been proposed in the geometry literature that aim to convert meshes to a set of ellipsoids. The work in [Liu *et al.*, 2007a] presents a method to approximate a mesh model with a hierarchical set of ellipsoids. Another method [Bischoff & Kobbelt, 2002] was proposed to approximate a mesh surface by a set of ellipsoids. The method consists in growing ellipsoids inside the mesh model and performing an optimization step to reduce the number of ellipsoids that highly overlap, therefore, demonstrating the ability of ellipsoids to compress a closed mesh model. Finally, the method [Liu *et al.*, 2007b] provides small modifications to the previous method [Bischoff & Kobbelt, 2002].

4.3.1 Methods

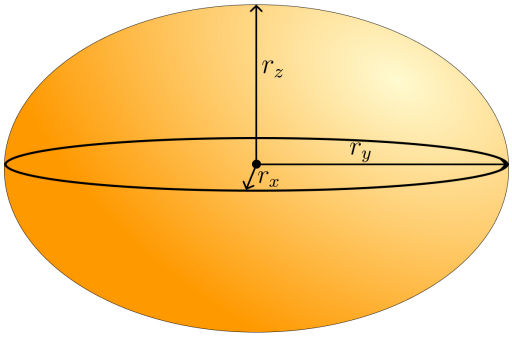

Figure 4.3: *A simple schematic of an axis-aligned ellipsoid with radii r_x, r_y, and r_z and a center represented by a black dot.*

Ellipsoids. An Ellipsoid is a geometrical generalization of the sphere. In the simpler case, when the ellipsoid is centered and axis-aligned, i.e., its main orientation follows the orthonormal vector of the 3D space, the ellipsoid can be described only by three parameters representing the different radii (r_x, r_y, and r_z) as illustrated in Fig.(4.3). However, in the most general case, when the ellipsoid has an arbitrary orientation and is not centered, it can be minimally described by a set of nine scalar parameters: its center coordinates, three parameters encoding the ellipsoid's orientation, and finally, three parameters for its radii values. Although a set of nine parameters is the most efficient to describe an arbitrary ellipsoid, this thesis uses a set of ten parameters instead. This choice is motivated by the ease of some ellipsoid-related calculations, such as evaluating whether a point belongs to a given ellipsoid, or whether an ellipsoid collides with another one or a 3D segment. The ellipsoid description using ten parameters is provided in the formula below, known as the algebraic equation of the ellipsoid:

$$Ax^2 + By^2 + Cz^2 + Dxy + Eyz + Fxz + Gx + Hy + Jz + K = 0 \tag{4.1}$$

The equation above has ten parameters $A, B, C, D, E, F, G, H, J$ and K such that the last parameter K is not a free parameter. When an arbitrary point (x_0, y_0, z_0) is evaluated using equation (4.1), the value obtained indicates if the point is located inside or outside the ellipsoid, i.e., negative values for points inside ellipsoid, the strictly positive values are points outside the ellipsoid, and finally, the points yielding null values lie exactly within the ellipsoid's surface. For the axis-aligned and origin-centered ellipsoid in Fig.(4.3), the equation (4.1) is simplified into:

$$\frac{x^2}{r_x^2} + \frac{y^2}{r_y^2} + \frac{z^2}{r_z^2} = 1 \tag{4.2}$$

By comparing the term in Eq.(4.1) and Eq.(4.2), it is possible to establish a relation between the coefficients A, B, C, D, E, F, G, H, J, K and the intuitive geometrical parameters of the ellipsoid: r_x, r_y and r_z. In the simple case of the axis-aligned and origin-centered ellipsoid, $A = \frac{1}{r_x^2}$, $B = \frac{1}{r_y^2}$, $C = \frac{1}{r_z^2}$, and $K = -1$.
It is also important to note that not all possible combinations of the parameters A, B, C, D, E, F, G, H, J, K will be an ellipsoid. In fact, the formula provided in Eq.(4.1) represents a

$$\mathcal{A} = \frac{1}{2}\begin{pmatrix} 2A & D & F & G \\ D & 2B & E & H \\ F & E & 2C & J \\ G & H & J & 2K \end{pmatrix} \quad X = \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Encodes ellipsoid main axes and radii

Encodes ellipsoid's center coordinates

Figure 4.4: *The general matrix representation of the quadratic surfaces with the focus on the ellipsoid's case. The ellipsoid main axes and radii are encoded in the sub-matrix outlined in red while the parameters outlined in blue encode the ellipsoid's center.*

general class of geometric surfaces known as quadratic surfaces. They include spheres, ellipsoids, hyperbolas, and parabolas. Each type of surface is obtained when a specific set of parameters A, B, C, D, E, F, G, H, J, and K is used. The same general formula in Eq.(4.1) can be re-arranged as follows:

$$X^t \mathcal{A} X = 0, \tag{4.3}$$

with $\mathcal{A}$ being the matrix containing the compressed parameters of the ellipsoid: The matrix $\mathcal{A}$ in Eq.(4.3) is illustrated in Fig.(4.4). Note that the matrix $\mathcal{A}$ is a 4x4 symmetric matrix: $\mathcal{A} = \mathcal{A}^t$. In order for the quadratic surface to be an ellipsoid, the upper left 3x3 symmetric sub-matrix of S should be positive-definite, i.e., the eigenvalues of the matrix selected in red in Fig.(4.4) should be strictly positive. The matrix eigenvectors are the ellipsoid's main directions, and their magnitudes are the squared inverses of the ellipsoid radii r_x, r_y, and r_z. The ellipsoid's center is encoded in the G, H, and J parameters as outlined in blue in Fig.(4.4).

The mesh skeleton. Also called a centerline, the mesh skeleton is a graph that represents the global shape of the underlying 3D geometry. It can be used for various purposes, such as shape marching, 3D animations, shape retrieval, and object recognition. The mesh skeleton is characterized by a set of connected 3D points as illustrated in Fig.(4.5). The same figure displays an example of skeleton segments, which are parts of the skeleton graph that form simple paths.

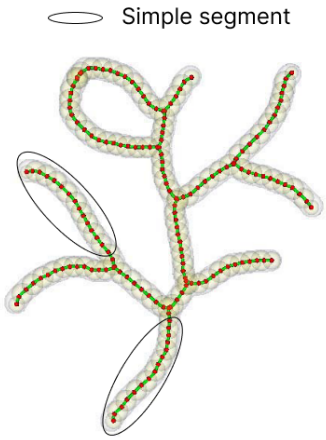

Figure 4.5: *Illustration of the skeleton of a mesh (image adapted from [Hassouna & Farag, 2005]).*

The Skeleton-Guided Ellipsoid-Atom Decomposition. The SGEAD method differs from the previously mentioned methods as it relies on mesh skeletons for the decomposition of 3D meshes.

The use of ellipsoids as components is motivated by their anisotropy and elliptical cross-sections. As ellipsoids have different radii, two of them can be used to model 2D sections of the axon, while the last one can be set according to the local properties of the fiber. The SGEAD method is tailored for neurons, axons, and glial cells and can also be applied to blood vessels. The second motivation is the gain in memory efficiency obtained from using ellipsoids as elements of the atom-based geometric representation [Bischoff & Kobbelt, 2002].
The SGEAD method works in five steps: cell skeleton segmentation, mesh-slicing, noise-filtering, ellipse fitting on the cross-section of interest, and finally, 3D reconstruction of the ellipsoid.

1. Skeleton segmentation. The cell skeleton complex graph is decomposed into simple paths in this step. A simple path is basically a connected sub-graph such that no node has more than two connections. The set of all simple paths forms a disjoint partition of the skeleton edges.

2. Mesh-slicing. In this step, the mesh is sliced along the segmented simple paths. For each simple path, one slice is performed at the center of each edge in the simple path. The slicing plane's origin is located at the center of the edge, and the plane orientation is the same as the edge's orientation in the case of a line-segment-shaped edge; otherwise, the orientation is taken as the tangential vector at the edge's center.

3. Noise-filtering. In many cases, not only one cross-section is obtained after performing the mesh-slicing step. Instead, multiple cross-sections result from the plane's intersection with other parts of the cell. This step consists of filtering the noisy cross-sections and only keeping the relevant cross-section stemming from the used edge in the mesh-slicing step. The filtering relies on calculating bounding boxes over the cross-sections and keeping the bounding box containing the cutting plane origin, i.e., the center of the edge used in the mesh-slicing step.

4. Ellipse-fitting. After filtering the noise, the cross-section of interest is used to fit an ellipse. Note that the cross-section points are co-planar since they result from the intersection of the 3D mesh with a cutting plane.

5. Ellipsoid reconstruction. The fitted 2D ellipses are transformed back to the 3D space using the cell skeleton. The ellipsoid's two main axes are fixed from the fitted 2D ellipse. As for the last ellipsoid's axis -oriented towards the fiber's direction-, its length is a free parameter and can be tuned according to the cell or skeleton parameters.

The first four steps of the SGEAD method are displayed in Fig.(4.6). The first illustration in the top left, i.e., panel *a*) displays a simplified sketch of a cell skeleton with the cell body. The next illustration in panel *b*) displays the same cell after the skeleton segmentation step. Each simple path in the skeleton has a different color, totaling 11 segments. The panel *c*) displays the cell's skeleton, the cell membrane, the cutting plane represented as a red-dashed line, and the cutting plane origin illustrated with a light green box. The panel *d*) displays multiple cross-sections obtained from the mesh-slicing step. Only the cross-section located in the center is the main cross-section of interest. The noisy cross-sections are filtered if their corresponding bounding

box does not contain the cutting plane origin. The central cross-section is used to fit an ellipse represented in red in the panel *d*), thus illustrating both the third and fourth steps.

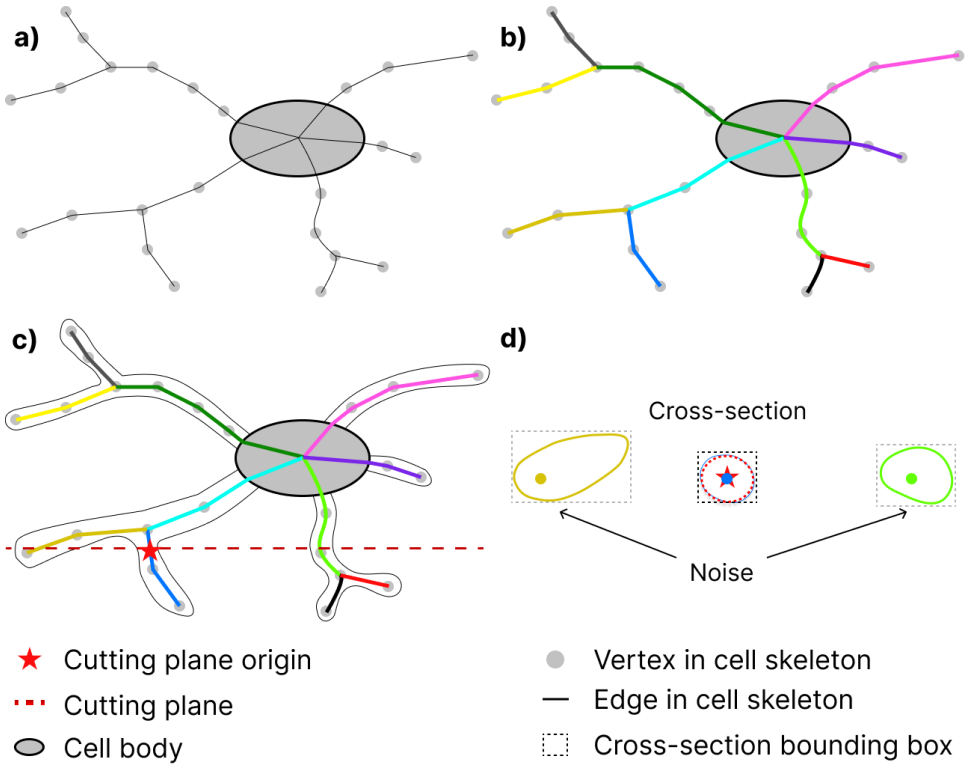

Figure 4.6: *Illustration of the first four steps of the SGEAD method. a) The cell skeleton with the cell body. b) Skeleton segmentation, where each simple skeleton segment is represented by a specific color. c) Illustration of the mesh-slicing along the segment in dark blue. d) Illustration of 2D cross-sections resulting from mesh-slicing with a fitted red ellipse on the cross-section of interest and the noise-filtering of cross-sections in light green and dark yellow.*

4.3.2 Dataset

The dataset used in this first study is the H01 dataset presented in the previous chapter. It contains over 49000 digital brain cells reconstructed from electron microscopy volumes. The reconstruction process relies on both manual and automatic segmentations of the 3D volumes. There are different segmentations of the 3D volumes, each adopting bias toward fewer or more false mergers. In the course of this thesis, the C3 segmentation layers were used. The dataset is open and available through the Google Cloud Platform (GCP). The cells can be accessed using the *gsutil* tool, which enables to access buckets in the Google Cloud Storage. More information can be found in the following documentation link: `https://cloud.google.com/storage/docs/gsutil`. The C3 segmentation layers contain both mesh-based cells, and their corresponding skeletons can be found using their bucket links:

- **Meshes.** `gs://h01-release/data/20210601/c3/mesh`
- **Skeletons.** `gs://h01-release/data/20210601/c3/mesh`

The total volume of the meshes is around 6.5 Tb, while the total volume of the skeletons is around 241 Gb.

The H01 dataset contains the following cell types:

- Oligodendrocytes
- Pyramidal neurons
- Microglia Oligodendrocyte precursor cells
- Astrocytes
- Interneurons
- Spiny stellate
- Blood vessel cells
- Spiny atypical cells
- C-shaped cells
- Unclassified neurons
- Unknown cells

These cell types are present in different proportions, with oligodendrocytes being the most frequent and pyramidal cells being the second most frequent. More insights about the cells from C3 segmentation layers are presented in Fig.(4.7).

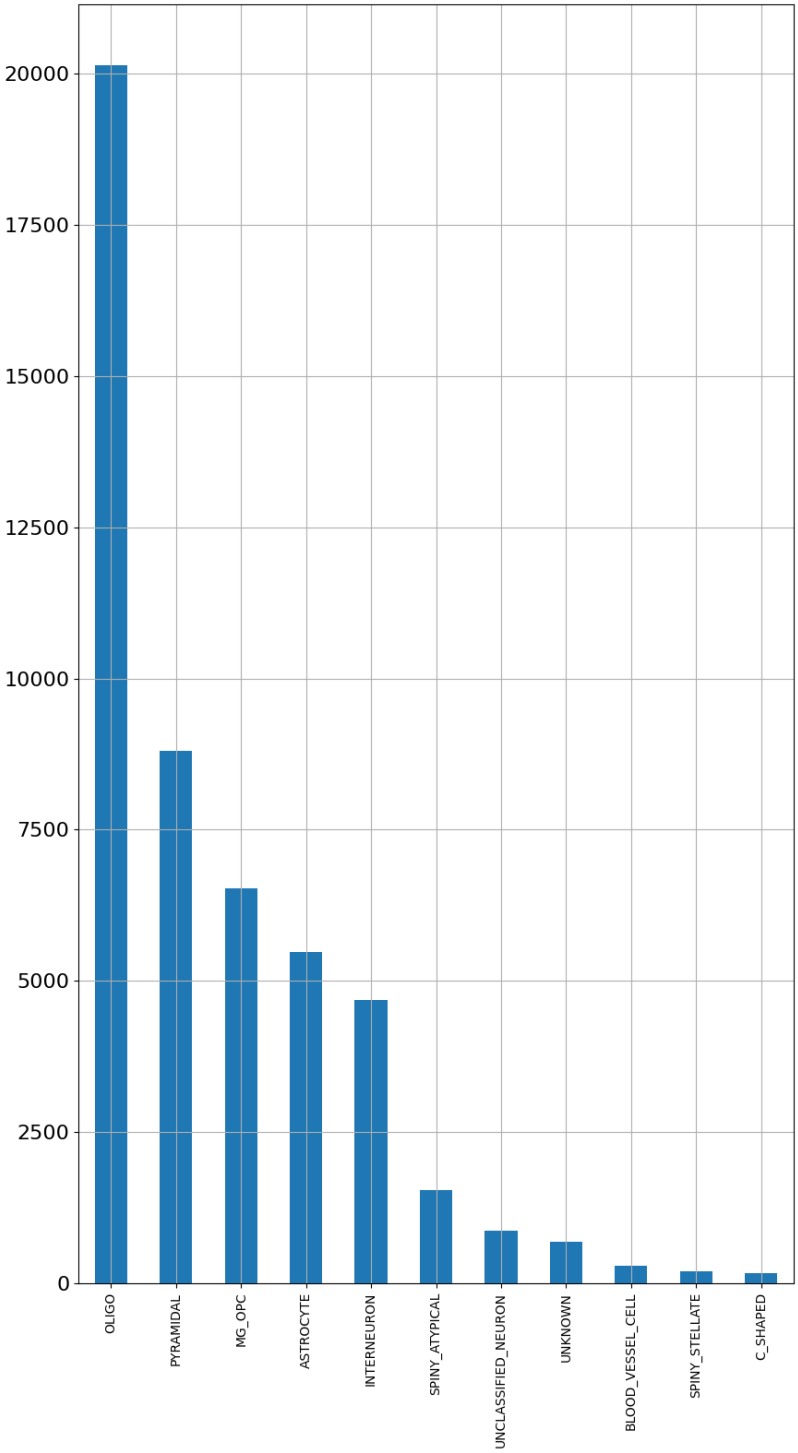

Figure 4.7: *The cell types distribution of the C3 segmentation layers of the H01 dataset.*

In the course of this thesis, only a fraction of this dataset was used to validate the SGEAD method.

The used fraction of the H01 dataset consists in a set of 842 cell meshes with a total memory size of 186 GB. The cells are downloaded one by one to a local disk space using their unique identifiers (ID). For this, the *cloudvolume* Python library was installed, and the following Python script was used:

```
from cloudvolume import CloudVolume

H01_DOWNLAOD_LINK = 'gs://h01-release/data/20210601/c3/'

def download_cell(link,cell_ID):

    cv = CloudVolume(
    link,
    progress=True, # shows progress bar
    cache=False, # cache to disk to avoid repeated download
    # parallel=True, # uncomment to try parallel download!
    )
    mesh = cv.mesh.get(cell_ID)
    skeleton = cv.skeleton.get(cell_ID)

    return mesh, skeleton
```

Listing 4.1: A sample Python code to download the cell mesh and skeleton from H01 dataset given its unique ID.

The script above allows the mesh and skeleton to be stored as variables but not saved to disk. The mesh is written to disk using the Python library *meshio* [Schlömer, n.d.]. Among the available mesh file formats listed in Fig.(4.1), we opted for the Wavefront ".obj" file format for compatibility with internal visualization tools. Regarding the cell skeleton, the cell segmentation processing in step 1 of the SGEAD method was applied before saving the skeleton to disk. We found this approach to be convenient as the first step was not computationally expensive and was in the worst cases comparable to the download time. This allows the saved meshes and skeletons to be ready for the rest of the SGEAD method steps. The skeletons were converted to VTK PolyData data structure and saved as ".vtk" files. The PolyData data structure provides a convenient way to model skeletons, which are represented as a set of vertices and lines. The lines in the PolyData contain the simple paths of the cell skeleton. A detailed distribution of the cell memory sizes is provided in Fig.(4.8) below:

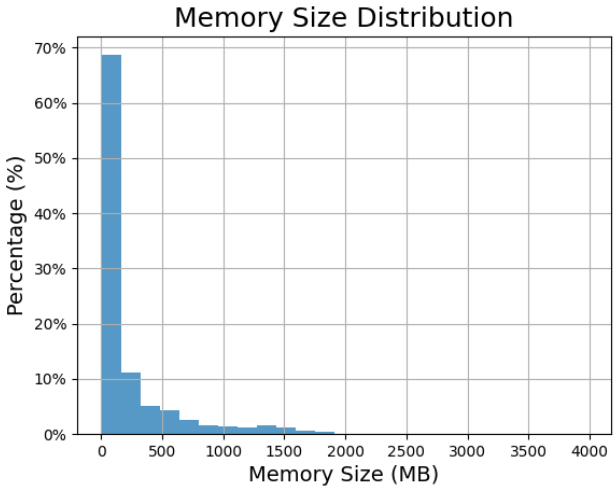

Figure 4.8: *The memory size distribution of the 842 cell meshes from the C3 layers segmentation of the H01 dataset.*

The decomposed cells are of different cell types, and their distribution is illustrated below:

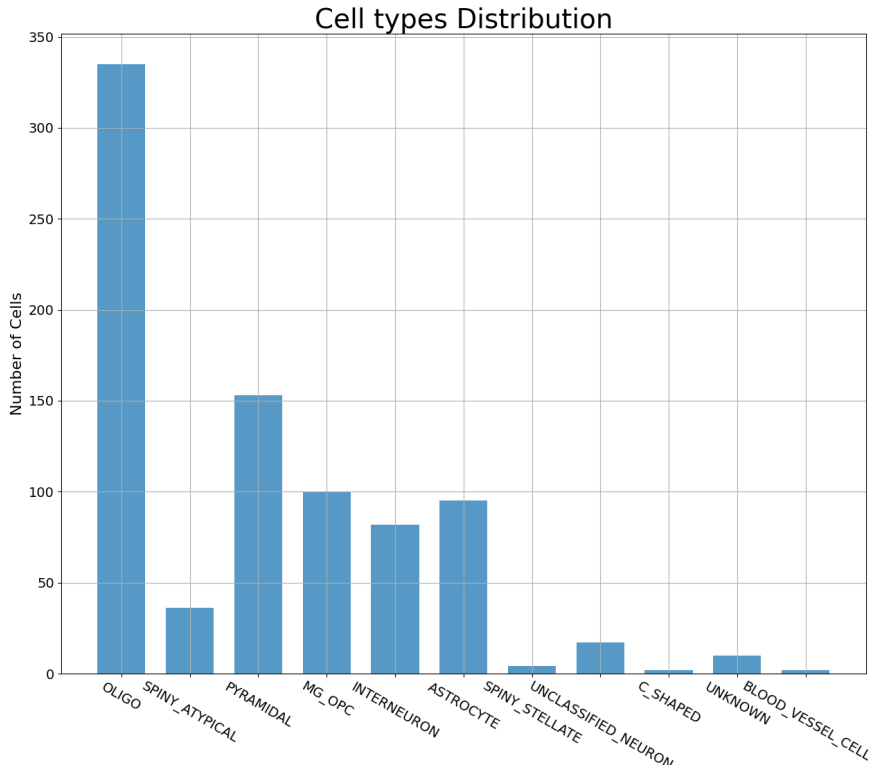

Figure 4.9: *The cell types histogram of the 842 cell meshes from the C3 layers segmentation of the H01 dataset.*

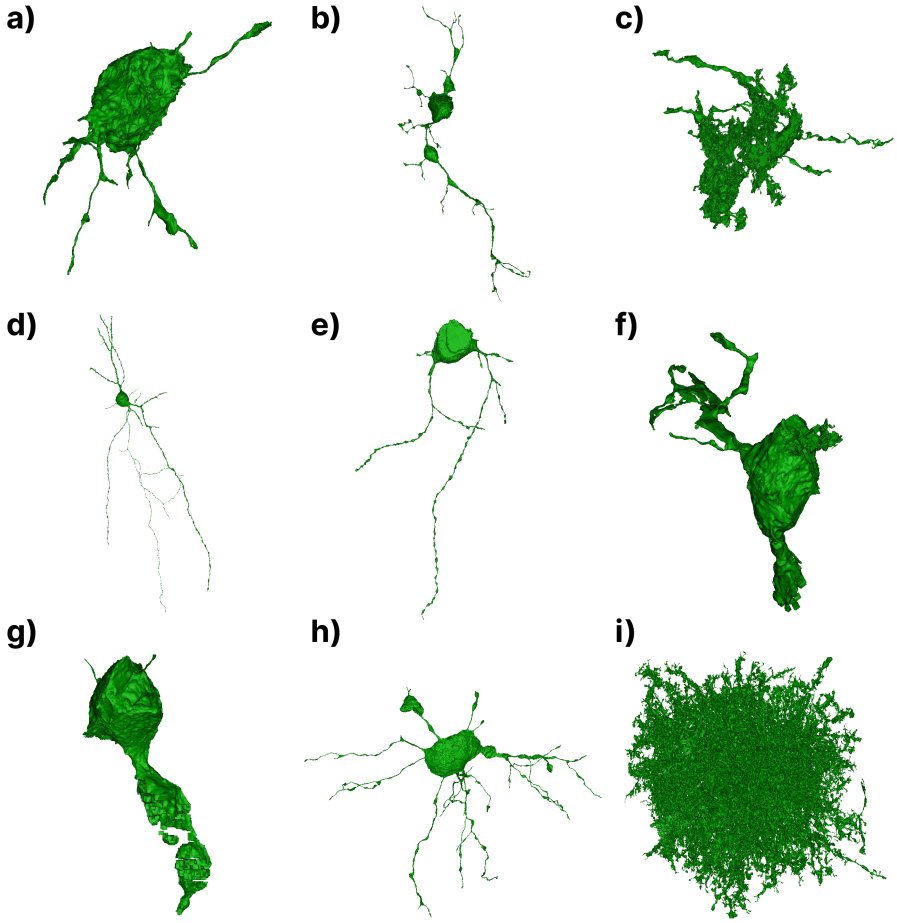

Figure 4.10: *An overview of some of the cell meshes provided in the C3 layers segmentation of the H01 dataset. The cell types and identifiers (IDs) for each panel are the following: a) Oligodendrocyte (*41563544088*) b) Oligodendrocyte (*42412474991*) c) Oligodendrocyte (*43512104318*) d) Interneuron (*45761229225*) e) Interneuron (*51454684170*) f) Microglia Oligodendrocyte precursor cell (*63304914763*) g) Oligodendrocyte (*3908308877*) h) Oligodendrocyte (*4012822944*) i) Astrocyte (*40647769629*).*

Finally, a brief overview of some of the downloaded cells from the H01 dataset is provided in Fig.(4.10). The cells display different forms, some of which are very complex, as shown in panels i) and c). Others display a cell body, as seen in panels a), b), d), e), f), g), and h).

4.3.3 Results

Before using the H01 dataset, the SGEAD method was first tested on an axon fragment obtained from electron microscopy of a mouse brain sample. The axon fragment was obtained from `http://brainiac2.mit.edu/SNEMI3D/home` in November 2021. Unfortunately, this website is no longer available; however, an archive version can be accessed from the following link: `https://web.archive.org/web/20210301013203/http://brainiac2.mit.edu/SNEMI3D/home`. This axon fragment was used to initially test and implement the SGEAD method, and its skeleton was obtained using a centerline module from the 3D-Slicer medical imaging software[Fedorov *et al.*, 2012].

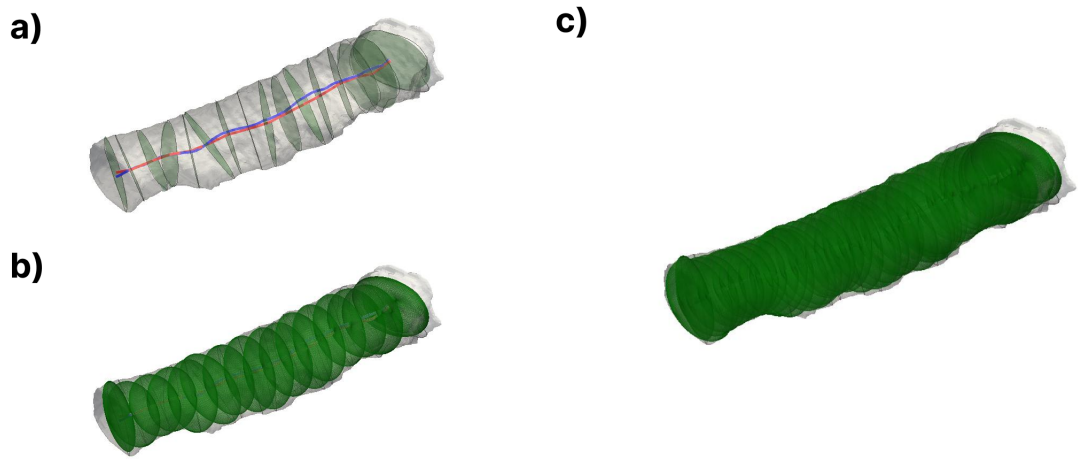

Figure 4.11: *The first proof of concept of the SGEAD method applied on an axonal segment of a mouse brain. Panel a) displays the fragment of an axon with 16 elliptical cross-sections in green, with the blue line being the axonal skeleton and the line in red being the line passing through the elliptical cross-section centers. Panel b) shows the 3D reconstructed ellipsoids from the 16 elliptical cross-sections. Finally, panel c) displays the SGEAD method using a denser representation with 40 cross-sections.*

The first result of the SGEAD method is displayed in Fig.(4.11). In this case, the third step of the SGEAD method of noise-filtering is not required. This is due to the absence of any ancillary intersections of the cutting plane with the axonal segment. Only the main axonal cross-section is obtained. The blue axonal skeleton in Fig.(4.11) is subdivided into equidistant smaller segments that define the centers of the cutting plane for the axonal cross-sections. The fitted 2D ellipses on the cross-sections are displayed in green color in panel a). Finally, these elliptical-cross sections are transformed into ellipsoids, as seen in panel b). With 16 ellipsoids, the first decomposition offers a sparse representation. Another decomposition was performed on the same sample with 40 ellipsoids; this denser representation is displayed in panel c).
Given the satisfactory results using the axonal segment, we moved on to using the cell meshes stemming from the C3 layers segmentations of the H01 dataset. The first result in Fig.(4.12) shows the original mesh geometry (ID: 5043316825) in the left panel and its ellipsoid-represented version obtained from the SGEAD method in the right panel. Indeed, the ellipsoid representation provides a lightweight representation compared to the mesh-based representation, as mentioned previously in the following study [Bischoff & Kobbelt, 2002]. Although a memory compression factor of 300 relative to the mesh-based representation was achieved, this reduction factor is not the same for all cells. It depends on different elements such as the spatial span of the cell, its skeleton topology, i.e., the number of edges and nodes in the skeleton graph, and the complexity of the cellular surface. The same result in Fig.(4.12) shows the same zoomed part of the interneuron in both representations. This allows to show the ability of the ellipsoid to accurately represent cellular mesh-based geometries.
Another result of the SGEAD decomposition is shown in Fig.(4.13), which displays ellipsoid-based representations of pyramidal cells.

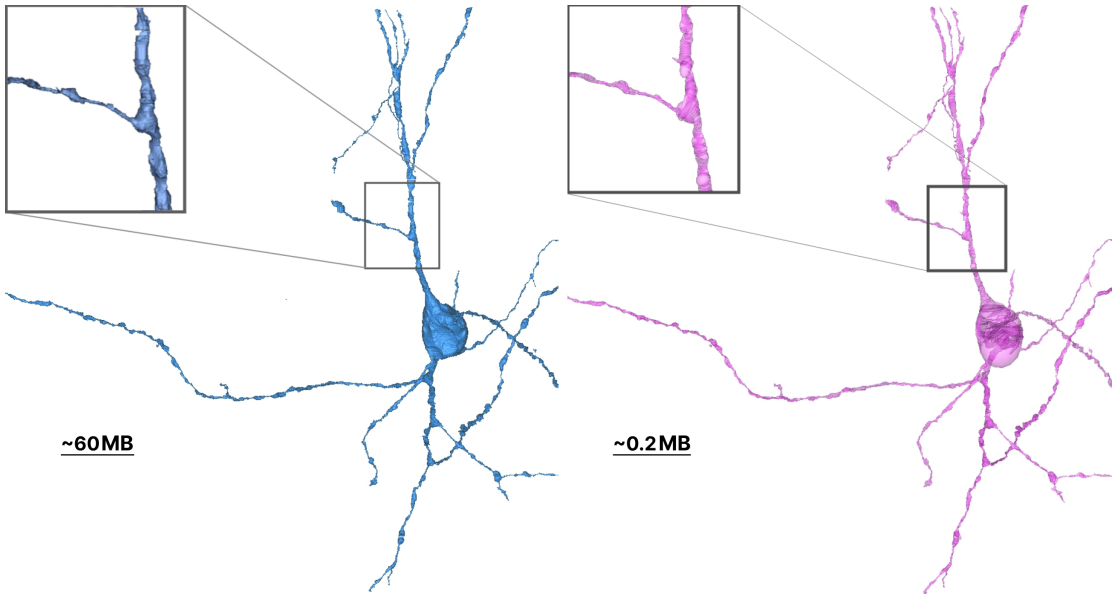

Figure 4.12: *Illustration of a mesh-based and ellipsoid-based representation of an interneuron from the H01 dataset. The left panel displays the original mesh of ID 5043316825, weighing around 60 Mb. The right panel shows a lightweight ellipsoid-based version. A zoomed part of the cell is displayed in both representations to enable the comparison of fine-grained details of both representations.*

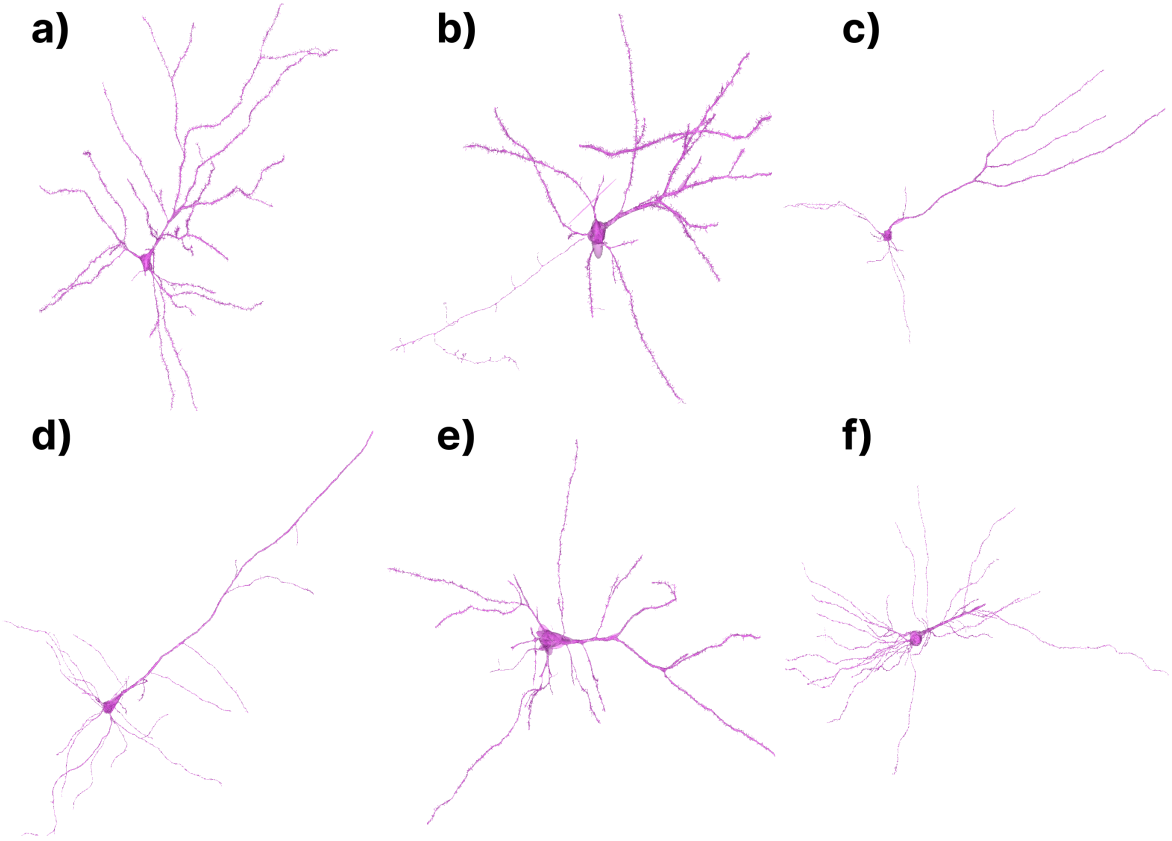

Figure 4.13: *Illustration of some ellipsoid-represented pyramidal cells from the H01 dataset decomposed with the SGEAD method. Their corresponding cell IDs are the following: a) 1581843873 b) 4189772777 c) 4726753206 d) 5057641324 e) 5791413754 f) 6322366087.*

The total memory size of the mesh-based representation of the pyramidal cells in Fig.(4.13) is 1872.5 Mb. In contrast, the total memory in the case of ellipsoid-based representations is only around 6.8 Mb, which corresponds to a 99.64% memory size reduction and showcases the advantage of using the ellipsoid-based representation.
During the first batch of cell processing using the SGEAD method, the runtime performance was evaluated on a workstation with the following specs: 128Gb RAM and an Intel Core i9-10885H CPU processor @ 2.40GHz × 16. The evaluation was done using a set of 89 cells with various memory sizes to generate the runtime performance results below:

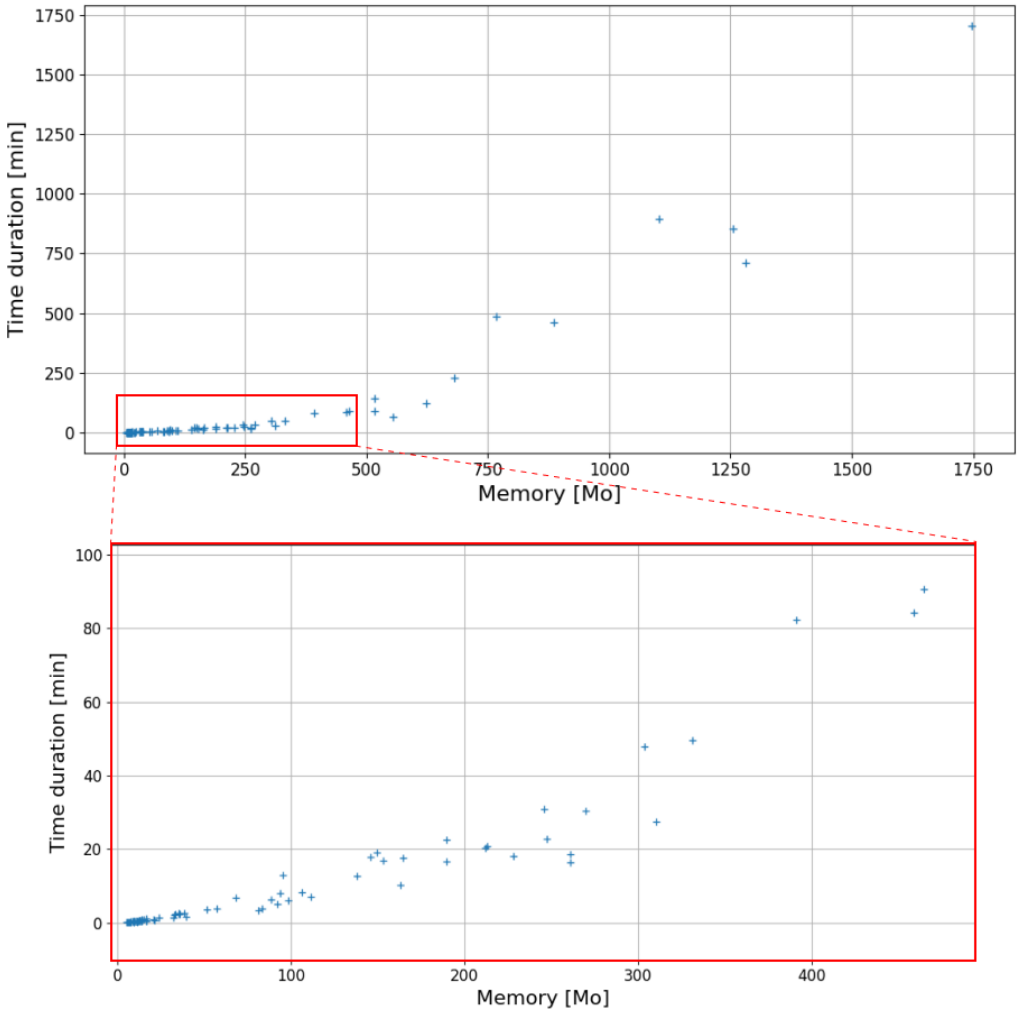

Figure 4.14: *Illustration of the runtime performance for a set of 89 cells from the H01 dataset with different memory sizes. The first plot shows the runtime for steps 2 to 4 of the SGEAD method with a zoomed box in red for lower memory size meshes displayed in the bottom part.*

The runtime results in Fig.(4.14) are only for steps two, three, and four of the SGEAD method since the first step is done before saving the mesh files to disk and the last step, i.e., the fifth, is fast and its runtime is negligible compared to the others. Also, from an implementation point of view, the fifth step was performed separately from the others. This allowed us to run a campaign of extracting cross-sections from various cells, which were used to perform 3D reconstructions in a later step. This was convenient for debugging the SGEAD method and gathering potential

problems from different cells to iteratively improve the decomposition pipeline. The total memory size of the 89 cells used for the runtime benchmark is around 43 Gb for the mesh-based version, and in the case of the ellipsoid-based representation, these cells occupy only around 64 Mb of memory, which provides a large compression ratio.

4.3.4 Discussion

In this section, the results of the SGEAD method were presented for both a fragment of axon in Fig.(4.11) as well as for gray matter cells from the H01 dataset in Fig.(4.12). The SGEAD implementation has known different modifications in the course of this thesis. Although the initial version worked well for the axonal segment, it was not ready to be used for the H01 dataset for many reasons, such as the data formats and, most importantly, the noise-filtering step, which was not implemented in the initial SGEAD version. As the SGEAD method processes 3D geometries, an important part of the development process was dedicated to finding convenient visualization tools. This was a crucial step in the development of the SGEAD method as it allows debugging and visually inspecting the obtained results. For this, we relied on two visualization tools: the first one from the PyVista Python library, and the second is the Anatomist with a custom internal module for visualization. The first tool provided a simple, easy-to-use interface; however, the second tool enabled a faster visualization of larger geometries, i.e., geometries with a high number of ellipsoids (over 6k).
Upon different iterations, the SGEAD's implementation massively relies on exception handling. In fact, throughout the development, many edge cases have been found, such that one of the SGEAD steps can return an error. A brief list of such edge cases is presented below:

1. A small missing portion of the cell mesh
2. A part of the skeleton located outside the mesh
3. Cases where the cross-sections have less than five points

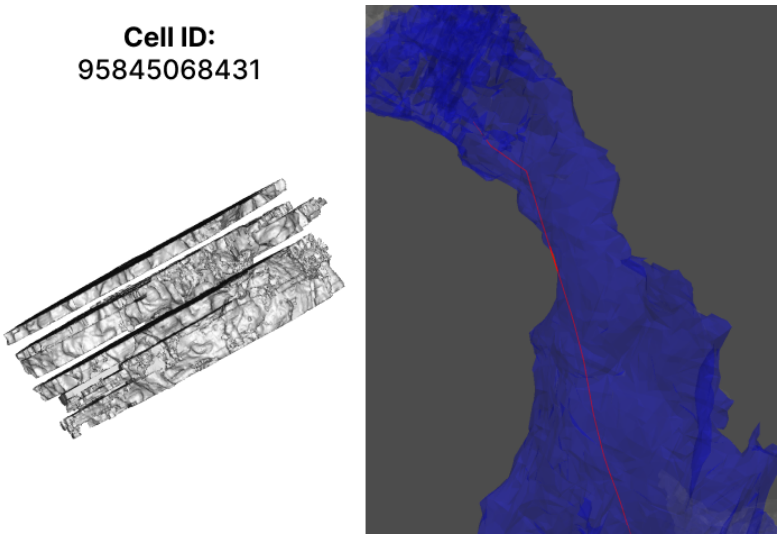

Figure 4.15: *Illustration of some of the edge cases leading to potential raised errors, which justify an increased usage of exception handling in the implementation of the SGEAD method.*

Two of these edge cases are illustrated in Fig.(4.15). The cell in the left panel displays an example of an incomplete cell with missing slices, and the fragment of the cell in the right panel showcases an example of a skeleton, displayed in red, that is not completely located inside the cell mesh. The first case of a missing cell fragment can potentially return an error at the mesh-slicing or noise-filtering steps. Also, when a skeleton part is located outside the cell mesh, provided that the cutting plane's origin is located outside the mesh, the noise-filtering step does not return the main cross-section. Finally, for the last edge case, when the number of points in the main cross-sections is less than five, the problem of fitting an ellipse is ill-posed. Such cases can be faced in small cell parts such as dendrites terminals. The first two points in the edge cases list can be grouped into the general term "data quality", which determines the quality of the ellipsoid-represented decompositions.

The memory size aspects of the ellipsoid-based representation of the H01 cells demonstrate the convenience of using ellipsoids as a unit of representation. The shown results confirm prior studies for the ability of ellipsoids to compress 3D geometries [Bischoff & Kobbelt, 2002]. This compression ability is particularly suited for tubular geometries such as axons or neurons. However, the mesh representations can be more memory efficient for other cases, i.e., trivial geometries like boxes. Therefore, in the case of microstructure substrate modeling, the ellipsoid-based representations obtained using the SGEAD method allow to reduce the memory burden associated with mesh-based 3D geometries. Thus helping to reduce the memory bottleneck in dMRI simulations as mentioned in Dang Van Nguyen's thesis [Nguyen, 2014].

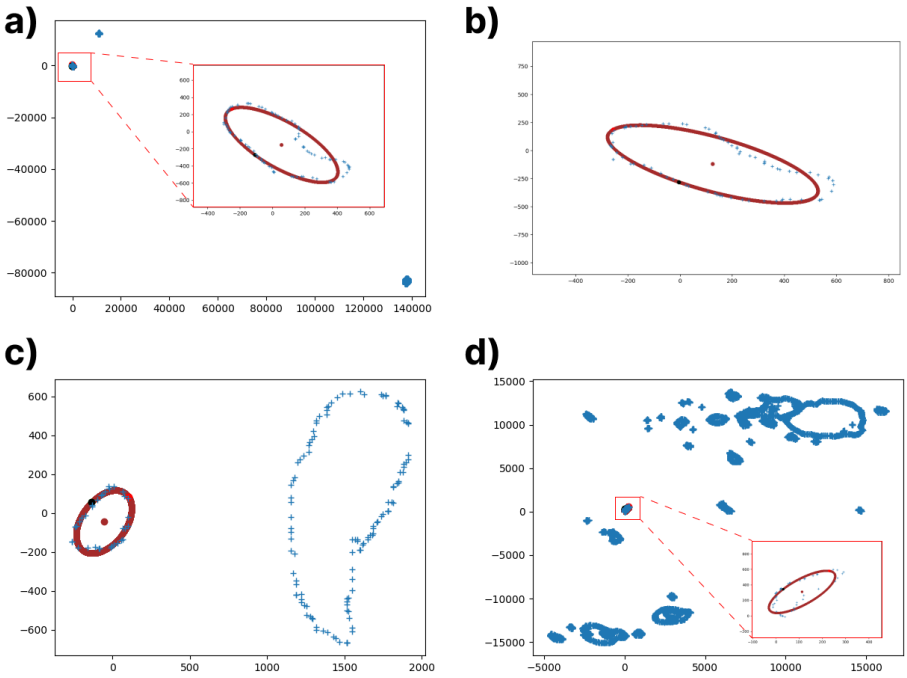

Figure 4.16: *Illustration of some good cases of ellipse fit and noise-filtering. a-b) Two 2D cross-sections from the cell IDs 5043316825 and 5043316825 with their corresponding fitted 2D ellipses. c) Another example from the same cell with only one noisy cross-section. d) A very noisy 2D cross-section using the cell ID 6917983029 with the detected cross-section and the fitted 2D ellipse.*

The non-circularity of 2D cross-sections was also observed when processing the H01 dataset, as illustrated in Fig.(4.16). As long as the skeleton is inside the mesh, the cross-section contains a relatively large number of points and has a fairly simple shape; good ellipse fits are obtained. However, there are also edge cases where the approximation of cross-sections with ellipses is not the best, and this will be discussed in more detail in the limitations of the SGEAD method.
The SGEAD runtime performance results in Fig.(4.14) show moderate decomposition times for the H01 cells. The decomposition time does not exceed 40 minutes for cells weighing less than 300 Mb. This is particularly convenient given the memory size distribution in Fig.(4.8). In fact, since most of the cells weigh less than 300 Mb, only large cells take significant amounts of time. The performance study is by no means a thorough performance analysis as it considers only the memory size and abstracts other aspects of the cell, such as the skeleton's topology. As described, the SGEAD method inputs the cell mesh and its corresponding skeleton. Therefore, the mesh-slicing is only as fine-grained as the skeleton edges are: the smaller the skeleton segments, the finer the reconstruction. There is also a trade-off between the amount of detail required and the computational cost of the method. Performing a detailed 3D reconstruction of the cell mesh requires a detailed cell skeleton, leading to more mesh-slicing and ellipse-fitting steps and, thus, higher computational costs. Also, the most time-consuming step is mesh-slicing. The ellipse-fitting and noise-filtering runtime is negligible compared to the mesh-slicing step.
Although the benefits of the SGEAD method are numerous, there are also different limitations:
Sub-optimal memory compression. The memory efficiency of the geometric representation using ellipsoids is sub-optimal and can be further optimized. This is for various reasons. First, the skeleton used for the SGEAD was directly taken from the H01 dataset with no further processing. This can be limiting as, in some cases where the skeleton is straight, and the cross-section does not significantly change, only few mesh-slicing steps are required. If that is not the case, i.e. the skeleton contains different sub-segments, this not only leads to increased runtime for unnecessary mesh-slicing steps but also redundant ellipsoids. This is also true for the soma. Different cell soma could have been efficiently approximated only by a small set of ellipsoids. Instead, given the actual skeleton, the number of ellipsoids representing the soma is significantly larger.

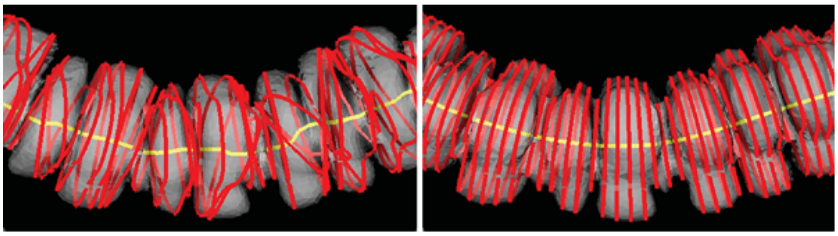

Figure 4.17: *Cross-sections in red stemming from two different skeletons in yellow (Image taken from[Ma et al., 2018]).*

A previous study in the computational geometry literature [Ma *et al.*, 2018] provides a method to deform a skeleton to obtain better cross-sections as illustrated in the right panel of Fig.(4.17). Thus, in order to overcome the sub-optimal memory compression limitation, skeleton processing methods should be tested and applied before using the SGEAD method. Also, from a runtime perspective, it is important for the skeleton processing step to be efficient with small runtimes relative to the SGEAD method. In addition, the soma's decomposition can be improved by

modifying the skeleton or adopting a decomposition scheme different from the SGEAD method. Methods from the following study [Zhou *et al.*, 2015] can also help improve the decomposition method.

Runtime efficiency. Although a large number of cells took less than an hour to be individually processed, the runtime performance of the SGEAD method is still limited and can be improved. First, the implementation of the SGEAD method in Python prevented the achievement of the best runtime efficiency. Second, through additional testing of the SGEAD method, we found that the mesh-slicing step was extremely inefficient and could be drastically improved as follows: instead of slicing the whole mesh, selecting only a small subset of the mesh can substantially reduce the runtime of the mesh-slicing step. For this, we suggest a simple heuristic for sub-mesh selection. It consists in using the skeleton segment used for the mesh-slicing step to extract two planes. The planes are defined using the endpoints of the skeleton segments and their normals. The proposed simple heuristic selects the sub-mesh between the two planes.

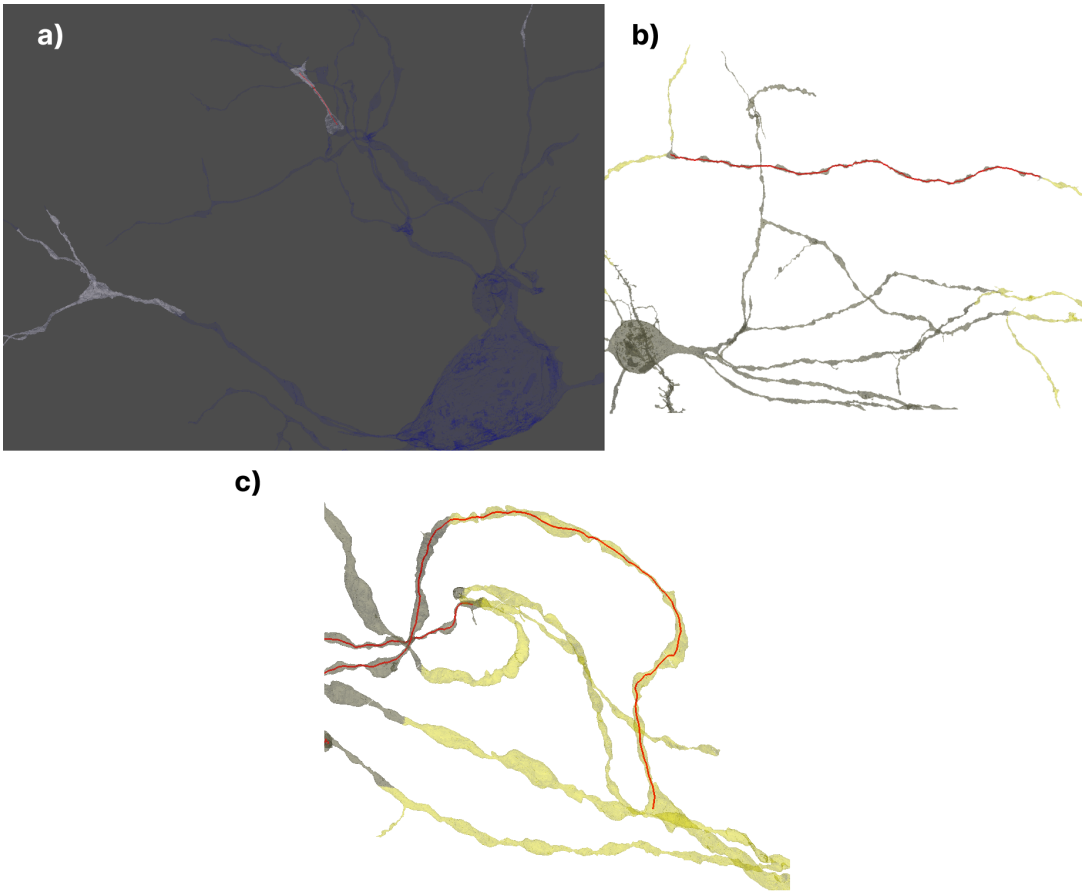

Figure 4.18: *Illustration of the used sub-mesh selection heuristic. a) The skeleton segment is in red, and the selected sub-mesh is in gray while the discarded part of the cell is in blue. b) The selected sub-mesh is displayed in blue and the discarded part of the cell is in yellow. c) An edge case when the simple selection heuristic does not work.*

We found that in most cases, this heuristic provided satisfactory results with important reductions in the runtime of the mesh-slicing step. However, there are also a few edge cases, as illustrated in panel c) of Fig.(4.18) such that the selected sub-mesh does not cover the whole skeleton segment. Note that the selected sub-mesh using this heuristic selects other parts of

the cell that are irrelevant for the cross-section of interest, leading to noisy cross-sections as in Fig.(4.16). The ideal sub-mesh extraction should only select the parts of the mesh that directly cover the skeleton segment along which the mesh-slicing step is performed. Finally, it is essential that the sub-mesh selection process is runtime-efficient. The reduced mesh-slicing time obtained from using a sub-mesh should be larger than the required time for sub-mesh extraction.

Memory consumption. We found that the SGEAD Python implementation required a large memory usage, especially for large cells. A C++ implementation of the SGEAD method offers better runtimes as well as a fine-grained control of memory, potentially reducing memory consumption.

Limitations of elliptical cross-sections. While ellipses offer a good approximation of axonal cross-sections. The fitting quality can be reduced for the soma. This limitation goes in hand with the sub-optimal memory compression mentioned previously and the skeleton's topology. The H01 cell processing campaign revealed cases where ellipses did not approximate some cross-sections accurately. These cross-sections were mainly located in or close to cell bodies.

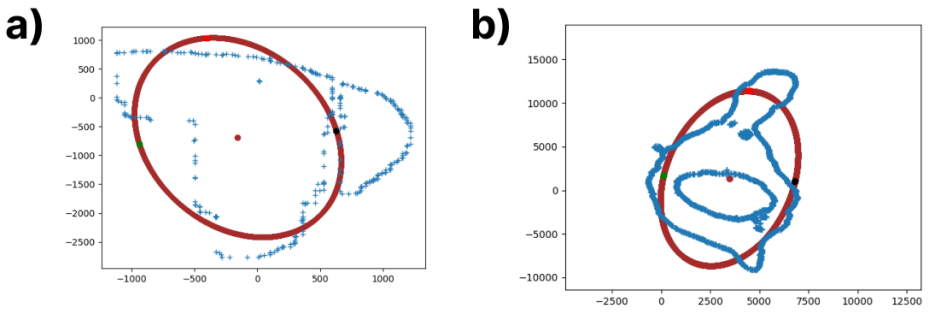

Figure 4.19: *Illustration of some of the problematic cross-sections using the SGEAD method. a) A cross-section sample with a fitted ellipse from cell ID 2931136508. b) A cross-section with a fitted ellipse from cell ID 3358332472.*

The illustration in Fig.(4.19) reveals edge cases when the elliptical geometry is not the best approximation for the cellular cross-section. A potential improvement could be to use different strategies such as using multiple ellipsoids to approximate the cross-sections or using ellipsoid growth-based approaches as presented in the previous study [Bischoff & Kobbelt, 2002].

4.3.5 Conclusion

This section presented the first contribution relative to microstructure geometry modeling. The proposed ellipsoid-based representation obtained from the SGEAD method allows a lightweight representation of the cells and approximates the original mesh-based version. This new representation based on ellipsoids allows to reduce the memory-related burdens previously mentioned in the dMRI simulation literature. Future work in this area includes the implementation of the suggestions presented previously to improve the SGEAD method. Another interesting aspect that was not explored is the SGEAD decomposition of purely synthetic microstructure substrates.

4.4 The Neuromorpho sphere-based reconstruction

In the same line of research of integrating large datasets for gray matter microstructure modeling. This section presents a second contribution, which consists of a reconstruction method of the cells stemming from the NEUROMORPHO dataset. The development of this method is motivated by different points. First, the computational advantages of using a sphere-based representation in the context of dMRI simulations as adopted in the MEDUSA framework [Ginsburger *et al.*, 2019]. Second, in order to build a robust microstructure parameters inference model, it is important for the model to be independent of the method used to generate the substrates. Such robustness can be achieved by diversifying the microstructure substrates. This can be done using different strategies, such as using different synthetic microstructure generation methods or using different datasets to build virtual microstructure substrates. By diversifying the sources of cells, potential model inference limitations such as overfitting can be mitigated. In the course of this thesis, we opted for the second approach for diversifying sources of cells by using different datasets: the H01 dataset presented in the previous section and the NEUROMORPHO dataset presented in the current one. As the reconstruction method is tightly linked to the dataset's format, a detailed description of the NEUROMORPHO dataset is presented first, followed by the reconstruction method and results.

4.4.1 Dataset

NEUROMORPHO is a large repository of virtual brain cells reconstructed from different species. The distribution of cells according to the species is provided below:

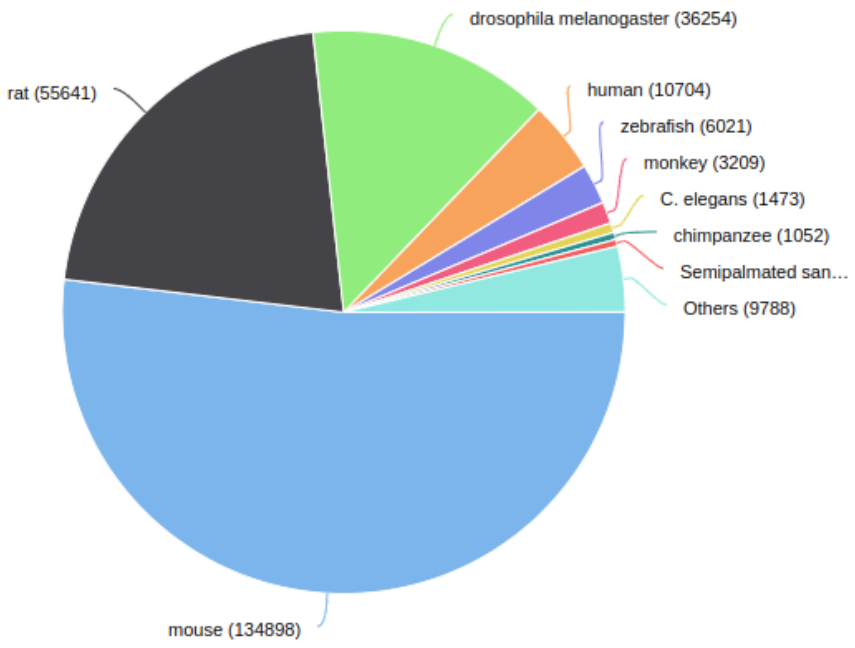

Figure 4.20: *The cell species distribution of the NEUROMORPHO dataset (image taken from https://neuromorpho.org/byspecies.jsp).*

The cell species distribution illustrated in Fig.(4.20) shows a large proportion of brain cells stemming from mice and rats.
The cells are stored in the .SWC format. It is a widely adopted format within the community, and it is used to describe complex cells such as neurons or glial cells.

Sample number	Sample identifier	x y z	radius	Parent identifier
An incremental integer value starting from 1	0 - Undefined 1 - Soma 2 - Axon 3 - (basal) dendrite 4 - Apical dendrite 5 - Custom	Spatial coordinates of the node, represented as a sphere	The radius of the sphere	An integer representing the identifier of the parent node. -1 for the first node of a soma

Figure 4.21: *A summary of the ".SWC" file format.*

The .SWC format describes brain cells using the cell's skeleton information. The cell is considered a tree, i.e. a graph with vertices connected by edges. The nodes within the tree represent regions of the cell, and they contain the following information:

- **The node identifier.** Also called the sample number. It basically provides a count for the number of nodes in the cell graph and the count starts from one.

- **The sample identifier.** It is represented as an unsigned integer and usually takes values between zero and four included. This defines the part of the cell that is described by the node. The values 0, 1, 2, 3, and 4 code respectively for an undefined part of the cell, the soma, the axon, a basal dendrite, and an apical dendrite. Finally, values over five can code for custom parts of the cell.

- **The node position.** This is represented by a set of three floating numbers representing the 3D coordinates of the node.

- **The node's radius.** It represents the radius of the cellular cross-section at the node's position.

- **The parent identifier.** Since each node in the tree graph has a parent, this field contains the parent node identifier. For the first node in the tree with no parent node, its identifier is -1.

A summary of the ".SWC" file format is illustrated in Fig.(4.21). This format provides a highly sparse representation of the cell as it only considers the cell's skeleton and additional information such as the radii. The ".swc" file format uses an ASCII encoding, i.e., a regular text file to store the cellular information. A sample of a ".SWC" file is provided below:

```
# Sample_num, sample_ID, x, y, z, r, parent_ID
1  1  2.14 1.73 -0.15      12.61  -1
2  1  2.14 14.34 -0.15     12.61   1
3  1  2.14 -10.87 -0.15    12.61   1
4  4  -13.83 -11.57 18.79  2.94    1
5  4  -14.67 -12.08 18.79  2.94    4
6  4  -17.63 -9.48 19.47   0.59    5
7  4  -19.82 -8.47 19.47   0.59    6
```

Figure 4.22: *A small sample of a ".SWC" file.*

4.4.2 Methods

This section presents the reconstruction method for the neural virtual cells stemming from the NEUROMORPHO dataset. First, few geometric notions are revisited then the reconstruction method is presented.

The spherical geometry. The sphere can be viewed as a restricted case of an ellipsoid with equal radii, i.e., $r_x = r_y = r_z$. A sphere of radius r in the 3D space is described by the following equation:

$$(x - x_c)^2 + (y - y_c)^2 + (z - z_c)^2 = r^2 \tag{4.4}$$

The terms (x_c, y_c, z_c) represent the coordinates of the sphere's center.

Interpolation between spheres. Given two spheres in the 3D space, the interpolation can be performed in two steps: first, interpolating their centers, and second, interpolating the radii. The sphere center interpolation is provided by the formula below:

$$\mathbf{x}_t = (1 - t)\mathbf{x}_a + t\mathbf{x}_b \tag{4.5}$$

The same formula can be used for the radius:

$$r_t = (1 - t)r_a + tr_b \tag{4.6}$$

Although other formulas have been previously proposed in the literature for the radius interpolation, such as $((1-t)\sqrt{r_a}+t\sqrt{r_b})^2$ in [Takahashi *et al.*, 2007], we opted for the linear interpolation in Eq.(4.6) for its simplicity.

Overlapping-ratio. This measure characterizes the quality as well as the sparsity of the sphere-based geometric representation. For a basic cylindrical geometry, when represented as a set of spheres, different configurations of spheres can be used to represent the same cylinder. The overlapping ratio is defined as:

$$O_r = \sqrt{1 - \left(\frac{d}{2R}\right)^2}\,, \tag{4.7}$$

with d in Eq.(4.7) is the distance between two consecutive spheres representing the cylinder of radius R.

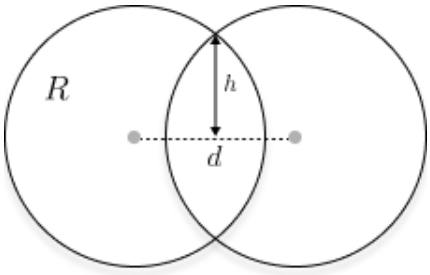

Figure 4.23: *Illustration of the overlapping ratio between two spheres of radius R separated by a distance $d < R$.*

The overlapping ratio is also equal to the ratio $O_r = \frac{h}{R}$ where h is the height at the intersection as illustrated in Fig.(4.23). The overlap ratio O_r has bounded values such that $0 \leq O_r < 1$. The overlapping-ratio value of 1 is never attained since it means that the value of h is exactly one, which is the case of perfectly overlapping spheres.

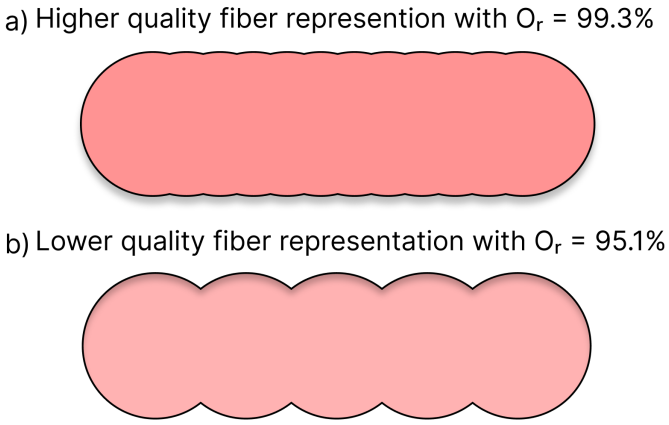

Figure 4.24: *Schematic of a sphere-based representation of a cylinder fragment with two overlapping-ratio values.*

Thus, the overlapping ratio, as illustrated in Fig.(4.24), is a measure of the "bumpiness" of the cylinder due to the use of a sphere-based representation. Higher values of the overlapping ratio O_r mean lower "bumpiness" but also a denser representation of the cylinder containing a large number of spheres. In contrast, smaller overlapping ratio values mean a sparser representation with few spheres and also an increased amount of "bumpiness".
The reconstruction method starts with reading a ".swc" file and constructing a graph from the cell. Once the graph is reconstructed, each edge within the graph is used to create a sphere-based segment of the cell. This segment reconstruction step inputs a parameter for the overlapping ratio to set the reconstruction's quality. Given an overlapping ratio O_r, a number of interpolation steps t_i is calculated. This process is done for all the edges within the cell graph to finally obtain a sphere-based representation for the whole cell.

4.4.3 Results and discussion

The implementation of the reconstruction method is made open source in the following Git repository: https://framagit.org/cpoupon/gkg/-/tree/master/python/simulation.

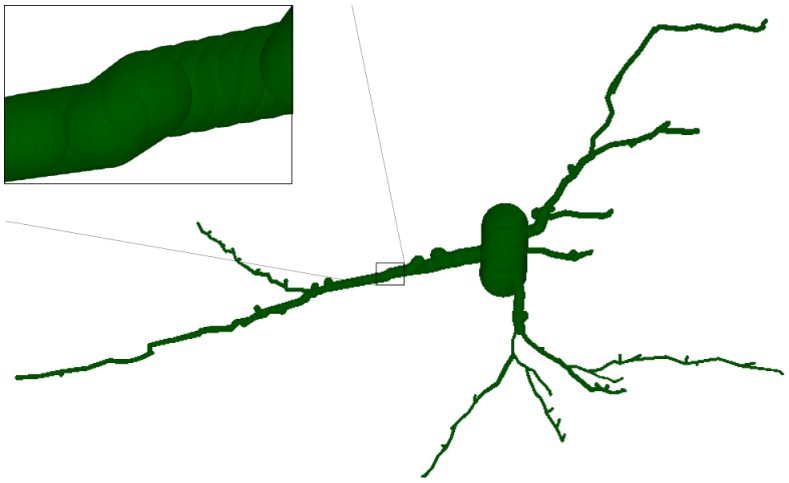

Figure 4.25: *A pyramidal neuron from the NEUROMORPHO dataset with a zoomed fragment of the cell in the top left panel. The reconstruction uses an overlapping ratio of* $O_r = 98\%$

A first example of a reconstructed pyramidal cell from the NEUROMORPHO dataset is illustrated in Fig.(4.25). The detailed zoom in the top left of the figure allows to visualize the sphere-based representation of the pyramidal cell segment. Note that the used overlapping ratio of $O_r = 98\%$ falls between the two cases of higher and lower quality fiber shown in Fig.(4.24). The choice of this value is motivated by having a moderately good fiber quality while at the same time maintaining a reasonable number of spheres representing the cells.
The sphere-based reconstruction method was applied to 264 human brain cells from the NEUROMORPHO dataset. The following cells belonged to the following three different cell types: Pyramidal, Spiny, and Aspiny. The reconstructed cell types distribution is illustrated in Fig.(4.26).

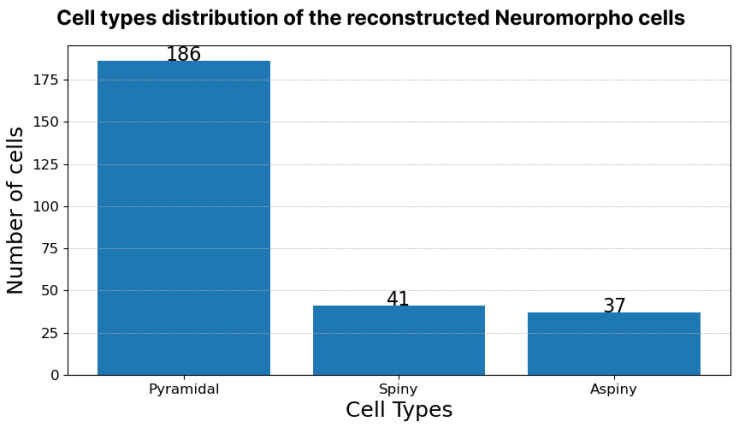

Figure 4.26: *The cell types distribution of the reconstructed NEUROMORPHO cells (Pyramidal, Spiny, and Aspiny).*

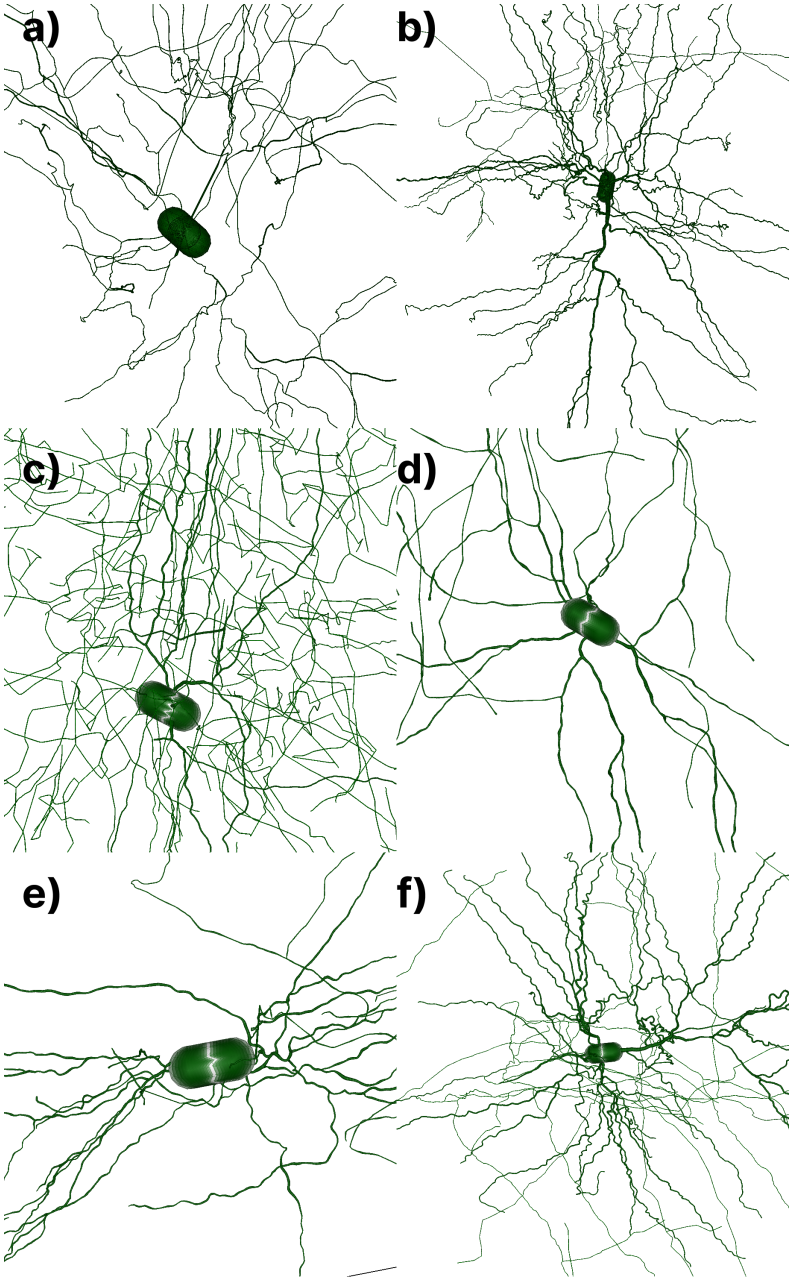

Figure 4.27: *Different sphere-reconstructed cells from the NEUROMORPHO dataset. Panels a) and b) show pyramidal neurons with the following IDs: 541549258 and 541557114, respectively. Panels c) and d) show examples of aspiny neurons of IDs* $H15-06-016-01-03-02\_549378277$, $H15-06-016-01-12-01\_623320232$. *Finally, panels e) and f) represent spiny neurons with the following IDs:* $H16-03-005-01-04-03\_697383950$ *and* $H16-03-003-01-19-02\_556380210$.

Another visual of six sample cells reconstructed from the NEUROMORPHO dataset is illustrated in Fig.(4.27). This illustration showcases the diversity of the available cells within the NEUROMORPHO dataset. Each cell occupies a given amount of space depending on the extent of its processes. For instance, some cells have a large extent as a consequence of their very long processes, while other cells have a smaller spatial extent as they have shorter processes. The co-existence of large and small cellular features in neurons has important memory implications in virtual gray matter microstructure, especially when the small cellular features, i.e., processes, have

a large spatial extent. To shed light on this, a simple memory analysis was performed regarding the memory size of the reconstructed cells.

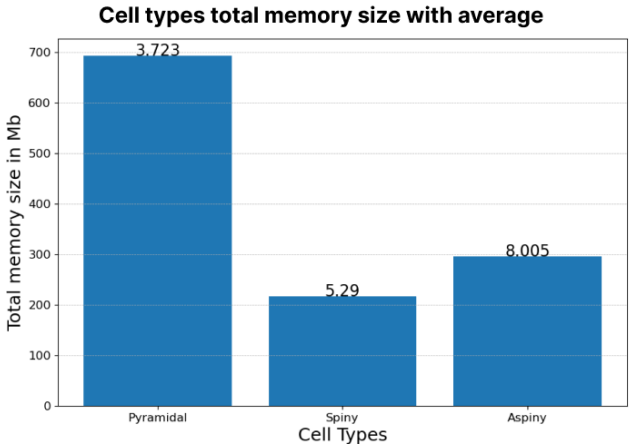

Figure 4.28: *The total memory sizes in Megabytes for each cell type and the average memory sizes per cell are displayed above each bar.*

The illustration in Fig.(4.28) shows the aggregate memory size for each type of cell. The per-cell memory size average is larger for the aspiny cells, with an average of 8 Megabytes per cell, then in second place comes the spiny cells, with an average size of 5.29 Megabytes per cell. Finally, the pyramidal cells are the lightest cells in terms of memory storage. The memory size of cells is directly obtained from the number of spheres used to represent the cells. This number depends on the topological graph structure of the cell, i.e., the number of edges and their radii, and on the spatial extent of the cell.

Discussion. The public NEUROMORPHO dataset offers a convenient source of neurons that can be reconstructed efficiently using the proposed method, which allows a tailored representation quality through the overlapping ratio parameter. Although a thorough runtime performance analysis was not performed, the reconstruction time was moderately fast and did not exceed a few seconds. The reconstructed cells have different memory sizes with the pyramidal cells being the lightest type of cells. In addition, in contrast to the H01 dataset presented previously, the NEUROMORPHO cells have relatively simple shapes with limited realism: The soma of the cells stemming from the NEUROMORPHO dataset have a capsule-like geometry, and the processes have circular cross-sections. Although given these limitations, the NEUROMORPHO dataset remains widely adopted for its simple format, community support which appends the dataset with new cells, and finally, its user-friendliness.

4.4.4 Conclusion

This section presented a new step towards integrating more datasets for gray matter microstructure. This was done by using a sphere-based reconstruction method for the widely adopted NEUROMORPHO cell repository. In addition, the use of the overlapping ratio metric allows a simple controlled way to choose the desired sphere-based representation's quality. As the reconstruction uses ".SWC" files, future directions of this work include reconstructing purely synthetic cells generated using the method proposed in [Palombo *et al.*, 2021].

4.5 Generating virtual gray matter hybrid microstructure

The previous sections consisted in reconstructing different gray matter cells to obtain sphere or ellipsoid-based geometries. Although given a large pool of virtual cells, but the transition from such individual cells to 3D realistic microstructure substrates is a challenging research problem still under investigation. This section presents our contributions to generating virtual gray matter substrates. First, the MEDUSA framework is presented in detail. Then, the virtual gray matter substrates generation method is presented.

4.5.1 The MEDUSA framework

MEDUSA stands for **M**icrostructure **E**nvironment **D**esigner with **U**nified **S**phere **A**toms. It is a toolkit initially introduced to generate white matter substrates using a sphere-based representation. This choice was motivated by the efficiency of the substrate generation process and for verifying collisions between different cells. The MEDUSA toolkit provides a generative framework for white matter realistic phantoms, the generation process follows these three steps:

1. **Generate virtual substrates.** This step consists in reading a JSON-like parameters input file which contains the microstructure generative parameters of a target virtual tissue. More precisely, it contains a list of cell populations to be generated and their corresponding properties and their corresponding values. The cell types that can be generated using the MEDUSA framework are axonal fibers, astrocytes, and oligodendrocytes. A detailed description of the used parameters for each cell type is illustrated in Fig.(4.29). Once the parameter file is read, virtual cells are generated in a volume of interest (VOI). The generation method depends on the type of cell being generated, and after the generation process, the sphere-based cell is appended to the VOI. The cell generation process is repeated until an ending condition is satisfied, for example, a target volume fraction of a given cell population is met, or a maximum number of spheres in the VOI is reached. At the end of this step, a VOI containing a set of cells represented as a set of spheres is obtained. However, these cells can show a high degree of overlap. This overlap is treated in the next step.

2. **Removing overlaps between different cells.** Given a VOI containing cells represented with spheres. This step consists in removing collisions between different cells. As cells are represented with a set of overlapping spheres, collisions between spheres of the same cells are considered to be undesired overlaps. In fact, only pairs of spheres belonging to different cells are counted as cases of overlapping. Since each sphere in the VOI has two identifiers, one for the population and the other for the cell type it represents, the overlap can efficiently be detected by simply running a collision test between two spheres. When a collision between two spheres $\mathbf{s}_1$ and $\mathbf{s}_2$ belonging to two different cells is detected, an optimization step is performed by calculating a repulsion force $F(\mathbf{s}_1, \mathbf{s}_2)$ with a direction following the straight line connecting the two colliding spheres and a magnitude given by:

$$\begin{cases} O(\mathbf{s}_1, \mathbf{s}_2) = \max(r_1 + r_2 - d(x_1, x_2)) \\ F(\mathbf{s}_1, \mathbf{s}_2) = \frac{O(\mathbf{s}_1, \mathbf{s}_2)}{2} \frac{x_1 - x_2}{|x_1 - x_2|}. \end{cases} \tag{4.8}$$

A GPU-based implementation of this de-overlapping scheme allows an efficient minimization of collisions between different cells.

a) Generative parameters for an axon population

Geometrical characteristics	Control parameter names	Parameters description
Axon diameter distribution	$\mathbf{u}$	Mean orientation of the population
	Φ_f	Target volume fraction of the population
	$\langle D_f \rangle$	Axon diameter mean value
	σ_{D_f}	Axon diameter standard deviation
Global angular dispersion	GAD	Target global angular dispersion
Local tortuosity	LAD	Additional angular dispersion due to tortuosity
Ranvier nodes	$\langle R \rangle$	Mean ratio (internodal length)/ (node width)
	σ_R	Ratio standard deviation
Beading	$< BS >$	Beading spacing mean value
	σ_{BS}	Beading spacing standard deviation
	$< BA >$	Beading amplitude mean value
	σ_{BA}	Beading amplitude standard deviation
Myelin sheath	$< g >$	Mean g-ratio value
	σ_g	g-ratio standard deviation

b) Generative parameters for an astrocyte population

Geometrical characteristics	Control parameters names	Parameters description
Astrocyte distribution	Φ_a	Volume fraction
	$\langle D_{at} \rangle$	Total diameter mean
	$\sigma_{D_{at}}$	Total diameter standard deviation
Astrocyte body	$< D_{ab} >$	Body diameter mean
	$\sigma_{D_{ab}}$	Body diameter standard deviation
Astrocyte processes	$< D_{ap} >$	Processes diameter mean
	$\sigma_{D_{ap}}$	Processes diameter standard deviation
	BF_a	Balancing factor
	AT_a	Angular threshold

c) Generative parameters for an olygodendrocyte population

Geometrical characteristics	Control parameter names	Parameters description
Oligodendrocyte distribution	Φ_o	Volume fraction
	$< D_{ot} >$	Total diameter mean
	$\sigma_{D_{ot}}$	Total diameter standard deviation
Oligodendrocyte body	$< D_{ob} >$	Body diameter mean
	$\sigma_{D_{ob}}$	Body diameter standard deviation
Oligodendrocyte processes	$< D_{op} >$	Processes diameter mean
	$\sigma_{D_{op}}$	Processes diameter standard deviation
	S_r	Axons search radius
	P_c	Processes count
	BF_o	Balancing factor

Figure 4.29: *The generative parameters for white matter cells in the MEDUSA framework (image adapted from [Ginsburger et al., 2019]).*

The number of iterations of the de-overlapping minimization algorithm depends on the geometric complexity of the generated substrate. For simple geometries, the minimization algorithm converges and removes the overlaps within a finite number of steps. However, in more complex geometries, convergence may not be attained. Thus, a maximum number of iterations, typically a few hundreds, is specified by the user in the JSON-like parameters file. Also, when the de-overlapping may not converge, the selection of the maximal number of iterations is a tradeoff between computational cost and the number of collisions in the substrate. When set low, the number of iterations does not significantly reduce collisions between different cells. On the other hand, when set very high, the optimization step can become computationally expensive.

3. **Post processing the virtual substrate.** After de-overlapping the virtual substrate, this step performs post-processing. There are different types of post-processing that can be performed and these are specified by the user in the JSON-like parameters input file used in step 1. Some of the possible post-processings are: smoothing the virtual substrate, connecting parts of the cells that were disconnected after the de-overlapping step, and others. Also, some of these post-processing methods are cell-specific and they can also be combined.

JSON-like parameters file structure

JSON-like parameters structure without full details

```
attributes = {
  "field_of_view": [-50.0, -50.0, -50.0, 50.0, 50.0, 50.0],
  "grid_resolution": 3.0,
  "apply_remove_overlaps": 1,
  "apply_atom_regularization": 0,
  "apply_add_details": 0,
  "atom_overlap_solver_maximum_iteration_count": 120,
  "maximum_force_norm": 0.25,
  "repulsion_force_moving_average_window_size": 10,
  "minimum_repulsion_force_attenuation_percentage": 10.0,
  "repulsion_force_stddev_percentage_threshold": 1.0,
  "maximum_atom_count": 700000000,
  "populations": {
    "fiber_population_01": {
      "parameters": {
        "axon": {
          "diameter_distribution": {
            ...
          },
          "global_angular_dispersion_in_degrees": 13.36,
          "has_beading": 0,
          "has_tortuosity": 1,
          "mean_orientation": [0.0, 0.0, 1.0],
          "tortuosity": {
            ...
          },
        "has_myelin_sheath": 0,
        "myelin_sheath": {
          "g_ratio_distribution": {
            ...
          },
          "has_ranvier_nodes": 0,
          "internodal_length_to_node_width_distribution": {
            ...
          }
        },
        "sphere_atom_oversampling_ratio": 20.0,
        "volume_fraction": 0.85
      },
      "type": "fiber-population"
    }
  }
}
```

Details of the fiber population parameters

```
{
  "fiber_population_01": {
    "parameters": {
      "axon": {
        "diameter_distribution": {
          "parameters": {
            "mean": 2.0,
            "multiplicator": 1000.0,
            "stddev": 0.45,
            "upper_boundary": 10.0
          },
          "type": "gamma-distribution"
        },
        "global_angular_dispersion_in_degrees": 13.36,
        "has_beading": 0,
        "has_tortuosity": 1,
        "mean_orientation": [0.0, 0.0, 1.0],
        "permeability_distribution": {
          "parameters": {
            "mean": 2.0,
            "stddev": 1.0
          },
          "type": "gamma-distribution"
        },
        "tortuosity": {
          "angular_dispersion_in_degrees": 1.5,
          "magnitude": 0.05,
          "wave_length": 10.0
        },
        "diffusivity_distribution": {
          "type": "gamma-distribution",
          "parameters": {
            "mean": 0.0,
            "stddev": 0.1
          }
        }
      },
      "has_myelin_sheath": 0,
      "myelin_sheath": {
        "g_ratio_distribution": {
          "parameters": {
            "mean": 0.6,
            "stddev": 0.06
          },
          "type": "gamma-distribution"
        },
        "has_ranvier_nodes": 0,
        "internodal_length_to_node_width_distribution":
          "parameters": {
            "mean": 40.0,
            "stddev": 0.5
          },
          "type": "gamma-distribution"
        }
      },
      "sphere_atom_oversampling_ratio": 20.0,
      "volume_fraction": 0.85
    },
    "type": "fiber-population"
  }
}
```

Figure 4.30: *Illustration of the JSON-like parameters file used to generate a fiber populations virtual substrate in the MEDUSA framework.*

To shed more light on the generation process of a virtual tissue, a sample JSON-like parameters file is presented in Fig.(4.30). The parameters file comprises general parameters about the generation such as the *"field_of_view"* field specifying the size of the VOI, the *"atom_overlap_solver_-*

maximum_iteration_count" setting the value of the maximum number of iterations of the de-overlapping optimization, etc.
The substrate-related parameters in the JSON-like parameters file are stored in the *"populations"* dictionary which contains the set of all the cell populations and their parameters. The illustrated parameters file in Fig.(4.30) enables to generate a substrate with one single population of fibers. The MEDUSA toolkit has proven to be an efficient way to generate complex white matter virtual substrates such as the sample illustrated in Fig.(4.31).

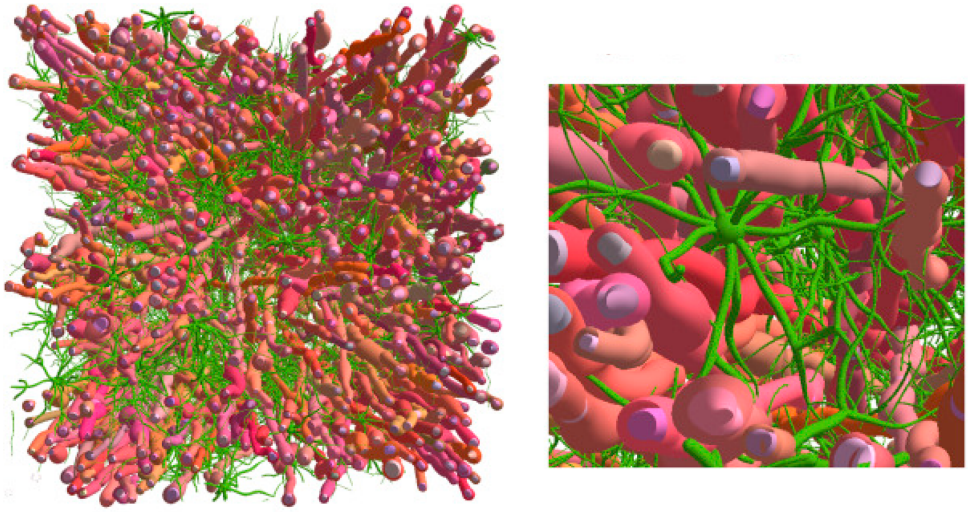

Figure 4.31: *Illustration of a white matter substrate generated with the MEDUSA framework (Image adapted from [Ginsburger et al., 2019]).*

4.5.2 Methods

1. Ellipsoids integration in MEDUSA. This section describes the integration of ellipsoids in the MEDUSA framework and the implementation of ellipsoid-related methods to generalize the framework. This part of the contributions aims to additionally include ellipsoids as units of geometric representation in the MEDUSA framework.
There are different elements to be considered for integrating the ellipsoid-based geometric representation into the MEDUSA framework:

- **Initialization.** This step consists simply in initializing arbitrary ellipsoids, set their positions, orientations, and perform basic transformations such as rotations and rescaling. In addition, in order to increase the memory efficiency, it is better to use the ellipsoid parameters stemming from the algebraic equation of the ellipsoid in Eq.(4.1). Also, it is necessary to be able to switch between different types of representations of the ellipsoid. For the initialization, given a set of ellipsoid parameters such as the center, axes, and radii, matrices are generated to get the compressed parameters (A, B, C, D, E, F, G, H, J, and K) of the ellipsoid.
  First, a scaling matrix S is initialized based on the radii (r_x, r_y, r_z) of the ellipsoid. Second, a rotation matrix R is constructed based on the main axes of the ellipsoid. Finally, a

translation matrix T is initialized based on the ellipsoid center $\mathbf{c} = (c_x, c_y, c_z)$.

$$S = \begin{pmatrix} \frac{1}{r_x^2} & 0 & 0 & 0 \\ 0 & \frac{1}{r_y^2} & 0 & 0 \\ 0 & 0 & \frac{1}{r_z^2} & 0 \\ 0 & 0 & 0 & -1 \end{pmatrix} T = \begin{pmatrix} 1 & 0 & 0 & -c_x \\ 0 & 1 & 0 & -c_y \\ 0 & 0 & 1 & -c_z \\ 0 & 0 & 0 & 1 \end{pmatrix} R = \begin{pmatrix} e_{1x} & e_{2x} & e_{3x} & 0 \\ e_{1y} & e_{2y} & e_{3y} & 0 \\ e_{1z} & e_{2z} & e_{3z} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \tag{4.9}$$

Given the three matrices S, R, and T, the matrix $\mathcal{A}$ is initialized as follows:

$$\mathcal{A} = TRSR^tT^t \tag{4.10}$$

The matrix $\mathcal{A}$ is the same in Eq.(4.3) and its elements encode the compressed coefficients of the ellipsoid. Thus, given the meaningful geometric parameters of the ellipsoid, one can obtain the compressed representation using the formula in Eq.(4.10).

- **Ellipsoid collision test.** To perform the de-overlapping step, it is important to test the intersection between two ellipsoids from different cells. For this, the intersection test between two ellipsoids [Alfano & Greer, 2003] was implemented in the course of this thesis. Given two ellipsoids with matrices $\mathcal{A}$ and $\mathcal{B}$, the collision verification method uses the eigenvalues of the following matrix: $\mathcal{A}^{-1}\mathcal{B}$. Based on the numerical values of the eigenvalues of the product, their degeneracy, and whether the numerical values are real or complex, it is possible to detect the intersection and its type as well, i.e., single or multiple intersection points.

The listed elements above are essential to integrate ellipsoids in the MEDUSA framework but not sufficient. Other software-related considerations were taken into account such as design patterns and abstractions to improve the code's design and allow both flexibility and ease of generalization for possible future geometric units of representations. For instance, an abstract class named Atom was used to encapsulate the geometric unit of representation, i.e., a sphere or an ellipsoid.

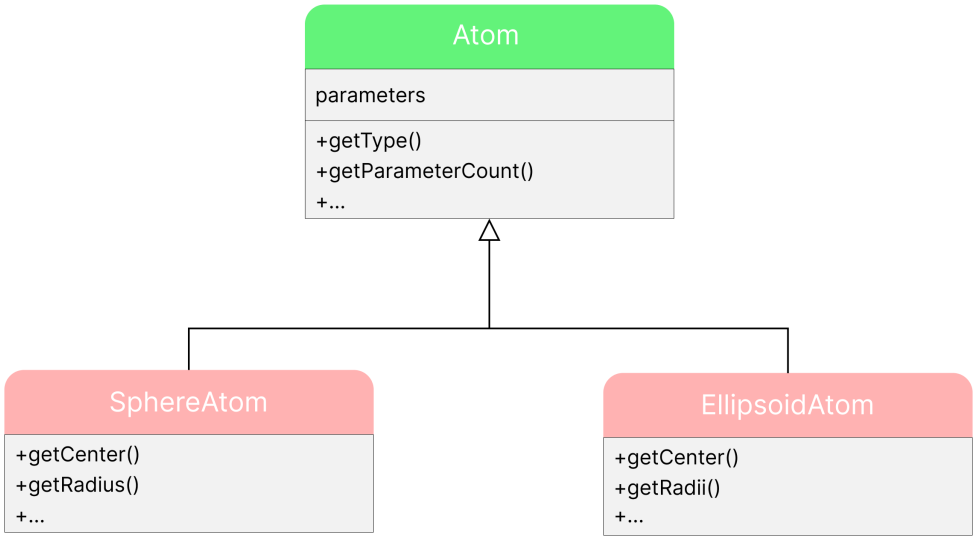

Figure 4.32: *The Unified Modeling Language (UML) diagram for a portion of the MEDUSA framework for the geometric unit of representation modeled by the class Atom.*

In addition, different factory design patterns were used to store different methods related to the

Atom objects. For example, a class AtomColliderFactory was developed to contain atom-related collision methods necessary to generate virtual substrates such as sphere-sphere, sphere-ellipsoid, and ellipsoid-ellipsoid collision tests. The use of these factories facilitates the handling of the generation process of virtual substrates regardless of the type of the geometric unit of representation used, thus, virtually allowing the generation of substrates containing both spheres and ellipsoids. A Test-driven development (TDD) approach was adopted during the development of the ellipsoid-related methods in the MEDUSA framework. For instance, regarding the development of the ellipsoid collision test method, tests were created to cover different scenarios: non-intersecting ellipsoids, two intersecting ellipsoids, and the same two intersecting ellipsoids but translated in space. The TDD approach was paramount to many ellipsoid-related developments, including the ellipsoid initialization and retrieving the intuitive geometric parameters such as the center and the three axes of the ellipsoid from the compressed parameters in the matrix representation $\mathcal{A}$. In this case, an ellipsoid was initialized from the given parameters: center and axes are used to calculate the compressed parameters which then are used to obtain back the original parameters used for the initialization. These developments can be found open source on the Git repository in the following link: https://framagit.org/cpoupon/gkg/-/tree/master/simulation/src/library/gkg-simulation-virtual-tissue

2. External data integration. This step represents an important contribution in the frame of this work and consists of adding the possibility of using external data sources in the MEDUSA framework. For this, the following steps were performed:

- **Adapt external cells to a MEDUSA-compatible file format.** This step consists in using the decomposed cells stemming from both the H01 and NEUROMORPHO datasets -see previous sections-. For each dataset, morphometrics features were extracted or simply collected when available. Then, the decomposed cells were written to disk using a specific file format named "*.medusageometry*" used in the MEDUSA framework to store virtual tissues. A simplified version of this file format is illustrated below:

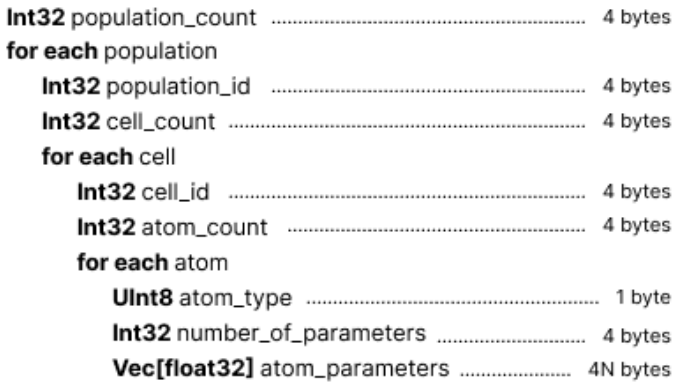

Figure 4.33: *Illustration of the structure of the medusageometry binary file format.*

A cell is described using two files, the ".medusageometry" file and another file containing meta-information about the cell such as the number of atoms, its bounding box coordinates, the volume of the soma, and the radii of the effective ellipsoid representing the soma.

- **MEDUSA-related developments.** In this part, the cells written to disk as *medusageometry* files were stored in folders. Cells within each dataset were saved in folders according to their cell types. For example, pyramidal cells from the NEUROMORPHO dataset were stored in `Neuromorpho/pyramidal`. In addition, two versions of cells from each dataset were created. The first version contains the original decomposed cells and the second contains the same cells with a single ellipsoid representing the soma. This choice was motivated by the need to have exact correspondence between microstructural parameters and the soma shape; this is to make sure that in case of potential bias in a microstructure estimating model, the bias is related to another factor than the actual soma-to-effective ellipsoid discrepancy. This also mitigates potential error from the method that approximates the soma to an effective ellipsoid. Further development was performed to enable reading the ".medusageometry" files in the MEDUSA framework.

3. The MEDUSA generation method. After organizing the decomposed cells into collections stored in folders, this part aims to generate virtual microstructure substrates that combine external data cells with synthetic cells. To generate such hybrid virtual microstructures, the user can specify the type of cells in the JSON-like MEDUSA input parameter file. A new possibility was added to select cells from a specific dataset. In addition, other parameters, such as the type of soma to be used, i.e. ellipsoid-equivalent of the soma or the regular decomposition, the average radius of the soma, its variance, and finally, the volume fraction.

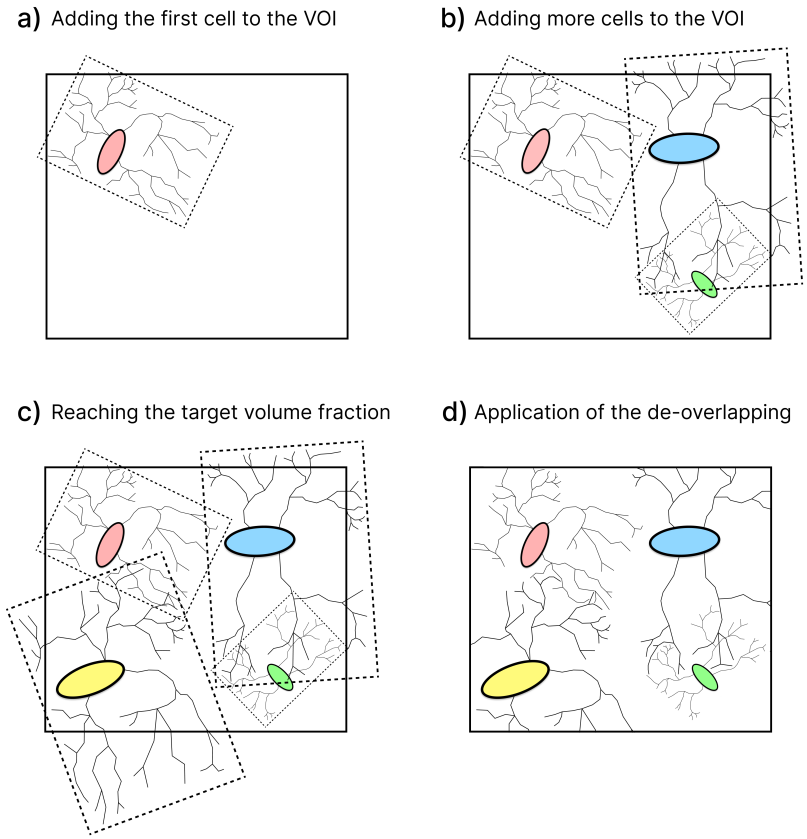

Figure 4.34: *Illustration of the generation steps of virtual substrates using external cells integrated into the MEDUSA framework.*

The generation process using cells from various datasets is the following:

1. Sample a random cell given the dataset source, its type, and the soma type.
2. Scale the cell according to the average diameter and its variance specified in the parameters file.
3. Perform random rotation of the whole cell.
4. Place the cell in a random position inside the volume of interest (VOI).
5. Repeat this process until the target volume fraction of cells in the VOI is reached.

The rotation and translation for each ellipsoid are applied to its center, axes, and radii which are obtained by converting the compressed parameters. Then, the new values of the center, axes, and radii are transformed back to the compressed representation with the set of ten parameters (A, B, C, D, E, F, G, H, J, and K). Note that the combination of these generation steps with the de-overlapping process allows to obtain highly diversified substrates as individual cells are deformed to avoid overlaps. Thus, it is likely that when the same cell is sampled twice inside a VOI, each copy will undergo different transformations and deformations due to the de-overlapping steps, leading to different cell shapes. A schematic summarizing the generation of virtual substrates from cells collected from open datasets is illustrated in Fig.(4.34).

4.5.3 Results

This section presents the results related to the extensions of the MEDUSA framework in the course of this thesis. It also includes the new generation process for virtual hybrid gray matter microstructure.

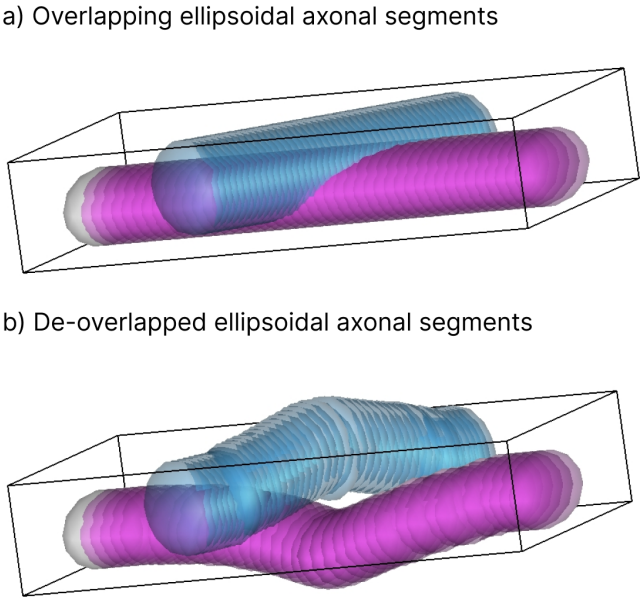

Figure 4.35: *Illustration of a crossing-fibers substrate sample before and after the de-overlapping process.*

Removing overlaps for ellipsoid-based geometries. First, a simple synthetic substrate was generated to showcase the generalization of the MEDUSA de-overlapping process. The substrate consists of two segments of axons that substantially overlap which was used to test the de-overlapping process.

Two intersecting axonal segments, one in clear blue and the other in pink displayed in panel a) of Fig.(4.35), are made of a set of overlapping ellipsoids such that each segment has 45 similar ellipsoids -atoms-. For the first segment, the values of the ellipsoid radii are $[1, 2.4, 1.2]$, while for the second segment, the ellipsoid radii are $[1, 2.25, 1.5]$. The illustration in panel b) of Fig.(4.35) clearly shows the generalization of the de-overlapping process for the case of ellipsoid-atoms.

Proof of concept of hybrid substrates. In this part, four different substrates were generated to establish a proof-of-concept for the ability of the MEDUSA toolkit to generate hybrid substrates mixing cells from existing datasets with purely synthetically generated cells. The four generated substrates for the demonstration are the following:

- **case a) Single H01 pyramidal cell population.** A first demonstration of the possibility to generate substrates from the H01 cells in the MEDUSA framework. A sample substrate is illustrated in Fig.(4.36).
- **case b) Single NEUROMORPHO Pyramidal cell population.** A first substrate made of pyramidal cells stemming from the NEUROMORPHO dataset. A sample substrate is illustrated in Fig.(4.37).
- **case c) A multipolar H01 neuron population with synthetic microglia.** A first hybrid substrate mixing multipolar neurons from the H01 dataset with a purely synthetic microglia population. A sample substrate is illustrated in Fig.(4.38).
- **case d) A pyramidal and multipolar neuron populations from the H01 dataset.** Finally, a substrate containing two cell types from the same H01 dataset is illustrated in Fig.(4.39).

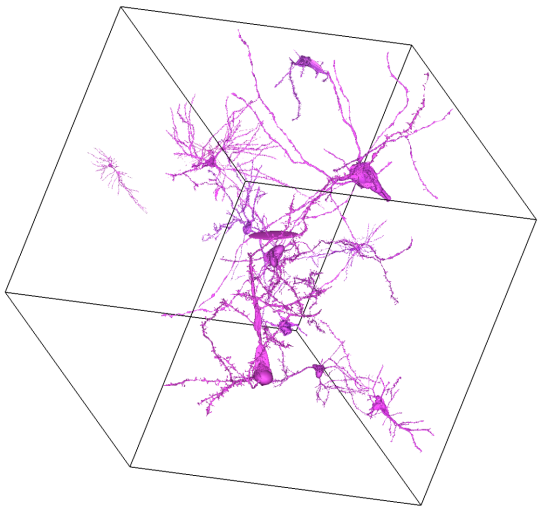

Figure 4.36: *Illustration of the first proof of concept of a MEDUSA-generated substrate with pyramidal cells from the H01 dataset corresponding to case a).*

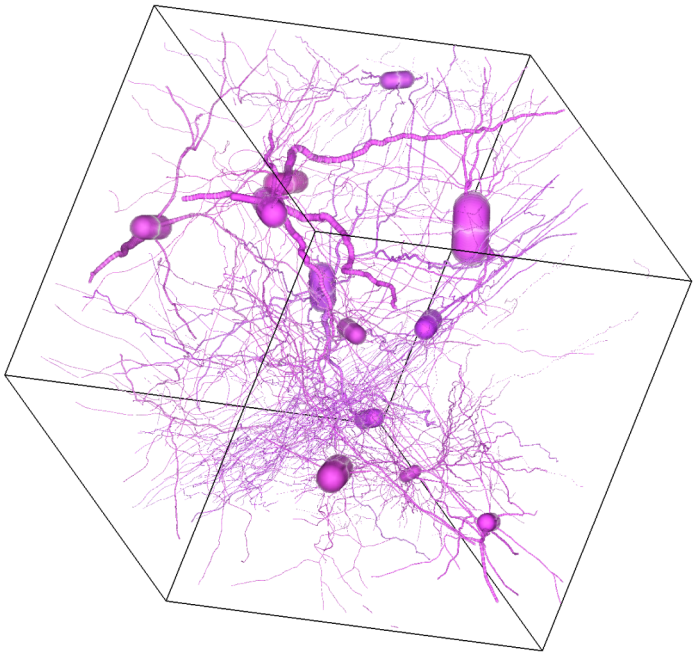

Figure 4.37: *Illustration of the second proof of concept of a MEDUSA-generated substrate with pyramidal cells from the NEUROMORPHO dataset corresponding to case b).*

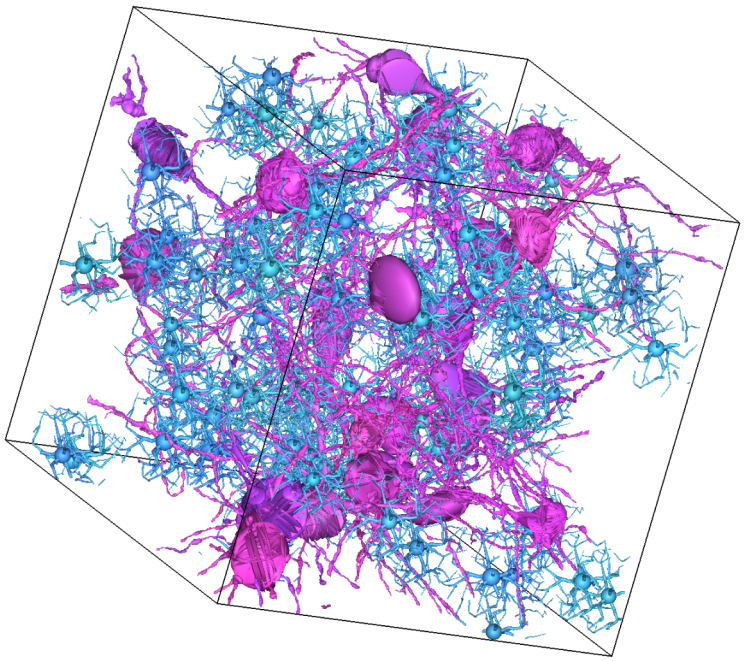

Figure 4.38: *Illustration of the first proof of concept of a hybrid MEDUSA-generated substrate with H01-based multipolar neurons in pink and purely synthetic microglia in light blue corresponding to case c).*

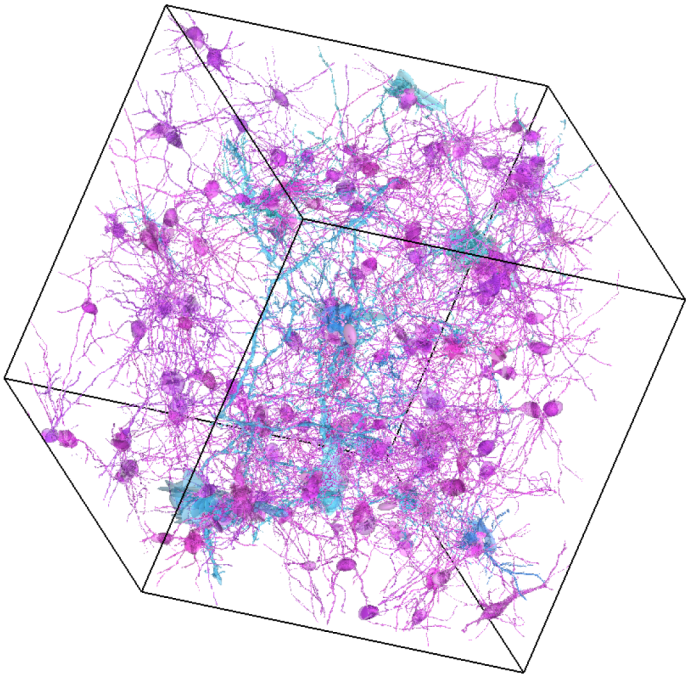

Figure 4.39: *Illustration of the first proof of concept of a hybrid MEDUSA-generated substrate with two cell types from the H01 dataset: multipolar (pink) and pyramidal (light blue) neurons. The substrate represents case d).*

Towards more realistic gray matter substrates. The second part of the results targets the generation of more realistic substrates with microstructural parameters close to those observed in the human brain. For this, four different types of substrates were considered:

1. **White matter.** A classic MEDUSA-generated white matter axon-only virtual substrate with a volume fraction (VF) of 85%, an average diameter $< d_a >=$ 2µm and an angular dispersion of 1.49°.

2. **Gray matter hybrid substrate.** A first hybrid substrate combining axons with pyramidal neurons from the NEUROMORPHO dataset. The axons have a VF of 42% and an average diameter of 2µm, while the neurons have a VF of 6% and an average diameter of 17µm.

3. **Gray matter hybrid substrate.** A second hybrid substrate similar to substrate 2 but in different volume fractions (VFs). In this case, the white matter axons have the same properties as substrate 2. and the neuron population are slightly denser with a VF of 12%.

4. **Neuron-only virtual substrate.** A final substrate containing only pyramidal neurons from the NEUROMORPHO dataset with an average diameter of 17µm and a VF of 15%.

Generated virtual substrates are shown in Fig.(4.40) such that each sample represents a specific substrate from the four types listed above.

4.5.4 Discussion

In this part of the work, we demonstrated the ability of the extended MEDUSA framework to generate not only white matter virtual substrates but also gray matter ones. Our first contribution

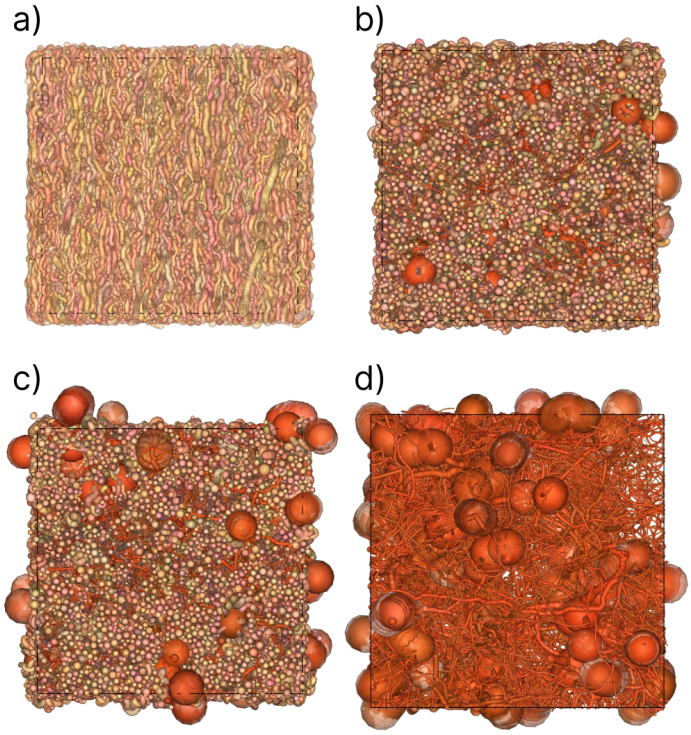

Figure 4.40: *Illustration of four substrate types: substrate a) models a white matter only substrate, panels b) and c) are hybrid substrates mixing axons with pyramidal neurons from the NEUROMORPHO dataset. Finally, the substrate d) contains a single NEUROMORPHO-based pyramidal neuron population.*

regarding the generalization of the MEDUSA framework to include ellipsoids allows to increase the realism of the virtual substrates while at the same time maintaining a memory-efficient way to store and handle geometries. In addition, the demonstrated hybrid approach is particularly important. Since the EM-reconstructed cells are limited due to the expensive data collection process, pure synthetic cells can be advantageous in order to overcome the data scarcity problem. However, hybrid virtual substrates can provide a bridge between purely synthetic and human-reconstructed microstructure substrates which allows building more diverse datasets for training and testing microstructure inference models. Again, ideally, a microstructure inference model's performance should not depend on the substrate generation method, whether it is fully synthetic or obtained from EM data. Also, following the same reasoning, a microstructure inference model's performance should not drop when estimating microstructure from substrates generated with different methods (MEDUSA, CONFIG, CACTUS, or others). Moreover, the ability of the MEDUSA framework to integrate cells from external data sources is particularly interesting and opens up new possibilities. For instance, integrating purely synthetic neurons from the following research works [Palombo *et al.*, 2019]. Although its feasibility was proven in this work, the integration of purely synthetic cells into the MEDUSA framework to generate new substrates was left for future work directions. It is also important to notice that in the case of hybrid substrates using external sources of cells, when the same cell is sampled twice or more times, the combination of random rotation, scaling, and the de-overlapping process ensures a certain variability between the various transformed versions of the same cell. This effect is more pronounced for the de-overlapping which

introduces tortuosity in different parts of the cells.
In addition, this work demonstrated that the generated hybrid substrates using cells from the NEUROMOPHO dataset display similar microstructure parameter values to those observed in the human brain according to recent human brain histology research, thus, getting a step closer towards more realistic virtual gray matter substrates.
Despite the progress in generating virtual gray matter microstructure, there are still some limitations preventing the generation of such substrates at large scale. First, we found that the de-overlapping process using ellipsoids was significantly more computationally expensive than spheres. Thus, in a large scale simulations setting, for the same compute budget, using the ellipsoid-based reprentations can lead to lower number of samples compared to using sphere-based ones. Although some efforts to reduce the de-overlapping runtime for ellipsoids, such as using bounding boxes as a proxy collision test, for highly packed geometries, still, the number of collision remained high leading to increased de-overlapping times. Therefore, regarding this challenge, future work directions aim at exploring learning-based methods to perform the de-overlapping of cells. Instead of performing an iterative optimization process as in MEDUSA, a neural network can learn to perform the de-overlapping process given a dataset of overlapped geometries and their corresponding de-overlapped version. Architectures such as graph networks may be more adapted to the de-overlapping tak, we also orient our reader towards this research work [Shao *et al.*, 2021] which combines graph networks and a classical iterative solver to speedup an optimization process.
In the same discussion, performing the de-overlapping is only expensive when used to substantially minimize the collisions between different cells. This is based on the assumption that the number of contacts and collisions between cells in the human brain is minimal. However, according to recent research [Shapson-Coe *et al.*, 2021], this assumption may not necessarily hold since cell collision cases are present in different forms in the human brain. Some of these cases are illustrated in Fig.(4.41) which include different parts of cells can be inside another cell, this also showcases the complexity of human brain microstructure.
Another limitation associated with relying on a hybrid approach to generate virtual microstructures is the limited control of certain parameters. For instance, when using the NEUROMORPHO dataset as an external cellular data source, it is not possible by design to vary neuron-related parameters such as the number of branches or to set the dendrite's average diameter independently of the soma diameter. Thus, the hybrid approach using open datasets does not allow fine-grained control over parameters of the brain microstructure. This limitation can be overcome by using purely synthetic cells as an external data source integrated with the MEDUSA framework.

4.5.5 Conclusion

This section demonstrated the new capabilities of the extended version of the MEDUSA framework in integrating custom cells from different datasets, supporting the use of ellipsoids for both reading and generating microstructure geometries. The integration of external cells allowed the generation of gray matter microstructure with parameter values close to the ones from human brain samples, marking a step closer to the scalable generation of gray matter microstructure

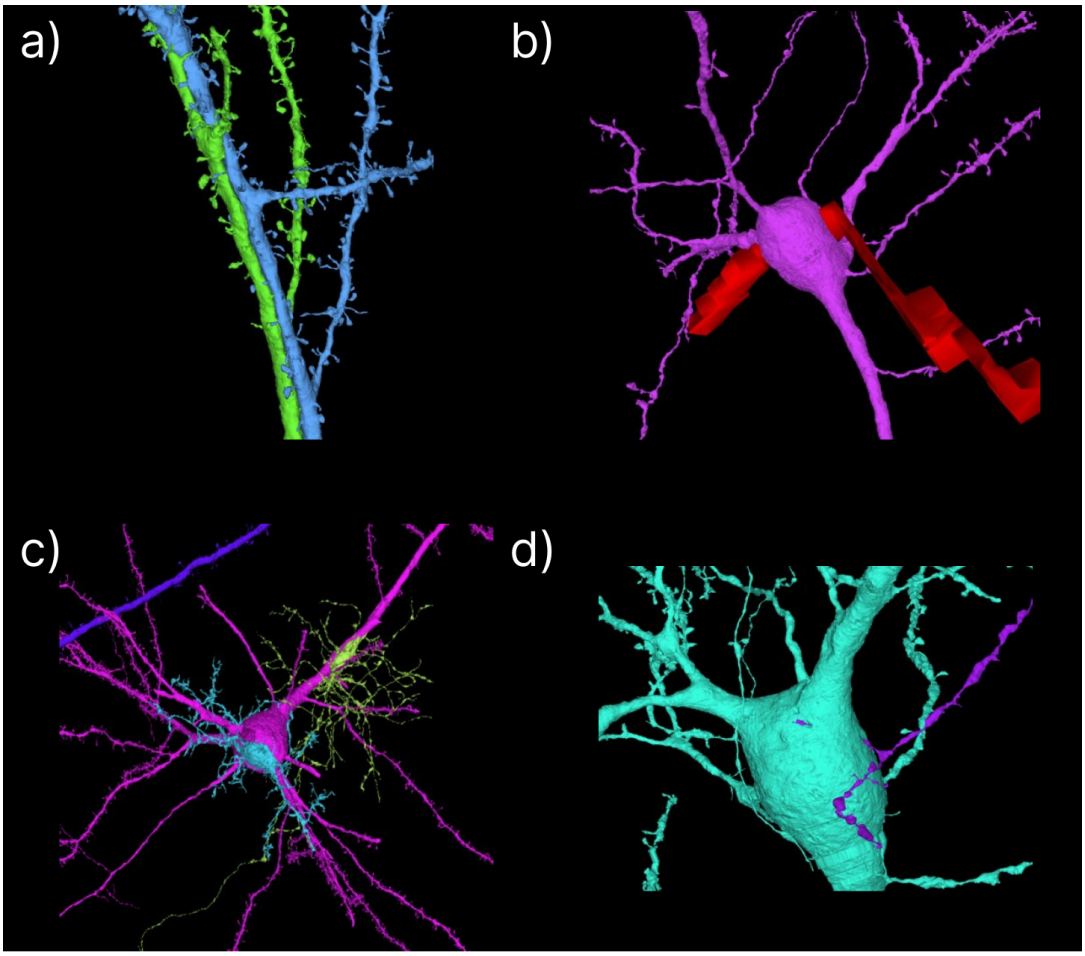

Figure 4.41: *Illustration of collision cases reported in the H01 dataset. Panels a) dendrites from two separate axons closely entangled and running side by side, b) a pyramidal neuron soma wrapped around a blood vessel, c) many glial cells appear to be glued to a pyramidal cell, and finally, panel d) represents a dendrite located inside a soma (images adapted from* `https://h01-release.storage.googleapis.com/gallery.html` *).*

substrates.

4.6 Summary and outlook

This chapter demonstrated the convenience of using atom-based representations to describe virtual substrates. Their compression ability offers a lightweight memory alternative to describe substrates. In addition, the SGEAD decomposition method allowed the conversion of mesh-based to ellipsoid-based geometries. The SGEAD decomposition applied to the H01 dataset and the sphere-based reconstruction of the NEUROMORPHO dataset allowed the construction of a diverse pool of atom-based cells. Combining the decomposed cells with the MEDUSA framework allowed the generation of hybrid substrates, thus pushing the state-of-the-art in gray matter microstructure modeling methods. Future contributions in this area include a more efficient C++-based implementation of the SGEAD method and the decomposition of purely synthetic substrates generated with other methods such as CONFIG [Callaghan *et al.*, 2020], CACTUS [Villarreal-Haro *et al.*, 2023], or others. While the current chapter presented details about the microstructure substrate generation process with atom-based representations, it did not present any details about performing Monte Carlo dMRI simulations which is important to efficiently characterize different microstructures and to learn a computational dMRI signal model.

Chapter 5

Contributions in diffusion MRI simulations

This chapter focuses on the contributions made in dMRI simulations in the course of this thesis. It covers different aspects and details about performing dMRI simulations on the types of virtual substrates presented previously. First, a scheme of performing dMRI simulation on ellipsoid-based geometries is presented, and then, in a second place, the dMRI simulation method is validated for known cases. Finally, a proof of concept is made to showcase the ability of the MEDUSA toolkit to obtain dMRI signals from complex virtual microstructure substrates.

5.1 Introduction

Acquiring actual MRI data is the most straightforward approach to study microstructure. However, as discussed previously, limitations related to collecting brain samples and extracting microstructure from them constitute challenges yet to overcome. In addition, restrictions of imaging systems in terms of attainable values of gradient magnitudes, slew rates, and other hardware parameters make the simulation approach more convenient. This approach not only allows the exploration of different parameters in a flexible manner but also the study of different microstructures in an economical way. Therefore, the simulation approach becomes an interesting alternative. Since most current diffusion MRI simulators are mesh-based and given some of their limitations in the context of substrate generation and dMRI simulations, there is an increasing need for new dMRI simulators inputting substrates with different types of geometric representations. Particularly the sphere and ellipsoid-based geometric representations as they provide a memory-efficient representation of substrates. In addition, the previously discussed contributions in gray matter microstructure generation only make sense if one is able to extract dMRI signals from them. This chapter presents in more detail the previous developments of the MEDUSA dMRI simulator and answers the question of how to obtain simulated dMRI signals from atom-based substrates. In addition, we address important questions such as the effect of using sphere or ellipsoid-based geometries on the dMRI signals. Also, the question of whether there are any significant differences between the simulated dMRI signals when using an atom-based or the classical mesh-based dMRI simulator.

5.2 MEDUSA dMRI simulator

It is important to emphasize that the MEDUSA simulator does not only generate virtual substrates but also performs Monte Carlo simulations to obtain dMRI signals. Although only the microstructure generation part was published in [Ginsburger *et al.*, 2019], previous versions of the MEDUSA simulator date back to the DMS simulator [Yeh *et al.*, 2013, Yeh, 2011]. The original DMS simulator relies on a mesh-based representation and is implemented in C++ using an object-oriented design with parallel processing capabilities.

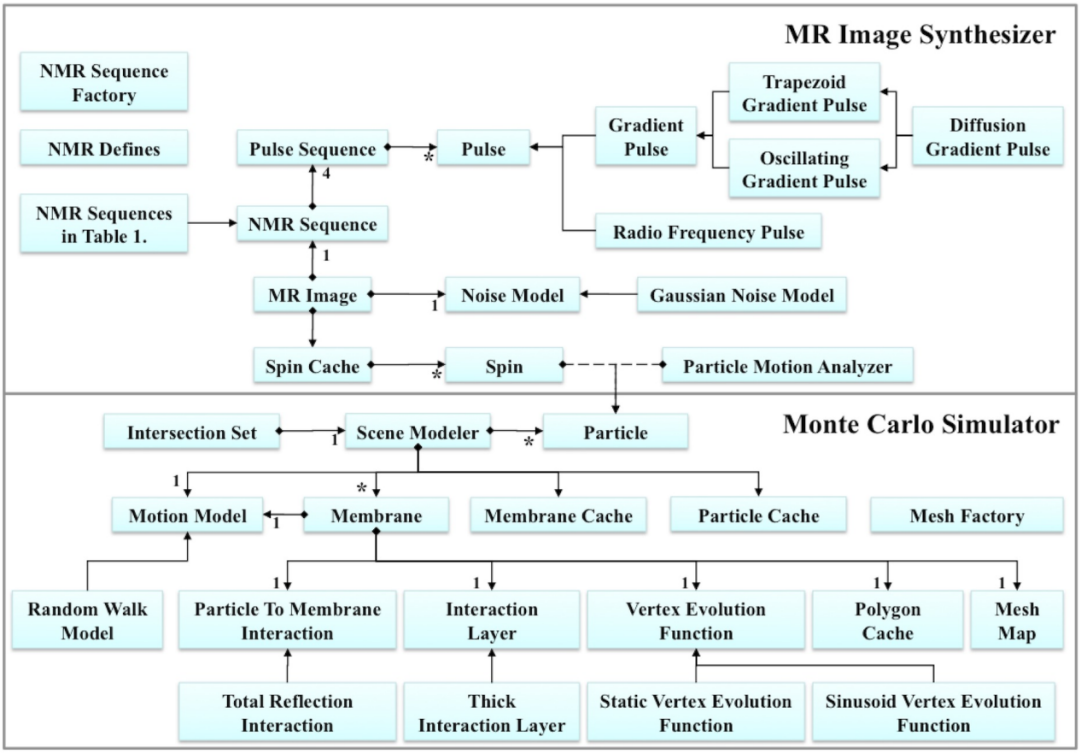

Figure 5.1: *The UML diagram of the original DMS software with separate entities: the top one for the MR Image synthesis and the other for Monte Carlo simulations (image taken from [Yeh, 2011]).*

The UML diagram depicted in Fig.(5.1) shows a separation between the Monte Carlo simulation and MR Image synthesizer modules. This separation implied the need to write the simulation particle trajectories to disk and to read them again to generate synthetic dMRI signals. Writing and reading particle trajectory data can be cumbersome and time-consuming. To overcome this limitation, the two separate entities of the DMS simulator were merged during K. Ginsburger's thesis to circumvent the need to write and read large trajectory data. This marks the beginning of the new MEDUSA dMRI simulator. In addition to the merging of the MR Image synthesizer and the MC simulator, the DMS code base was re-written in both **CUDA** and **C++** using the **Kokkos** framework [Trott *et al.*, 2022] to enable HPC portability [Ginsburger, 2019]. Performance benchmarks [Ginsburger, 2019] showed that the Kokkos implementation was superior in terms of the advantages it offers. The Kokkos implementation had similar performance as CUDA on GPU while maintaining at the same time good performances on CPU. In addition, the portability aspect of the **Kokkos** framework allows running the dMRI simulation code to almost any

HPC facility, disregarding the type of device used, i.e. CPU or GPU. Thus, the Kokkos implementation offers a convenient means to maintain a good performance while at the same time reducing the efforts needed to adapt the codebase to specific HPC facilities. The **Kokkos** implementation has provided a successful proof of concept and consequently guided the choice of re-designing the MEDUSA simulator. The new design intends to jointly improve the code performance and readability, leading to a new software architecture to which further improvements were made by A. Brullé and myself under the direction of C. Poupon. In the new design, the use of Kokkos was also generalized to the substrate generation part.

Volume of Interest (VOI) subdivision. The MEDUSA Monte Carlo dMRI simulator relies

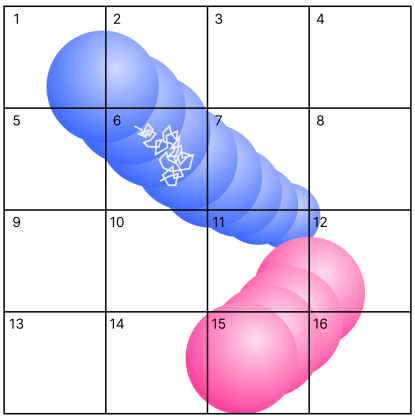

Figure 5.2: *Illustration of the data structures used in MEDUSA to perform MC dMRI simulations.*

on sphere-based geometries to synthesize dMRI signals. For this, the substrate is located inside a VOI (Volume of Interest) which is simply the substrate's bounding box. This bounding box is further subdivided into multiple smaller cuboids. The cuboids' exact size, named the grid resolution, is a hyperparameter specified by the user when performing dMRI simulations. Such a subdivision of the substrate is crucial and allows an efficient parallel implementation of collision detection between different objects. In the first place, the case of collision detection of interest was between a sphere and a 3D segment representing the spin's random displacement in 3D space. Such subdivisions are well-known in computer graphics and game development communities and more information can be found in the following link: https://developer.nvidia.com/gpugems/gpugems3/part-v-physics-simulation/chapter-32-broad-phase-collision-detection-cuda.

A simple 2D illustration of such a subdivision of the substrate is presented in Fig.(5.2). In this case, the grid resolution has the same size in all dimensions, and the subdivision has a cubic form or simply a square in the 2D space, as illustrated in Fig.(5.2). Given the subdivision, voxel-related data are stored in flat Kokkos arrays. For this, the number of atoms is computed for each voxel. For instance, in the case of the illustration depicted above, voxels of IDs 1, 2, and 12 have, respectively, an atom count of 3, 4, and 4. The atom count per voxel allows the initialization of a new Kokkos flat array which contains the atom IDs per voxel.

Initialization of spins. Given the subdivided structure illustrated in Fig.(5.2) and the flat array of atom IDs per voxel. The initialization step of spins requires the following parameters: the number of spins and the type of initialization. The main types of spin initialization are intracellular, extra-cellular, and everywhere. However, these are not the only types used in the course

of this thesis. Other customized initializations were developed for validation purposes and will be presented in the next sections of this chapter. Given a certain type of particle initialization, intra-cellular, for example, voxels can be processed in parallel to randomly sample new spin positions inside the voxel. More precisely, given the sampled random position, its voxel ID, and the array of atom IDs in the given voxel ID, the random position can be tested to check if it belongs to the spheres within the voxel ID. When the generated position is found to belong to one of the atoms, the spin is kept, and its ID becomes equal to the atom's ID to which it belongs. Otherwise, when no atom contains the sampled spin's position, a new random is resampled. This process is repeated until the target total number of particles is reached. A detailed description of the intra-cellular particle initialization pseudo-algorithm is displayed in Alg.(3).

procedure InitializeOneParticleIntraCellular
    $\mathbf{r} \leftarrow$ Random initialized position
    $N_a \leftarrow$ Atom count in the voxel of interest
    $A_{ID} \leftarrow$ Array of Atom IDs in the voxel
    $i \leftarrow 0$
    ParticleOutside $\leftarrow$ true
    while ParticleOutside **and** $i < N_c$ **do**
        if
            $PositionInsideAtom(\dots)$ **then**
            ParticleOutside $\leftarrow$ false;
            $UpdateSpinID(\dots)$
        $++i$;

Algorithm 3: Pseudo-algorithm for a one particle intra-cellular initialization used in MEDUSA.

The Kokkos implementation of the intracellular particle initialization step is efficient and allows the initialization of particles in a parallel fashion.

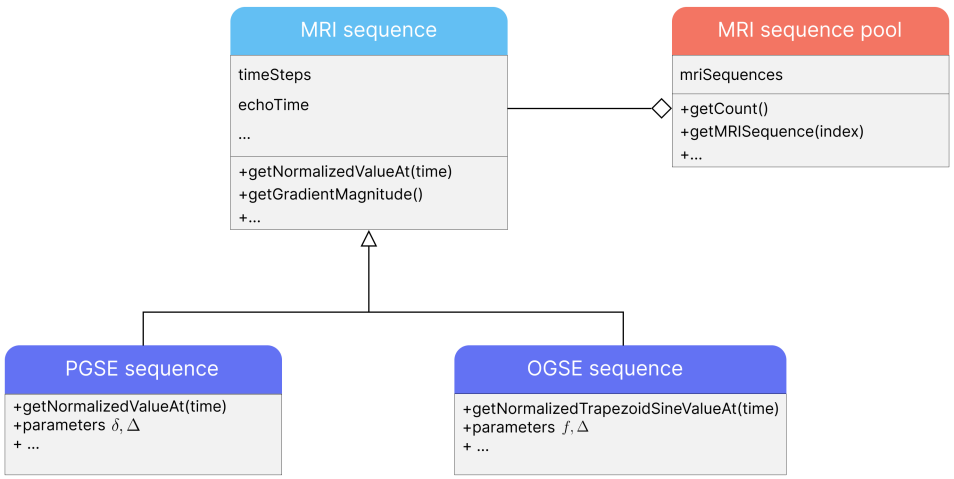

Figure 5.3: *Illustration of the data structures used in MEDUSA to perform MC dMRI simulations.*

MRI Sequences. To perform dMRI simulations on a substrate, it is important to specify the MRI sequences used. For this, specific classes are created in the MEDUSA framework to handle the initialization of MRI sequences from user-specified input files.
The UML design of the MRI sequences in the MEDUSA framework is presented in Fig.(5.3). In

addition to the presented design, the MRI sequences module in MEDUSA is equipped with a factory pattern that facilitates the creation and initialization of sequences.
The initialization of sequences is based on a JSON-like parameter file containing the specifications and the details of the MRI sequences used. Two JSON-like parameter files are required for the dMRI simulations in the MEDUSA framework. The first file contains the **diffusion process**-related parameters and the second one contains the list of **MRI sequences** for the MC dMRI simulations. An example of the two-parameter files used in the MEDUSA framework is illustrated in Fig.(5.4). The first file contains general information such as the total time duration of the simulation, the time step δt used in the MC dMRI simulation, and other parameters such as the number of spins, the initialization type, and finally, the session count parameter which will be further detailed in the next section. The second file contains sequence-related parameters, which in the case of the illustration is a PGSE sequence with parameters such as δ, Δ, the maximum slew rate, the echo time T_E, the gradient magnitudes, and finally, the gradient orientations.
dMRI signal synthesis. To obtain dMRI attenuation signals, individual phase contributions from each spin are added. We recall the formula previously mentioned (Chapter 3) for the phase shift resulting from a set of N spins:

$$\begin{cases} \Delta\phi_i = \gamma \mathbf{G}(\mathbf{r}(t_i), t_i).\mathbf{r}\delta t \\ \\ \frac{S}{S_0} = \frac{1}{N_s}\sum_{n=1}^{N_s} \exp\left(-i\sum_j^{N_t} \Delta\phi_j\right) \end{cases}$$

The dMRI signal calculation relies mainly on the phase terms ϕ_j, which in turn rely on the gradient, the gyromagnetic ratio γ, the time step of the simulation δt, and finally, the spin's position $\mathbf{r}$. To avoid redundant calculations, the following term $\gamma\mathbf{G}\delta t$ is pre-computed as all the following parameters γ, $\mathbf{G}$, and δt are known.

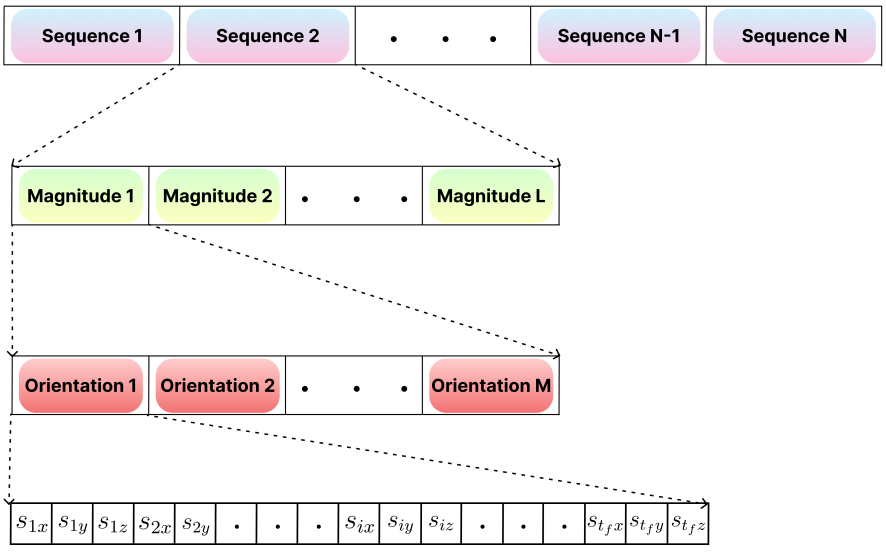

Figure 5.6: *Illustration of storage of the set of all MRI sequences used for a single dMRI simulation in the MEDUSA framework. A large vector view contains all sequences such that each sequence contains the set of all gradient magnitudes used such that each gradient magnitude contains the set of orientations, which in turn contain the final values of the sequence.*

a) Diffusion-process parameter file example

```
attributes =
{
    "particle_initialization": "all-particles-at-field-
of-view-center",
    "session_count": 1,
    "particle_diffusivity_in_um2_per_us": 2e-3,
    "particle_count": 1000000,
    "grid_resolution_in_um": [
        3.0,
        3.0,
        3.0
    ],
    "time_step_in_us": 100.143,
    "total_duration_in_us": 70200.0,
    "particle_backup_percentage": 2.0,
}
```

b) Diffusion sequences parameters file example

```
attributes = {
    "sequence-1": {
        "type": "pgse",
        "parameters": {
            "echo_time_in_ms": 70.2,
            "time_offset_to_diffusion_module_in_ms": 0,
            "little_delta_in_ms": 30.0,
            "big_delta_in_ms": 40.0,

"maximum_gradient_slew_rate_in_tesla_per_m_per_s": 200,
            "gradient_time_resolution_in_us": 100,
            "gradient_magnitudes": [
                (
                    1.0, 10.0, 30.0, 50.0,
                    1e2, 2e2, 3e2, 4e2,
                     5.0e2, 6.0e2, 7.0e2,
                     8.0e2, 9.0e2, 1.0e3
                )
            ],
            "gradient_orientations": [
                (
                  (1.0,0.0,0.0)
                )
            ],
            "noise_standard_deviation": 0.0
        }
    },
    “sequence-2”: {
     ...
   }
}
```

Figure 5.4: *Illustration of the JSON-like parameter files used in the MEDUSA framework. Panel a) is the first file containing parameters related to the diffusion process, such as the medium's diffusivity coefficient. Panel b) shows the sequence-related parameters file containing a dictionary of sequences. Each sequence's dictionary contains a parameters dictionary describing the details of the used sequence type (OGSE or PGSE).*

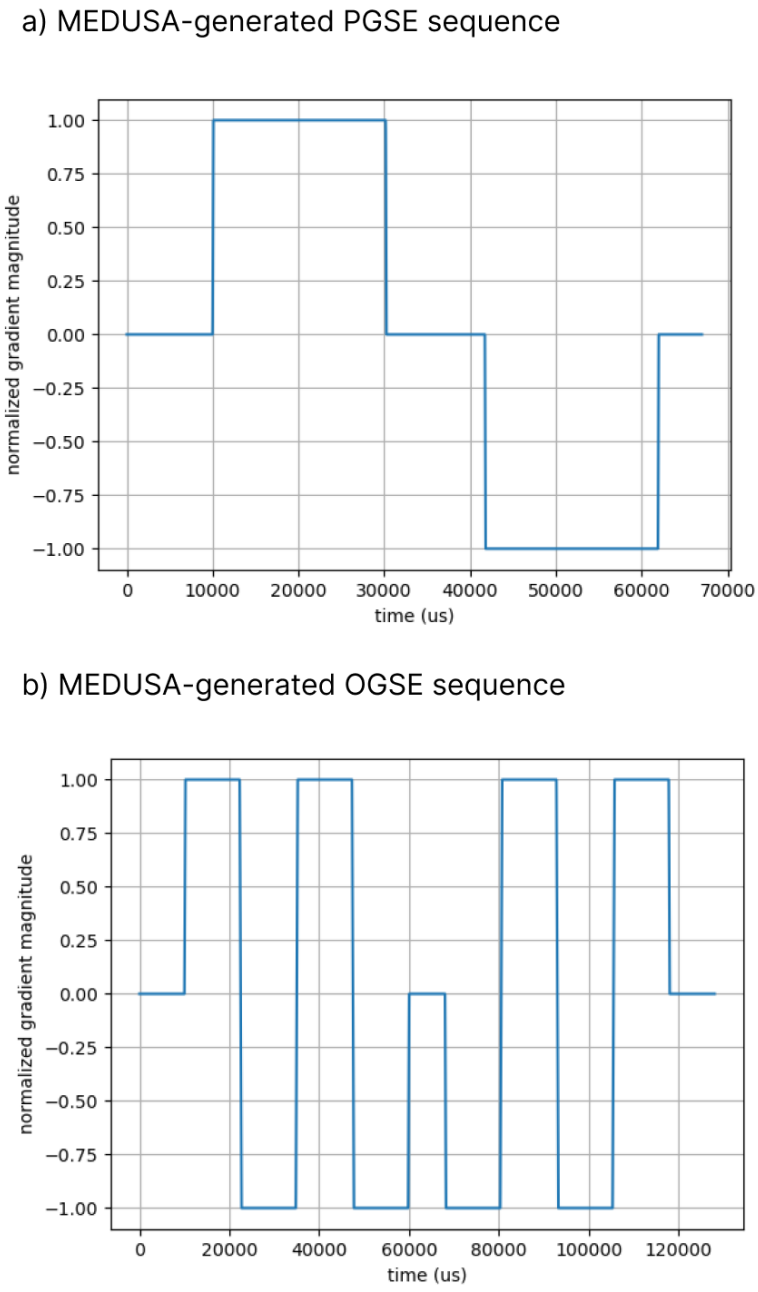

Figure 5.5: *Illustration of two different MRI sequences generated using the MEDUSA framework.*

The pre-computation results are stored in a vector view, i.e. a Kokkos-based data structure for vectors, to allow fast and parallel computations. The pre-computed term has the same structure as the set of all MRI sequences used for dMRI simulations provided in Fig.(5.6).

Another important parameter previously mentioned is the number of particle sessions; see Fig.(5.4). As the possible number of spins N_s used in a simulation is limited due to hardware-related memory constraints, the fragmentation of one dMRI simulation into multiple batches allows virtually the use of a very large number of spins, i.e. $N_s \gg 1$. The subdivision of the simulation into multiple batches is a consequence of the signal attenuation formula. The phase shifts sum $\sum_{n=1}^{N_s} \exp\left(-i \sum_j^{N_t} \Delta\phi_j\right)$ can be rearranged into $\sum_{s=1}^{n_S} \left(\sum_{k=1}^{N_p} \exp\left(-i \sum_j^{N_t} \Delta\phi_j\right)\right)$, with n_S being the number of sessions used and N_p the number of spins per session.

5.3 Performing dMRI simulation on ellipsoid-based substrates

This section presents contributions in the course of this thesis to enable the extraction of dMRI signals from ellipsoid-based substrates. At a high level, dMRI simulators require to have a set of methods in order to synthesize dMRI signals. These elements can be summarized in the following list:

1. Verify if a spin is inside a geometry
2. Detect collision with a geometry
3. Calculate the normal vector at the collision point

In addition, these elements need to be adapted to the ellipsoid-based representation case.
To simplify the dMRI simulations for ellipsoid-based geometries, a simple schematic for spins diffusing inside an ellipsoid-based axonal segment is illustrated below:

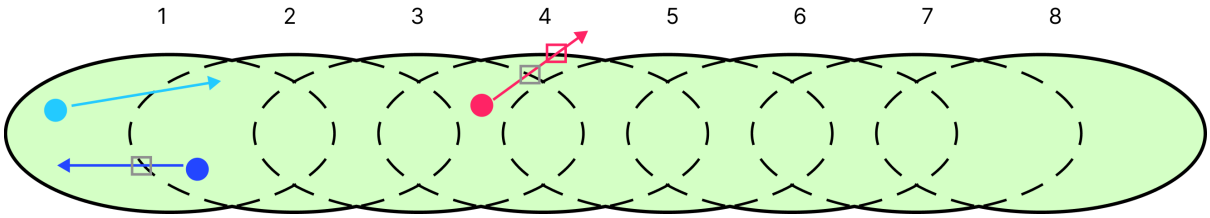

Figure 5.7: *Illustration of a simple virtual ellipsoid-based geometry with their identifiers (IDs) and three different particles representing different scenarios. a) The particle in light blue belonging to the ellipsoid of ID 1 represents the first scenario where the particle remains in the same ellipsoid. b) The particle in dark blue belonging to the ellipsoid of ID 2 represents the case of a fictitious collision. Finally, c) represents the last scenario for the particle in red belonging to the ellipsoid of ID 3 in which the particle collides with the geometry's membrane at the ellipsoid of ID 4.*

The illustration in Fig.(5.7) shows the frequent particle interaction scenarios encountered in dMRI simulation in ellipsoid-based geometries. As atoms in the MEDUSA framework, i.e. spheres and ellipsoids, are labeled with unique identifiers (IDs), these IDs are used to tag particles in the dMRI simulation. More precisely, each particle has an identifier equal to the atom's ID to which it belongs. Thus, before moving, the three particles in light blue, dark blue, and red in Fig.(5.7) have the following IDs: 1, 2, and 3, respectively. The first particle in light blue is the simplest scenario,

where the particle's position after the displacement, depicted by a light blue arrow, remains in the same atom of ID 1. The first simple scenario is also the most frequent one for small step lengths relative to the geometry in which the particle diffuses. The second particle in dark blue with an ID of 2 shows a collision with the second atom's membrane, and the particle ends in the first atom of ID 1. Although a collision is encountered in the case of the dark blue particle, in this case, it is a fictitious collision: a collision that does not happen with the outer membrane of the geometry, which is depicted by a solid line in Fig.(5.7). Basically, all collisions of the particle displacements with the black dashed lines in Fig.(5.7) represent cases of fictitious collisions. Finally, the last scenario represents the case when the particle collides with the geometry's outer membrane, as shown in Fig.(5.7) for the red particle.

5.3.1 Methods

This part presents the ellipsoid-related methods necessary to perform dMRI simulations.

Verifying whether a particle is inside an ellipsoid. As mentioned in the previous chapter, the ellipsoid can be represented by a set of ten parameters extracted from the expression of its quadratic surface provided in Eq.(4.1). The evaluation of the quadratic surface on an arbitrary point (x_p, y_p, z_p) is given by:

$$Ax_p^2 + By_p^2 + Cz_p^2 + Dx_py_p + Ey_pz_p + Fx_pz_p + Gx_p + Hy_p + Jz_p + K$$

The evaluation of the formula above informs about the status of a given particle with the position vector $\mathbf{r_p} = (x_p, y_p, z_p)$ relative to a given ellipsoid with parameters $(A, B, C, D, E, F, G, H, J, K)$. If the evaluated term is negative, the particle's position $\mathbf{r_p}$ is inside the ellipsoid, and it is outside when the term is positive. Finally, the last case is when the evaluated term is null, the position $\mathbf{r_p}$ lies exactly on the ellipsoid's surface.

The ellipsoid-segment intersection. Verifying the collisions between the particles and the membranes can be reduced to the ellipsoid and line segment collision test. To determine the collision, the parametric equation of the line segment and the ellipsoid are grouped together to obtain the following collision condition below:

$$\begin{cases} \mathbf{x_t} = \mathbf{x_0} + t\mathbf{s}, t \in [0, 1] \\ \\ \begin{pmatrix} \mathbf{x_t} & 1 \end{pmatrix} \mathcal{A} \begin{pmatrix} \mathbf{x_t} \\ 1 \end{pmatrix} = 0 \end{cases} \tag{5.1}$$

The expressions in Eq.(5.1) represent the parametric equation of a line segment parameterized by t. The segment starts from the initial position $\mathbf{x_0}$ following the direction of $\mathbf{s}$ which represents the step vector in the dMRI simulation with a norm equal to the step length previously defined as $\sqrt{6D_0\delta t}$. The collision condition is given by plugging the line segment parametric equation into the quadratic equation of the ellipsoid. This yields the equation for the collision condition between the line segment and the ellipsoid presented in its matrix form in Eq.(5.1). The matrix

$\mathcal{A}$ is the symmetric 4x4 matrix encoding the ellipsoid parameters $(A, B, C, D, E, F, G, H, J, K)$. Thus, the second equation admits a solution in the case of collision.
The expansion of the second expression in Eq.(5.1) yields the following second degree polynomial equation:

$$\begin{cases} a\,t^2 + b\,t + c = 0 \\ a = \mathbf{s}^T \mathcal{A}_{3\times3}\,\mathbf{s} \\ b = \left(2\mathbf{x_0}^T \mathcal{A}_{3\times3}\,\mathbf{s} + \mathbf{s}^T \begin{pmatrix} G \\ H \\ J \end{pmatrix} \right) \\ c = \begin{pmatrix} \mathbf{x_0} & 1 \end{pmatrix} \mathcal{A} \begin{pmatrix} \mathbf{x_0} \\ 1 \end{pmatrix} \end{cases}, \tag{5.2}$$

which admits at most two solutions. The term $\mathcal{A}_{3\times3}$ in Eq.(5.2) is the upper symmetric three by three submatrix of the ellipsoid representation matrix $\mathcal{A}$.
The number of solutions for the ellipsoid-segment intersection depends on the particle's position relative to the given ellipsoid. For instance, when the particle is located inside the ellipsoid, the first equation in Eq.(5.2) admits at most one solution. However, when the particle is located outside the ellipsoid, it is possible to encounter cases with two intersection points. To distinguish between different cases of intersections, further reasoning based on the numerical values of the obtained solutions is required.
The general solution to the ellipsoid-segment collision equation is the following:

$$\begin{cases} \Delta = \sqrt{b^2 - 4ac} \geq 0 \\ t_{1,2} = \frac{-b \pm \sqrt{\Delta}}{2a} \end{cases} \tag{5.3}$$

Note that these solutions are subject to the following constraint $t \in [0, 1]$. Thus, if $\Delta < 0$, there is no intersection between the segment and the ellipsoid. There is also no intersection when $\Delta > 0$ but $t_{1,2} > 1$, or $t_{1,2} < 0$, or finally, when $t_i < 0$ or $t_j > 1$ with $i \neq j$.
Most of the encountered cases of segment-ellipsoids are depicted in Fig.(5.8). Notice that even when $\Delta \geq 0$, it does not necessarily imply that there is a collision. In such cases, the actual 3D line intersects with the ellipsoid but not the segments. These cases are characterized by a solution set t_i that does not belong to the interval $[0, 1]$.

Calculating the normal at the surface. When using the elastic reflection to model the particle-membrane interaction in dMRI simulations, the particle's trajectory is reflected after encountering the membrane. The computation of the reflected trajectory requires the knowledge of the normal vector to the surface at the collision point. For this, the quadratic surface of the ellipsoid is used to extract the normal vector $\mathbf{n}$ to the surface at a given point $\mathbf{x_c}$ located on the ellipsoid's surface. The normal vector $\mathbf{n}$ is obtained by applying the nabla operator $\vec{\nabla}$ on the

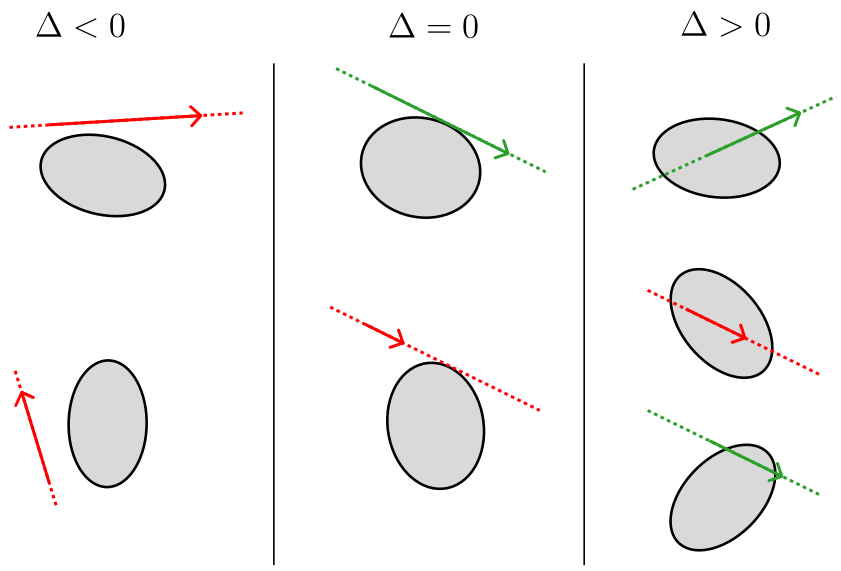

Figure 5.8: *Illustration of different segment-ellipsoid scenarios for different values of* Δ. *Solid arrows represent line segments, and dashed lines represent the line carrying them. The color of the segments represents whether there is an ellipsoid-segment collision. Green arrows exhibit collisions with the ellipsoid, while red ones represent non-collision cases.*

quadratic surface of the ellipsoid. This yields the following expression:

$$\mathbf{n} = \begin{pmatrix} 2Ax_c + Dy_c + Fz_c + G \\ 2By_c + Dx_c + Ez_c + H \\ 2Cz_c + Ey_c + Fx_c + J \end{pmatrix}, \tag{5.4}$$

which enables to compute the normal vector to the ellipsoid's surface at the collision point $\mathbf{x_c}$. It is also important that the collision point $\mathbf{x_c}$ belongs to the ellipsoid's surface, which can be checked by verifying the following equality $\begin{pmatrix} \mathbf{x_c} & 1 \end{pmatrix} \mathcal{A} \begin{pmatrix} \mathbf{x_c} \\ 1 \end{pmatrix} = 0$.

Atom-based dMRI simulations with elastic reflections. Now that the important elements required to perform dMRI simulation with elastic-membrane collisions are set, we will describe how these methods are used in the MEDUSA framework to enable the extraction of dMRI simulations. For this, two separate cases are distinguished: when the particle is in the intracellular space and when it is in the extracellular space.

- **Particle in the intracellular space.** When a particle is located inside a cell, it has an identifier ($\mathrm{ID}_{particle}$) that is equal to that of the atom ID_A to which it belongs as depicted in Fig.(5.7). First, the potential final position $\mathbf{x_f}$ of the particle is computed as follows: $\mathbf{x_f} = \mathbf{x_0} + \mathbf{s}$. Then, a test is performed to verify if the potential final position $\mathbf{x_f}$ belongs to the same atom with identifier ID_A. If the test is positive, i.e. the final position remains in the same original atom of start, then the step is accepted and another particle is processed. This case represents the most encountered scenario for intracellular particles and is also depicted in light blue in Fig.(5.7). When the particle's potential final position $\mathbf{x_f}$ is not inside the original atom of ID_A, an array of atom indices in the voxels corresponding to the positions $\mathbf{x_0}$

and $\mathbf{x_f}$ is browsed. More precisely, atoms that belong to the same cell and with an identifier different than ID_A are iterated over such that for each atom, a test is run to check whether the atom contains the potential final position $\mathbf{x_f}$. If one of the iterated atoms contains the position $\mathbf{x_f}$, then the particle is displaced and its identifier ($\text{ID}_{particle}$) is updated to take the same value as the ID of the atom containing it: this is the case depicted in dark blue in Fig.(5.7). In the other case, then a segment-ellipsoid test is run when the atoms iterated over do not contain the position $\mathbf{x_f}$. When a collision is detected, the collision coordinates and the distance to the collision point are stored in variables. These variables are updated as long as the atoms iterated over do not contain the position $\mathbf{x_f}$ and that it intersects with the displacement segment with a larger collision distance than previously stored. Finally, after iterating over all the atoms belonging to the same cell within the selected voxels, if no atom containing the position $\mathbf{x_f}$ is found, then a collision with the actual cell's membrane has occurred. This is exactly the case depicted in red in Fig.(5.7) which shows a fictitious collision characterized by a smaller collision distance. In this case, the collision coordinates are used to calculate the normal vector at the collision point using the formula in Eq.(5.4). The particle's position is updated from $\mathbf{x_0}$ to $\mathbf{x_c}$, and the step vector is also updated to $\mathbf{s_r}$ which is the reflected direction with a norm $||\mathbf{s_r}|| = \sqrt{6D_0\delta t} - ||\mathbf{x_c} - \mathbf{x_0}||$.

- **Particle in the extracellular space.** This case is slightly different from the intracellular one. While the particles in the intracellular space have identifiers equal to the identifiers of the atoms containing them, in the extracellular space, all the particles have the same identifier $\text{ID}_{particle} = -1$. To simulate particles in the extracellular space, the list of atoms in the voxels containing the positions $\mathbf{x_0}$ and $\mathbf{x_f}$ is iterated over. For each atom, a segment-ellipsoid collision test is performed such that when an intersection is detected, variables storing the collision point and the collision distance are updated if the new collision distance is smaller than the previously stored value. The variable containing the collision distance is initialized first at infinity and then updated whenever a collision with a smaller distance is detected. After iterating over all the atoms, if no collision is detected, this means that the particle has remained in the extracellular space. Thus, its final position is exactly $\mathbf{x_f}$. However, in the other case, when collisions are encountered, the collision point $\mathbf{x_c}$ with the smaller distance is the one kept to update the particle's position. In a similar fashion as the intracellular case, the particle's position is updated to $\mathbf{x_c}$ and a new step vector $\mathbf{s_r}$ is used such that its norm is $||\mathbf{s_r}|| = \sqrt{6D_0\delta t} - ||\mathbf{x_c} - \mathbf{x_0}||$.

Adding tissue properties. This part discusses tissue properties related features that were implemented in the MEDUSA framework.

- **Permeability.** Until now, the discussed case of dMRI simulations concerned cases where the particles don't penetrate the cellular membranes. However, in order to increase the realism of the simulation, it is important to account for cases in which the simulated cellular membranes have a non-null permeability κ, thus allowing transitions of spins from the intra to the extra-cellular spaces and vice versa. For this, the membrane permeability model in the following research paper [Lee *et al.*, 2021] was implemented in the MEDUSA framework and adapted to the case of atom-based substrates.

According to the same research [Lee *et al.*, 2021], the work argues that the membrane permeability set in the simulations κ_0 is different than the effective one κ. For small membrane transition probabilities $P << 1$, the input permeability κ_0 is the same as the genuine permeability κ, and the transition probability is given by:

$$P \simeq \frac{\kappa||\mathbf{s}||}{D_0} C_d, \tag{5.5}$$

with C_d being a dimension-dependent constant:

$$C_d = \begin{cases} 1, & d = 1 \\ \pi/4, & d = 2 \\ 2/3, & d = 3 \end{cases}. \tag{5.6}$$

The approximation in Eq.(5.5) is valid as long as the following condition is satisfied: $\kappa_0 << \sqrt{\frac{D_0}{2d\delta t}} \frac{1}{C_d}$.
The more general case of membrane permeability between two media with different diffusivity values $D_{1,2}$ and two step vectors $\mathbf{s_{1,2}}$. is provided below:

$$P_{1\to 2} = \frac{\frac{\kappa||\mathbf{s_1}||}{D_1} C_d}{1 + \frac{\kappa}{2}\left(\frac{||\mathbf{s_1}||}{D_1} + \frac{||\mathbf{s_2}||}{D_2}\right) C_d}. \tag{5.7}$$

In the MEDUSA framework, the expression in Eq.(5.7) is used with a value of $C_d = 2/3$ since the substrates are in a 3D space.

- **Intracellular diffusivity.** Although the bulk diffusivity parameter D_0 is already specified in the diffusion process JSON-like parameter file in the MEDUSA framework as depicted in Fig.(5.4), it is not clear whether the intracellular diffusivity is the same as the bulk one D_0. Thus, in order to achieve more flexibility in the dMRI simulations, a new feature was implemented in MEDUSA to enable the attribution of customized diffusivity values. For this, modifications were added to the MEDUSA framework to support reading JSON-like parameter files containing the cellular tissue properties.
  The tissue properties parameter file illustrated in Fig.(5.9) allows to set custom values for intra-cellular diffusivities and cellular permeabilities. Note that in order for this to work, it is important for the parameters in the substrate and tissue properties input files to be consistent. For this, tests were implemented to verify if the number of cellular populations in the substrate's geometry and tissue properties input files are equal and of the same type. In addition, as depicted in Fig.(5.9), some of the tissue properties can be set as distributions. This allows to randomly sample different values for different cells of the same type. In addition, the actual sampled tissue property values can be saved into a .CSV file which is convenient for further data processing.

5.3.2 Results

The methods previously discussed were implemented in C++ in the following open-source library for the MEDUSA dMRI simulator: `https://framagit.org/cpoupon/gkg/-/tree/master/`

Tissue properties parameter file example

```
attributes = {
    "populations": {
        "fiber-population-01":
          {

“diffusivity_in_um2_per_us_distribution”:
              {
                “type”: “uniform-distribution”,
                “parameters”:
                  {
                    “lower_boundary”: 2e-3,
                    “upper_boundary”: 3e3
                  }
              },
            “permeability_distribution”:
              {
                “type”: “constant-distribution”,
                “parameters”:
                  {
                    “constant_value”: 0.0
                  }
              }
          }
        “neuron-population-01”:
          {

“diffusivity_in_um2_per_us_distribution”:
              {
                “type”: “uniform-distribution”,
                “parameters”:
                  {
                    “lower_boundary”: 2e-3,
                    “upper_boundary”: 3e3
                  }
              },
            “permeability_distribution”:
              {
                “type”: “uniform-distribution”,
                “parameters”:
                  {
                    “lower_boundary”: 0.0,
                    “upper_boundary”: 5e-1
                  }
              }
          }
      }
    }
```

Figure 5.9: *Schematic of a sample of a JSON-like tissue properties parameter file in MEDUSA in the case of a substrate with one axon and one neuron populations.*

simulation/src/library.

The first part of the results shows the runtime efficiency for using atoms, i.e., spheres and ellipsoids, compared to the classical case of mesh-based dMRI simulations. This part consists simply of capturing the runtime for segment-triangle, segment-sphere, and segment-ellipsoid collision tests. These collision tests were implemented in Python such that for each elementary geometry, a number of randomized trials of $N_t = 1000$ were used, and the runtime was averaged over these trials.

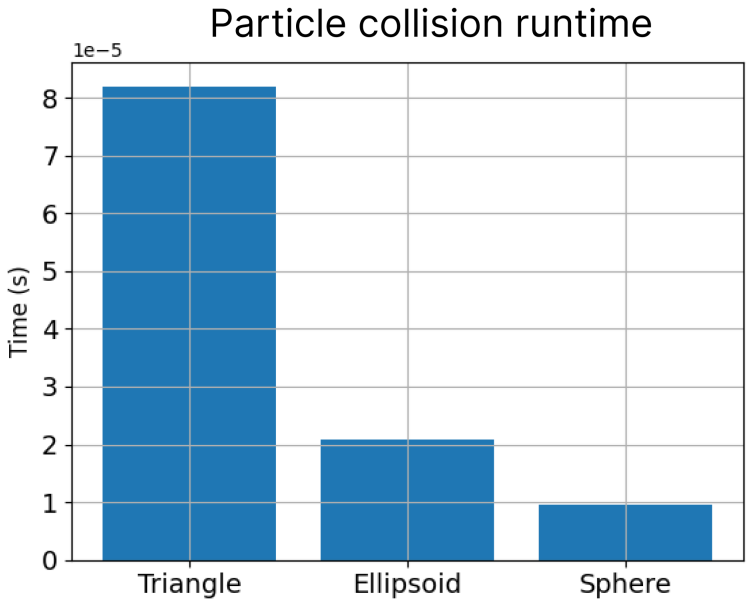

Figure 5.10: *Averaged runtimes for a particle colliding with simple geometries.*

The second part of the results consists in simulating the dMRI process in two simple geometries: a sphere and an ellipsoid. An elastic reflection is used for the particle-membrane interaction type in this case. The aim of this experiment is to simply verify if the particles remain inside the geometry after reflection.

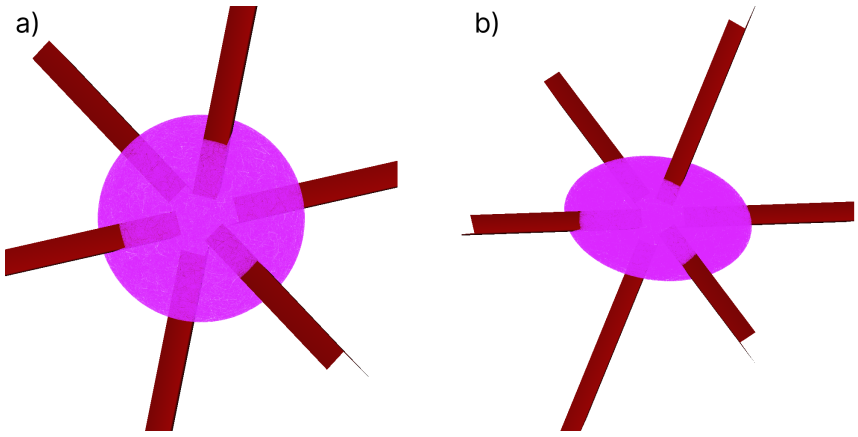

Figure 5.11: *An illustration of the spin trajectories resulting from simulation using elastic reflections in two simple geometries: a sphere for panel (a) and an ellipsoid for panel (b).*

The results in Fig.(5.11) are obtained using a sphere of radius $R = 3\mu m$ for panel a) with a number of particles $N_s = 10000$ and using an ellipsoid for panel b) with the following radii $[2, 3, 4]$ in µm

and a number of particle $N_s = 100000$.
The final experiment investigates the runtime performance of the MEDUSA dMRI simulator. It consists in performing dMRI simulations for different numbers of spins and tracking the runtime performance measured as the elapsed time duration in seconds.

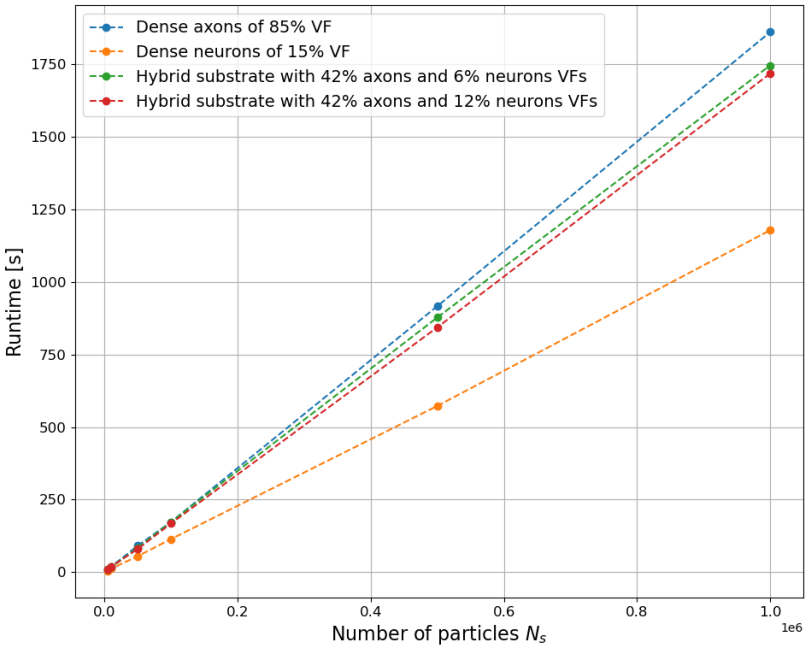

Figure 5.12: *A graph of the runtime performance of the MEDUSA dMRI simulator as a function of the number of spins N_s for four different substrate types.*

The substrates used to perform the runtime measurement are the ones generated in the previous chapter in Fig(4.40). For each substrate, six different simulations were performed, each having a different number of spins $N_s \in \{5 \times 10^3, 10^4, 5 \times 10^4, 10^5, 5 \times 10^5, 10^6\}$. These simulations were performed on a computer with the following characteristics: Intel® Core i9-10885H CPU @ 2.40 GHz x 16 using all cores for parallel computations.

5.3.3 Discussion

The first result on the collision runtimes in Fig.(5.10) shows that, indeed, atom-based geometries not only provide a memory-efficient representation of substrates but are also runtime-efficient for particle-membrane collisions. The particle-sphere collision detection is the most efficient, and then at the second place comes the particle-ellipsoid. The aim of this experiment is to compare the relative differences between collision runtimes for different geometries. Note also that as this experiment was performed in Python, thus, it does not reflect accurate absolute runtimes as the particle-membrane collision methods are implemented in C++ in the MEDUSA framework.
The second part of the results in Fig.(5.11) allows to visually inspect the spin trajectories remaining inside the substrate after elastic collisions with the substrate's surface. Note that the dMRI simulations are performed in the intracellular space.
Finally, the runtime performance results displayed in Fig.(5.12) show, as expected, an increasing runtime with the number of particles used in dMRI simulations. Note that the substrates used

for this benchmark are relatively complex as they represent more realistic white and gray matter substrates with a large VOI of (100x100x100) µm^3. The same results show that for the maximal number of spins used $N_{s_{max}} = 10^6$, the simulation runtime did not exceed 30 minutes. In addition, the runtime differences can be attributed to the complexity of the substrates, i.e. the number of spheres and their radii distribution. For instance, the dense axons substrate with a volume fraction of 85% has the highest runtime, while the dense neurons-only substrate has the lowest runtime. These runtime differences can partly be explained by the number of atoms, i.e. spheres in this case, which for the dense axons is around 2.7 M spheres while only 0.71 M for the case of the dense neurons-only substrate.

5.3.4 Conclusion

This section presented a method to simulate dMRI signals in ellipsoid-based geometries with elastic membrane reflections. The method was integrated into the MEDUSA framework for dMRI simulations, which is made open-source in the following repository: https://framagit.org/cpoupon/gkg/-/tree/master/simulation/src/library. Indeed, the use of atom-based substrates also has dMRI-related advantages, such as reduced collision times compared to mesh-based geometries. In addition, it allows to obtain simulated dMRI signals within reasonable durations, i.e. less than 30 minutes, given moderate computational resources: one station with a CPU with 16 cores.

Finally, the presented results focused on the advantages of atom-based compared to mesh-based geometries and on establishing runtime benchmarks. These results do not allow to make statements about the accuracy of the dMRI signals obtained. Therefore, knowing whether the obtained signals from the MEDUSA framework are biased is of equal importance. This will be addressed in the next section.

5.4 Validating the MEDUSA dMRI simulator

This section presents the contributions in the frame of this thesis to validate the MEDUSA dMRI simulator. This consists in ensuring that the signals obtained from the simulator are meaningful. For this, different experiments were conducted to verify the validity of the obtained results. This section presents first the methods used and then the experiments conducted and their results. Finally, the implications of these results and the use of atom-based geometries are discussed in detail.

5.4.1 Methods

To validate a dMRI simulator, a checklist with different tests is prepared. Each test within the checklist verifies a single aspect of the dMRI simulator. For a detailed review about validating dMRI simulators, we orient the reader toward the following work [Fieremans & Lee, 2018].
In this work, the following tests were used to validate the MEDUSA diffusion MRI simulator.

1. **Free diffusion.** This is the first and simplest test done. It consists in running dMRI simulations in free space and verifying the following two points: the measured diffusivity is constant along a given direction $\hat{\mathbf{n}}$ and equal to the value D_0 set in the input parameters. The empirical diffusivity can be obtained from the simulated particle trajectories according to the following formula:
$$D\left(t, \hat{\mathbf{n}}\right) = \frac{\left\langle \left(\Delta \mathbf{r}.\hat{\mathbf{n}}\right)^2 \right\rangle}{2t}. \tag{5.8}$$
The term $\Delta\mathbf{r}$ is the position difference between a given particle at time t and the initial time $t_0 = 0$, $\hat{\mathbf{n}}$ is the direction along which the empirical diffusivity is measured, and finally, the term $\langle \dots \rangle$ refers to the average over the particles used in the simulation. The measured value $D\left(t, \hat{\mathbf{n}}\right)$ obtained from the trajectories should be constant over time and equal to the diffusivity value D_0 set in the parameters.

2. **Simple cases of restricted diffusion.** In this part, dMRI simulations are performed on simple substrates such as parallel cylinders, a sphere, or a cylinder. The dMRI signal stemming from such simple substrates can be obtained analytically using the short pulse gradient (SPG) approximation (see Chapter 2 Eq.(2.33)). We also recall the analytic formula for the dMRI signal stemming from a cylindrical geometry:
$$\frac{S}{S_0} = 8\left(2\pi q R\right)^2 \sum_{k=1}^{\infty}\sum_{m=0}^{\infty} \alpha_{km}^2 [J_m'(2\pi q R)]^2 \frac{\exp\left(-\left(\frac{\alpha_{km}}{R}\right)^2 D\Delta\right)}{[\alpha_{km}^2 - (2\pi q R)^2]^2(\alpha_{km}^2 - m^2)}.$$
To validate the MEDUSA dMRI simulator, the analytical formula above was implemented as a function and compared to the actual simulation results. It is important to emphasize that the formula above is obtained using the SPG approximation; therefore, the used PGSE sequence in the simulations should satisfy the condition $\delta \ll \Delta$.

3. **Benchmarks using open-source dMRI simulators.** In addition to evaluating the dMRI simulator on analytically known cases, signals from a dMRI simulator can be compared to

existing solutions. In this part, a benchmark was performed using the open-source Disimpy dMRI simulator [Kerkelä *et al.*, 2020]. Since the Disimpy framework relies on mesh-based geometries, a virtual substrate was specifically designed in two versions: mesh and atom-based versions of the same substrate were created so that it is possible to obtain dMRI signals from both simulators. The substrate consists of a set of non-overlapped spheres randomly placed inside a cubical voxel of size 80 µm. The spheres in the substrate are randomly sampled following a Gamma distribution such that the average radius is 8 µm with a standard deviation of 1 µm.

In addition to the validation part, an experiment was conducted to investigate the effect of using atom-based geometric representations.

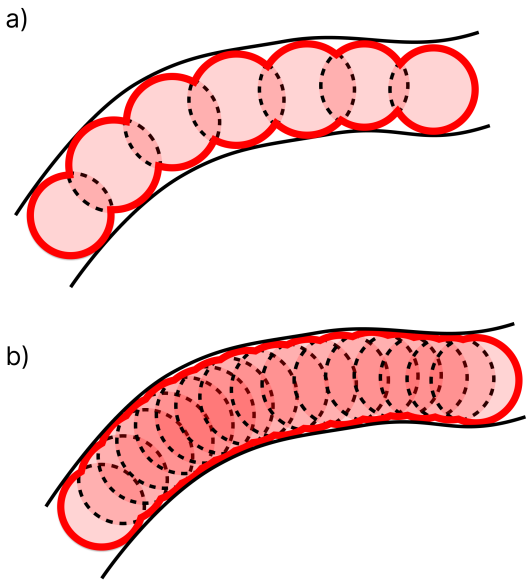

Figure 5.13: *Illustration of two different representations of an axonal fragment drawn with a bold black line. The first one in panel a) depicted in red represents a sparse atom-based representation, while panel b) depicts a denser and more accurate representation.*

The experiment aims to characterize the overlapping ratio for atom-based representations such that the obtained signal from the substrate is indistinguishable from an ideal mesh-based version of the same substrate. As depicted in Fig.(5.13), two versions of the same fiber fragment are drawn with solid black lines. The upper version shows a sparse approximation of the fiber while the bottom version is denser with a higher overlapping ratio, i.e. $O_{r_a} < O_{r_b}$. Both approximations are represented with thick red contours. While the first version is computationally more convenient as it requires fewer collision tests, the "bumpiness" resulting from the sphere-based approximation can lead to altered dMRI signals. Thus, to quantify the overlapping-ratio threshold over which the simulated dMRI do not change, a specific experiment was designed to answer this question for a given set of MRI sequences. The experiment consists in performing dMRI simulation on cylinders with different radii and different overlapping ratios. For each radius, a very high value of overlapping ratio was used to obtain an unbiased reference signal, which was used to compare with the dMRI signals stemming from other cylinders with the same radius but different overlapping ratio values.

5.4.2 Results

Free diffusion. The first part of the results displays the trajectories of different particles initialized at the origin of the space in the case of free diffusion. The corresponding results illustrated in Fig.(5.14) mark the first element in the validation checklist of dMRI simulators.

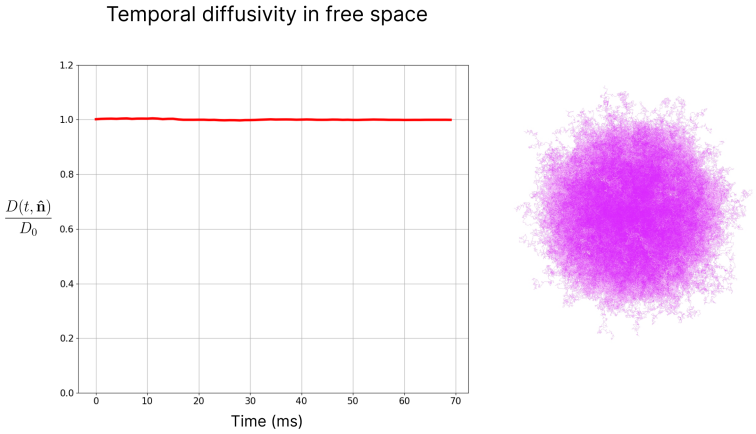

Figure 5.14: *Illustration of the measured temporal diffusivity over D_0 in free space from particle trajectories obtained from the MEDUSA dMRI simulator. The right panel shows a fraction of the particle trajectories used to generate the temporal diffusivity profile in the left panel.*

The second part of the results is also related to the case of free diffusion. In this part, a dMRI signal was synthesized using a PGSE sequence with the following parameters: echo time $t_E = 70.2$ ms, $\delta = 30$ ms, $\Delta = 40$ms, orientation following the x-axis, i.e. $(1, 0, 0)$, and different gradient magnitudes such that b-values are between 0 and around 2900 s/mm^2. A diffusivity coefficient $D_0 = 0.002\mu\text{m}^2/\mu\text{s}$, with a number of particles $N_s = 10^6$ were used. The results of this experiment are shown in Fig.(5.15) in red circular markers fitted using the following signal expression $S/S_0 = \exp(-bD_0)$. Note that a log scale was used and the obtained fit in blue matches the simulated signals.

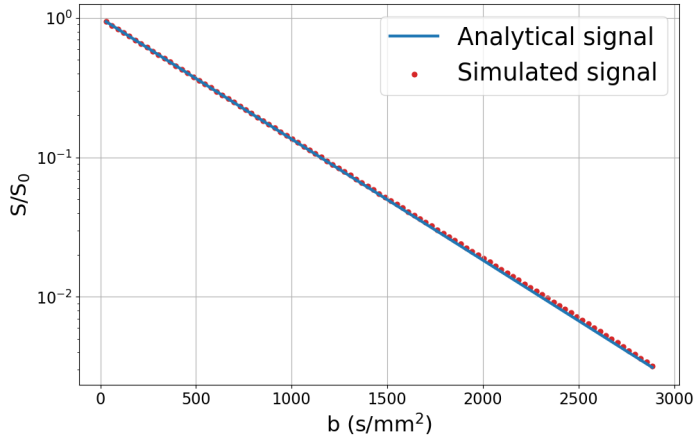

Figure 5.15: *A graphic of the free diffusion MRI signal obtained using different b-values and a fitted analytical formula of exponential decay of the signal. A log scale is used for the signal's axis.*

Validation using a cylinder. In addition to the free space validation, a long sphere-based cylindrical substrate of radius $r = 10$ µm was created to perform restricted diffusion validation of the MEDUSA dMRI simulator. The cylinder is approximated with a set of 7926 spheres of the same radius with a high overlapping ratio of $O_r = 99.8\%$ to best approximate an ideal cylinder. The used PGSE sequence to validate the dMRI signals stemming from the cylinder have the following parameters: a short pulse with $\delta = 1$ms with a large spacing of $\Delta = 498$ms, a high slew rate of $10^4\,\mathrm{T.m^{-1}.s^{-1}}$, an echo time $T_E = 500$ms. The PGSE sequence has an orientation following the x-axis which is perpendicular to the cylinder's main direction (z-axis). For the diffusion process, the following parameters were used: diffusion coefficient $D_0 = 0.002\mu\mathrm{m}^2/\mu\mathrm{s}$, a number of particles $N_s = 10^6$, and finally a simulation time step $\delta t = 50\mu$s. Two simulations were performed using two different particle-membrane interaction types: an elastic particle-membrane collision and the **Equal-step Random Leap** (ERL) which consists in rejecting steps in which a particle encounters a membrane and simply resamples the step vector $\mathbf{s}$.

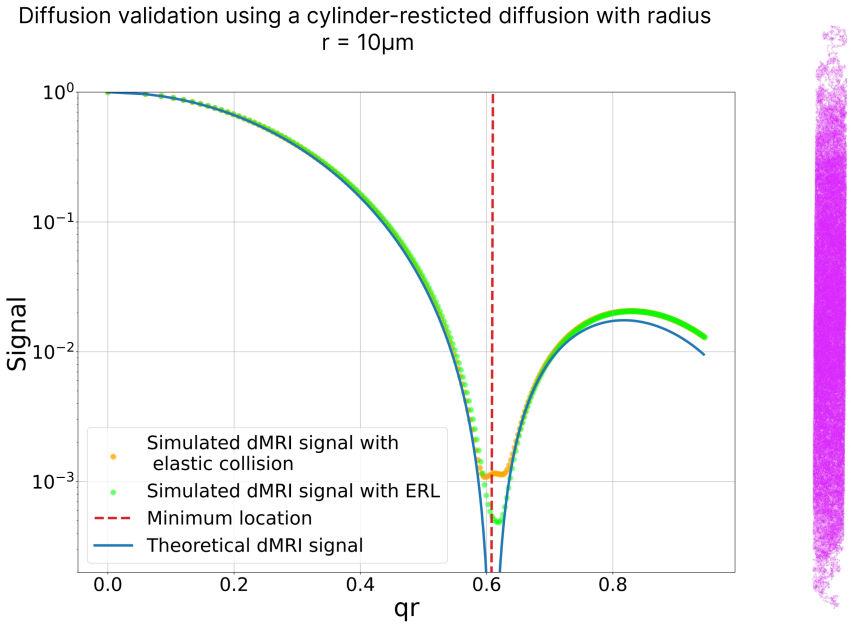

Figure 5.16: *Illustration of the MEDUSA dMRI simulation validation using diffusion within a cylinder.*

The dMRI signals for the cylinder-restricted dMRI simulations were compared with the analytical formula previously provided. Although the presented formula has infinite sums over two indices k and m, these sums were truncated at the following values: $k_{max} = 10$ and $m_{max} = 20$. In practice, we found this truncation provided a good approximation and all the terms for which $k > k_{max}$ and $m > m_{max}$ showed a null contribution to the signal. The results of these experiments are depicted in Fig.(5.16) in the left panel which shows agreement between simulated results and the analytical formula. In addition, the minimum location dashed line in red agrees with the minimum value mentioned in prior dMRI research [Callaghan, 1995]. The right panel of Fig.(5.16) shows a fraction of the particle trajectories obtained from the same experiment.

Signal comparison with Disimpy. The substrate used to extract the dMRI signals from both simulators is displayed in Fig.(5.17). A set of non-overlapping spheres providing a simplistic model of a soma population are randomly distributed with an average radius of 8µm. For this case, two simulations were performed, one using intracellular only and the other using extracellular signals. For the intracellular case, the following diffusion process parameters were used: $N_s = 30\,000$ spins, a diffusivity coefficient $D_0 = 0.003$ µm^2/µs, and a time step of $\delta t = 10$ µs. For the MRI sequence used, a PGSE sequence oriented towards the x-axis was used with the following parameters, $\delta = 30$ms, $\Delta = 40$ms, $T_E = 70.1$ms, a slew rate of 123 T.m^{-1}.s^{-1} and 41 different gradient magnitudes ranging from 0.4 to 70.25 mT.m^{-1}.

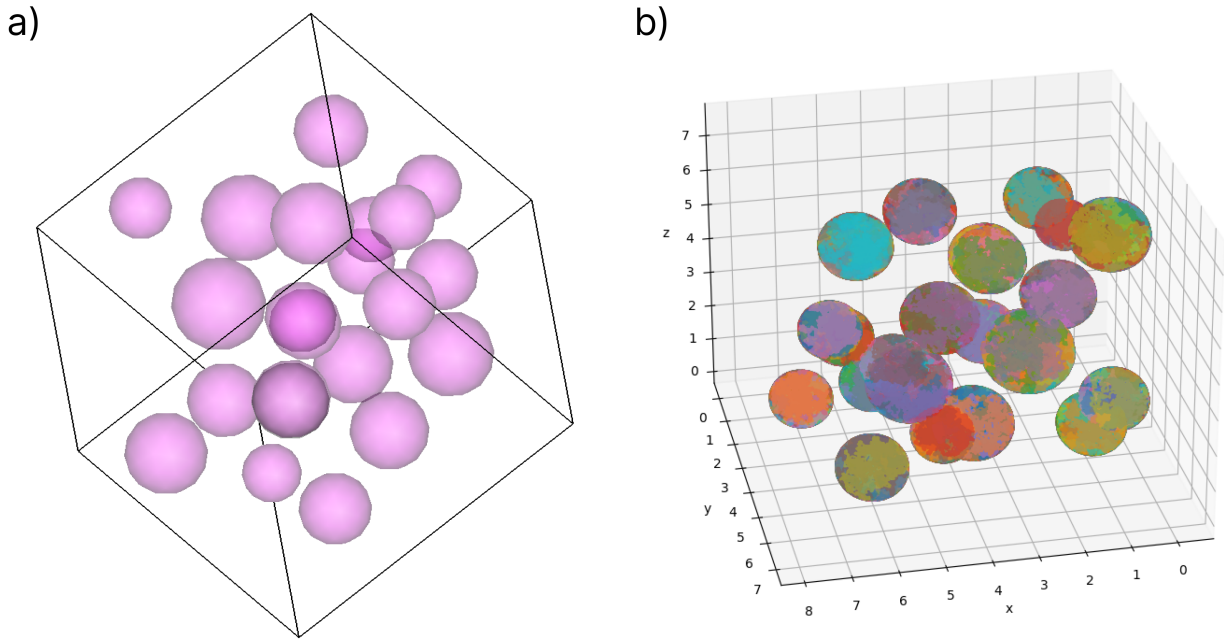

Figure 5.17: *Illustration of the virtual substrate used to compare the dMRI signals between the MEDUSA and Disimpy dMRI simulators. Panel a) shows the used substrate and panel b) shows trajectories generated using the Disimpy simulator.*

The same diffusion process and MRI sequence parameters of the intracellular case were used for the second simulation in the extracellular space. In addition, a reflective boundary condition was used in the Disimpy dMRI simulator: it consists of elastically reflecting particles colliding with the bounding box of the used substrate.
The results in Fig.(5.18) show the intracellular signals obtained from both the MEDUSA and Disimpy dMRI simulators for the substrates depicted in Fig.(5.17). The intracellular signal values show an agreement between the two simulators for different b-values.
The simulated extracellular signal values from both MEDUSA and the Disimpy simulators are depicted in Fig.(5.19). These results also show an agreement between the MEDUSA and Disimpy simulators for extracellular signals. It is also important to mention that in MEDUSA, the particles in the extracellular space do not intersect with the substrate's bounding box; instead, the step is simply rejected and another displacement vector is sampled.

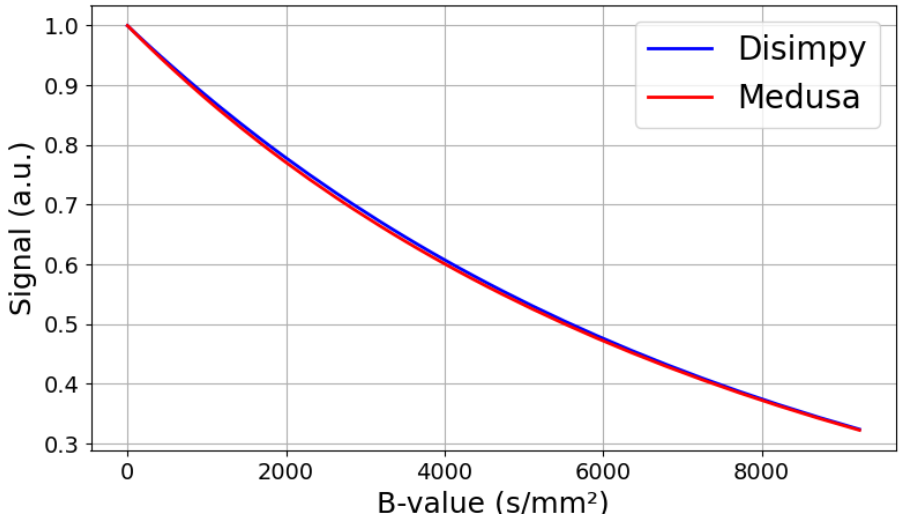

Figure 5.18: *A graph showing a comparison of the intracellular dMRI signals using the Disimpy and MEDUSA simulators and a substrate made of non-colliding spheres.*

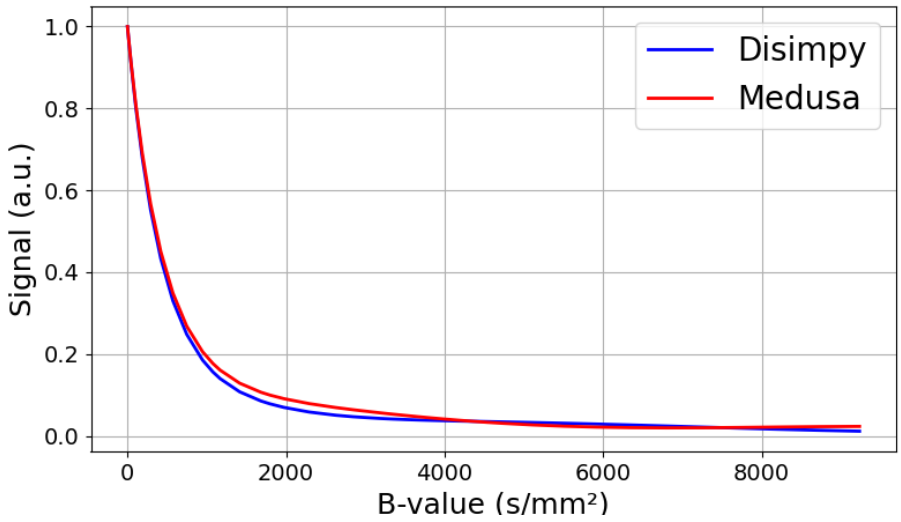

Figure 5.19: *A graph showing a comparison of the extracellular dMRI signals using the Disimpy and MEDUSA simulators and a substrate made of non-colliding spheres.*

The varying overlapping-ratio experiment. This part presents the results related to the characterization of the overlapping-ratio values which lead to reasonably unbiased dMRI signals. For this, different sphere-based cylindrical substrates were generated capturing three cases: the first one is a set of cylinders with a large radius of 10µm and with different overlapping ratio values $0.65 \leq O_r \leq 0.998$. The second case represents the scenario of medium-sized fibers with a radius of 5µm with varying overlapping ratios in the same range as the first case. Finally, the last case represents the scenario of small fibers with a radius of 2µm and different overlapping ratios also in the same range as the first case. For each case, the signal from the cylinder with the highest overlapping ratio was used a reference signal S^* to compute the relative signal error from cylinders with the same radius but lower overlapping ratios. The relative signal error from the reference and other signals was calculated using the following expression:

$$\epsilon_r(O_r) = \frac{|S(O_r) - S^*|}{S^*}. \tag{5.9}$$

For the signal synthesis, the following diffusion process parameters were used: a number of particles $N_s = 0.5 \times 10^6$, a simulation time step of $\delta t = 40$µs, and a diffusivity coefficient of $D_0 = 0.0025$ µm^2/µs. In addition, for the used dMRI sequences, a set of 31 PGSE sequences was adopted for this experiment. The sequences had an echo time $T_E = 37.1$ms and different values of δ and Δ leading to the following b-values: $[20, 50, 100, 150, 200, 300, 400, 500, 600, 700, 800, 900, 1000, 1500, 2000, 2500, 3000, 4000, 5000, 6000, 7000, 8000, 9000, 10000, 15000, 20000, 25000, 30000, 40000, 50000, 60000]$.

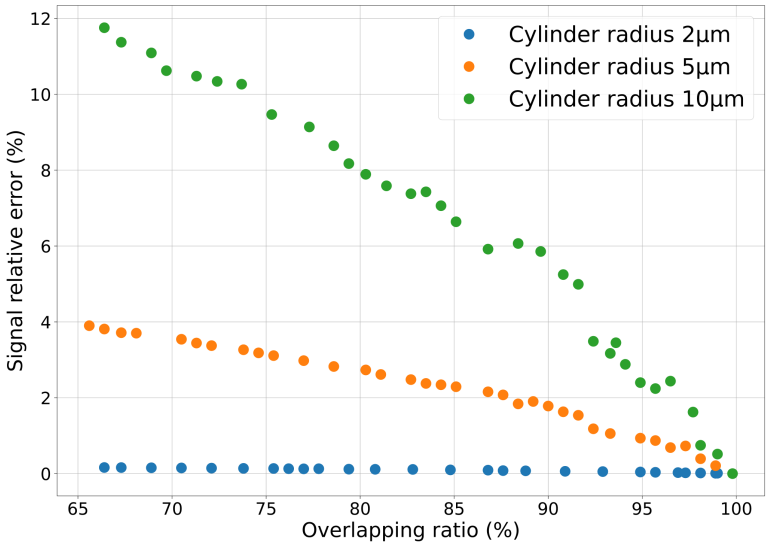

Figure 5.20: *The effect of the cylinder's "bumpiness" measured by the overlapping ratio on the dMRI signals for cylinders of three different radii: 2, 5, and 10 µm.*

The results of the varying overlapping ratio are displayed in Fig.(5.20). The same results show that the effect of the overlapping ratio O_r is more apparent for larger fibers than small fibers given the larger increase in the relative error ϵ_r for larger cylinders.

A proof-of-concept using the MEDUSA dMRI simulator. In this part, the MEDUSA dMRI simulator was used to extract signals from different MEDUSA-generated substrates. For this, dMRI simulations were performed on the substrates generated in the previous chapter, see Fig(4.40). For each substrate type, ten different samples were generated using the same microstructural parameters and then, dMRI simulations were performed on each sample. The obtained dMRI signals were orientation-averaged and then averaged over all the samples of the same substrate type. Standard deviations were also calculated from the orientation-averaged signals for each substrate type. For this experiment, the diffusion process parameters are the following: a number of particles $N_s = 0.5 \times 10^6$, a simulation time step $\delta t = 10\mu s$ and a diffusivity of $D_0 = 0.003\ \mu m^2/\mu s$ with particles initialized everywhere, i.e. including both intra and extracellular spaces. In addition, PGSE sequences were used with the following parameters: a slew-rate of 9600 $m\text{T.m}^{-1}$, an echo time of $T_E = 20\text{ms}$, the δ and Δ parameters are selected to achieve the following b-values: $[30, 60, 80, 120, 150, 180, 210, 240, 270, 300, 550, 700, 1000, 1700, 3200, 4000]$ with the following number of orientations $[6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 30, 30, 30, 30, 60, 60, 60]$ respectively. These parameters are similar to a preclinical 17.2 T Brucker MRI system.

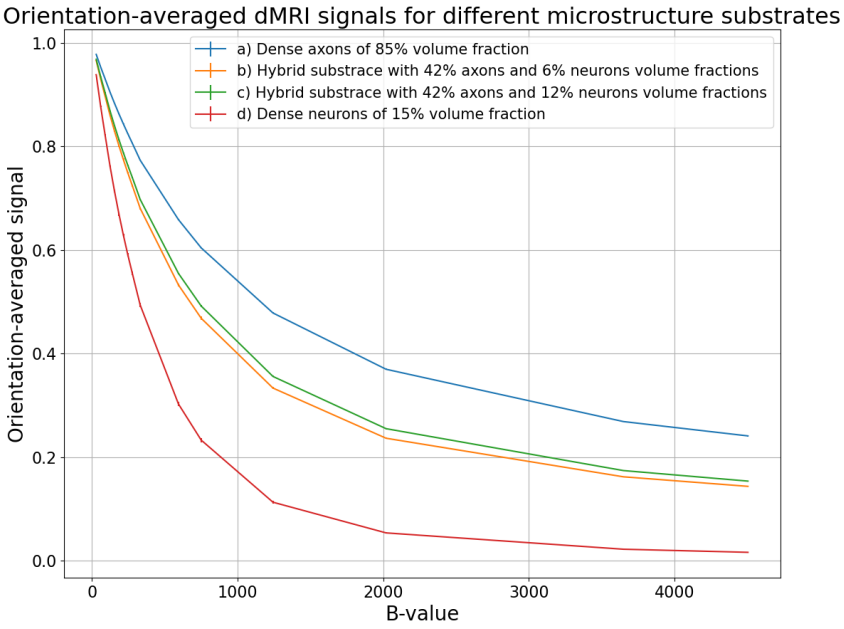

Figure 5.21: *The oriented-averaged dMRI signals obtained from the four substrate types with results averaged over ten substrate samples from each type.*

The orientation and type averaged signals were plotted as a function of the b-values and depicted in Fig.(5.21).

5.4.3 Discussion

In this section, the MEDUSA dMRI simulator was validated using different tests. The MEDUSA dMRI signals were obtained for both free and restricted diffusion cases. In addition, the signals were compared with the open-source Disimpy simulator in both intra and extracellu-

lar spaces. The presented results show a good agreement between the MEDUSA-simulated and ground truth signals, thus showcasing the ability to obtain valid dMRI signals from atom-based geometries that are comparable to those obtained from mesh-based alternatives.
The varying overlapping-ratio experiment also shows interesting mesh versus atom-based geometries results. The experiment's results in Fig.(5.20) show that the sensibility to the effect of the overlapping ratio O_r depends on the radius of the cylinder. This effect is most apparent for the largest radius of 10 µm for which the relative signal error ϵ_r increases with decreasing overlapping ratios O_r. The same effect is observed for the medium-sized cylinder of radius 5 µm, although more attenuated. Finally, for the smallest cylinder of 2 µm, the signal relative error is almost constant, and the effect of using different overlapping ratios can not be distinguished. This experiment shows that for the MRI sequences used, an overlapping ratio $O_{r_{min}} = 96\%$ is sufficient to represent substrates such that the effects on the signal can be neglected. In addition, for the case of small fibers, using even lower overlapping ratios is possible to represent them as it would not significantly impact the simulated dMRI signals. This finding suggests that it is possible to reach sparser microstructure geometries by tuning the overlapping ratio for each axonal segment generated.
Finally, the last result in Fig.(5.21) shows that it is possible to perform Monte Carlo dMRI simulations on relatively large substrates of dimensions (100×100×100) μm^3.

5.4.4 Conclusion

This section presented the validation of the MEDUSA dMRI simulator for free and restricted diffusion cases. Thus, the MEDUSA framework allows the Monte Carlo dMRI simulation to be performed on sphere and ellipsoid-based substrates and provides great flexibility regarding the simulation parameters. For instance, it is possible to set customized cell-dependent diffusivity and permeability values, choose specific particle-membrane interaction types, and also use a very high number of particles through the particle sessions mechanism.

5.5 Summary and outlook

As mentioned previously, the use of sphere and ellipsoid-based geometries allows a lightweight memory representation of the complex microstructure substrates. However, in the context of studying microstructure, this representation is only of use if it is possible to perform simulations to extract dMRI signals from such substrates. This chapter presented a simulation method that uses the elastic reflection for the particle-membrane interaction and that is adapted to the atom-based geometries. The method extended the MEDUSA framework and was tested and validated using ground truth cases. In addition, dMRI simulations in the MEDUSA framework can be performed efficiently which sets the ground to generate large synthetic datasets to study and decode microstructure.

Finally, although implemented in the Kokkos framework, the MEDUSA toolkit still only uses CPU-based parallel computing when it can additionally include GPU-based simulations as well. Therefore, future work perspectives in this direction include introducing code changes to the MEDUSA framework to support both CPU and GPU execution. This step does not require re-writing the whole dMRI simulator in CUDA, but instead, adapting the Kokkos-related datastructures to support both types of processing units. This potentially allows to perform fair runtime benchmarks with other dMRI simulators such as RMS [Lee *et al.*, 2021] and Disimpy [Kerkelä *et al.*, 2020].

Chapter 6

Large scale simulation campaign

The previous chapter presented the ability to generate hybrid substrates and to perform dMRI simulations that were validated against ground truth cases. This chapter investigates the scalability of the MEDUSA framework and its ability to generate microstructure substrates at a large scale. For this, a large simulation campaign was designed and its corresponding results were analyzed. In addition, a further investigation of the effect of the soma density on the dMRI signals was performed. This chapter starts with a general introduction, a brief presentation of the TGCC HPC facility used, the methods, the experiments and finally, their corresponding results are presented and discussed.

6.1 Introduction

The dMRI modality offers an interesting approach to probe the brain non-invasively. This has increasingly gained scientists' interest in studying brain-related diseases. Moreover, different studies have addressed the challenge of decoding the brain microstructure, i.e. estimating cell-related features from dMRI signals. Although it can be simply formulated, decoding the microstructure is one of the most challenging problems in the field of dMRI. Its difficulty is tightly linked to the data collection process, which is not only cost-prohibitive but also very complex. This process is twofold: first, acquiring dMRI signals from small brain samples, and then extracting the microstructural parameters from these samples. The first part requires designing a dMRI signal acquisition campaign: selecting an MRI system and adjusting the acquisition protocol with appropriate MRI sequences and other acquisition-related parameters. The second part requires extracting micrometer-scale features from samples whose size is in the millimeter scale. These steps are challenging to implement, especially the microstructural feature extraction step which is not only required to be efficient but also scalable. In addition, imaging modalities such as electron microscopy provide detailed insights into brain microstructure but also require high-capacity memory storage systems as demonstrated in [Shapson-Coe *et al.*, 2024].
Prior methods in decoding brain microstructure can be broadly subdivided into two classes, analytical and computational models. Although analytically tractable, analytical models are limited as they rely on oversimplistic models of brain microstructure, i.e. spherical soma and straight cylindrical axon shapes. However, computational methods are not bound to such simplistic geometries and offer great flexibility in terms of the possible substrates that can be simulated. In addition, as the number of parameters to estimate increases, new challenges related to fitting

the model's parameters emerge. Also, in contrast to analytical models, computational models are not bounded by specific regimes, i.e. a set of hypotheses such that only when satisfied then an explicit analytical formula can be derived. Thus, the computational approach offers more flexibility beyond the regimes found in analytical models. Despite this flexibility, the use of computational methods is still limited, given the mesh-based approach discussed previously, which prevents performing simulations at a large scale. This chapter challenges the scalability of the proposed methods previously presented and provides a large-scale test of the MEDUSA framework for the generation of different substrate types as well as the Monte Carlo (MC) simulation of the dMRI signals.

6.2 The TGCC HPC facility

The TGCC is an acronym for: "Très Grand Centre de calcul du CEA". It is a large High Performance Computing (HPC) facility designed to accomodate advanced computing systems in France. Currently, the TGCC hosts the 21 PFlops supercomputer named Joliot-Curie after the French physicist. The TGCC supercomputer is part of GENCI (Grand Equipement National De Calcul Intensif), which represents the French contribution to the Partnership for Advanced Computing in Europe (PRACE) infrastructure. To enable fast I/O, the computing nodes are connected to the private Lustre file storage system in addition to Global Lustre filesystems which are shared between different supercomputers. The computing resources are accessible to French researchers through the DARI portal http://www.edari.fr. They are also accessible for multinational research groups and industrial partners through the PRACE platform http://www.prace-ri.eu/how-to-apply.

Environment management. The TGCC can be accessed through a Secure SHel (SSH) protocol. Once connected, the user can set up the development environment by using the Modules tool which allows to load different software packages required for the computations that will be executed. The uploaded modules can be saved so that the environment is set, thus avoiding re-installing the required modules upon every connection.

```
# Load modules
#####################
$ Module load mpi
$ module load python3
$ module load cuda


# Check loaded modules
#####################
$ module list

# Saving environment
# with the loaded modules
#####################
$ module save
```

Figure 6.1: *Illustration of some of the Module-related terminal commands to load, list, and save modules.*

Some examples of the Modules-related commands are depicted in Fig.(6.1) in which the MPI,

Python3, and CUDA modules are loaded.

Supercomputer architecture. The supercomputer is organized into partitions. Each partition is a collection of nodes with similar hardware characteristics, i.e. amount of RAM, CPU architecture, and the existence of GPU. The Irene supercomputer has three kinds of nodes: the regular KNL compute nodes, large memory nodes, and finally, GPU nodes. The partition selection depends on the hardware needs of the computation job, i.e. the code submitted by the user for execution. For instance, a GPU-based partition is the most suitable for AI-related applications. Some of the available partitions are listed below:

- **Skylake.** Nodes used for regular computations. Equipped with 2x24-cores Intel Skylake@2.7GHz (AVX512) CPUs and 45 cores per node. This partition has a set of 1653 nodes totaling 79344 cores with 180 GB of RAM per node and 3075 GB of RAM per core.
- **Rome.** This partition is equipped with 2286 nodes with 2x64 AMD Rome@2.6Ghz (AVX2) CPUs. It has 128 cores per node and a RAM of 1.8 GB per core.
- **V100.** This partition is equipped with 4 Nvidia Tesla V100 GPUs and 2x20-cores Intel Cascadelake@2.1GHz (AVX512) CPUs. It has a total of 32 nodes with 40 cores per node and 4.4 GB of RAM per core.

The partitions list provided above does not form an exhaustive list of all the available partitions in the TGCC. More information can be found in the TGCC documentation in https://www-hpc.cea.fr/tgcc-public/en/latex/tgcc-public.pdf.

Data spaces. The TGCC is equipped with different file systems reachable from the supercomputer's login and compute nodes. The file systems are the following: Home, Scratch, Work, and Store. Each file system has different characteristics and its own purpose. For instance, the Home file system has a low data transfer rate and a quota of 5 GB per user. One of its purposes is to host small source codes and job submission scripts. In contrast, the Scratch file system is very fast, with a data transfer rate of 300 GB per second. It is also big and has a quota of 100 TB of disk space. The Work file system has a fast data transfer rate of 70 GB per second and has a medium quota of 5TB of disk space with a maximal number of files of 500 000. Finally, the Store file system provides a huge storage file system with a high data transfer rate of 200 GB per second. Its purpose is to store large files with a quota of 50 000 files per user group with an expected file size ranging from 10 GB to 1 TB. The disk quotas are available to the user through the following terminal command:

```
$ ccc_quota
```

which shows information about the file systems.

Job submission. Two elements need to be specified when submitting a job: the code execution instructions and the required resources for the submitted job. The first one is related to the code to be executed, i.e. a simulation code or model training algorithms. The second one specifies the compute resources to be allocated for the executed code. As the computing nodes

are shared between different users, different jobs can be submitted by multiple users. Thus, the jobs are handled by a scheduler named SLURM, which stacks the jobs in a queue. The scheduler manages the submitted jobs and decides their execution order according to a scheduling policy. The execution order is established using a priority metric that relies on three components: starting with the Quality of Service (QoS), which is a user-specified parameter that defines the user's priority of the submitted job. The second component is the project's fair share, this reflects the consumption state of the computational resources of the project. This value will be high if the project is under-consumption state and low in the opposite case. Finally, the last element used to determine the job's priority is the job's age: the older the job, the higher its priority. The age-related job priority is capped after a delay of seven days.
To submit a job to a supercomputer, the following command is used:

```
$ ccc_msub the_job_script
```

The *ccc_msub* terminal command allows the submission of a job to the scheduler which then decides on its priority. The job file named *the_job_script* specifies the user-requested computational resources and the program to be executed. As depicted in Fig.(6.2), the job file contains different information, such as the job's name and the number of tasks: one for sequential execution in the left panel a) and the other for parallel execution in panel b). In addition, an elapsed job time T has to be set such that once the job is executed, it will only run for T seconds and be interrupted after the elapsed time, regardless of whether the code execution has finished or not.

a) Sequential Job

```
#!/bin/bash
#MSUB -r SequentialJob   <-- Job name
#MSUB -n 1               <-- Number of tasks
#MSUB -T 600             <-- Ellapsed job time in seconds
#MSUB -o example_%I.o    <-- Standard output
#MSUB -e example_%I.e    <-- Error output
#MSUB -q <partition>     <-- Partition name
#MSUB -A <project>       <-- Project ID

set -x

cd ${BRIDGE_MSUB_PWD} # Contains the
directory from which the script is
submitted

ccc_mprun ./exec.out
```

b) Parallel Job

```
#!/bin/bash
#MSUB -r ParallelJob
#MSUB -n 32
#MSUB -T 600
#MSUB -o example_%I.o
#MSUB -e example_%I.e
#MSUB -q <partition>
#MSUB -A <project>

set -x

cd ${BRIDGE_MSUB_PWD} # Contains the
directory from which the script is
submitted

ccc_mprun ./exec.out
```

Figure 6.2: *Illustration of two job submission scripts. Panel a) represents a sequential job with one single task and panel b) represents a parallel job with 32 tasks.*

Also, two types of outputs need to be specified. The first is the standard output resulting from the execution of the code, and the second contains supercomputer-related errors. The partition name has to be specified using the flag *-q* as illustrated in Fig.(6.2). Finally, the last supercomputer-related parameter is the project ID which is set using the flag *-A*.
The executable file named *exec.out* is run using the *ccc_mprun* command to execute the code given the listed computational resources requested.

Virtualization. This is an important step in distributing and running programs. The combined use of virtual machines and containerization tools allows to simply and quickly deploy and execute programs, thus reducing the overhead of setting the proper environment for the code and installing packages with OS compatibility constraints. Tools like **Docker** are widely used to create container images that are usually distributed in the OCI standard format. The TGCC is equipped with the **PCOCC** tool, which is an acronym for Private Cloud On a Compute Cluster. It allows users of an HPC cluster to host their own clusters of VMs on compute nodes alongside regular jobs. The **PCOCC** tool can be used to import Docker images and to deploy them efficiently on the compute cluster. A compressed Docker image in ".tar" format can be imported using the following command:

```
$ pcocc-rs image import docker-archive:my_docker_image.tar my_docker_image
```

To verify if a Docker image was successfully imported, the following command can be used:

```
$ pcocc-rs image list
```

This command allows to list all the imported images which can be run using a job script similar to the one in Fig.(6.2) with the following modified run command:

```
$ ccc_mprun -n 2 -c 2 -C my_pcocc_image -- <cmd>
```

6.3 Methods

This section presents the methods used to perform large-scale simulations in the TGCC using the MEDUSA framework.

Dockerization. The first part consisted in Dockerizing the MEDUSA framework by establishing a Docker file listing all the needed dependencies to run MEDUSA. The Docker file relies on an Ubuntu 22.04 base image and other Debian packages to establish a working container image. The generated image was tested using the following command to execute the MEDUSA framework:

```
$ sudo docker run -it -d -v $HOME/data_input:/home/ginkgo/data_input:ro -v
  $HOME/data_output:/home/ginkgo/data_output medusa_img
```

Note that the command uses a flag -v to mount volumes for input and output files. The inputs include parameter files for the geometry or diffusion simulations and an output folder for saving the generated results.

The simulation pipeline. It is a Python-based pipeline that wraps the MEDUSA framework to launch the simulations developed by A. Brullé in his work [Alexis *et al.*, 2022]. This pipeline allows to perform end-to-end simulations. First, it generates MEDUSA-compatible input parameter files and performs substrate generation, dMRI simulations or both. The generation of the parameter files is done using a higher-level parameter file named the Meta-JSON, which contains probabilistic distributions used to sample values for the different parameters of the generated MEDUSA-compatible parameter files.

The simulation pipeline inputs the Meta-JSON, which generates different MEDUSA-compatible parameter files used to run simulations. The generated parameter files can contain randomly distributed values sampled according to a given probability distribution defined in the Meta-JSON parameter file. A simplified schematic of the simulation pipeline is depicted in Fig.(6.3).

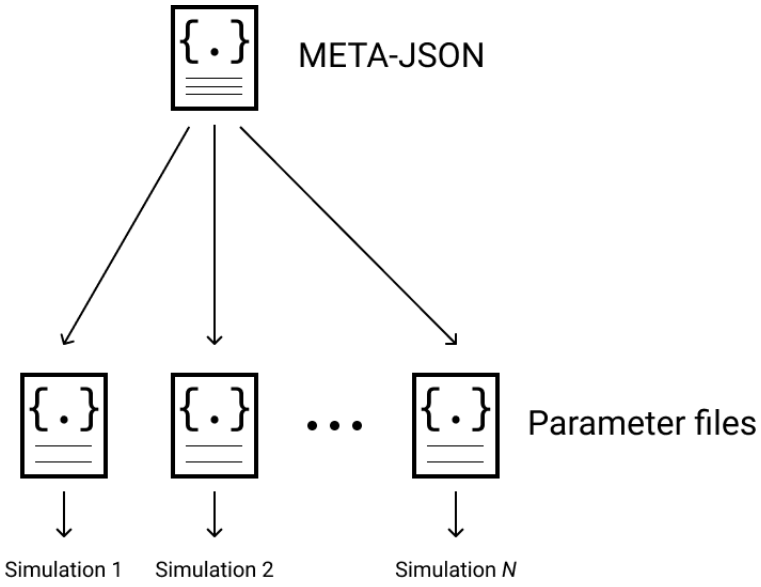

Figure 6.3: *Illustration of the simulation pipeline inputting the higher-level parameter file: the Meta-JSON.*

The Meta-JSON file also allows to set relational constraints between different parameters. For instance, it is possible to link the VOI size of the substrate to the average diameter of cells such that virtual substrates are generated in smaller VOIs for low cell diameters. Note also that the Meta-JSON parameters file contains both substrate and MRI-sequence-related parameters. This allows to perform substrate synthesis and dMRI simulations in one go. The general structure of the Meta-JSON parameters file is the following:

Meta-JSON parameter file

```
{
    "the_simulation_campaign_name": {
        “target_sample_count”: 50, # The number of simulation samples
        “geometry”: {
            ... # All the geometry parameters
        },
        “diffusion_process”: {
            ... # Parameters such as the diffusivity, number of particles, etc...
        },
        “mri-sequences”: {
            ... # MRI sequences used for MC simulations
        },
    }
}
```

Figure 6.4: *Illustration of the structure of the Meta-JSON file used to launch the large-scale simulations.*

The simple skeleton of the Meta-JSON depicted in Fig.(6.4) shows its structure which is composed of the name of the simulation campaign, the number of simulations represented by the key "target_sample_count", the virtual substrate's parameters, the diffusion process, and finally, the MRI sequences used for MC dMRI simulations. The Meta-JSON parameter file allows to set

parameters according to probability distributions and also to specify conditional rules for a parameter given other parameter values. The use of conditional rules in the Meta-JSON parameters can be activated using the key "linked" as depicted in panel a) of Fig.(6.5). This informs the JSON parser that the selected parameters is not free and the details of the dependency are specified in the dictionary corresponding to the value of the "linked" key. This value contains the type of the dependency and the set of parameters it requires. Each type of dependency corresponds to a given function inputting parameters that should match the dictionary values of the "parameters" key in the Meta-JSON file.

Examples of related parameters and probability distribution

a) Relation between the diameter and the VOI size

```
{
... # Geometry parameters
"field_of_view":
       {
         "linked":
         {
           "type" : "field_of_view_ratio",
           "parameters":
           {
             "coefficient": 10,
             "linked_paths":
             [
               [
              "the_simulation_campaign_name",
                 "geometry",
                 "populations",
                 "fiber_population_01",
                 "parameters",
                 "axon",
                 "diameter_distribution",
                 "parameters",
                 "mean"
               ],
               [
                 "the_simulation_campaign_name",
                 "geometry",
                 "populations",
                 "fiber_population_02",
                 "parameters",
                 "axon",
                 "diameter_distribution",
                 "parameters",
                 "mean"
               ]
             ]
           }
         }
       }
}
```

b) Use of the uniform distribution to sample values

```
{
... # Geometry parameters
“populations”:
     {
       “fiber_population_01”:
           {
              ...
              "has_tortuosity": 1,
              "tortuosity": {
"magnitude": 0.02,
"angular_dispersion_in_degrees": {
                    "random": {
                      "lower_value": 2.5,
                      "upper_value": 2.5
                    }
                  },
                  "wave_length": 25.0
                }
           },
       “diameter_distribution”: {
                "type": "gamma-distribution",
                "parameters": {
                  "mean": {
                    "random": {
                      "lower_value": 0.5,
                      "upper_value": 3.0
                    }
                  },
                  "stddev": 0.5

           }
     }
}
```

Figure 6.5: *Illustration of the structure of the Meta-JSON file used to launch the large-scale simulations.*

The same illustration in panel a) of Fig.(6.5) shows a dependency example where "field_of_view" parameter is related to the axonal diameters. In this case, the substrate has two axon populations and the field of view -the VOI size- is set as the average axonal diameter multiplied by a constant whose value is precised in the "coefficient" key. Note also that the "linked_paths" key contains the paths to the desired numerical values. Also, different relational functions are already implemented in the pipeline, thus covering frequent cases where such relation rules between different parameters are required.

A brief list of some of the existing relational rules is provided below:

- **Field of view ratio.** Already presented in Fig.(6.5), it allows to set the size of the field of view given the average axonal diameters in the substrate.
- **Time step function.** This allows to tune the value of the simulation time step δt given the bulk diffusivity value D_0 and the minimal axonal diameter present in the substrate.
- **Geometric grid scaling.** This tunes the grid size of the lookup table (LUT) used to localize atoms of the virtual substrate.

Finally, this mechanism is suitable as it allows to set custom relationships between different parameters while at the same time keeping a JSON file format.

Data extraction pipeline. A Python-based data extraction pipeline was developed in the course of this thesis. This pipeline extracts and aggregates the results of the large-scale simulation campaign. For each simulation results folder, the pipeline extracts the parameters related to the substrate's geometry and the dMRI simulations, i.e. dMRI signals and b-values. The extracted values for each simulation are aggregated and written to a ".CSV" file.

6.4 Experiments

This section describes the details of the large scale simulations performed at the TGCC using the MEDUSA framework. The simulations were twofold: microstructure substrate generation and MC dMRI simulations.

Large campaign of substrate generation. The first part of microstructure substrate generation comprised the following four classes of microstructure types:

- **Gray matter only substrates (GM).** These substrates are made of a single population of pyramidal neurons with different average soma diameters and volume fractions. The substrates are generated inside a cubic VOI of size 120 μm. The volume fraction VF is sampled randomly between 7% and 25%. The soma diameter follows a Gamma distribution with an average value sampled randomly between 14 and 20 μm and a standard deviation of 1.0 μm.
- **Mixture of pyramidal neurons and axons (WM_GM).** This type of microstructure models cortical layers containing mixtures of soma and axons. Substrates within this class are hybrid and contain a synthetic axon population and a pyramidal neuron population stemming from the NEUROMORPHO dataset. The axons are characterized by diameters following a Gamma distribution with an average sampled at random uniformly in the interval between 0.5 et 3 μm and a standard deviation also sampled randomly uniformly between 0.5 and 1.5 μm. The axon population has a global direction towards the z-axis with an angular dispersion sampled at random uniformly in the interval 2° and 15°. The axonal VF is sampled randomly between 5% and 42%. The axonal population also has a tortuosity characterized by an angular dispersion randomly sampled between 0.5° and 2.5°, with varying magnitudes randomly sampled between 0.01 and 1 and a wavelength between 8 and 12

μm. The pyramidal neuron population is characterized by a diameter following a Gamma distribution with a mean value randomly selected between 14 and 22 μm and a VF ranging between 6% and 20%.

- **White matter only with one axon population (WM).** In this case, a single axon population is generated with diameters following a Gamma distribution with mean values ranging between 0.5 and 3 μm. and a VF between 40% and 85%. The size of the VOI is selected dynamically as a function of the average axonal diameter. This selection method was used to avoid the case of large VOI sizes containing very small fibers.

- **White matter only with two perpendicular axon populations (WM_2Pops).** Finally, the last class of microstructure models cases of two axonal populations perpendicular to each other. The axonal populations have the same diameter distributions but different VFs. For the first axon population, the axon diameters follow a Gamma distribution with an average sampled randomly between 0.5 and 3 μm, and the VF is distributed between 15% and 75%. The second axon population has the same diameter distribution as the first one but a different volume fraction distribution. The VF of the second axon population is such that the total VF of both axon populations does not exceed 90%.

A summarized version of the generated substrate types is provided in Table.(6.1).

	Soma avg diameter [μm]	Soma VF	Axonal avg diameter [μm]	Axonal VF	VOI size [μm]
GM	[14, 20]	[0.07, 0.25]	-	-	120
WM_GM	[14, 22]	[0.06, 0.2]*	[0.5, 3]	[0.05, 0.42]	3.5 x avg diameter
WM	-	-	[0.5, 3]	[0.4, 0.8]	19 x avg diameter
WM_2Pops	-	-	[0.5, 3]	[0.15, 0.9]	19 x avg diameter

Table 6.1: *Summary of the four substrate types generated during the large TGCC simulation campaign and their parameters. The star in the VF values of the WM_ GM substrate class indicates the range of values for the large majority of the generated substrates.*

Campaign of dMRI simulations. This is the second part of the large-scale simulations. It consists in performing MC dMRI simulations using the generated microstructure substrates. For this, the diffusion process parameters used are the following: a set of $N_s = 400\,000$ walkers initialized in both intra and extracellular spaces; the equal-step random leap for the particle-membrane interaction (steps with a particle-membrane collision are rejected and a new step length vector is sampled), and a simulation time step $\delta t = 5.0\mu$s.
The illustration in Fig.(6.6) shows all the diffusion process-related parameters used for the dMRI simulation campaign. Note also that the total number of spins N_s was distributed over ten particle sessions, each using 40 000 random walkers.
The MRI sequences used consist of PGSE sequences with the following b-values: [29.99, 59.97, 89.91, 119.99, 149.99, 180.12, 209.98, 240.03, 270.09, 300.18, 550.34, 700.95, 999.72, 1700.38, 3199.29] in s/mm^2. The PGSE sequences have the following number of orientations respectively: [6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 30, 30, 30, 60, 60]. The time separation $\Delta = 31.72$ ms is constant and the same for all the sequences. This is also the case of the δ parameter which is constant for all

sequences $\delta = 20.24$ ms. A constant echo time was used for all sequences $T_E = 67$ ms. Finally, a finite slew rate -rise time- was used $R_t = 200$ $\mathrm{Tm^{-1}s^{-1}}$ across all sequences.

Diffusion process parameters

```
attributes =
{
    "grid_resolution_in_um": [
        1.0,
        1.0,
        1.0
    ],
    "particle_backup_percentage": 10.0,
    "particle_count": 40000,
    "particle_diffusivity_in_um2_per_us": 0.003,
    "particle_initialization": "everywhere",
    "particle_to_membrane_interaction_type": "equal-step-length-random-leap",
    "session_count": 10,
    "time_step_in_us": 5.0,
    "total_duration_in_us": 67000.0
}
```

Figure 6.6: *Illustration of the diffusion process parameters used for the dMRI simulation campaign.*

6.5 Results

This section presents the results of the large simulation campaign using the MEDUSA framework in the TGCC HPC facility. The first part of the results showcases details and statistics about the generated substrates, and the second part is dedicated to dMRI simulation-related results.

6.5.1 The generated substrates.

As mentioned in the previous section, four substrate types were generated. These four types are denoted by the following abbreviations: **GM**, **WM_GM**, **WM**, and finally **WM_2Pops**. To characterize the generated substrates, different statistics were extracted and illustrated.

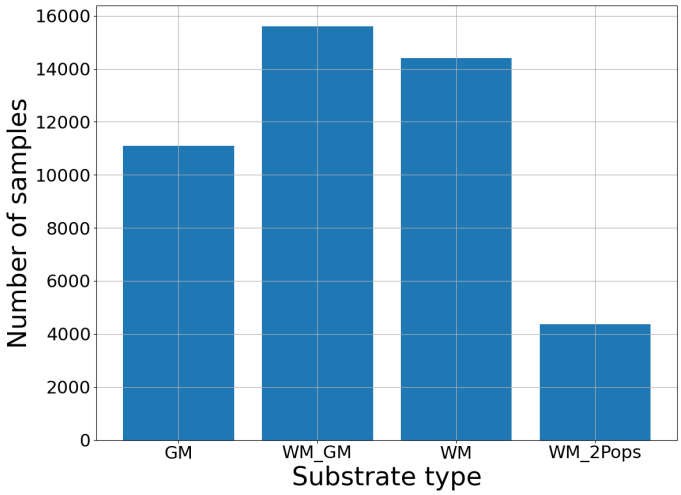

Figure 6.7: *Illustration of the frequency distribution of the generated substrates in the large-scale simulation campaign at the TGCC.*

First, the per type samples distribution, then, more detailed information such as the distribution of microstructural parameters across substrates generated for each substrate type. A total set of 45457 substrates for all the types were generated, totaling around 1.4 TB of data, including the geometry files, dMRI simulation parameter files, and a terminal Log file. The distribution of these substrates is displayed in Fig.(6.7). For the three types GM, WM, and WM_GM there are over 10 000 samples each to ensure a large pool of substrates covering different configurations.
The samples from each substrate type cover a wide range of parameters as specified in the previous section. A more detailed investigation of the distribution of the substrate microstructural parameters was done.

GM. To visualize the parameters of the generated substrate containing pyramidal neurons, a two-dimensional histogram containing marginal frequency distributions was used. The histogram is displayed in Fig.(6.8) and shows the pyramidal neurons VF and their soma average diameter distributions for the generated substrates. The same illustration also shows the marginal frequency distributions for both VFs and soma average diameters.
The parameter distribution shows a relatively even distribution of the substrates over the two selected parameters -VF and average soma diameters- except in a region with low values of average soma diameters and high VF values. The gap region can be qualitatively explained by the relation between the soma diameter and their volume in the neurons stemming from the NEUROMORPHO dataset and by the increased number of atoms required when generating a substrate with a small and dense population of neurons. However, a further investigation is required.

WM. In the case of substrates containing a single axon population, a similar strategy was used to probe the generated substrates and verify whether the microstructural parameters fall within the specified ranges defined in the Meta-JSON input parameters file. The parameters of interest in this class of substrates is the axonal average diameters and VFs. The distribution of these parameters is illustrated in Fig.(6.9) and shows an even distribution of the substrate samples across the parameter ranges. Note also that in this type of substrates, the size of the VOI depends on the average axonal diameter. This allows to adapt the size of the VOI such that substrate with small axonal diameters are generated inside smaller VOIs. When the size of the VOI is not adapted, large substrates containing small fibers can be computationally prohibitive.

WM_GM. This class of substrates is hybrid and contains both pyramidal neurons from the NEUROMORPHO dataset and purely synthetic axons. Thus, in contrast to the previously presented substrate types, this class of microstructure is characterized by four important parameters instead of two: the axonal VF, the pyramidal neurons VF, the axonal average diameter, and finally, the soma average diameter. To inspect the generated substrates, the distribution of the average axon and soma diameters was plotted as a function of the total volume fraction: the sum of the axonal and pyramidal neurons VFs.
The illustration in Fig.(6.10) shows the 2D histogram of the axonal average diameters and the total VF. The distribution is not as even as the one obtained from the **WM** samples. The same illustration shows a vertical gap for samples with an average axonal diameter less than 1 μm. This gap becomes slightly larger for increased values of the total VF.

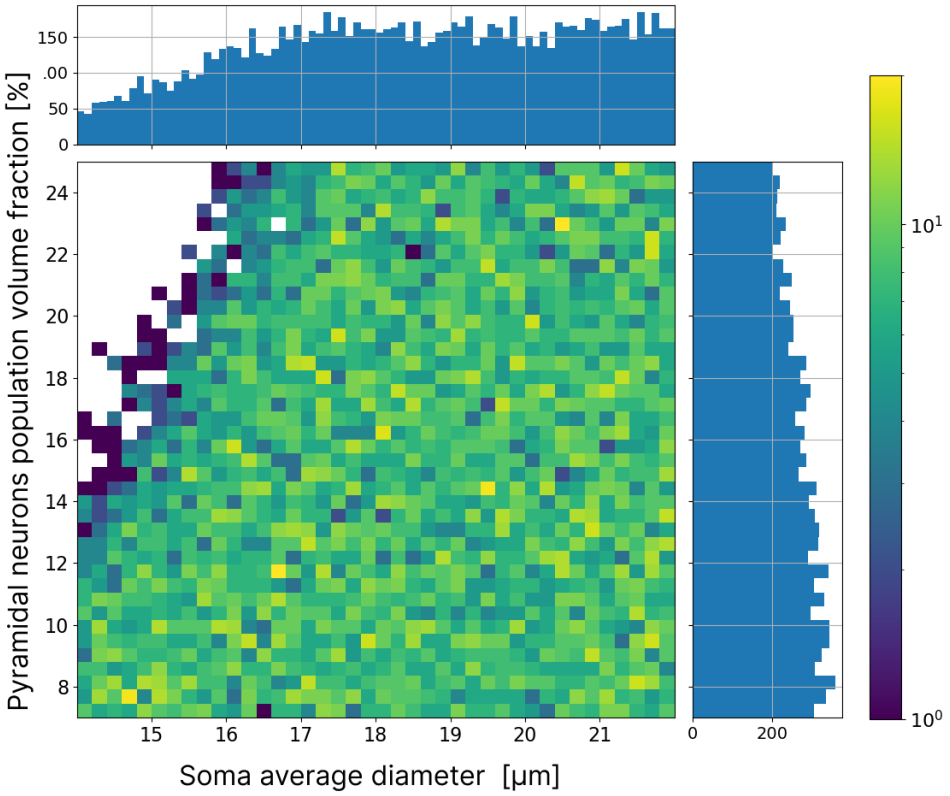

Figure 6.8: *The 2D histogram for the generated substrates of the GM class shows the soma average diameter and the VF with marginal distributions.*

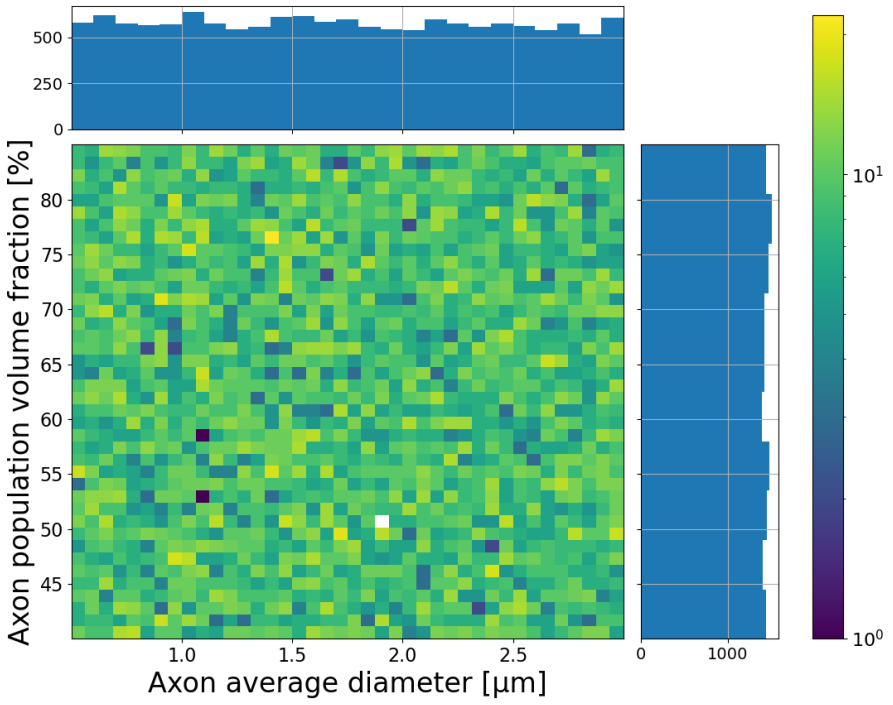

Figure 6.9: *The 2D histogram for the generated substrates of the WM class shows the axonal average diameter and the axonal VF with their marginal distributions.*

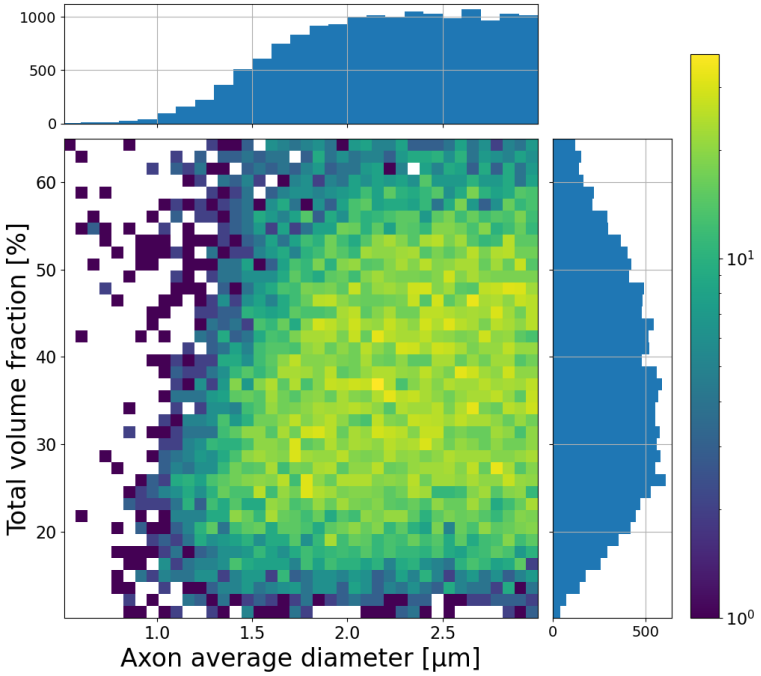

Figure 6.10: *The 2D histogram for the generated substrates of the WM_GM class shows the average axonal diameter and the total VF with their marginal distributions.*

The gap can be explained by the difficulty of generating very thin axons in a large VOI. This matter will be discussed in more detail in the next section. Aside from the thin axon-related gap, the generated samples are distributed over a wide range of parameters.
To characterize the average pyramidal soma diameters, the same visualization process was used. The 2D histogram of soma average diameters is illustrated in Fig.(6.11). It shows a relatively even spread of substrate samples across all the soma-related parameter ranges. The results also show an agreement between the specified parameters in the Meta-JSON input file and the actual simulated samples which show that the soma average diameters take values between 14 and 22 μm as specified.
To further investigate the generated *WM_GM* samples, another visualization was performed to illustrate the distribution of both axonal and pyramidal neurons VFs. This aims to verify if the volume fraction of each compartment lies in their corresponding range specified in the Meta-JSON input file.
The VF distribution of each component illustrated in Fig.(6.12) shows two regimes: first a low sampled triangular region used as a stress test for the MEDUSA simulator for substrates with a large VF of pyramidal neurons, second, a regular regime with a rectangular shape covering a large proportion of the generated samples. The 2D VF histogram shows that the parameter values of the generated samples are evenly distributed across all the specified parameters in the regular regime.

WM_2Pops. For this class of substrates, the same approach is adopted as for the *WM_GM* class. Different visualizations were performed to probe the distribution of the axonal diameters of the two perpendicular axons and their corresponding VFs.

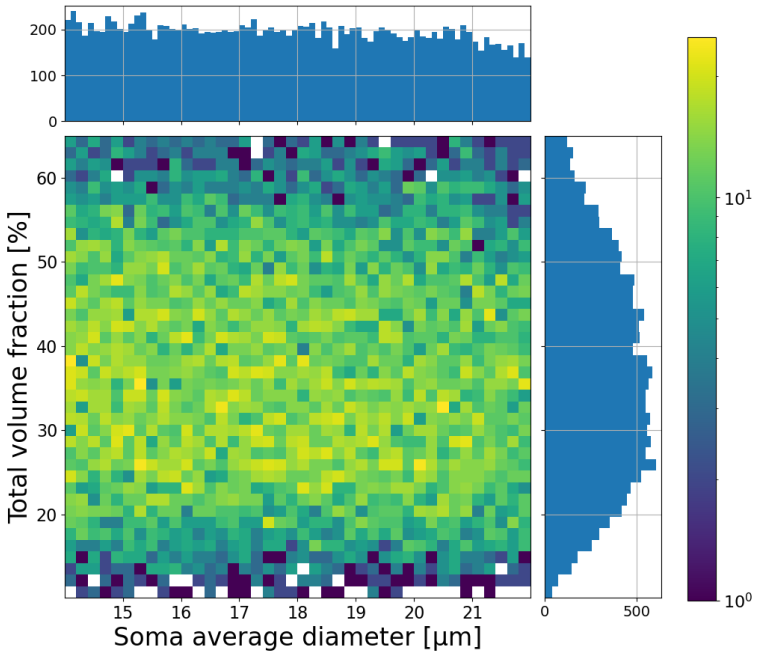

Figure 6.11: *The 2D histogram for the generated substrates of the WM_GM class shows the average soma diameter and the total VF with marginal distributions.*

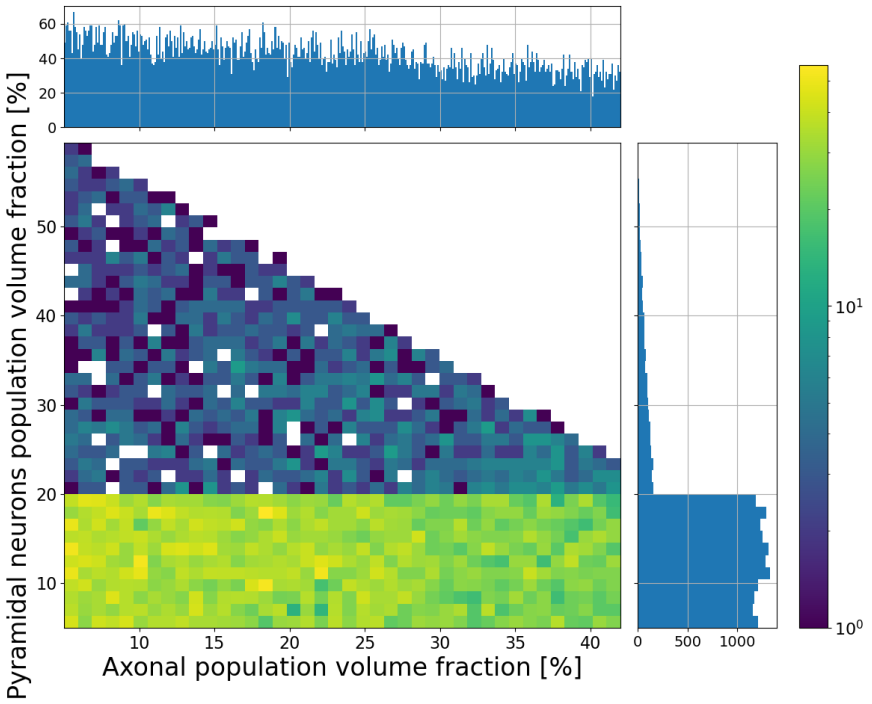

Figure 6.12: *The 2D histogram showing the distributions of the pyramidal neuron and axonal populations and their marginal distributions for the WM_GM substrate class.*

The results related to the axonal diameter distributions are illustrated in Fig.(6.13) and Fig.(6.14).

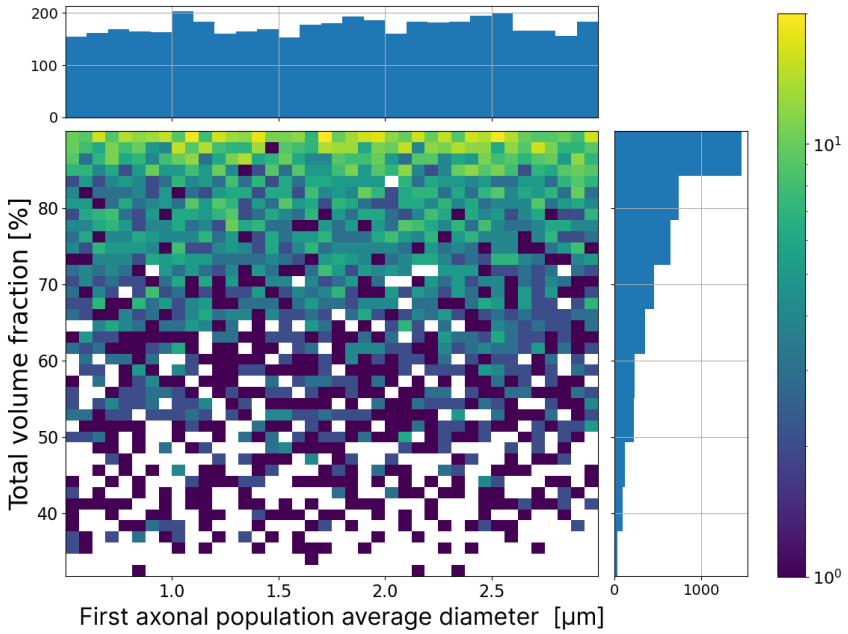

Figure 6.13: *The 2D histogram for the microstructural parameters in the case of the substrate class WM_2Pops showing the axonal diameter of the first axonal population and the total VF values and their marginal distributions.*

The average axonal diameter of the first population in Fig.(6.13) shows an increasing number of samples as the total VF increases. This result is related to the relationship between the two axon populations embedded in the Meta-JSON. Note that the marginal distribution of the diameter of the first axonal population is evenly distributed.
The average axonal diameter for the second axon population is displayed in Fig.(6.14). This distribution is similar to the one of the first axonal population shown in the previous illustration in Fig.(6.13). This distribution also displays similar properties: increasingly more samples as the total VF increases and an even marginal distribution of the second axonal population average diameters.
Finally, an investigation of the VFs of both axonal populations was done to show how the generated substrates are distributed on different VF values. The results displayed in Fig.(6.15) show a triangular-shaped 2D histogram such that the total VF does not exceed 90%.

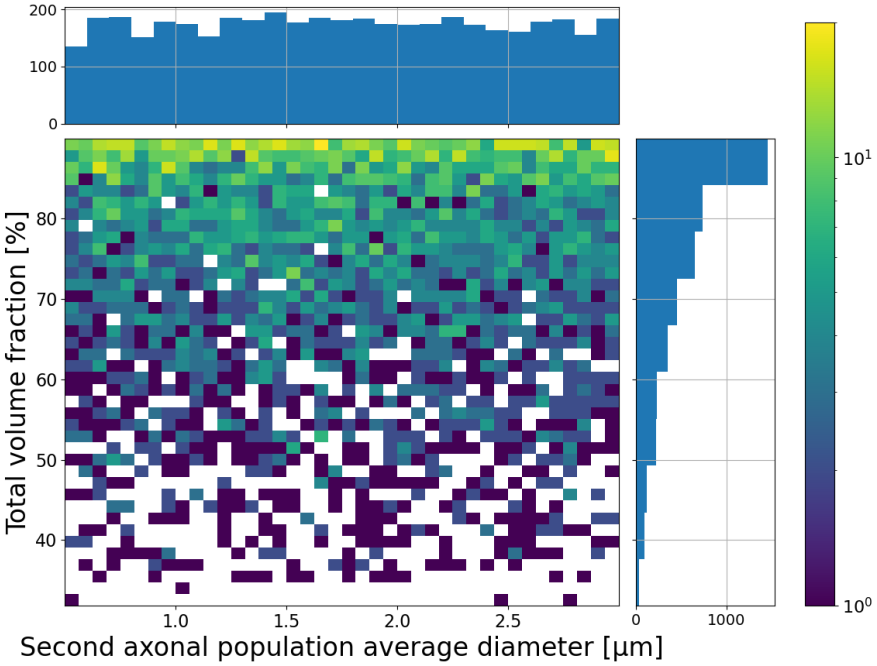

Figure 6.14: *The 2D histogram for the microstructural parameters in the case of the substrate class WM_2Pops showing the axonal diameter of the second axonal population and the total VF values and their marginal distributions.*

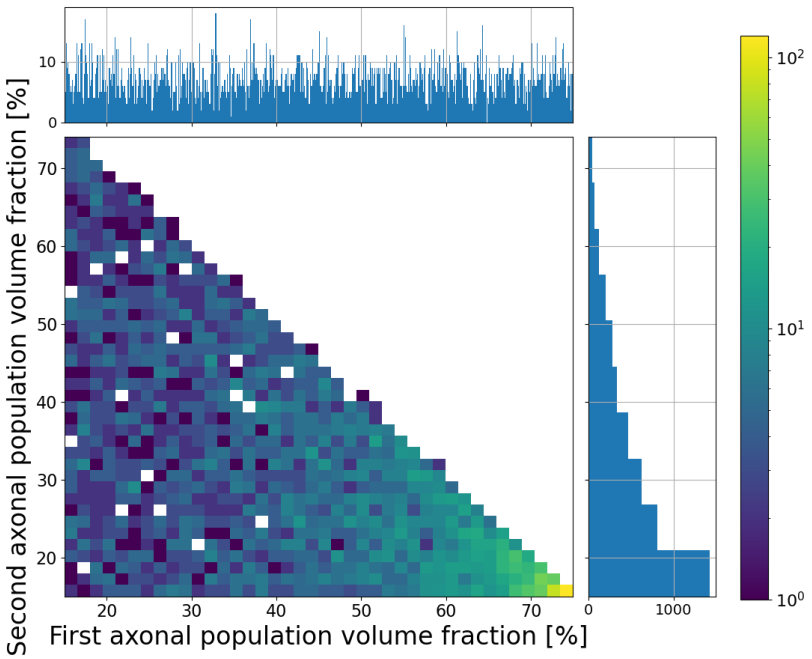

Figure 6.15: *The 2D histogram for the VF of the two axonal populations in the WM_2Pops substrate class with their marginal distributions.*

6.5.2 DMRI simulation results.

This part represents the second step of the large-scale simulation campaign performed at the TGCC. First, insights about the simulated samples are presented, and then, dMRI results of different samples are shown.
For the dMRI simulations, a total of 3856 substrates from different substrate types were used to extract dMRI signals.

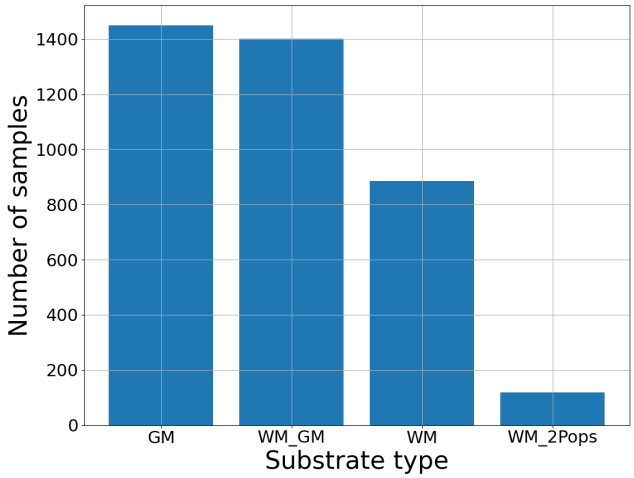

Figure 6.16: *Illustration of the frequency distribution of the generated substrates for which the dMRI signals were simulated.*

The simulated substrates stem from different substrate classes, and the per-class distribution is displayed in Fig.(6.16). Around 74% of the simulated substrates involve neuronal cells, GM and WM_GM samples combined.

GM. For this class of substrates, 1450 substrates containing populations of pyramidal neurons were used to extract dMRI signals. First, the parameters of these substrates were characterized by the distribution of their microstructural parameters: soma average diameter and VF values.
The parameter distribution of the simulated substrates is illustrated in the left panel of Fig.(6.17). The histogram shows a similar distribution to that of the total samples from the GM substrate class in Fig.(6.8). The 2D histogram of simulated substrates has four red rectangles outlining four subsets representing extreme cases of microstructural parameters: a) small soma average diameters and high VF, b) large soma average diameters and high VF, c) small soma average diameters and low VF, and finally, d) small soma average diameters and high VF. For each case, the orientation-averaged dMRI signals were calculated and averaged across the simulated samples within the same type.
The orientation-averaged dMRI signals are shown in the right panel of Fig.(6.17) with error bars reflecting the variance resulting from different samples. The number of samples used to calculate the average values of the orientation-averaged signals are the following: 18, 34, 46, and finally, 40 samples for cases a), b), c), and, d) respectively.

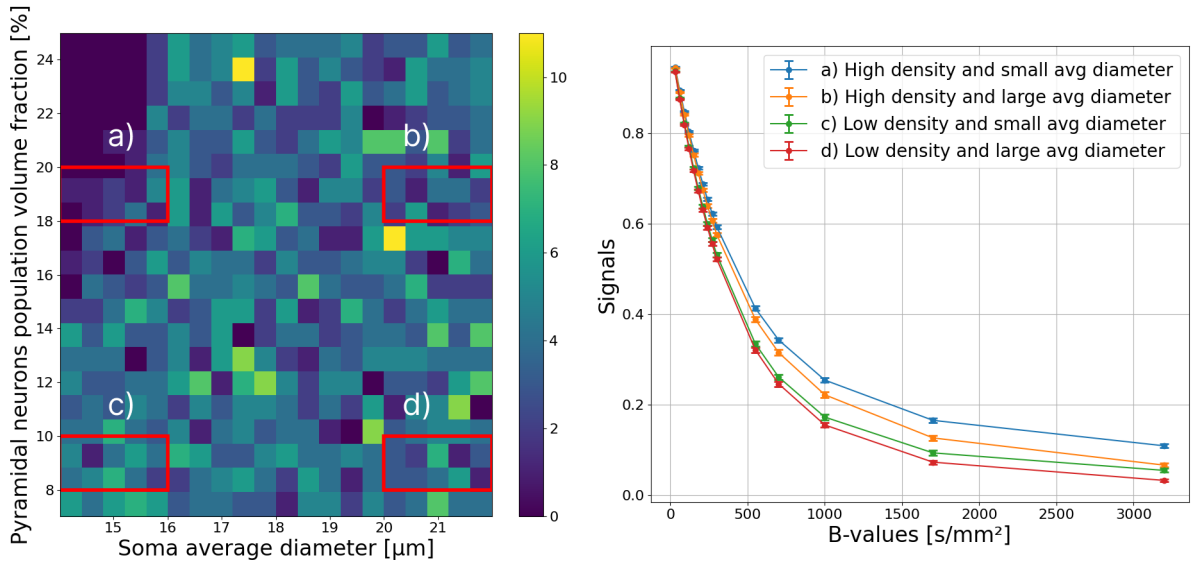

Figure 6.17: *Illustration of the microstructural parameter distribution of the simulated substrates from the GM class and their corresponding orientation-averaged dMRI signals.*

WM. In this case, a similar approach to the substrate class GM was used. The simulated substrate samples with one single axon population were characterized using the average axonal diameter and the VF.

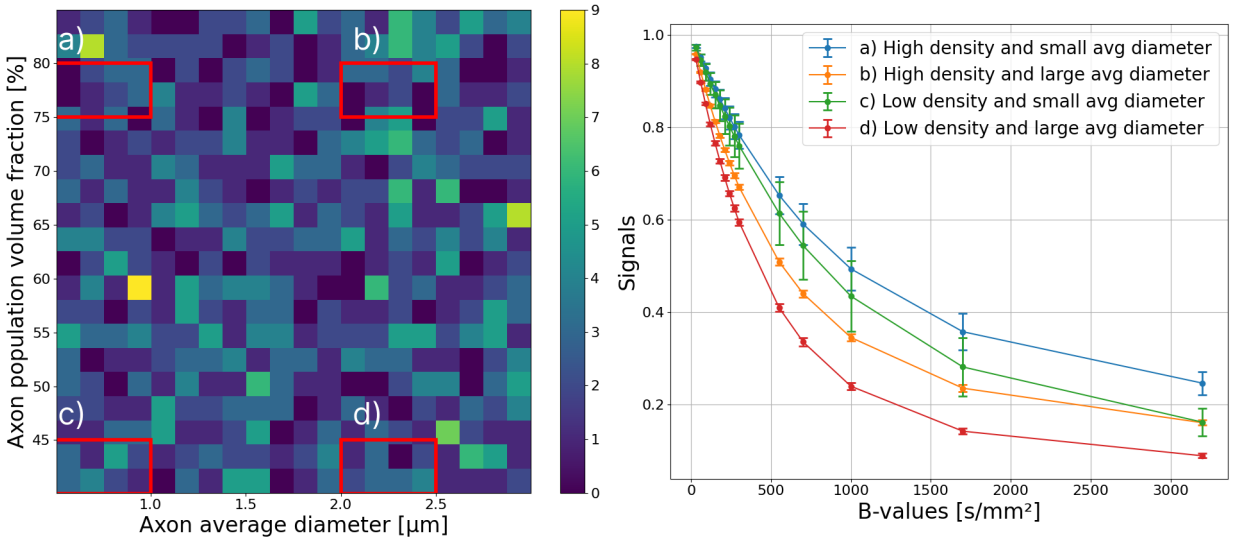

Figure 6.18: *Illustration of the microstructural parameter distribution of the simulated substrates from the WM class and their corresponding orientation-averaged dMRI signals.*

In the same fashion, four different regions representing extreme microstructural parameter cases were selected as shown in the left panel of Fig.(6.18). Their corresponding average orientation-averaged dMRI signals are illustrated in the right panel of the same figure. The number of substrate samples used for each case are the following: 17, 18, 21, and 23 for the cases a), b), c), and d), respectively. Note also from the dMRI signals that for the case c) of low-density and small axonal diameter substrates, the variance is large and shows an important overlap with the

signal curve of case a) representing high-density substrates with small average diameters.

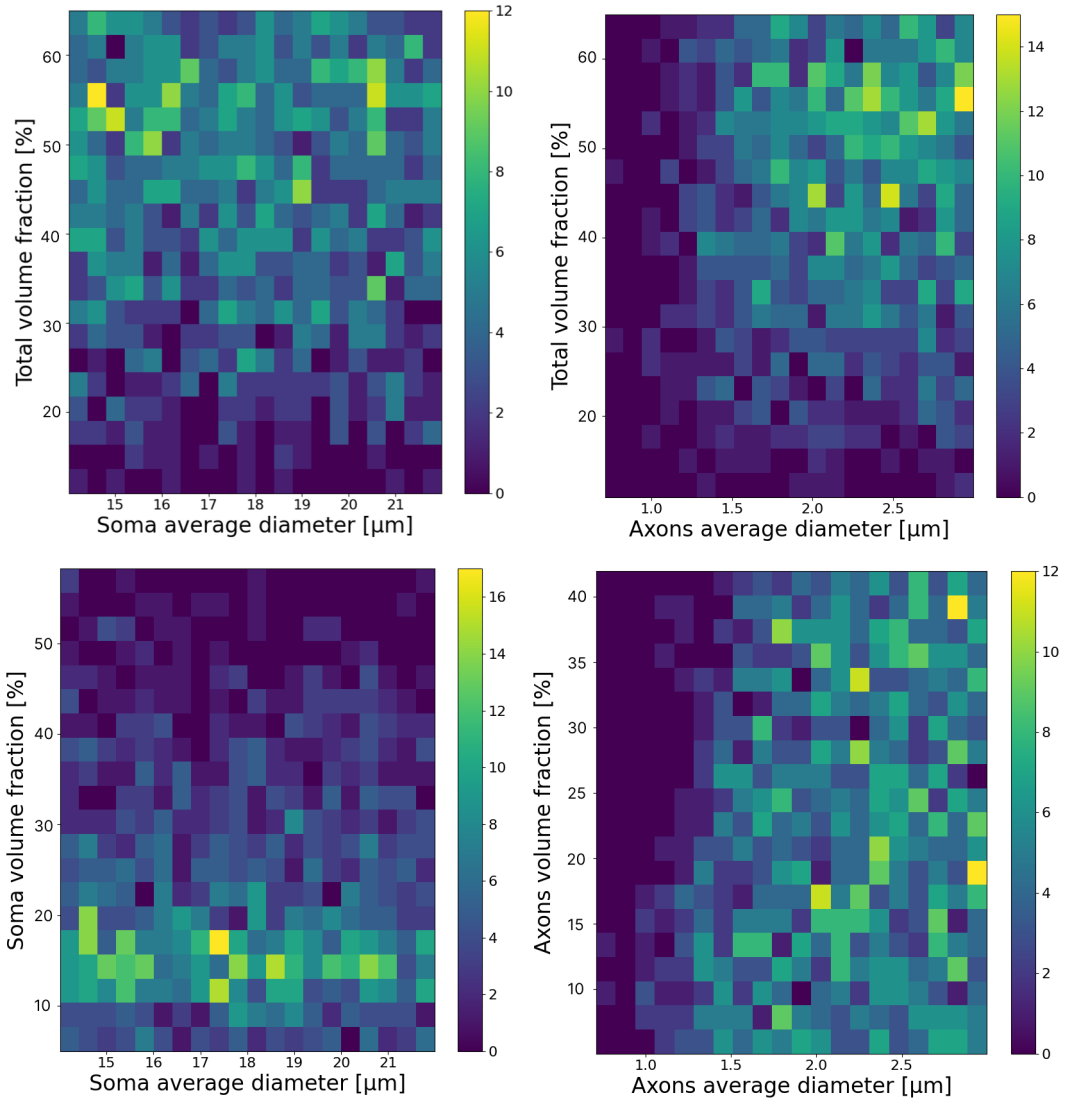

Figure 6.19: *Illustration of the microstructural parameter distribution of the simulated substrates from the WM_GM class. The top left panel shows the 2D histogram of the pyramidal neuron's average soma diameter and the total VF. The top right panel shows the 2D histogram of the average axonal diameters and the total VF. The bottom left panel shows the 2D histogram of the soma VF and its average diameter. Finally, the bottom right panel shows the 2D histogram of the axonal average diameters and their VF.*

WM_GM. For this substrate class, dMRI signals were simulated for a total of 1402 substrate samples, covering different ranges of microstructural parameters. First, these parameters were characterized by showcasing their distribution, and then, the results of the dMRI signals obtained using this class of substrates are presented.
The second part of the results investigates the effects of the soma volume fraction on the orientation-averaged dMRI signals. For this, two subsets of the simulated samples were selected. Each subset has microstructural parameters in the same range except for the VF of the neurons which is what distinguishes the two subsets from each other. Both subsets have the following set of parameters:

- **Subset a).** Has axonal VF in the interval $[0.35, 0.42]$, the axonal average diameter values are in the $[1.0, 3.0]$ μm range, the average diameter values of the soma is in the interval $[14, 18]$ μm. In this case, the substrates have higher neuron VF values in the range $[0.2, 0.25]$.
- **Subset b).** Has the same parameters as the substrates of case a) but with lower VF values for the pyramidal neurons which lie in the $[0.07, 0.11]$ range.

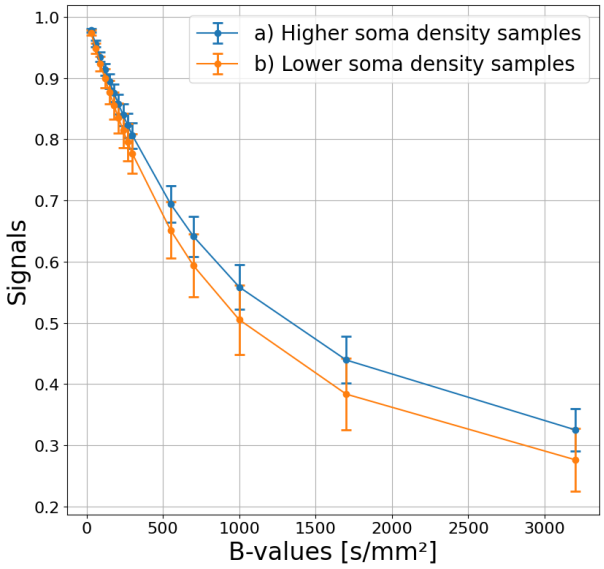

Figure 6.20: *The average orientation-averaged dMRI signals for both subsets of simulated substrates of the WM_GM class showcasing the difference between high and low neuron VF values.*

6.6 Discussion

The results presented in the previous section demonstrate the ability of the MEDUSA framework to generate and perform dMRI simulations at a large scale. This scalable generation allowed by the TGCC HPC facility is not only restricted to white matter substrates but also gray matter ones. In addition, the combination of virtualization tools such as Docker and the Kokkos framework indeed allows performance portability: the execution of the code adapts across different computer systems with various compute resources while maintaining at the same time a near-optimal performance. The generated substrates have shown the coverage of a wide range of microstructural parameters as seen in different 2D histograms in Fig.(6.8), Fig.(6.9), Fig.(6.10), and Fig.(6.11), Fig.(6.13), Fig.(6.14), and Fig.(6.15). Although different values of microstructural parameters were covered, for the hybrid substrates of class WM_GM, average values of axonal diameters below 1 μm are still not reached. We argue that this limitation can be overcome by using a diameter-dependent overlap ratio, as mentioned in the discussion section of the previous chapter. By introducing this feature in the MEDUSA framework, it is possible to further reduce the memory size of substrates and speed up the de-overlapping process. In addition, the use of a diameter-dependent overlap ratio O_r can enable the generation of substrates from the WM_GM

class with lower axonal diameters reaching closer values to those of the human brain. Also, the diameter-dependent overlap ratio relation can be tuned according to the used MRI sequences to minimize dMRI signal bias.
The second part of the results showed the possibility of performing dMRI simulation at a large scale using the MEDUSA framework. Although only a subset of the generated substrates were used to extract dMRI signals, this choice was driven by the time constraints of this project. The dMRI simulations presented can be considered as our first exploratory results of how the dMRI signals change with different microstructural parameters. First, the exploration of the microstructural parameters of the substrates for which dMRI simulations were performed shows a wide coverage of the parameter values with distributions similar to those of the large pool of the generated substrates. The diverse set of microstructural parameters allowed the selection of four regions for the WM and GM substrate classes representing extreme cases of parameter values: low average diameters and high density, high average diameters and high density, low average diameters and low density, and finally, low average diameters and high density. These preliminary results show for the substrate classes WM and GM that for similar average radii distributions, the dMRI signals are attenuated more with lower cell densities. This finding was also consistent with the experiment conducted on the substrate of class WM_GM which investigates the effect of soma density on the orientation-averaged dMRI signals for a wide range of microstructural parameters. These results seem also to be in agreement with the meta-analysis study in [Chen *et al.*, 2013] showing a negative correlation between the ADC and tumor cellularity.

6.7 Conlusion

This chapter presented the results of a large simulation campaign using the MEDUSA framework for both substrate generation and MC dMRI simulations. This demonstrates the ability of this framework to perform end-to-end computations starting from a set of microstructural parameters and MRI sequences. The results presented in this work indeed confirm the sensitivity of the dMRI signals to the volume fraction of the soma. Therefore, this opens new perspectives on probing microstructural parameters using dMRI signals and dedicating more focus on performing real-to-simulation validation of the dMRI signals.

Chapter 7

Conclusion and outlook

This chapter provides broad conclusions and future perspectives in the context of this thesis. It is structured as follows: first, the contributions made in the frame of this work are listed. Second, from the made contributions, a broad conclusion is presented. Finally, future work directions and perspectives are discussed.

7.1 Main contributions and conclusion

Different contributions were made in the course of this thesis in areas including microstructure modeling, 3D geometric processing, and Monte Carlo dMRI simulations. A summary of the contributions is provided below:

- **The Skeleton-Guided Ellipsoid Atom Decomposition (SGEAD) method.** This method consists of converting cell meshes into a set of ellipsoids that approximates the mesh surface. Its application to the H01 dataset demonstrated its ability to convert large cell meshes reconstructed from electron microscopy to a memory-lightweight version made of ellipsoids with compression rate values over 200. The ellipsoid-based cells are also compatible with the dMRI simulator extended in the course of this thesis. This contribution provided a thorough proof of concept of the ability to convert highly realistic volumes of human brain microstructure to a lightweight format, thus facilitating the comparison between simulated and acquired dMRI signals.

- **Generation of synthetic gray matter microstructure substrates.** Prior to this thesis, very few methods have addressed the synthesis of realistic virtual gray matter microstructures. In this thesis, a hybrid approach was proposed to generate virtual gray matter substrates. The approach consists of combining purely synthetic cell populations with neurons stemming from open-access datasets. For this, different extensions were made to the MEDUSA framework: first, processing and integrating cells from both the H01 and NEUROMORPHO datasets into the MEDUSA framework. Second, extensions were also made to support the use of ellipsoid-based geometries and generalize the virtual microstructure generation process such that ellipsoid-based substrates can be generated with the same workflow. The proposed hybrid substrate generation method was shown to generate substrates inside 3D cubic voxels of size 120 μm with microstructural parameter values (density and average diameters) close to those observed in the human brain. Although cellular components such

as blood vessels were not accounted for, the generation method still marks an important step towards the synthesis of gray matter substrates.

- **DMRI simulations on atom-based geometries.** This part consisted of two main contributions. First, a method to perform dMRI simulation on atom-based geometries using elastic particle-membrane collisions was developed and integrated into the MEDUSA framework. Second, the MEDUSA dMRI simulator was validated using both analytical and empirical benchmarks. The results of the validation show that the dMRI signals stemming from atom-based geometries lead to comparable results to their mesh-based counterparts while at the same time benefiting from the computational and memory efficiencies gained with the use of atom-based geometries.

- **Large simulation campaign using the TGCC HPC facility.** In this part, two simulation campaigns were performed. The first one is a large-scale microstructure substrate generation campaign and the second one is a dMRI simulation campaign to extract attenuation signals from the generated substrates. This contribution demonstrates the scalability of the previous contributions for generating gray matter microstructure substrates. It also demonstrates the ability of the MEDUSA toolkit to be used on a large scale and deployed to high-performance computing platforms. The preliminary results of the large simulation campaign showed the impact of the average diameters of soma and their densities on the dMRI signals.

The ensemble of contributions made in this thesis provides a notable advancement in modeling gray matter microstructure and performing dMRI simulations. The presented methods combined an interdisciplinary approach and addressed the scalability issue related to the use of mesh-based representations. Finally, the contributions made allow getting closer to decoding gray matter microstructure.

7.2 Future perspectives

The combination of the different methods developed in the course of this thesis opens exciting future perspectives and work directions. The focus of this work was to improve and generalize the substrate generation methods by using atom-based geometries such as spheres or ellipsoids to efficiently represent substrates. This type of representation allows to efficiently test for collisions and whether a point belongs to a cell, which are key ingredients for building a Monte Carlo dMRI simulator. Although a large-scale simulation campaign was conducted in the course of this thesis, further investigations are required to compare the obtained results with the different models proposed in the literature. A list of interesting future research projects is provided below:

- **Implementing the suggested improvements.** This part consists in implementing the suggested features to improve the SGEAD method and the MEDUSA framework. For instance, for the SGEAD method, explore a C++ implementation of the SGEAD to unlock better computational performance. For the case of the MEDUSA framework, implement a diameter-dependent overlapping-ratio feature. This feature allows to generate substrates

with a larger voxel size or, for a given voxel size, generate hybrid substrates with lower axonal diameters.

- **Comparisons with analytical models.** The comparison of gray matter dedicated models such as SANDI [Palombo *et al.*, 2020] with dMRI signals generated using the MEDUSA framework. This approach is not only limited to gray matter substrates but can also be used to verify if the estimated white matter microstructural parameters from analytical models such as NODDI [Zhang *et al.*, 2012] are in agreement with the white matter substrates generated using the MEDUSA framework.

- **Training a neural network to establish a computational model.** Another interesting project is to perform more dMRI simulations on gray matter microstructure substrates and build a computational model, i.e., a deep learning network to map the dMRI signals to the microstructural parameters that also include different values of the cell permeability, a parameter that was not accounted for in the simulation campaign. This step also includes benchmarks with published microstructure computational models, such as the model presented in this study [Fang *et al.*, 2023].

- **Validating simulated dMRI signals against acquired signals.** An important research direction is validating the simulation results against real substrates. In our view, this remains a major challenge but also crucial and needs to be addressed. The contribution of the SGEAD method to convert microstructure meshes into ellipsoid-based geometries can facilitate the validation process against real substrates. Thus future works in this direction include the acquisition of dMRI signals from small *ex-vivo* human brain samples and digitizing them using electron microscopy techniques as performed in [Shapson-Coe *et al.*, 2024]. The dMRI acquisition signals can be compared with the simulated dMRI signals on the numerically-reconstructed cells which were converted to atom-based geometries. This process enables the direct comparison between the acquired dMRI signals and the simulated ones. This validation step can potentially help bring successful computational models to clinical practice and decode microstructure substrates *in-vivo*.

Communications

List of communications:

- Bachiri A., Brullé A., Uszynski I., and Poupon C., Ginkgo: Increasing the realism of in silico cortex substrates for diffusion MRI simulation. Abstract 3460. International Society for Magnetic Resonance in Medicine (2023).
- Bachiri A., Brullé A., Uszynski I., and Poupon C., Ginkgo: Increasing the realism of in silico cortex substrates for diffusion MRI simulation. Abstract 1609. Organization for Human Brain Mapping (2023).
- Bachiri A., Brullé A., Uszynski I., and Poupon C., Ginkgo: A novel MEDUSA ultra-realistic and memory-efficient neuron population model for the synthesis of gray matter substrates. Organization for Human Brain Mapping (2022).
- Brullé A., Bachiri A., Destrieux C., Wiber G., Delzescaux T., Uszynski I., and Poupon C., Ginkgo: Large-scale simulations to create large collections of realistic white matter samples using MEDUSA. Abstract 2726. International Society for Magnetic Resonance in Medicine (2022).
- Brullé A., Bachiri A., Destrieux C., Wiber G., Delzescaux T., Uszynski I., and Poupon C., Ginkgo: Large-scale simulations to create large collections of realistic white matter samples using MEDUSA. Abstract 449. Organization for Human Brain Mapping (2022).

Bibliography

[typ, n.d.] *Types of Neurons.* https://qbi.uq.edu.au/brain/brain-anatomy/types-neurons. [Accessed 04-02-2024].

[tau, 2024] 2024. *Coronal sections of the brain | Neuroanatomy — neuroanatomy.sites.tau.ac.il.* https://www.neuroanatomy.sites.tau.ac.il/copy-of-blood-vessels-of-the-brain. [Accessed 26-01-2024].

[mha, 2024] 2024. *Neurons: How the Brain Communicates — mhanational.org.* https://mhanational.org/neurons-how-brain-communicates. [Accessed 04-02-2024].

[Abdollahzadeh *et al.*, 2019] Abdollahzadeh, Ali, Belevich, Ilya, Jokitalo, Eija, Tohka, Jussi, & Sierra, Alejandra. 2019. Automated 3D Axonal Morphometry of White Matter. *Scientific Reports*, **9**(1), 6084.

[Alexander, 2008] Alexander, Daniel C. 2008. A general framework for experiment design in diffusion MRI and its application in measuring direct tissue-microstructure features. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, **60**(2), 439–448.

[Alexis *et al.*, 2022] Alexis, Brullé, Anas, Bachiri, Christpohe, Destrieux, Gilles, Wiber, Thierry, Delzescaux, Ivy, Uzsynski, & Cyril, Poupon. 2022 (05). Large-scale simulations to create large collections of realistic white matter samples using MEDUSA. *In: International Society for Magnetic Resonance in Medicine.*

[Alfano & Greer, 2003] Alfano, Salvatore, & Greer, Meredith L. 2003. Determining if two solid ellipsoids intersect. *Journal of guidance, control, and dynamics*, **26**(1), 106–110.

[Allman, 1999] Allman, J.M. 1999. *Evolving Brains.* Library series]. Scientific American Library.

[Ascoli *et al.*, 2007] Ascoli, Giorgio A., Donohue, Duncan E., & Halavi, Maryam. 2007. NeuroMorpho.Org: A Central Resource for Neuronal Morphologies. *Journal of Neuroscience*, **27**(35), 9247–9251.

[Assaf *et al.*, 2004] Assaf, Yaniv, Freidlin, Raisa Z, Rohde, Gustavo K, & Basser, Peter J. 2004. New modeling and experimental framework to characterize hindered and restricted water diffusion in brain white matter. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, **52**(5), 965–978.

[Assaf *et al.*, 2008] Assaf, Yaniv, Blumenfeld-Katzir, Tamar, Yovel, Yossi, & Basser, Peter J. 2008. AxCaliber: a method for measuring axon diameter distribution from diffusion MRI.

Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine, **59**(6), 1347–1354.

[Azevedo *et al.*, 2009] Azevedo, Frederico A.C., Carvalho, Ludmila R.B., Grinberg, Lea T., Farfel, José Marcelo, Ferretti, Renata E.L., Leite, Renata E.P., Filho, Wilson Jacob, Lent, Roberto, & Herculano-Houzel, Suzana. 2009. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *Journal of Comparative Neurology*, **513**(5), 532–541.

[Balls & Frank, 2009] Balls, Gregory T., & Frank, Lawrence R. 2009. A simulation environment for diffusion weighted MR experiments in complex media. *Magnetic Resonance in Medicine*, **62**, 771–778.

[Bammer, n.d.] Bammer, Roland. *Basic principles of diffusion-weighted imaging.*

[Basinger & Hogg, 2019] Basinger, Hayden, & Hogg, Jeffery P. 2019. *Neuroanatomy, brainstem.* StatPearls.

[Basser *et al.*, 1994] Basser, P.J., Mattiello, J., & LeBihan, D. 1994. MR diffusion tensor spectroscopy and imaging. *Biophysical Journal*, **66**(1), 259–267.

[Behrens *et al.*, 2003] Behrens, Timothy EJ, Woolrich, Mark W, Jenkinson, Mark, Johansen-Berg, Heidi, Nunes, Rita G, Clare, Stuart, Matthews, Paul M, Brady, J Michael, & Smith, Stephen M. 2003. Characterization and propagation of uncertainty in diffusion-weighted MR imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, **50**(5), 1077–1088.

[Berlin, 2024] Berlin, Wias. 2024. *TetGen example: Dragon.* [Online; accessed 01-March-2024].

[Bischoff & Kobbelt, 2002] Bischoff, S, & Kobbelt, L. 2002. Ellipsoid decomposition of 3D-models. *Proceedings - 1st International Symposium on 3D Data Processing Visualization and Transmission, 3DPVT 2002*, 480–488.

[Bloch, 1946] Bloch, F. 1946. Nuclear Induction. *Phys. Rev.*, **70**(Oct), 460–474.

[Bloch *et al.*, 1946] Bloch, F., Hansen, W. W., & Packard, M. 1946. The Nuclear Induction Experiment. *Phys. Rev.*, **70**(Oct), 474–485.

[Bradshaw & O'Sullivan, 2004] Bradshaw, Gareth, & O'Sullivan, Carol. 2004. Adaptive Medial-Axis Approximation for Sphere-tree Construction. *ACM Transactions on Graphics*, **23**(01).

[Breazeal & Scassellati, 2000] Breazeal, Cynthia, & Scassellati, Brian. 2000. Challenges in Building Robots That Imitate People. *Imitation in Animals and Artifacts*, 10.

[Brodmann, 1909] Brodmann, K. 1909. *Vergleichende Lokalisationslehre der Grosshirnrinde in ihren Prinzipien dargestellt auf Grund des Zellenbaues von Dr. K. Brodmann ...* J.A. Barth.

[Callaghan, 1994] Callaghan, A. Coy; P. T. 1994. Use of spin echoes in a pulsed magnetic-field gradient to study anisotropie, restricted diffusion and flow. *The Journal of Chemical Physics*, **101**, 4599–4609.

[Callaghan, 1995] Callaghan, P T. 1995. Pulsed-Gradient Spin-Echo NMR for Planar, Cylindrical, and Spherical Pores under Conditions of Wall Relaxation. *Journal of Magnetic Resonance, Series A*, **113**, 53–59.

[Callaghan *et al.*, 2020] Callaghan, Ross, Alexander, Daniel C., Palombo, Marco, & Zhang, Hui. 2020. ConFiG: Contextual Fibre Growth to generate realistic axonal packing for diffusion MRI simulation. *NeuroImage*, **220**, 117107.

[Carr & Purcell, 1954] Carr, H. Y., & Purcell, E. M. 1954. Effects of Diffusion on Free Precession in Nuclear Magnetic Resonance Experiments. *Phys. Rev.*, **94**(May), 630–638.

[Caskey *et al.*, 2023] Caskey, Eric, Gilbert, Catherine, & Chen, Victor. 2023. Chapter 9 - Visual pathways and hydrocephalus. *Pages 117–137 of:* Tubbs, R. Shane, Iwanaga, Joe, Rizk, Elias B., D'Antoni, Anthony V., & Dumont, Aaron S. (eds), *Cerebrospinal Fluid and Subarachnoid Space*. Academic Press.

[Chen *et al.*, 2013] Chen, Lihua, Liu, Min, Bao, Jing, Xia, Yunbao, Zhang, Jiuquan, Zhang, Lin, Huang, Xuequan, & Wang, Jian. 2013. The Correlation between Apparent Diffusion Coefficient and Tumor Cellularity in Patients: A Meta-Analysis. *PLOS ONE*, **8**(11), 1–9.

[Chomiak & Hu, 2009] Chomiak, Taylor, & Hu, Bin. 2009. What Is the Optimal Value of the g-Ratio for Myelinated Fibers in the Rat CNS? A Theoretical Approach. *PloS one*, **4**(11), e7754.

[Community, 2018] Community, Blender Online. 2018. *Blender - a 3D modelling and rendering package*. Blender Foundation, Stichting Blender Foundation, Amsterdam.

[Cover & Mathur, 2020] Cover, Kara, & Mathur, Brian. 2020. Axo-axonic synapses: Diversity in neural circuit function. *Journal of Comparative Neurology*, **529**(12).

[Descoteaux & Poupon, 2014] Descoteaux, Maxime, & Poupon, Cyril. 2014. Comprehensive Biomedical Physics. *Comprehensive Biomedical Physics*, 01, 81–97.

[Di Paola *et al.*, 2010] Di Paola, Margherita, Spalletta, Gianfranco, & Caltagirone, Carlo. 2010. In Vivo Structural Neuroanatomy of Corpus Callosum in Alzheimer's Disease and Mild Cognitive Impairment Using Different MRI Techniques: A Review. *Journal of Alzheimer's disease : JAD*, **20**(02), 67–95.

[Döring *et al.*, 2021] Döring, A, Afzali, M, Kleban, E, Kreis, R, & Jones, DK. 2021. Realistic simulations of diffusion MR spectroscopy: The effect of glial cell swelling on non-Gaussian and anomalous diffusion. *In: Proceedings of the International Society for Magnetic Resonance in Medicine (ISMRM).*

[Einstein, 1905] Einstein, A. 1905. Über die von der molekularkinetischen Theorie der Wärme geforderte Bewegung von in ruhenden Flüssigkeiten suspendierten Teilchen. *Annalen der Physik*, **322**(8), 549–560.

[Fang *et al.*, 2020] Fang, Chengran, Nguyen, Van Dang, Wassermann, Demian, & Li, Jing Rebecca. 2020. Diffusion MRI simulation of realistic neurons with SpinDoctor and the Neuron Module. *NeuroImage*, **222**, 117198.

[Fang *et al.*, 2023] Fang, Chengran, Yang, Zheyi, Wassermann, Demian, & Li, Jing-Rebecca. 2023. A simulation-driven supervised learning framework to estimate brain microstructure using diffusion MRI. *Medical Image Analysis*, **90**, 102979.

[Fatterpekar *et al.*, 2002] Fatterpekar, Girish M, Naidich, Thomas P, Delman, Bradley N, Aguinaldo, Juan G, Gultekin, S Humayun, Sherwood, Chet C, Hof, Patrick R, Drayer, Burton P, & Fayad, Zahi A. 2002. Cytoarchitecture of the human cerebral cortex: MR microscopy of excised specimens at 9.4 Tesla. *AJNR Am. J. Neuroradiol.*, **23**(8), 1313–1321.

[Fedorov *et al.*, 2012] Fedorov, Andriy, Beichel, Reinhard, Kalpathy-Cramer, Jayashree, Finet, Julien, Fillion-Robin, Jean-Christophe, Pujol, Sonia, Bauer, Christian, Jennings, Dominique, Fennessy, Fiona, Sonka, Milan, Buatti, John, Aylward, Stephen, Miller, James V., Pieper, Steve, & Kikinis, Ron. 2012. 3D Slicer as an image computing platform for the Quantitative Imaging Network. *Magnetic Resonance Imaging*, **30**(9), 1323–1341. Quantitative Imaging in Cancer.

[Fick, 1855] Fick, Adolph. 1855. V. On liquid diffusion. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, **10**(63), 30–39.

[Fick, 2017] Fick, Rutger. 2017 (03). *Advanced dMRI Signal Modeling for Tissue Microstructure Characterization*. Ph.D. thesis, Université Côte d'Azur.

[Fieremans *et al.*, 2013] Fieremans, E., Benitez, A., Jensen, J.H., Falangola, M.F., Tabesh, A., Deardorff, R.L., Spampinato, M.V.S., Babb, J.S., Novikov, D.S., Ferris, S.H., & Helpern, J.A. 2013. Novel White Matter Tract Integrity Metrics Sensitive to Alzheimer Disease Progression. *American Journal of Neuroradiology*, **34**(11), 2105–2112.

[Fieremans & Lee, 2018] Fieremans, Els, & Lee, Hong-Hsi. 2018. Physical and numerical phantoms for the validation of brain microstructural MRI: A cookbook. *NeuroImage*, **182**, 39–61. Microstructural Imaging.

[Gardier *et al.*, 2023] Gardier, Rémy, Haro, Juan Luis Villarreal, Canales-Rodríguez, Erick J., Jelescu, Ileana O., Girard, Gabriel, Rafael-Patiño, Jonathan, & Thiran, Jean Philippe. 2023. Cellular Exchange Imaging (CEXI): Evaluation of a diffusion model including water exchange in cells using numerical phantoms of permeable spheres. *Magnetic Resonance in Medicine*, **90**(10), 1625–1640.

[Ginsburger, 2019] Ginsburger, Kévin. 2019 (Aug.). *Modeling and simulation of the diffusion MRI signal from human brain white matter to decode its microstructure and produce an anatomic atlas at high fields (3T)*. Theses, Université Paris Saclay (COmUE).

[Ginsburger *et al.*, 2019] Ginsburger, Kévin, Matuschke, Felix, Poupon, Fabrice, Mangin, Jean-François, Axer, Markus, & Poupon, Cyril. 2019. MEDUSA: A GPU-based tool to create realistic phantoms of the brain microstructure using tiny spheres. *NeuroImage*, **193**, 10–24.

[Hahn, 1950] Hahn, E. L. 1950. Spin Echoes. *Phys. Rev.*, **80**(Nov), 580–594.

[Hall & Alexander, 2009] Hall, Matt G., & Alexander, Daniel C. 2009. Convergence and Parameter Choice for Monte-Carlo Simulations of Diffusion MRI. *IEEE Transactions on Medical Imaging*, **28**(9), 1354–1364.

[Hassouna & Farag, 2005] Hassouna, M.S., & Farag, A.A. 2005. Robust centerline extraction framework using level sets. *Pages 458–465 vol. 1 of: 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 1.

[Henry Gray (1825–1861), 2024] Henry Gray (1825–1861). 2024. *Anatomy of the Human Body. 1918.* [Online; accessed 2-Feb-2024].

[Jawabri & Sharma, 2023] Jawabri, Khalid H, & Sharma, Sandeep. 2023. *Physiology, cerebral cortex functions.* StatPearls Publishing.

[Jelescu *et al.*, 2015] Jelescu, Ileana O., Veraart, Jelle, Adisetiyo, Vitria, Milla, Sarah S., Novikov, Dmitry S., & Fieremans, Els. 2015. One diffusion acquisition and different white matter models: How does microstructure change in human early development based on WMTI and NODDI? *NeuroImage*, **107**, 242–256.

[Jelescu *et al.*, 2022] Jelescu, Ileana O., de Skowronski, Alexandre, Geffroy, Françoise, Palombo, Marco, & Novikov, Dmitry S. 2022. Neurite Exchange Imaging ((NEXI): A minimal model of diffusion in gray matter with inter-compartment water exchange. *NeuroImage*, **256**(8).

[Jensen *et al.*, 2005] Jensen, Jens H, Helpern, Joseph A, Ramani, Anita, Lu, Hanzhang, & Kaczynski, Kyle. 2005. Diffusional kurtosis imaging: the quantification of non-gaussian water diffusion by means of magnetic resonance imaging. *Magnetic Resonance in Medicine: An Official Journal of the International Society for Magnetic Resonance in Medicine*, **53**(6), 1432–1440.

[Jones, 2010] Jones, PhD, Derek K. 2010. *Diffusion MRI: Theory, Methods, and Applications.* Oxford University Press.

[Jäkel & Dimou, 2017] Jäkel, Sarah, & Dimou, Leda. 2017. Glial Cells and Their Function in the Adult Brain: A Journey through the History of Their Ablation. *Frontiers in Cellular Neuroscience*, **11**.

[Kayalioglu, 2009] Kayalioglu, Gulgun. 2009. Chapter 3 - The Vertebral Column and Spinal Meninges. *Pages 17–36 of:* Watson, Charles, Paxinos, George, & Kayalioglu, Gulgun (eds), *The Spinal Cord.* San Diego: Academic Press.

[Kerkelä *et al.*, 2020] Kerkelä, Leevi, Nery, Fabio, Hall, Matt, & Clark, Chris. 2020. Disimpy: A massively parallel Monte Carlo simulator for generating diffusion-weighted MRI data in Python. *Journal of Open Source Software*, **5**(8), 2527.

[Kärger, 1985] Kärger, Jörg. 1985. NMR self-diffusion studies in heterogeneous systems. *Advances in Colloid and Interface Science*, **23**, 129–148.

[Kärger *et al.*, 1988] Kärger, Jörg, PFEIFER, HARRY, & HEINK, WILFRIED. 1988. Principles and Application of Self-Diffusion Measurements by Nuclear Magnetic Resonance. Advances in Magnetic and Optical Resonance, vol. 12. Academic Press.

[Lanciego *et al.*, 2012] Lanciego, José L, Luquin, Natasha, & Obeso, José A. 2012. Functional neuroanatomy of the basal ganglia. *Cold Spring Harb. Perspect. Med.*, **2**(12), a009621.

[Le Bihan *et al.*, 1986] Le Bihan, D, Breton, E, Lallemand, D, Grenier, P, Cabanis, E, & Laval-Jeantet, M. 1986. MR imaging of intravoxel incoherent motions: application to diffusion and perfusion in neurologic disorders. *Radiology*, **161**(2), 401–407. PMID: 3763909.

[Lee *et al.*, 2020] Lee, Hong-Hsi, Jespersen, Sune N., Fieremans, Els, & Novikov, Dmitry S. 2020. The impact of realistic axonal shape on axon diameter estimation using diffusion MRI. *NeuroImage*, **223**, 117228.

[Lee *et al.*, 2021] Lee, Hong Hsi, Fieremans, Els, & Novikov, Dmitry S. 2021. Realistic Microstructure Simulator (RMS): Monte Carlo simulations of diffusion in three-dimensional cell segmentations of microscopy images. *Journal of Neuroscience Methods*, **350**, 109018.

[Li *et al.*, 2019] Li, Jing-Rebecca, Nguyen, Van-Dang, Tran, Try Nguyen, Valdman, Jan, Trang, Cong-Bang, Nguyen, Khieu Van, Vu, Duc Thach Son, Tran, Hoang An, Tran, Hoang Trong An, & Nguyen, Thi Minh Phuong. 2019. SpinDoctor: A MATLAB toolbox for diffusion MRI simulation. *NeuroImage*, **202**, 116120.

[Liewald *et al.*, 2014] Liewald, Daniel, Miller, Robert, Logothetis, Nikos, Wagner, Hans-Joachim, & Schüz, Almut. 2014. Distribution of axon diameters in cortical white matter: an electron-microscopic study on three human brains and a macaque. *Biological Cybernetics*, **108**(5), 541–557.

[Liu *et al.*, 2007a] Liu, Shengjun, Wang, Charlie C L, Hui, Kin Chuen, Jin, Xiaogang, & Zhao, Hanli. 2007a. Ellipsoid-tree construction for solid objects. *Proceedings - SPM 2007: ACM Symposium on Solid and Physical Modeling*, **1**, 303–308.

[Liu *et al.*, 2007b] Liu, Shengjun, Jin, Xiaogang, Wang, Charlie C L, & chuen Hui, Kin. 2007b. Ellipsoidal-blob approximation of 3D models and its applications. *Computers and Graphics (Pergamon)*, **31**, 243–251.

[Long & Corfas, 2014] Long, Patrick, & Corfas, Gabriel. 2014. To learn is to myelinate. *Science*, **346**(6207), 298–299.

[Lorensen & Cline, 1987] Lorensen, William E., & Cline, Harvey E. 1987. Marching cubes: A high resolution 3D surface construction algorithm. *SIGGRAPH Comput. Graph.*, **21**(4), 163–169.

[Lundgaard *et al.*, 2014] Lundgaard, I., Osório, M.J., Kress, B.T., Sanggaard, S., & Nedergaard, M. 2014. White matter astrocytes in health and disease. *Neuroscience*, **276**, 161–173. Secrets of the CNS White Matter.

[Ma *et al.*, 2018] Ma, Ruibin, Zhao, Qingyu, Wang, Rui, Damon, James, Rosenman, Julian, & Pizer, Stephen. 2018. Deforming generalized cylinders without self-intersection by means of a parametric center curve. *Computational Visual Media*, **4**(4), 305–321.

[Magnain *et al.*, 2019] Magnain, Caroline, Augustinack, Jean C, Tirrell, Lee, Fogarty, Morgan, Frosch, Matthew P, Boas, David, Fischl, Bruce, & Rockland, Kathleen S. 2019. Colocalization of neurons in optical coherence microscopy and Nissl-stained histology in Brodmann's area 32 and area 21. *Brain Struct. Funct.*, **224**(1), 351–362.

[Mandonnet *et al.*, 2018] Mandonnet, Emmanuel, Sarubbo, Silvio, & Petit, Laurent. 2018. The Nomenclature of Human White Matter Association Pathways: Proposal for a Systematic Taxonomic Anatomical Classification. *Frontiers in Neuroanatomy*, **12**.

[Nazeri *et al.*, 2020] Nazeri, Arash, Schifani, Christin, Anderson, John A.E., Ameis, Stephanie H., & Voineskos, Aristotle N. 2020. In Vivo Imaging of Gray Matter Microstructure in Major Psychiatric Disorders: Opportunities for Clinical Translation. *Biological Psychiatry: Cognitive Neuroscience and Neuroimaging*, **5**(9), 855–864.

[Nedjati-Gilani *et al.*, 2014] Nedjati-Gilani, Gemma L., Schneider, Torben, Hall, Matt G., Wheeler-Kingshott, Claudia A. M., & Alexander, Daniel C. 2014. Machine Learning Based Compartment Models with Permeability for White Matter Microstructure Imaging. *Pages 257–264 of:* Golland, P, Hata, N, Barillot, C, Hornegger, J, & Howe, R (eds), *MEDICAL IMAGE COMPUTING AND COMPUTER-ASSISTED INTERVENTION - MICCAI 2014, PT III.* Lecture Notes in Computer Science, vol. 8675. HEIDELBERGER PLATZ 3, D-14197 BERLIN, GERMANY: SPRINGER-VERLAG BERLIN, for Harvard Med Sch. 17th International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI), Massachusetts Inst Technol, Boston, MA, SEP 14-18, 2014.

[Neher *et al.*, 2014] Neher, Peter F., Laun, Frederik B., Stieltjes, Bram, & Maier-Hein, Klaus H. 2014. Fiberfox: Facilitating the creation of realistic white matter software phantoms. *Magnetic Resonance in Medicine*, **72**(5), 1460–1470.

[Neuman, 1974] Neuman, C. H. 1974. Spin echo of spins diffusing in a bounded medium. *The Journal of Chemical Physics*, **60**(11), 4508–4511.

[Nguyen, 2014] Nguyen, Dang Van. 2014 (Mar.). *A finite elements method to solve the Bloch-Torrey equation applied to diffusion magnetic resonance imaging of biological tissues.* Theses, Ecole Polytechnique X.

[Nguyen *et al.*, 2018] Nguyen, Khieu-Van, Hernández-Garzón, Edwin, & Valette, Julien. 2018. Efficient GPU-based Monte-Carlo simulation of diffusion in real astrocytes reconstructed from confocal microscopy. *Journal of Magnetic Resonance*, **296**, 188–199.

[Nieuwenhuys *et al.*, 2007] Nieuwenhuys, R., Voogd, J., & van Huijzen, C. 2007. *The Human Central Nervous System: A Synopsis and Atlas.* Steinkopff.

[Nilsson *et al.*, 2017] Nilsson, Markus, Lasič, Samo, Drobnjak, Ivana, Topgaard, Daniel, & Westin, Carl-Fredrik. 2017. Resolution limit of cylinder diameter estimation by diffusion MRI:

The impact of gradient waveform and orientation dispersion. *NMR in Biomedicine*, **30**(7), e3711. e3711 nbm.3711.

[Novikov *et al.*, 2019] Novikov, Dmitry S., Fieremans, Els, Jespersen, Sune N., & Kiselev, Valerij G. 2019. Quantifying brain microstructure with diffusion MRI: Theory and parameter estimation. *NMR in Biomedicine*, **32**, 1–53.

[Padamsey & Rochefort, 2023] Padamsey, Zahid, & Rochefort, Nathalie L. 2023. Paying the brain's energy bill. *Current Opinion in Neurobiology*, **78**, 102668.

[Palombo *et al.*, 2019] Palombo, Marco, Alexander, Daniel C., & Zhang, Hui. 2019. A generative model of realistic brain cells with application to numerical simulation of the diffusion-weighted MR signal. *NeuroImage*, **188**, 391–402.

[Palombo *et al.*, 2020] Palombo, Marco, Ianus, Andrada, Guerreri, Michele, Nunes, Daniel, Alexander, Daniel C, Shemesh, Noam, & Zhang, Hui. 2020. SANDI: A compartment-based model for non-invasive apparent soma and neurite imaging by diffusion MRI. *NeuroImage*, **215**, 116835.

[Palombo *et al.*, 2021] Palombo, Marco, Alexander, Daniel, & Zhang, Hui. 2021 (05). Large-scale analysis of brain cell morphometry informs microstructure modelling of gray matter. *In: Large-scale analysis of brain cell morphometry informs microstructure modelling of gray matter*.

[Paus & Toro, 2009] Paus, Tomas, & Toro, Roberto. 2009. Could Sex Differences in White Matter be Explained by g ratio? *Frontiers in neuroanatomy*, **3**(09), 14.

[Purcell *et al.*, 1946] Purcell, E. M., Torrey, H. C., & Pound, R. V. 1946. Resonance Absorption by Nuclear Magnetic Moments in a Solid. *Phys. Rev.*, **69**(Jan), 37–38.

[Rabi *et al.*, 1938] Rabi, I. I., Zacharias, J. R., Millman, S., & Kusch, P. 1938. A New Method of Measuring Nuclear Magnetic Moment. *Phys. Rev.*, **53**(Feb), 318–318.

[Rafael-Patino *et al.*, 2020] Rafael-Patino, Jonathan, Romascano, David, Ramirez-Manzanares, Alonso, Canales-Rodríguez, Erick Jorge, Girard, Gabriel, & Thiran, Jean-Philippe. 2020. Robust Monte-Carlo Simulations in Diffusion-MRI: Effect of the Substrate Complexity and Parameter Choice on the Reproducibility of Results. *Frontiers in Neuroinformatics*, **14**.

[Rossi *et al.*, 2021] Rossi, Alessandro, Barbiero, Marco, Scremin, Paolo, & Carli, Ruggero. 2021. Robust Visibility Surface Determination in Object Space via Plücker Coordinates. *Journal of Imaging*, **7**(06), 96.

[Schlömer, n.d.] Schlömer, Nico. *meshio: Tools for mesh files*.

[Sepehrband *et al.*, 2016] Sepehrband, Farshid, Alexander, Daniel C., Clark, Kristi A., Kurniawan, Nyoman D., Yang, Zhengyi, & Reutens, David C. 2016. Parametric Probability Distribution Functions for Axon Diameters of Corpus Callosum. *Frontiers in Neuroanatomy*, **10**.

[Servier Medical Art, 2024] Servier Medical Art. 2024. *Drawing arteries, brain inferior view*. [Online; accessed 19-Feb-2024].

[Shao *et al.*, 2021] Shao, Han, Kugelstadt, Tassilo, Hädrich, Torsten, Palubicki, Wojtek, Bender, Jan, Pirk, Soeren, & Michels, Dominik L. 2021. Accurately Solving Rod Dynamics with Graph Learning. *Pages 4829–4842 of:* Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P.S., & Vaughan, J. Wortman (eds), *Advances in Neural Information Processing Systems*, vol. 34. Curran Associates, Inc.

[Shapson-Coe *et al.*, 2021] Shapson-Coe, Alexander, Januszewski, Michał, Berger, Daniel R, Pope, Art, Wu, Yuelong, Blakely, Tim, Schalek, Richard L, Li, Peter, Wang, Shuohong, Maitin-Shepard, Jeremy, *et al.* 2021. A connectomic study of a petascale fragment of human cerebral cortex. *bioRxiv*.

[Shapson-Coe *et al.*, 2024] Shapson-Coe, Alexander, Januszewski, Michał, Berger, Daniel R., Pope, Art, Wu, Yuelong, Blakely, Tim, Schalek, Richard L., Li, Peter H., Wang, Shuohong, Maitin-Shepard, Jeremy, Karlupia, Neha, Dorkenwald, Sven, Sjostedt, Evelina, Leavitt, Laramie, Lee, Dongil, Troidl, Jakob, Collman, Forrest, Bailey, Luke, Fitzmaurice, Angerica, Kar, Rohin, Field, Benjamin, Wu, Hank, Wagner-Carena, Julian, Aley, David, Lau, Joanna, Lin, Zudi, Wei, Donglai, Pfister, Hanspeter, Peleg, Adi, Jain, Viren, & Lichtman, Jeff W. 2024. A petavoxel fragment of human cerebral cortex reconstructed at nanoscale resolution. *Science*, **384**(6696), eadk4858.

[Stanisz *et al.*, 1997] Stanisz, Greg J., Wright, Graham A., Henkelman, R. Mark, & Szafer, Aaron. 1997. An analytical model of restricted diffusion in bovine optic nerve. *Magnetic Resonance in Medicine*, **37**(1), 103–111.

[Stejskal, 1965] Stejskal, E. O. 1965. Use of spin echoes in a pulsed magnetic-field gradient to study anisotropie, restricted diffusion and flow. *The Journal of Chemical Physics*, **43**, 3597–3603.

[Stejskal & Tanner, 1965] Stejskal, E.O., & Tanner, J.E. 1965. Spin Diffusion Measurements: Spin Echoes in the Presence of a Time-Dependent Field Gradient. *Journal of Chemical Physics*, **42**(1), 288–292.

[Stepišnik, 1981] Stepišnik, Janez. 1981. Analysis of NMR self-diffusion measurements by a density matrix calculation. *Physica B+ C*, **104**(3), 350–364.

[Swaab *et al.*, 2012] Swaab, Dick F., Bao, Ai-Min, Garcia-Falgueras, Alicia, Hofman, Michel A., & Ishunina, Tatjana A. 2012. Chapter 21 - Sex Differences in the Forebrain. *Pages 739–758 of:* Mai, Jürgen K., & Paxinos, George (eds), *The Human Nervous System (Third Edition)*, 3rd edn. San Diego: Academic Press.

[Szczepankiewicz *et al.*, 2021] Szczepankiewicz, Filip, Westin, Carl-Fredrik, & Nilsson, Markus. 2021. Gradient waveform design for tensor-valued encoding in diffusion MRI. *Journal of Neuroscience Methods*, **348**, 109007.

[SÖDERMAN & JÖNSSON, 1995] SÖDERMAN, OLLE, & JÖNSSON, BENGT. 1995. Restricted Diffusion in Cylindrical Geometry. *Journal of Magnetic Resonance, Series A*, **117**(1), 94–97.

[Takahashi *et al.*, 2007] Takahashi, Tomokazu, Lina, Ide, Ichiro, Mekada, Yoshito, & Murase, Hiroshi. 2007. Interpolation Between Eigenspaces Using Rotation in Multiple Dimensions. *Pages 774–783 of:* Yagi, Yasushi, Kang, Sing Bing, Kweon, In So, & Zha, Hongbin (eds), *Computer Vision – ACCV 2007*. Berlin, Heidelberg: Springer Berlin Heidelberg.

[Torrey, 1956] Torrey, H. C. 1956. Bloch Equations with Diffusion Terms. *Phys. Rev.*, **104**(Nov), 563–565.

[Tramo *et al.*, 1995] Tramo, Mark Jude, Loftus, William C., Thomas, Catherine E., Green, Ronald L., Mott, Leila A., & Gazzaniga, Michael S. 1995. Surface Area of Human Cerebral Cortex and Its Gross Morphological Subdivisions: In Vivo Measurements in Monozygotic Twins Suggest Differential Hemisphere Effects of Genetic Factors. *Journal of Cognitive Neuroscience*, **7**(2), 292–302.

[Trott *et al.*, 2022] Trott, Christian R., Lebrun-Grandié, Damien, Arndt, Daniel, Ciesko, Jan, Dang, Vinh, Ellingwood, Nathan, Gayatri, Rahulkumar, Harvey, Evan, Hollman, Daisy S., Ibanez, Dan, Liber, Nevin, Madsen, Jonathan, Miles, Jeff, Poliakoff, David, Powell, Amy, Rajamanickam, Sivasankaran, Simberg, Mikael, Sunderland, Dan, Turcksin, Bruno, & Wilke, Jeremiah. 2022. Kokkos 3: Programming Model Extensions for the Exascale Era. *IEEE Transactions on Parallel and Distributed Systems*, **33**(4), 805–817.

[Tuch, 2004] Tuch, David S. 2004. Q-ball imaging. *Magnetic Resonance in Medicine*, **52**(6), 1358–1372.

[Vasile *et al.*, 2017] Vasile, Flora, Dossi, Elena, & Rouach, Nathalie. 2017. Human astrocytes: structure and functions in the healthy brain. *Brain Structure and Function*, **222**(5), 2017–2029.

[Villarreal-Haro *et al.*, 2023] Villarreal-Haro, Juan Luis, Gardier, Remy, Canales-Rodríguez, Erick J., Fischi-Gomez, Elda, Girard, Gabriel, Thiran, Jean-Philippe, & Rafael-Patiño, Jonathan. 2023. CACTUS: a computational framework for generating realistic white matter microstructure substrates. *Frontiers in Neuroinformatics*, **17**.

[Vogt, 1910] Vogt, Oskar. 1910. Le champ myéloarchitectonique du lobe frontal humain. *J Psychol Neurol*, **15**(4/5), 221–232.

[von Bartheld *et al.*, 2016] von Bartheld, Christopher S, Bahney, Jami, & Herculano-Houzel, Suzana. 2016. The search for true numbers of neurons and glial cells in the human brain: A review of 150 years of cell counting. *J. Comp. Neurol.*, **524**(18), 3865–3895.

[Walsh, 2016] Walsh, Corey. 2016. *Cuda-Accelerated Voxelizer*. [Online; accessed 29-February-2024].

[Wang *et al.*, 2006] Wang, Rui, Zhou, Kun, Snyder, John, Liu, Xinguo, Bao, Hujun, Peng, Qunsheng, & Guo, Baining. 2006. Variational sphere set approximation for solid objects. *The Visual Computer*, **22**(9), 612–621.

[Waxman, 1980] Waxman, Stephen G. 1980. Determinants of conduction velocity in myelinated nerve fibers. *Muscle & Nerve*, **3**.

[Webb, 2017] Webb, Wanda G. 2017. 2 - Organization of the Nervous System I. *Pages 13–43 of:* Webb, Wanda G. (ed), *Neurology for the Speech-Language Pathologist (Sixth Edition)*, 6th edn. Mosby.

[Wedeen *et al.*, 2005a] Wedeen, Van J., Hagmann, Patric, Tseng, Wen-Yih Isaac, Reese, Timothy G., & Weisskoff, Robert M. 2005a. Mapping complex tissue architecture with diffusion spectrum magnetic resonance imaging. *Magnetic Resonance in Medicine*, **54**(6), 1377–1386.

[Wedeen *et al.*, 2005b] Wedeen, Van J., Hagmann, Patric, Tseng, Wen-Yih Isaac, Reese, Timothy G., & Weisskoff, Robert M. 2005b. Mapping complex tissue architecture with diffusion spectrum magnetic resonance imaging. *Magnetic Resonance in Medicine*, **54**(6), 1377–1386.

[Wedeen *et al.*, 2000] Wedeen, VJ, Reese, TG, Tuch, DS, Weigel, MR, Dou, JG, Weiskoff, RM, & Chessler, D. 2000. Mapping fiber orientation spectra in cerebral white matter with Fourier-transform diffusion MRI. *Page 82 of: Proceedings of the 8th Annual Meeting of ISMRM, Denver.*

[Wikipedia contributors, 2024a] Wikipedia contributors. 2024a. *Afferent nerve fiber.* [Online; accessed 2-Feb-2024].

[Wikipedia contributors, 2024b] Wikipedia contributors. 2024b. *Brodmann Areas.* [Online; accessed 23-Feb-2024].

[Wikipedia contributors, 2024c] Wikipedia contributors. 2024c. *Circle of Willis.* [Online; accessed 19-Feb-2024].

[Wikiversity, 2020] Wikiversity. 2020. *WikiJournal of Medicine/Medical gallery of Blausen Medical 2014 — Wikiversity,.* [Online; accessed 13-August-2020].

[Wright *et al.*, 2012] Wright, Ben, Lai, James, & Sinclair, Alexandra. 2012. Cerebrospinal fluid and lumbar puncture: A practical review. *Journal of neurology*, **259**(01), 1530–45.

[Wu *et al.*, 2015] Wu, Yuwen, Dissing-Olesen, Lasse, MacVicar, Brian A., & Stevens, Beth. 2015. Microglia: Dynamic Mediators of Synapse Development and Plasticity. *Trends in Immunology*, **36**(10), 605–613. Special Issue: Neuroimmunology.

[Xing *et al.*, 2013] Xing, Haoyang, Lin, Fang, Wu, Qizhu, & Gong, Qiyong. 2013. Investigation of different boundary treatment methods in Monte-Carlo simulations of diffusion NMR. *Magnetic Resonance in Medicine*, **70**(4), 1167–1172.

[Yebga Hot, 2021] Yebga Hot, Raïssa. 2021 (Sept.). *Ultra-high field diffusion and quantitative MRI with strong gradients to explore the connectivity, cytoarchitecture and myeloarchitecture of animal and human brains at the mesoscale.* Theses, Université Paris-Saclay.

[Yeh, 2011] Yeh, Chun Hung. 2011 (Sept.). *Diffusion Microscopist Simulator - The Development and Application of a Monte Carlo Simulation System for Diffusion MRI.* Theses, Université Paris Sud - Paris XI ; National Yang-Ming University (Taiwan).

[Yeh *et al.*, 2013] Yeh, Chun-Hung, Schmitt, Benoît, Le Bihan, Denis, Li-Schlittgen, Jing-Rebecca, Lin, Ching-Po, & Poupon, Cyril. 2013. Diffusion Microscopist Simulator: A General Monte Carlo Simulation System for Diffusion Magnetic Resonance Imaging. *PLOS ONE*, **8**(10), 1–12.

[Yuste *et al.*, 2020] Yuste, Rafael, Hawrylycz, Michael, Aalling, Nadia, Aguilar-Valles, Argel, Arendt, Detlev, Armañanzas, Ruben, Ascoli, Giorgio A., Bielza, Concha, Bokharaie, Vahid, Bergmann, Tobias Borgtoft, Bystron, Irina, Capogna, Marco, Chang, YoonJeung, Clemens, Ann, de Kock, Christiaan P. J., DeFelipe, Javier, Dos Santos, Sandra Esmeralda, Dunville, Keagan, Feldmeyer, Dirk, Fiáth, Richárd, Fishell, Gordon James, Foggetti, Angelica, Gao, Xuefan, Ghaderi, Parviz, Goriounova, Natalia A., Güntürkün, Onur, Hagihara, Kenta, Hall, Vanessa Jane, Helmstaedter, Moritz, Herculano-Houzel, Suzana, Hilscher, Markus M., Hirase, Hajime, Hjerling-Leffler, Jens, Hodge, Rebecca, Huang, Josh, Huda, Rafiq, Khodosevich, Konstantin, Kiehn, Ole, Koch, Henner, Kuebler, Eric S., Kühnemund, Malte, Larrañaga, Pedro, Lelieveldt, Boudewijn, Louth, Emma Louise, Lui, Jan H., Mansvelder, Huibert D., Marin, Oscar, Martinez-Trujillo, Julio, Chameh, Homeira Moradi, Mohapatra, Alok Nath, Munguba, Hermany, Nedergaard, Maiken, Němec, Pavel, Ofer, Netanel, Pfisterer, Ulrich Gottfried, Pontes, Samuel, Redmond, William, Rossier, Jean, Sanes, Joshua R., Scheuermann, Richard H., Serrano-Saiz, Esther, Staiger, Jochen F., Somogyi, Peter, Tamás, Gábor, Tolias, Andreas Savas, Tosches, Maria Antonietta, García, Miguel Turrero, Wozny, Christian, Wuttke, Thomas V., Liu, Yong, Yuan, Juan, Zeng, Hongkui, & Lein, Ed. 2020. A community-based transcriptomics classification and nomenclature of neocortical cell types. *Nature Neuroscience*, **23**(12), 1456–1468.

[Zeng & Sanes, 2017] Zeng, Hongkui, & Sanes, Joshua R. 2017. Neuronal cell-type classification: challenges, opportunities and the path forward. *Nature Reviews Neuroscience*, **18**(9), 530–546.

[Zhang *et al.*, 2012] Zhang, Hui, Schneider, Torben, Wheeler-Kingshott, Claudia A., & Alexander, Daniel C. 2012. NODDI: Practical in vivo neurite orientation dispersion and density imaging of the human brain. *NeuroImage*, **61**(4), 1000–1016.

[Zhang *et al.*, 2023] Zhang, Wendiao, Jiang, Jiamei, Xu, Zhenhong, Yan, Hongye, Tang, Beisha, Liu, Chunyu, Chen, Chao, & Meng, Qingtuan. 2023. Microglia-containing human brain organoids for the study of brain development and pathology. *Molecular Psychiatry*, **28**(1), 96–107.

[Zhou *et al.*, 2015] Zhou, Yang, Yin, Kangxue, Huang, Hui, Zhang, Hao, Gong, Minglun, & Cohen-Or, Daniel. 2015. Generalized Cylinder Decomposition. *ACM Transactions on Graphics (Proc. of SIGGRAPH Asia)*, **34**(6), 171:1–171:14.

[Özarslan *et al.*, 2013] Özarslan, Evren, Koay, Cheng Guan, Shepherd, Timothy M., Komlosh, Michal E., İrfanoğlu, M. Okan, Pierpaoli, Carlo, & Basser, Peter J. 2013. Mean apparent propagator (MAP) MRI: A novel diffusion imaging method for mapping tissue microstructure. *NeuroImage*, **78**, 16–32.