Some contributions towards the parallel simulation of time dependent neutron transport and the integration of observed data in real time

Olga Mula Hernandez

Quelques contributions vers la simulation parallèle de la cinétique neutronique et la prise en compte de données observées en temps réel.

THÈSE DE DOCTORAT

présentée par

Olga MULA HERNÁNDEZ

pour obtenir le grade de

Docteur de l'Université Pierre et Marie Curie

Spécialité

Mathématiques Appliquées

sous la direction d'Yvon Maday

Soutenue publiquement le 24/09/2014 devant le jury composé de

M. Albert Cohen	UPMC	Examinateur
M. Wolfgang Dahmen	RWTH	Examinateur
Mme. Laurence Halpern	Université Paris XIII	Rapportrice
M. Jan Hesthaven	EPFL	Rapporteur
M. Jean-Jacques Lautard	CEA	Examinateur
M. Yvon Maday	UPMC	Directeur de thèse

Thèse effectuée aux :

Laboratoire Jacques-Louis Lions, UMR 7598

Adresse géographique :
Laboratoire Jacques Louis Lions
Bâtiments : 3ème étage – 15-16, 15-25, 16-26,
4 place Jussieu
75005 Paris, France
+33 (0)1 44 27 42 98 (Tél.)

Adresse postale :
Laboratoire Jacques-Louis Lions
Université Pierre et Marie Curie
Boîte courrier 187
75252 Paris Cedex 05 France

CEA Saclay

Adresse :
CEA Saclay
DEN/DANS/DM2S/SERMA/LLPR
91191 Gif-sur-Yvette CEDEX
France
+33 (0)1 64 50 10 00 (Tél.)

Quelques contributions vers la simulation parallèle de la cinétique neutronique et la prise en compte de données observées en temps réel.

Résumé

Dans cette thèse nous avons tout d'abord développé un solveur neutronique de cinétique transport 3D en géométrie déstructurée avec une discrétisation spatiale par éléments finis (solveur MINARET). L'écriture d'un tel code représente en soi une contribution importante dans la physique des réacteurs car il permettra de connaître de façon très précise l'état du coeur au cours d'accidents graves. Il jouera aussi un rôle très important pour des études de fluence de la cuve des réacteurs. D'un point de vue mathématique, l'apport le plus important dans l'écriture de ce solveur a consisté en l'implémentation d'algorithmes modernes adaptés aux architectures actuelles et à venir de calcul parallèle, permettant de réduire de façon significative les temps de calcul. Un effort particulier a été mené pour paralléliser de façon efficace la variable temporelle par l'algorithme pararéel en temps. Ce travail a consisté dans un premier temps à analyser les performances que le schéma classique de pararéel apporte dans la résolution de l'équation de transport de neutrons. Ensuite, nous avons cherché à améliorer ces performances en proposant un schéma de pararéel qui intègre de façon plus optimisée la présence de schémas itératifs autres que le pararéel dans la résolution de chaque pas de temps de l'équation du transport. L'idée principale de ce nouveau schéma consiste à limiter le nombre d'itérations internes pour chaque pas de temps du solveur fin et d'atteindre la convergence au cours des itérations pararéelles.

Dans un second temps, une réflexion a été entamée autour de la question suivante : étant donné le haut degré de précision que MINARET fournit dans la connaissance de la population neutronique, serait-il possible de l'utiliser en tant qu'outil de surveillance pendant l'opération d'un réacteur nucléaire ? Et, qui plus est, comment rendre un tel outil à la fois cohérent et complémentaire par rapport aux mesures prises *in situ* ? Une des difficultés majeures de ce problème réside dans le besoin de fournir les simulations en temps réel alors que, malgré nos efforts pour accélérer les calculs, les méthodes de discrétisation utilisées dans MINARET ne permettent pas des calculs de coeur à une telle vitesse.

Cette question a été abordée en développant tout d'abord une généralisation de la méthode Empirical Interpolation (EIM) grâce à laquelle on a pu définir un processus d'interpolation bien posé pour des fonctions appartenant à des espaces de Banach. Ceci est rendu possible par l'utilisation de formes linéaires d'interpolation au lieu des traditionnels points d'interpolation et une partie de cette thèse a été consacrée à la compréhension des propriétés théoriques de cette méthode (analyse de convergence sous hypothèse d'ensemble de petite dimension de Kolmogorov et étude de sa stabilité). Ce processus d'interpolation (appelé Generalized EIM) permet de reconstruire en temps réel des processus physiques de la façon suivante : étant donné un système pouvant être décrit par une EDP paramétrée et sur lequel des mesures peuvent être prises *in situ*, on construit d'abord une base d'interpolation constituée de solutions de cette EDP pour différentes valeurs du paramètre grâce à GEIM (ceci est fait par un algorithme greedy). On donne ensuite une approximation en temps réel de l'état du système via une fonction interpolée exprimée dans la base calculée et qui utilise des mesures acquises *in situ* comme données d'entrée (et modélisées mathématiquement par les formes linéaires). La méthode a été appliquée avec succès dans des exemples simples (équations de Laplace et de Stokes) et nous espérons que les développements actuels et à venir pourront mener à son emploi dans des cas réels plus complexes comme celui de la reconstruction de la population neutronique dans un coeur de réacteur avec MINARET.

Some contributions towards the parallel simulation of time dependent neutron transport and the integration of observed data in real time.

Abstract

In this thesis, we have first developed a time dependent 3D neutron transport solver on unstructured meshes with discontinuous Galerkin finite elements spatial discretization. The solver (called MINARET) represents in itself an important contribution in reactor physics thanks to the accuracy that it can provide in the knowledge of the state of the core during severe accidents. It will also play an important role on vessel fluence calculations. From a mathematical point of view, the most important contribution has consisted in the implementation of modern algorithms that are well adapted for modern parallel architectures and that significantly decrease the computing times. A special effort has been done in order to efficiently parallelize the time variable by the use of the parareal in time algorithm. For this, we have first analyzed the performances that the classical scheme of parareal can provide when applied to the resolution of the neutron transport equation in a reactor core. Then, with the purpose of improving these performances, a parareal scheme that takes more efficiently into account the presence of other iterative schemes in the resolution of each time step has been proposed. The main idea consists in limiting the number of internal iterations for each time step and to reach convergence across the parareal iterations.

A second phase of our work has been motivated by the following question: given the high degree of accuracy that MINARET can provide in the modeling of the neutron population, could we somehow use it as a tool to monitor in real time the population of neutrons on the purpose of helping in the operation of the reactor? And, what is more, how to make such a tool be coherent in some sense with the measurements taken *in situ*? One of the main challenges of this problem is the real time aspect of the simulations. Indeed, despite all of our efforts to speed-up the calculations, the discretization methods used in MINARET do not provide simulations at such a speed.

This question has been addressed by proposing an extension of the Empirical Interpolation Method (EIM) thanks to which a well-posed interpolation procedure has been defined for functions belonging to Banach spaces. This is possible thanks to the use of interpolating linear forms instead of the traditional interpolation points and a part of this thesis has been devoted to the understanding of the theoretical properties of this method (convergence analysis under the hypothesis of small Kolmogorov n-width and stability of the procedure). This interpolation process (called GEIM as for Generalized EIM) can be used to reconstruct in real time physical processes in the following manner: given a system that can be described by a parameter dependent PDE and over which measurements can be taken *in situ*, we start by building with GEIM an interpolation basis spanned by solutions of the PDE for different parameter values. This is performed by a greedy algorithm. Then, a real-time approximation of the state of the system is provided through the computation of an interpolating function expressed in the interpolating basis and that uses measurements (mathematically modelled by linear forms) acquired *in situ* as an input. This method has been successfully applied in simple cases (involving parameter dependent Laplace and Stokes equations) and we expect that the present developments will allow its use in more realistic and complex cases in the future, like the one of the reconstruction of the neutron population in a reactor core with MINARET.

Acknowledgments

Many friends have helped me in writing this book.
Virginia Woolf, Preface of Orlando.

Here am I, writing the acknowledgments of my PhD thesis. This is a strong sign that means that this beautiful period of my life is coming to an end. It is therefore with some regret that I write this lines, in order to thank the many people that have made, in various ways, this PhD to be such an extraordinary journey to me.

First and foremost, the very biggest thanks go to my supervisor, Yvon Maday, at the Laboratoire Jacques Louis Lions (LJLL). Without his guidance, patience, enthusiasm, generosity and, above all, incredible scientific knowledge and inexhaustible mathematical imagination, none of this would have been possible. Despite an agenda more complicated than the one of any minister, Yvon always made time for a meeting and managed to check my work. Yvon, I have been incredibly fortunate to have you as a supervisor and I am deeply grateful for all the support you have given me.

A huge thank you goes also to my industrial supervisor, Jean-Jacques Lautard, at CEA. With Jean-Jacques, I have not only worked with an outstanding researcher in neutronic computations but also with an extraordinary person. He always found time to kindly answer my never-ending lists of questions about neutronics and the MINARET code, for which I am most grateful. Jean-Jacques, I have also been extremely fortunate to have had the chance to have such a deep insight into neutronics thanks to you.

I am also very grateful to the other people that I have had the pleasure of collaborating with during this PhD. I sincerely hope that this is just the beginning of future works together. Thank you very much to Anne-Marie Baudron (and her exceptional programming skills that have saved my life in MINARET more than once). Thank you also to Gabriel Turinici: without your support, our analysis on the convergence rates of GEIM would never have been as accurate as it is now. Many thanks to Tony Patera and Masayuki Yano: it has been a real honor to collaborate with both of you during Tony's research stay at LJLL. It has also been a pleasure to work with with Benjamin Stamm in our "rectification bussiness". Thank you also to Mohamed Kamel Riahi and his important efforts to make our "para-iterations scheme" work in practice. Finally, I would like to thank Professor W. Dahmen and the members of RTWH for hosting me as a visitor in the very last months of this PhD.

I have had the opportunity to work in two outstanding different places (CEA/../SERMA/LLPR and LJLL), each welcoming and stimulating in their own way. I would like to thank both institutions and, in particular, the CEA for their generosity towards me concerning all the missions I have been involved in. I am very aware of the efforts you have made for me in this respect and I have always done my best to reach the heights of your expectations in this work. Thank you also to the administrative teams of LLPR (Jocelyne) and LJLL (Salima, Nadine and others).

Thank you to L'Oréal-UNESCO, not only for having honored me by awarding the works presented in this PhD, but also to have made me think more deeply about the importance of the presence of women in science.

I would also like to thank all the people that have been on my side during these three years of work. Thank you to Erell, Sébastien and Paul, for all your support, advices and help in any circumstances. Thank you also to all the members of our "Piscine Team" and, in particular, thank you to Dider, our beloved "Chef de l'Eau", for making possible our swimming sessions. Thank you to our "Pétanque Group" and especially to my patient team-mate Frédéric. I have also apreciated very much our afterworks between PhD students: thank you Karim, Clélia (never forget to do your "special Clélia" from time to time), Amélie (our beer expert), king Arthur, Kevin (my personal French to English translator), France and many others for creating such a nice atmosphere inside and outside our working hours.

Very special thanks to all my lifelong friends in Spain (Pilar, Rocío, Marco, Héctor, Ali, Maehtro Pablo...) and also to all my friends from Polytechnique (Alfredo, Claire, Hélène, Martin, Anna: your support has been very important for me).

Last, but certainly not least, I would like to thank my parents and the rest of my family for the continuous encouragement, unlimited patience and daily support. Mamá, te dedico esta tesis a ti y a la memoria de Papá, que siempre estará con nosotras.

A mis padres, Nati y Luis.

Revolutionär wird der sein,
der sich selbst revolutionieren
kann.

Ludwig Wittgenstein,
Vermischte Bemerkungen.

Contents

Introduction (Version française)

Motivations de ce travail

Tout comme les centrales à charbon, gaz, solaires thermiques ou d'incinération de déchets, les centrales nucléaires appartiennent à la famille des centrales thermiques. Dans l'objectif de produire de l'électricité, ces centrales chauffent un caloporteur (à l'état liquide ou gazeux) afin de générer de la vapeur dans des conditions thermodynamiquement adéquates pour sa détente dans une turbine. Ceci entraîne la rotation de la turbine, qui, grâce à son couplage avec un alternateur, permet de transformer l'énergie de rotation en énergie électrique. Suite à son passage par la turbine, la vapeur passe par un condenseur et est recirculée vers la source de chaleur.

La particularité des centrales nucléaires réside dans le fait que la source de chaleur est un réacteur nucléaire, c'est à dire que le caloporteur est chauffé par des réactions de fission nucléaire dans un endroit appelé le coeur du réacteur (dont un exemple est donné en figure 1). Au sein du coeur, des neutrons libres collisionnent avec des particules fissiles (comme par exemple des particules de U^{235} ou de Pu^{239}) qui sont contenues dans ce que l'on appelle le combustible nucléaire. Sous des conditions appropriées, la collision d'un neutron avec une particule fissile peut scinder la particule en deux atomes plus légers et relâcher 2 ou 3 neutrons libres dans le milieu (cf. formule (1) pour un exemple de réaction de fission). Ce type de réaction est exothermique (environ 200 MeV sont relâchés dans le milieu lors de la fission d'une particule de U^{235}) et constitue la source de chaleur pour le caloporteur (qui est de l'eau dans la plupart des cas).

$$\mathrm{U}_{92}^{235} + \mathrm{n}_{0}^{1} \longrightarrow \mathrm{Kr}_{36}^{92} + \mathrm{Ba}_{56}^{141} + 3\mathrm{n}_{0}^{1} \tag{1}$$

Les neutrons libérés dans le milieu peuvent, à leur tour, collisionner avec le combustible et engendrer de nouvelles fissions, d'où le phénomène de réaction en chaîne. Par ailleurs, les particules plus légères issues de la fission naissent en général dans un état excité et subissent des réactions de désexcitation pour atteindre un état plus stable. Certaines de ces réactions peuvent provoquer le relâchement de nouveaux neutrons libres dans le milieu et alimentent eux aussi la réaction en chaîne. Il semble donc clair que le contrôle de la population de neutrons libres est crucial afin de préserver la sûreté et la qualité du processus : une croissance incontrôlée de cette population pourrait conduire à un échauffement excessif du coeur, ce qui peut se transformer en une situation dangereuse pouvant mener à un accident. A l'inverse, la population de neutrons ne doit pas non plus être trop basse, car la chaleur (et donc l'électricité) produite ne serait pas suffisante pour satisfaire la demande électrique. Dans ce contexte, les simulations numériques jouent un rôle important pour la recherche de configurations sûres de coeurs de réacteurs et aussi pour la compréhension d'éventuelles situations accidentelles. Bien qu'il existe de nombreux phénomènes couplés qu'il est important de comprendre dans leur globalité (thermohydraulique, transfert thermique, irradiation et dommage de matériaux...), la parcelle de l'étude dédiée à la population neutronique est analysée par la neutronique et ce travail est une contribution dans ce domaine.

A un certain instant t, l'état d'un neutron de masse m peut être décrit par sa position spatiale $\boldsymbol{r}$

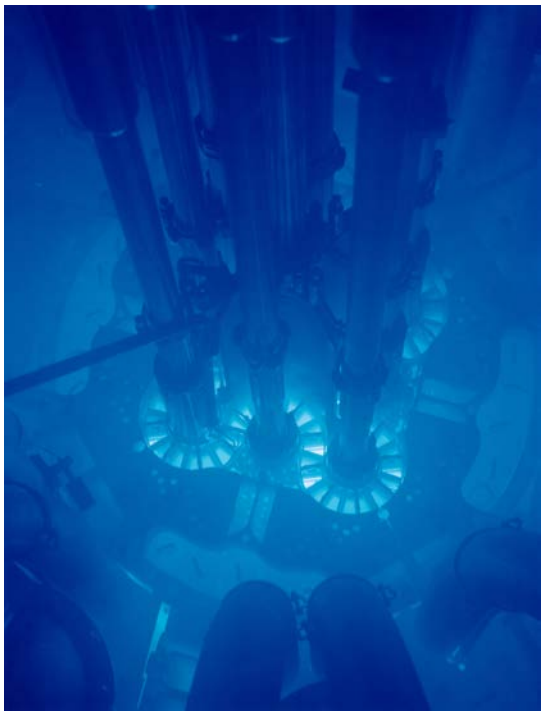


Figure 1 – Un exemple de coeur de réacteur nucléaire : l'Advanced Test Reactor core (Idaho National Laboratory)... avec l'hypnothique lumière bleue due à l'effet Cherenkov.

et sa vitesse $\boldsymbol{v}$, ou, de façon équivalente, par sa position $\boldsymbol{r}$, son énergie cinétique $E = m|\boldsymbol{v}|^2/2$ et sa direction de propagation $\boldsymbol{\omega} = \boldsymbol{v}/|\boldsymbol{v}|$. Nous souhaitons déterminer la densité $n(t, \boldsymbol{r}, \boldsymbol{v}) \equiv n(t, \boldsymbol{r}, \boldsymbol{\omega}, E)$ de neutrons libres par unité de temps et par unité de volume dans l'espace des positions et des vitesses. De façon équivalente, nous pouvons chercher à déterminer le flux angulaire $\psi(t, \boldsymbol{r}, \boldsymbol{v}) = |\boldsymbol{v}|n(t, \boldsymbol{r}, \boldsymbol{v}) \equiv \psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E)$, qui décrit aussi cette population. Comme il sera expliqué au chapitre 1, $\psi(t, \boldsymbol{r}, \boldsymbol{v})$ est la solution d'une équation de Boltzmann linéaire qui représente un bilan entre les neutrons libres qui apparaissent ou disparaissent dans le coeur suite à des réactions nucléaires. Bien que l'on pourrait dire que les fondements théoriques de cette équation sont de nos jours bien établis, il n'en va pas de même en ce qui concerne sa résolution numérique, car la réalisation de calculs dans des géométries tridimensionnelles de coeurs réalistes représente encore aujourd'hui un défi du point de vue de la mémoire allouée ainsi que du temps de calcul requis. En effet, après la discrétisation de toutes les variables, le nombre d'inconnues à traiter peut être de l'ordre de $\mathcal{O}(10^{14})$. Ce problème a traditionnellement été contourné en cherchant le flux scalaire $\phi(t, \boldsymbol{r}, E) = \int_{\mathbb{S}_2} \psi(t, \boldsymbol{r}, \boldsymbol{\omega'}, E) d\boldsymbol{\omega'}$, qui est une moyenne en angle du flux angulaire et qui peut être déterminé par la résolution d'une équation de diffusion. Une autre façon de limiter le nombre d'inconnues est de considérer le cas stationnaire, qui est important pour l'analyse des propriétés du coeur en régime normal de fonctionnement. Dans ce dernier cas, il n'est pas nécessaire de stocker ψ pour donner les outputs habituels du calcul, comme par exemple la puissance totale. En revanche, dans l'étude de transitoires rapides et d'accidents de réactivité, il n'est pas possible d'utiliser ces deux approximations et il est nécessaire de résoudre l'opérateur de Boltzmann sans aucune simplification en faisant face à la complexité numérique que cela entraîne.

Dans ce contexte, l'objectif de notre travail a été justement de traiter cette complexité numérique. Plus exactement, nous avons voulu montrer que la résolution de l'équation de cinétique transport neutronique dans des géométries 3D réalistes est de nos jours possible à effectuer dans

des temps de calcul raisonnables en employant des architectures modernes de calcul parallèle ainsi que des schémas numériques innovants.

Pour ce faire, nous avons travaillé dans un solveur appelé MINARET qui est développé au CEA dans le cadre du projet APOLLO3®. MINARET résout l'équation S_N de transport multi-groupe avec une discrétisation spatiale en éléments finis discontinus de Galerkin. Le développement d'un module pour des calculs concernant des configurations stationnaires de coeurs ayant été réalisé dans des travaux antérieurs à cette thèse (voir, par exemple, [89]), notre travail a donc commencé par l'extension du solveur existant en implémentant un module de résolution de problèmes à source. Ceci permet de traiter :

- des problèmes à source qui surviennent, par exemple, dans des calculs de fluence cuve,
- des problèmes de cinétique, comme par exemple des situations accidentelles.

Nous nous sommes concentrés dans ce travail sur l'accélération des cas de cinétique car les techniques employées dans ces problèmes s'appliquent aussi aux cas stationnaires. Par la même occasion, nous avons pu aussi aborder l'accélération de la variable temporelle, qui rallonge de façon très importante les temps de calcul. Les performances de certaines méthodes d'accélération séquentielles (extrapolation de Chebyshev et diffusion synthétique) ainsi que parallèles (pour la parallélisation des variables angulaire et temporelle) ont été analysées dans un benchmark classique de neutronique qui représente une éjection de barre de contrôle ([67]). Les résultats à ce sujet sont présentés dans les chapitres 2 et 3, où des réductions très significatives des temps de calcul sont montrées.

Le restant des chapitres de cette thèse constitue une contribution plus théorique et ils sont motivés par l'idée suivante : étant donné le haut degré de précision que MINARET peut fournir dans la modélisation de la population neutronique, pourrait-on utiliser ce solveur comme outil de suivi en temps réel de cette population afin d'aider au pilotage du réacteur en opération ? Cette question représente un très grand défi. Elle est même provocatrice étant donné que, malgré tous nos efforts pour accélérer le solveur, les calculs de MINARET sont loin de pouvoir être faits en temps réel ! En plus, notre outil devrait être cohérent en quelque sorte avec les mesures collectées pendant l'opération et provenant du coeur du réacteur lui-même et qui sont, jusqu'à présent, la seule information dont on dispose pour superviser le processus. Dans l'objectif de développer un tel outil, nous avons étendu une méthode d'interpolation déjà existante (appelée EIM pour Empirical Interpolation Method). En utilisant notre généralisation (appelée GEIM pour Generalized EIM) dans un contexte de bases réduites, il est possible de rassembler les mesures provenant d'expériences en temps réel avec des simulations numériques fondées sur des modèles mathématiques. La méthodologie est très générale et pourrait s'appliquer à la reconstruction d'un très grand nombre de processus physiques ou industriels.

Dans les paragraphes suivants, nous présentons un bref résumé de chaque chapitre de ce document. Nous aimerions précisier que le manuscrit est une compitation de quatre articles (chapitres 2, 4, 5 et 6) et deux travaux en cours (chapitres 3 et 7). Nous nous excusons donc par avance auprès du lecteur pour la répétition de certaines notions dans plusieurs chapitres.

Résumé des résultats par chapitres

Partie I : Chapitre 1

Ce premier chapitre a essentiellement pour but de résumer les connaissances actuelles que l'on peut trouver dans la bibliographie au sujet de l'équation du transport des neutrons. La plupart de ce qui est donc présenté n'est pas nouveau, mais le travail de rassembler toutes ces informations n'a pas été simple. Il nous a donc semblé intéressant de présenter cette compilation d'information, dans laquelle un effort particulier a été fait pour fournir les références les plus facilement trouvables.

Après la présentation de l'équation de Boltzmann dans le cas du transport neutronique, nous

rappellerons les principaux résultats théoriques concernant l'existence de solutions. Ensuite, nous discuterons sur les enjeux qui surgissent quant à la définition des conditions initiales de l'équation.

Les techniques de discrétisation des variables de l'équation seront ensuite présentées en insistant tout particulièrement sur les méthodes les plus répandues. Ceci nous permettra de passer au deuxième objectif de ce chapitre qui est celui de fournir des détails sur la construction du module de cinétique dans le solveur MINARET. La stratégie de discrétisation dans MINARET est plutôt classique et nous verrons que la résolution d'un pas de temps aboutit à la résolution d'un problème à source qui est numériquement abordé au moyen de deux schémas itératifs emboîtés : les itérations externes sont très similaires à des itérations de Gauss-Seidel et les itérations internes résolvent un schéma de Richardson.

En guise d'introduction pour la dernière partie du chapitre, nous rappellerons les principales simplifications de l'opérateur de Boltzmann qui existent en neutronique. Un accent tout particulier sera porté sur l'équation de diffusion cinétique, qui est utilisée pour modéliser l'évolution du flux scalaire ϕ dans l'industrie nucléaire.

La dernière partie du chapitre est consacrée aux techniques d'accélération qui peuvent s'employer pour accélérer un solveur comme MINARET, c'est à dire un code de cinétique transport S_N multi-groupe. Tout d'abord, les deux accélérations séquentielles implémentées dans MINARET seront présentées en détail (extrapolation de Chebyshev et diffusion synthétique). Nous finirons par analyser certaines méthodes d'accélération par parallélisation : nous expliquerons la stratégie suivie dans MINARET pour paralléliser les variables angulaire et temporelle. La parallélisation de la variable temporelle est particulièrement délicate étant donné que le temps est séquentiel par nature. Malgré cela, plusieurs stratégies ont été proposées à ce sujet-là dans la littérature (cf. [22], [44]) et nous nous sommes concentrés sur la méthode pararéelle (voir, par exemple, [72]) car c'est celle qui donne les meilleures performances. Le chapitre se finit par la présentation d'autres méthodes d'accélération par parallélisation qui n'ont pas été étudiées dans ce travail, mais qui semblent intéressants garder à l'esprit pour des travaux futurs.

Partie I : Chapitre 2

Le deuxième chapitre est un article qui résume les accélérations obtenues dans MINARET grâce aux accélérations séquentielles et parallèles décrites au chapitre 1. Les résultats principaux sont que l'extrapolation de Chebyshev combinée avec la diffusion synthétique réduit d'environ un facteur 100 les temps de calcul. Il est possible de réduire encore par trois ces temps de calcul en choisissant de bonnes initialisations pour les schémas itératifs employés.

La parallélisation des variables angulaire et temporelle a été étudiée séparément. La première fournit des speed-ups quasi optimaux pour un nombre réduit de processeurs. Les performances se dégradent quand le nombre de processeurs augmente non pas à cause du temps de communication, mais à cause de l'étape de diffusion synthétique qui n'a pas été parallélisée pour le moment (cette tâche est possible à faire et consisterait en la parallélisation d'un problème spatial elliptique par des techniques de décomposition de domaine comme celles qui sont présentées dans [4]).

Finalement, dans les exemples numériques que nous avons traités, l'utilisation de l'algorithme pararéel en temps peut accélérer les calculs d'environ un facteur 5 avec 40 processeurs. Du point de vue de l'efficacité, ces résultats ne sont pas aussi compétitifs que ce que la parallélisation de la variable angulaire fournit, mais comme il sera expliqué au chapitre 1, il existe des raisons théoriques qui expliquent la relativement basse efficacité de la méthode pararéelle. Pour cette raison, cette méthode devient intéressante pour atteindre des speed-ups additionnels dans un contexte où les autres techniques plus efficaces de parallélisation dont on peut disposer atteignent saturation (comme la parallélisation de la variable angulaire dans notre cas).

Partie I : Chapitre 3

Comme il sera expliqué au chapitres 1 et 2, la résolution de chaque pas de temps de l'équation de transport des neutrons utilise des schémas itératifs dans le solveur MINARET. Le nombre d'itérations est a priori inconnu et peut varier d'un instant de temps à autre. Lorsque la méthode pararéelle en temps est appliquée à ce problème, il se crée un déséquilibre en ce qui concerne le nombre d'itérations que chaque processeur doit traiter. Si l'on utilise un algorithme distribué pour implémenter la méthode pararéelle, chaque processeur P_n devra traiter la propagation des solveurs fin et grossier dans l'intervalle de temps $[T_n, T_{n+1}[$. Mais le coût en nombre d'itérations de ces propagations variera d'un processeur à l'autre en fonction de la complexité des événements qui auront lieu à l'instant $[T_n, T_{n+1}[$. Ce déséquilibre entraîne une dégradation des performances de la méthode.

Dans l'objectif d'aborder ce problème, nous présentons dans ce chapitre un travail en cours dans lequel nous étudions un schéma pararéel adapté dans lequel les itérations à chaque pas de temps sont tronquées et la convergence est atteinte de façon "globale" au cours des itérations pararéelles. Notre contribution peut être vue comme s'inscrivant dans la lignée de certains travaux précédents de M. Minion (cf. [87], [43]) dans lesquels la méthode pararéelle a déjà été couplée avec des itérations non linéaires.

A notre connaissance, aucune analyse de convergence n'existe à ce sujet dans la littérature et nous commencerons ce chapitre en présentant quelques résultats sur ce sujet. Par ailleurs, étant donné que le schéma nécessite du stockage de toutes les solutions à tous les instants de temps de l'itération pararéelle précédente, une stratégie de type bases réduites est proposée comme solution à ce problème de stockage qui peut être, dans de nombreux cas, impossible à faire. L'idée principale consiste à projeter les solutions dans une base réduite et de ne stocker que les projections.

Partie II : Chapitres 4 et 5

La seconde partie de cette thèse est consacrée au développement d'un outil de surveillance en temps réel de processus physiques ou industriels et qui combine des données mesurées avec des modèles mathématiques (représentés par des EDP paramétrées). En particulier, la méthode présentée pourrait être appliquée dans le futur pour coupler des calculs du solveur MINARET avec des mesures prises dans un coeur de réacteur. Cela permettrait de surveiller en temps réel la population neutronique en tout point du coeur. Pour développer un tel outil, il a été nécessaire tout d'abord d'étudier préalablement certains aspects théoriques et c'est ce qui est présenté dans la deuxième partie de ce manuscrit. Plusieurs exemples numériques simples seront aussi présentés dans le but d'illustrer la technique proposée ainsi que ses performances.

L'idée clé dans la construction de l'outil de surveillance que nous proposons repose sur l'extension de la méthode d'interpolation empirique (appelée EIM, pour Empirical Interpolation Method, cf. [11], [55], [80]). Dans la généralisation de EIM que nous proposons (appelée GEIM, pour Generalized EIM), les fonctions f à approcher appartiennent à un ensemble compact F d'un espace de Banach $\mathcal{X}$. La structure de F est supposée être telle que n'importe quel élément $f \in F$ puisse être approché par des combinaisons linéaires de petite taille. Ceci est quantifié par l'épaisseur de Kolmogorov $d_n(F, \mathcal{X})$ de F dans $\mathcal{X}$. Ce concept (qui sera défini rigoureusement dans cette deuxième partie du manuscrit) mesure jusqu'à quel point F peut être approché par des espaces de dimension finie n. La nouveauté de notre approche par rapport à EIM, c'est que nous travaillons avec des formes linéaires d'interpolation choisies dans un dictionnaire donné $\Sigma \in \mathcal{L}(\mathcal{X})$ au lieu de points d'interpolation. Ceci présente l'avantage majeur de pouvoir relaxer la traditionnelle condition nécessaire de continuité dans les fonctions à interpoler. De plus, les formes linéaires peuvent modéliser de façon plus fidèle les capteurs employés dans des expériences physiques en utilisant des moyennes locales.

Dans ce cadre, les chapitres 4 et 5 abordent les fondements de GEIM, en mettant un accent particulier sur le cas hilbertien. Ils sont présentés dans l'ordre chronologique afin de montrer nos progrès dans la compréhension de la théorie (interpolation bien posée, constante de Lebesgue, interprétation en tant que projection oblique) pendant ces trois années de travail. Le lecteur pourra observer que nous avons étendu la méthode du cas $\mathcal{X} = L^2(\Omega)$ au chapitre 4 au cas d'espaces de Banach au chapitre suivant. Dans le cas particulier (mais très important) où $\mathcal{X}$ est un espace de Hilbert, des avancées importantes ont été faites en ce qui concerne la compréhension de la condition de stabilité de l'interpolation généralisée (constante de Lebesgue) en la reliant à un problème $\inf - \sup$. Bien qu'il n'existe pas à l'heure actuelle de théorie sur l'impact du dictionnaire Σ sur la constante de Lebesgue, nous illustrerons son importante influence dans une application numérique simple en une dimension. En utilisant notre formule $\inf - \sup$, une croissance linéaire dans la constante de Lebesgue a été observée dans une application numérique en rapport avec un problème de Stokes (voir chapitre 5). Il est important de remarquer que ce résultat diffère beaucoup du comportement de la constante de Lebesgue présenté dans l'application numérique du chapitre 4. Cela est dû au fait que nous ne disposions pas d'une formule explicite pour la constante de Lebesgue au moment où l'article du chapitre 4 est paru.

Finalement, les chapitres 4 et 5 montrent des exemples d'application qui illustrent la méthodologie proposée pour la reconstruction en temps réel d'une expérience physique en utilisant GEIM dans un cadre de bases réduites.

Partie II : Chapitre 6

Dans GEIM, les espaces d'interpolation X_n de dimension n et les n formes linéaires d'interpolation sont donnés par un algorithme Greedy (tout comme dans la traditionnelle version d'EIM). Cet espace d'interpolation ne correspond pas en général au meilleur espace de dimension n qui pourrait être employé pour approcher les fonctions de F. Il est donc intéressant d'analyser la qualité de ces espaces d'interpolation X_n par rapport à le ou les espaces optimaux qui sont associés à l'épaisseur de Kolmogorov. Notre analyse sera faite en partant de l'hypothèse que $d_n(F, \mathcal{X})$ présente une décroissance de type exponentielle ou polynomiale.

Partie II : Chapitre 7

Dans ce chapitre, nous présentons un travail en cours dont le but n'est pas directement en rapport avec les précédents chapitres de cette thèse (nous présenterons néanmoins le rapport existant avec la méthode GEIM).

Nous souhaitons contribuer dans cette partie à la compréhension d'une méthode de post-traitement, dite de rectification, introduite dans [21] puis employée dans [59] dans le contexte de résolution d'EDP par des méthodes de bases réduites. Dans ces travaux, une approximation peu exacte mais peu coûteuse de type base réduite est post-traitée en faisant intervenir un certain nombre de snapshots et permet de retrouver, à moindre coût, une approximation beaucoup plus précise. Pour étudier cette méthode, nous nous sommes placés dans le cadre général suivant :

Soit $\mathcal{X}$ un espace de Hilbert et soit F un ensemble compact de $\mathcal{X}$ de petite épaisseur de Kolmogorov que l'on souhaite approcher au moyen d'éléments d'un espace $X_M \subset \mathcal{X}$ de petite dimension M. Supposons que l'on dispose de deux opérateurs d'approximation :

- $\pi_M : \mathcal{X} \mapsto X_M$ qui fournit une approximation très précise mais très coûteuse numériquement des éléments de F. C'est à dire que $\sup_{f \in F} \|f - \pi_M[f]\|_{\mathcal{X}}$ est suffisamment petit.
- $\mathcal{J}_M : \mathcal{X} \mapsto X_M$ qui donne une approximation peu coûteuse mais pas suffisamment précise des éléments de F. C'est à dire que $\sup_{f \in F} \|f - \mathcal{J}_M[f]\|_{\mathcal{X}}$ n'est pas suffisamment petit pour nos

critères.

Dans ce chapitre, nous présentons une méthode pour construire (offline) une application de rectification $R_M : X_M \mapsto X_M$ (dont l'utilisation online ne coûte que $\mathcal{O}\left(M^2\right)$ opérations). Sous certaines hypothèses qui seront présentées, l'application R_M permet d'avoir $\sup_{f \in F} \|f - (R_M \circ \mathcal{J}_M)[f]\|_{\mathcal{X}} \approx \sup_{f \in F} \|f - \pi_M[f]\|_{\mathcal{X}}$. Le point clé dans la définition de R_M réside dans le fait que $(R_M \circ \mathcal{J}_M)[f_i] = \pi_M[f_i]$ pour des éléments f_i, $i = 1, \ldots, M$, d'une base réduite de F.

L'intérêt de cette approche est que, une fois R_M construit, il est possible d'obtenir une approximation toute aussi précise que π_M mais à un coût très réduit (en utilisant $R_M \circ \mathcal{J}_M$).

Introduction (English version)

Motivations of this work

Like coal, gas, geothermal, solar thermal or waste incineration plants, nuclear power plants can be classified in the category of thermal plants. They generate electricity by heating a coolant (water or gas) and transform it into steam. The steam drives the rotation of a turbine that is coupled to an alternator where heat energy is transformed into electricity. After exiting the turbine, the steam is condensed in a condenser and recycled to the heat source.

The particularity of nuclear power plants relies on the fact that the heat source is a nuclear reactor, i.e. the coolant is heated in a nuclear core (see an example on figure 2) by nuclear fission reactions: inside the core, free neutrons collide with fissile particles (typically U^{235} or Pu^{239}) that are contained in the fuel material. Under appropriate conditions, the collision of a given neutron with a fissile particle can split this particle into lighter nuclei and also release 2 or 3 free neutrons in the medium (see formula (1) for an example of fission reaction). This kind of reaction is exothermic (about 200 MeV per fission of U^{235}) and is the heat source for the coolant (which is water in most cases).

$$\mathrm{U}_{92}^{235} + \mathrm{n}_0^1 \longrightarrow \mathrm{Kr}_{36}^{92} + \mathrm{Ba}_{56}^{141} + 3\mathrm{n}_0^1 \tag{1}$$

The released neutrons can in turn collide with the fuel again, giving rise to the so called nuclear chain reaction. The resulting lighter nuclei are also usually in an excited state and subject to further nuclear reactions to reach a more stable state. Some of these reactions can also involve the release of free neutrons to the medium. Thus it seems clear that controlling the population of free neutrons in the core is critical to preserve both the safety and the quality of the process: uncontrolled growth of this population can lead to an excessive heating of the core, a dangerous situation that can even cause an accident. Conversely, the population of neutrons must not be too low either, because the plant would not be generating the necessary heat to supply the electricity demand. In this framework, numerical simulations play an important role to design safe reactor core configurations and also help to understand accidental situations. Although there exists many complex coupled phenomena that need to be understood as a whole (thermalhydraulics, heat transfer, material irradiation and damage...), the study of the neutron population is carried out by neutronics and this work is a contribution in this field.

At a given instant t, a neutron of mass m can be described by its spatial position $\boldsymbol{r}$ and its velocity $\boldsymbol{v}$, or, equivalently, its position $\boldsymbol{r}$, its energy $E = m|\boldsymbol{v}|^2/2$ and its direction of motion $\boldsymbol{\omega} = \boldsymbol{v}/|\boldsymbol{v}|$. In particular, we wish to determine the density $n(t, \boldsymbol{r}, \boldsymbol{v}) \equiv n(t, \boldsymbol{r}, \boldsymbol{\omega}, E)$ of free neutrons per time unit and per spatial and velocity volumes. Equivalently, one can look for the so-called angular flux $\psi(t, \boldsymbol{r}, \boldsymbol{v}) = |\boldsymbol{v}|n(t, \boldsymbol{r}, \boldsymbol{v}) \equiv \psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E)$, which is also a representation of this population. As will be explained in chapter 1, $\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E)$ is the solution to a linear Boltzmann equation that represents a balance between the free neutrons that are created and that disappear in the core. Although it could be said that the main mathematical foundations of this equation are well understood nowadays, its numerical resolution in a full realistic three dimensional core still

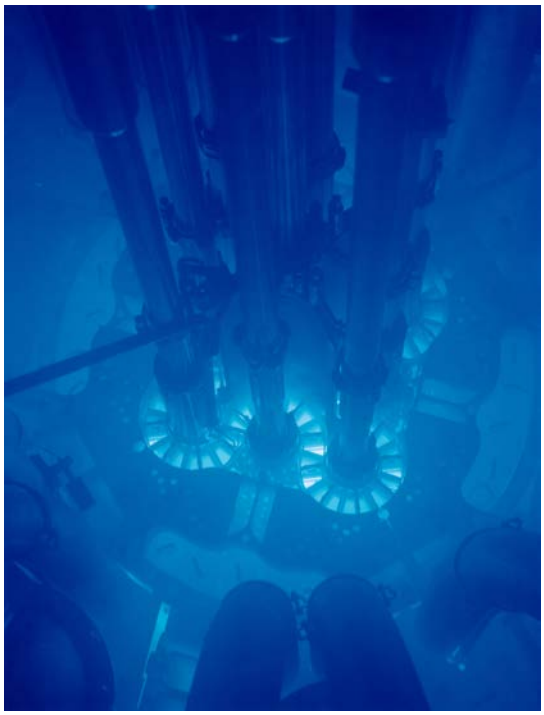


Figure 2: Example of a nuclear core: the Advanced Test Reactor core (Idaho National Laboratory)... with its mesmerizing blue color due to the Cherenkov radiation.

represents a challenge with respect to the memory storage and the required computational time. Indeed, after discretization of all the variables, the number of unknowns to be solved can easily reach $\mathcal{O}(10^{14})$ in realistic geometries. This issue has traditionally been circumvented by looking for the so-called scalar flux $\phi(t, \boldsymbol{r}, E) = \int_{\mathbb{S}_2} \psi(t, \boldsymbol{r}, \boldsymbol{\omega'}, E) d\boldsymbol{\omega'}$, an average of the angular flux that can be found through the resolution of a diffusion equation. Another way to limit the number of unknowns is to consider the stationary case, that is important for the analysis of the core under normal working conditions. This not only gets rid of the time variable, but such computations do not either require the storage of ψ to provide the standard outputs of neutronics such as the total power. However, for the study of fast transients or several types of reactivity accidents, these two approximations do not hold any more: the linear Boltzmann operator has to be treated without approximations and one has to cope with its computational complexity.

In this context, the aim of the present work has precisely been to deal with this computational complexity. Our purpose has been to show that the resolution of the time dependent neutron transport equation in realistic 3D geometries is feasible in a reasonable amount of time by the use of modern parallel computer architectures together with innovative numerical schemes.

For this, we have worked in a solver called MINARET, that is developed at CEA under the APOLLO3® project. MINARET solves the multigroup neutron transport S_N equation with a discontinuous Galerkin finite element discretization for the space. Since steady state calculations have already been implemented in previous works (see, e.g. [89]), our task started by implementing a source solver in MINARET. This lets us treat:

- source problems that arise, e.g., in vessel fluence calculations
- time dependent situations (for accidental situations).

We have in particular focused in speeding-up time dependent cases because the implemented techniques apply also to stationary source problems. At the same time, we have also dealt with the

time variable that significantly extends the computing times. Sequential acceleration techniques (Chebyshev extrapolation and Diffusion Synthetic Acceleration) as well as parallel ones (for the angular and time variables) have been explored through a classical benchmark that represents a rod withdrawal (see [67]). The results show a very significant reduction of the computing time and are presented on chapters 1, 2 and 3.

The remaining chapters of this thesis are a more theoretical contribution and are motivated by the following idea: given the high degree of accuracy that MINARET can provide in the modeling of the neutron population, could we somehow use it as a tool to monitor in real time the population of neutrons on the purpose of helping in the operation of the reactor? The question is highly challenging and even provocative given that, despite all our efforts to accelerate the solver, MINARET's computations are still far from being in real time. Besides, such a tool must be coherent in some sense with the measurements coming from the sensors that are inside the core and that are, up until now, the only information to supervise the process. With this idea in mind, we have developed an extension of an already existing interpolation methodology (called EIM, i.e. Empirical Interpolation Method). Used in a reduced basis framework, GEIM (as for Generalized EIM) allows to sensibly summarize measurements from the experiment with numerical simulations based on a mathematical model. The procedure is very general and could be applied to the reconstruction of any type of physical or industrial process.

In the following section, a summary of every chapter will be provided. We would like to point out that the present manuscript is a compilation of four articles (chapters 2, 4, 5 and 6) and two ongoing works (chapters 3 and 7). We therefore apologize to the reader in advance for the repetition of some of the notions at different chapters.

Summary of the results by chapters

Part I: Chapter 1

The aim of this first chapter is twofold: first and foremost, it is intended to provide a bibliographical overview of the time-dependent neutron transport equation. Most of what is stated here is not new, but gathering all the information that is presented here has sometimes been a hard task. It therefore seemed interesting to us to present this compilation of information, in which we have made a special effort to give as easily tractable references as possible.

After introducing the linear Boltzmann equation in the case of neutron transport, the main theoretical results regarding its existence will be presented. We will then explain the issue of deriving appropriate initial conditions to this equation.

We will continue by recalling the existing discretization techniques of the variables involved in the equation and a special emphasis will be put on the most widespread ones. This will let us come to the second objective of this chapter which is to provide some details about the construction of MINARET's time-dependent code. The discretization strategy followed in MINARET is rather classical and we will see that the resolution of a time-step is a source problem that is numerically solved by two embedded iterative schemes: outer iterations that are very similar to a Gauss-Seidel scheme and Richardson inner iterations.

As a preliminary introduction to the last part of this chapter, we continue by recalling the main approximations to the Boltzmann operator that exist in neutronics. In particular, we will detail the kinetic diffusion equation that is used to model the evolution of the scalar flux ϕ in the nuclear industry.

The last part of the chapter is devoted to the existing acceleration techniques that are available to accelerate a solver like MINARET, i.e. a time-dependent multigroup transport S_N code. First, the two traditional sequential accelerations implemented in MINARET will be presented in

detail (Chebyshev extrapolation and Diffusion Synthetic Acceleration). We finally discuss parallel acceleration techniques: we will present the strategy followed in MINARET to parallelize the angular and time variables. The parallelization of the time variable is particularly involved given the sequential nature of time. Despite this, several strategies have been proposed in the literature (see [22], [44]). We have focused in the parareal in time method (see, e.g. [72]) because it is the one that seems to provide the best performances. We finish by noting that other parallel accelerations that seem interesting to keep in mind for future works will also be outlined.

Part I: Chapter 2

The second chapter is an article that summarizes the speed-up performances that we have obtained in MINARET with the sequential and parallel acceleration techniques introduced in detail in chapter 1. The main results are that the use of the Chebyshev extrapolation combined with the Diffusion Synthetic Acceleration reduce by a factor of about 100 the computing time. Another factor of about 3 can also be gained by choosing an appropriate starting guess.

The parallelization of the angular and time variables has been tested separately. The first one provides almost optimal speed-ups for a reduced number of processors. The performances are degraded for higher numbers not because of communication times, but mainly because the Diffusion Synthetic Acceleration has not been parallelized yet (this task is nevertheless possible and it would involve spatial domain decomposition methods like the ones outlined in [4]).

Finally, in the numerical examples that we have treated, the inclusion of the parareal in time algorithm can speed-up the calculations by a factor of about 5 with 40 processors. From an efficiency point of view, these results are not as competitive as the high efficiency that the parallelization of the angular variable provides, but, as it will be explained in chapter 1, there are theoretical reasons that explain the relatively low efficiency of the parareal in time method. Because of this fact, parareal is an useful technique in the context where other more efficient parallelization techniques reach saturation (such as the parallelization of the angular variable in our case) as a way to obtain additional speed-ups.

Part I: Chapter 3

As will be presented in detail in chapters 1 and 2, the resolution of each time step of the neutron transport equation is performed by iterative techniques in the MINARET solver. The number of iterations is not a priori known and can vary from one time step to another. When applying the parareal in time algorithm to this problem, an imbalance is created in the complexity of the tasks addressed by each processor. If we use a distributed algorithm to implement the parareal algorithm, processor P_n will deal with the propagations of the coarse and fine solvers in the time slice $[T_n, T_{n+1}[$. But the cost of these propagations can vary from one processor to another depending on the numerical complexity (number of internal iterations) that takes place in each time slice $[T_n, T_{n+1}[$. This imbalance results in a degradation of the speed-up performances.

In an attempt to address this issue, this chapter presents an ongoing work in which we look for an adapted parareal numerical scheme where the iterations inside a time step are truncated and the convergence is reached "globally" across the parareal iterations. The work can be seen as an extension of previous works of M. Minion (see [87], [43]) in which parareal has already been coupled with non linear iterations.

To the best of our knowledge, a convergence analysis of such a scheme does not exist in the literature and we will start by presenting some results in this respect. Furthermore, since the scheme requires the storage of the solutions at all times of the previous parareal iteration, a reduced basis strategy is explored as a remedy to this problem that can be, in many cases, an unaffordable

requirement. The main idea consists in projecting the previous solutions in a reduced basis and storing the projections.

Part II: Chapters 4 and 5

The second part of this thesis is devoted to the development of a numerical tool to supervise in real time an industrial or physical process by combining sensor measurements and a mathematical model (via a parameter dependent PDE). In particular, the technique presented here could be applied in the future to couple MINARET's calculations with internal measurements in a reactor to monitor in real time the neutron population inside the core. The development of the method has required the analysis of some theoretical aspects beforehand and this is what is presented in this second part of this manuscript. Nevertheless, several simple numerical examples will be presented with the purpose of illustrating the technique and its performances.

The key idea to build our monitoring tool relies in an extension of the so-called Empirical Interpolation Method (EIM, [11], [55], [80]). In the Generalized EIM that we have explored, the functions f to approximate belong to a compact set F of a general Banach space of functions $\mathcal{X}$. The structure of F is supposed to make any $f \in F$ be approximable by finite expansions of small size. This is quantified by the Kolmogorov $n-$width $d_n(F, \mathcal{X})$ of F in $\mathcal{X}$ (a concept that will be precised later on), whose smallness measures the extent to which F can be approximated by some finite dimensional space $X_n \subset \mathcal{X}$ of dimension n. The novelty, in comparison with the traditional EIM, relies in the fact that we work with interpolating continuous linear functionals chosen in a given dictionary $\Sigma \subset \mathcal{L}(\mathcal{X})$ instead of interpolating points. This presents the major advantage of relaxing the classical continuity requirement in the target functions. Besides, the linear functionals could model real sensors by the use of local averages in a more faithful manner.

In this framework, chapters 4 and 5 deal with the foundations of GEIM and a particular focus has been placed on the hilbertian case. They are presented in chronological order with the purpose of showing our advances in the understanding of the theory (well-posedness, Lebesgue constant, interpretation as an oblique projection...) during our three years of work on this topic. In this regard, a special effort has been done to enlarge the theory from the case $\mathcal{X} = L^2(\Omega)$ (chapter 4) to Banach spaces (chapter 5). In the particular, albeit very important, case of Hilbert spaces, significant advances have been made in the understanding of the stability condition of the generalized interpolant (the Lebesgue constant) by relating it to an $\inf - \sup$ problem. Although there is no theory on the impact of the dictionary Σ on the Lebesgue constant, we illustrate its critical influence through a one-dimensional simple case. Thanks to the $\inf - \sup$ formula, a linearly increasing Lebesgue constant in the numerical application of chapter 5 has been observed. Note that this result differs greatly from the behavior of the estimated Lebesgue constant of chapter 4. This is due to the fact that we did not have at our disposal the explicit formula of the Lebesgue constant at the time when the article of chapter 4 was written.

Chapters 4 and 5 also present two numerical examples that illustrate the reconstruction procedure of an experiment thanks to the use of GEIM in a reduced basis framework.

Part II: Chapter 6

In GEIM, the interpolating n-dimensional spaces X_n and the n interpolating linear forms are provided by a Greedy algorithm (just like in the traditional EIM). This interpolating space does not correspond, in general, to the best n-dimensional space that one could use to approximate the functions of F. Hence, it is interesting to analyze the quality of the generalized interpolating spaces X_n built by the Greedy selection procedure. On this purpose, the accuracy of our interpolation in X_n will be compared in this chapter to the best possible performance that is given by the

Kolmogorov n-width. This analysis will be conducted in the case where $d_n(F, \mathcal{X})$ is supposed to present an exponential or polynomial decay rate.

Part II: Chapter 7

In this chapter we present an ongoing study whose main goal is not directly related to the previous chapters of this thesis (we will nevertheless briefly discuss its possible connection with the works on GEIM).

Our main purpose here is to shed some light about a successful post-processing strategy first presented in [21] and used in [59] in the framework of reduced basis simulation of PDE's. In these works, some cheap and non optimal reduced basis approximation is post-processed through some snapshots and allows to recover a very accurate approximation. To analyze the method, we work in the following framework:

Let $\mathcal{X}$ be a Hilbert space and let F be a compact subset of $\mathcal{X}$ of small Kolmogorov n-width that we wish to approximate accurately by elements of a finite dimensional subspace $X_M \subset \mathcal{X}$ of small dimension M. Suppose that we have at our disposal two approximation operators:

- $\pi_M : \mathcal{X} \mapsto X_M$ that provides a computationally expensive, but accurate approximation of the elements of F, i.e. such that $\sup_{f \in F} \|f - \pi_M[f]\|_{\mathcal{X}}$ is small enough for the application under consideration,
- $\mathcal{J}_M : \mathcal{X} \mapsto X_M$ that provides a cheap, but inaccurate approximation of the elements of F, i.e. such that $\sup_{f \in F} \|f - \mathcal{J}_M[f]\|_{\mathcal{X}}$ is not small enough for our standards.

In this chapter, we will present a method to build (offline) a rectification operator $R_M : X_M \mapsto X_M$ (the online application of which costs $\mathcal{O}(M^2)$ computations) such that, under several hypothesis that will be discussed, $\sup_{f \in F} \|f - (R_M \circ \mathcal{J}_M)[f]\|_{\mathcal{X}} \approx \sup_{f \in F} \|f - \pi_M[f]\|_{\mathcal{X}}$. The key point to build R_M is that $(R_M \circ \mathcal{J}_M)[f_i] = \pi_M[f_i]$ for the elements f_i, $1 \leq i \leq M$, of a reduced basis of F.

The interest of this approach is that, once R_M has been built, one may circumvent the computational cost of π_M but nevertheless recover its accuracy (using $R_M \circ \mathcal{J}_M$). After the theoretical introduction we finish by presenting some numerical example.

Part I

Numerical models for time dependent neutron transport for safety studies

Chapter 1

Overview and modern challenges of neutronic calculations

This first chapter is intended to be a bibliographic summary about the time-dependent neutron transport equation: an overview of some theoretical results, discretization, numerical and acceleration methods to solve the equation are presented. We also explain some traditional approximations to the Boltzmann operator like diffusion. Most of what this chapter contains is not new and what could make it of any value is the difficulty of gathering all the bibliographical references. This is the reason why we would like to share with the reader this compilation of information.

To illustrate the main theoretical results existing in the theory of neutron transport, we have chosen to cite the theorems or lemmas from the literature that can be found in a very explicit and direct way. As it will be made clear, some of these results are formulated under hypothesis that are sometimes not entirely realistic. In that case, we will cite other references where one can find elements that could bring to the formulation of similar results but with more realistic hypothesis.

Paradoxically enough, although the fundamentals of neutron transport are nowadays very well established, the time-dependent transport equation has almost never been implemented in three-dimensional geometries because of long computational times. In this context, the main contribution of this work has been to explore sequential and parallel acceleration techniques to reduce this computational time (see chapters 2 and 3) and also to think about numerical methods that could one day make such computations be performed in real time (see the second part of this manuscript).

1.1 The time-dependent neutron transport equation

1.1.1 The equation

The evolution of the angular flux ψ of neutrons in a reactor core $\mathcal{R}$ is governed by a linear Boltzmann equation whose terms physically express a balance between the free neutrons that are created and that disappear in the core. We will consider here the three-dimensional case ($\mathcal{R} \subset \mathbb{R}^3$) where ψ depends on 7 variables, namely the time $t \in [0, T]$, the position within the reactor denoted as $\boldsymbol{r} \in \mathcal{R}$, the velocity of the neutrons $\boldsymbol{v} = \sqrt{2E/m}\,\boldsymbol{\omega}$ where $E \in [E_{\min}, E_{\max}]$ stands for the energy of the neutron, $\boldsymbol{\omega} = \dfrac{\boldsymbol{v}}{|\boldsymbol{v}|}$ stands for the direction of the velocity and m is the mass of the neutron. We will have $\boldsymbol{v} \in \mathcal{V} = \mathbb{S}_2 \times [E_{\min}, E_{\max}]$, where $\mathcal{V}$ is a compact subset of $\mathbb{R}^3$ and $\boldsymbol{\omega}$ in the unit shpere $\mathbb{S}_2$. In order to take into account the presence of radioactive isotopes (also called precursors) that emit neutrons with a given delay, the time-dependent neutron transport equation needs to be complemented with a set of first order ODE's expressing the evolution in $\mathcal{R}$ of the precursors' concentration that will be denoted as $\mathbf{C} = \{C_\ell\}_{\ell \in \{1,\dots,L\}}$.

The set $(\psi, \mathbf{C})$ is thus the solution to the following initial value problem over the domain $\mathcal{D} = \{(t, \boldsymbol{r}, \boldsymbol{\omega}, E) \in [0, T] \times \mathcal{R} \times \mathbb{S}_2 \times [E_{\min}, E_{\max}]\}$:

$$\begin{cases} \dfrac{1}{|\boldsymbol{v}|}\partial_t\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) + (L - H - F - Q)\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) = S(t, \boldsymbol{r}, \boldsymbol{\omega}, E) \\ \partial_t C_\ell(t, \boldsymbol{r}) = -\lambda_\ell C_\ell(t, \boldsymbol{r}) \\ \qquad\qquad + \displaystyle\int_{E'=E_{\min}}^{E_{\max}} \beta_\ell(t, \boldsymbol{r}, E')(\nu\sigma_f)(t, \boldsymbol{r}, E')\phi(t, \boldsymbol{r}, E')dE', \ \forall \ell \in \{1, \dots, L\}, \end{cases} \tag{1.1}$$

where $\phi(t, \boldsymbol{r}, E) = \displaystyle\int_{\mathbb{S}_2} \psi(t, \boldsymbol{r}, \boldsymbol{\omega'}, E)d\boldsymbol{\omega'}$ is the scalar flux and the following operator notations have been used:

- $L\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) = (\boldsymbol{\omega}.\nabla + \sigma_t(t, \boldsymbol{r}, E))\,\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E)$ is the advection operator,
- $H\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) = \displaystyle\int_{\mathbb{S}_2}\int_{E'=E_{\min}}^{E_{\max}} \sigma_s(t, \boldsymbol{r}, \boldsymbol{\omega'} \to \boldsymbol{\omega}, E' \to E)\psi(t, \boldsymbol{r}, \boldsymbol{\omega'}, E')dE'd\boldsymbol{\omega'}$ is the scattering operator,
- $F\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) = \dfrac{\chi_p(t, \boldsymbol{r}, E)}{4\pi}\displaystyle\int_{E'=E_{\min}}^{E_{\max}} (1 - \beta(t, \boldsymbol{r}, E'))\,(\nu\sigma_f)(t, \boldsymbol{r}, E')\phi(t, \boldsymbol{r}, E')dE'$ is the prompt fission operator,
- $Q\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) = \displaystyle\sum_{\ell=1}^{L} \lambda_\ell\chi_{d,\ell}(t, \boldsymbol{r}, E)C_\ell(t, \boldsymbol{r})$ is the delayed fission source,
- $S(t, \boldsymbol{r}, \boldsymbol{\omega}, E)$ is an external source that designates an angular density of neutrons in $(\boldsymbol{r}, \boldsymbol{\omega}, E)$ at time t per time unit. It is therefore a positive quantity. There are basically two types of time-dependent calculations in the nuclear industry:
  - analysis of safety issues, such as accident scenarios, in which the external source is negligible.
  - reactor start-up, in which this source is not zero.

In the enlisted terms, $\sigma_t(t, \boldsymbol{r}, E)$ denotes the total cross-section and $\sigma_s(t, \boldsymbol{r}, \boldsymbol{\omega'} \to \boldsymbol{\omega}, E' \to E)$ is the scattering cross-section from energy E' and direction $\boldsymbol{\omega'}$ to energy E and direction $\boldsymbol{\omega}$. $\sigma_f(t, \boldsymbol{r}, E)$ is the fission cross-section. $\nu(t, \boldsymbol{r}, E)$ is the average number of neutrons emitted per fission and $\chi_p(t, \boldsymbol{r}, E)$ and $\chi_{d,\ell}(t, \boldsymbol{r}, E)$ are respectively the prompt spectrum and the delayed spectrum of precursor ℓ. λ_ℓ and $\beta_\ell(t, \boldsymbol{r}, E)$ are respectively the decay constant and the delayed neutron fraction of precursor ℓ . Finally: $\beta(t, \boldsymbol{r}, E) = \displaystyle\sum_{\ell=1}^{L} \beta_\ell(t, \boldsymbol{r}, E)$.

Equation (1.1) is complemented with initial conditions ψ^0 and $C_{\ell,0}$ at $t = 0$ and boundary conditions over $\partial\mathcal{R}$. In the following subsections, several forms of boundary conditions will briefly be recalled and a theorem from the literature about the existence and uniqueness of the resulting Cauchy problem will be presented to account for its well-posedness.

1.1.2 Boundary conditions

We will assume that $\mathcal{R}$ is a bounded open set of $\mathbb{R}^3$. We denote as ds the surface measure on the boundary $\partial\mathcal{R}$ and assume that $\partial\mathcal{R}$ is continuously differentiable. We define the following partitions of the set $\Gamma = \partial\mathcal{R} \times \mathcal{V}$:

$$\begin{cases} \Gamma_0 = \{(\boldsymbol{r}, \boldsymbol{v}) \in \partial\mathcal{R} \times \mathcal{V};\ \boldsymbol{v}.\boldsymbol{n}(\boldsymbol{r}) = 0\} \\ \Gamma_+ = \{(\boldsymbol{r}, \boldsymbol{v}) \in \partial\mathcal{R} \times \mathcal{V};\ \boldsymbol{v}.\boldsymbol{n}(\boldsymbol{r}) > 0\} \\ \Gamma_- = \{(\boldsymbol{r}, \boldsymbol{v}) \in \partial\mathcal{R} \times \mathcal{V};\ \boldsymbol{v}.\boldsymbol{n}(\boldsymbol{r}) < 0\}\,, \end{cases} \tag{1.2}$$

where $\boldsymbol{n}(\boldsymbol{r})$ is the outward unit normal vector to $\partial\mathcal{R}$ at point $\boldsymbol{r}$. The set Γ_- (resp. Γ_+) will therefore correspond to the set of the phase space for which particles are incoming (resp. exiting).

Regarding Γ_0, it indicates the space of tangent directions to $\mathcal{R}$ and we will assume that it has zero measure over Γ for the measure $dsd\boldsymbol{v}$.

We list here some of the more usual boundary conditions that are associated to problem (1.1) and that lead to a well posed Cauchy problem.

Vacuum boundary conditions

We assume that $\mathcal{R}$ is surrounded by vacuum and therefore there are no incoming particles in $\mathcal{R}$:

$$\psi(t,\boldsymbol{r},\boldsymbol{v}) = 0, \forall t \in [0,T] \text{ and } (\boldsymbol{r},\boldsymbol{v}) \in \Gamma_-. \tag{1.3}$$

In the following sections, we will put special stress on this type of boundary conditions because it is the one that has been used throughout our studies with the MINARET solver.

Non homogeneous boundary conditions

In this condition, there is a given incoming angular flux g_{in} in Γ_-:

$$\psi(t,\boldsymbol{r},\boldsymbol{v}) = g_{in}(t,\boldsymbol{r},\boldsymbol{v}), \forall t \in [0,T] \text{ and } (\boldsymbol{r},\boldsymbol{v}) \in \Gamma_-. \tag{1.4}$$

Reflective and albedo boundary conditions

In the reflective boundary conditions, we make the hypothesis that the incoming angular flux is equal to the exiting flux through the relation:

$$\psi(t,\boldsymbol{r},\boldsymbol{v}) = \psi(t,\boldsymbol{r},\boldsymbol{v}') \text{ with } \boldsymbol{v}' = \boldsymbol{v} - 2\boldsymbol{n}(\boldsymbol{n}.\boldsymbol{v}) \quad \forall t \in [0,T] \text{ and } (\boldsymbol{r},\boldsymbol{v}) \in \Gamma_- \tag{1.5}$$

These conditions consist in supposing that the boundary reflects "like a mirror" the particles and they are often employed for assembly simulations where we usually need to work in an infinite periodic medium.

A more involved boundary condition (that is in fact closer to the real physics of the core) is the albedo condition in which some of the exiting particles are "reflected" inside the domain $\mathcal{R}$ and others definitely leave the medium. The condition reads:

$$\psi(t,\boldsymbol{r},\boldsymbol{v}) = \int_0^t \int_{\Gamma_+} \beta(t',\boldsymbol{r}',\boldsymbol{v}',t,\boldsymbol{r},\boldsymbol{v})\psi(t',\boldsymbol{r}',\boldsymbol{v}')d\Gamma'_+dt' \quad , \forall t \in [0,T] \text{ and } (\boldsymbol{r},\boldsymbol{v}) \in \Gamma_-, \tag{1.6}$$

where $\beta(t',\boldsymbol{r}',\boldsymbol{v}',t,\boldsymbol{r},\boldsymbol{v})$ is a given function that stands for the flux that, at time t, enters the domain at $\boldsymbol{r}$ with velocity $\boldsymbol{v}$ as a result of the interaction with the external media of a unit flux of particles, which, at time t', exit the domain at $\boldsymbol{r}'$ with velocity $\boldsymbol{v}'$.

Periodic boundary conditions

Periodic boundary conditions are often used to simulate a large system by modeling a small part that is far from its edge. These conditions consist in enforcing a relation of the form

$$\psi(t,\boldsymbol{r},\boldsymbol{v}) = \psi(t,\boldsymbol{r}',\boldsymbol{v}), \tag{1.7}$$

for $\boldsymbol{r}$ and $\boldsymbol{r}'$ in $\partial\mathcal{R}$. For instance, if $\mathcal{R} = [0,L]$ in a 1D case, the condition would read:

$$\psi(t,0,\boldsymbol{v}) = \psi(t,L,\boldsymbol{v}), \quad \forall\ (t,\boldsymbol{v}) \in [0,T] \times \mathcal{V}. \tag{1.8}$$

1.1.3 Existence theorems

The theoretical properties of existence, uniqueness, positiveness of the solution to equation (1.1) have been investigated by several authors in the literature. We recall here one of the most important theorem on this topic that can be found in chapter XXI, section 3.1 of [33]. The theorem proves the existence, uniqueness and regularity of the solution $(\psi, \mathbf{C})$ in the case where there are no delayed neutrons and that the cross-sections do not vary in time. An extension of this result to the case in which we do not make these hypothesis and for more involved boundary conditions can be found in [111].

The result requires the reformulation of equation (1.1) in a form that uses the variables $(t, \boldsymbol{r}, \boldsymbol{v}) \in [0,T] \times \mathcal{R} \times \mathcal{V}$ instead of $(t, \boldsymbol{r}, \boldsymbol{\omega}, E) \in [0,T] \times \mathcal{R} \times \mathbb{S}_2 \times [E_{\min}, E_{\max}]$. For this, we define:

$$\begin{cases} \Psi(t,\boldsymbol{r},\boldsymbol{v}) & := \dfrac{m}{|\boldsymbol{v}|}\psi(t,\boldsymbol{r},\boldsymbol{\omega},E), \quad \text{with } E = \dfrac{1}{2}m|\boldsymbol{v}|^2; \quad \boldsymbol{\omega} = \boldsymbol{v}/|\boldsymbol{v}| \\ \sigma(t,\boldsymbol{r},\boldsymbol{v}) & := |\boldsymbol{v}|\sigma_t(t,\boldsymbol{r},E), \\ f(t,\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v}) & = m\dfrac{|\boldsymbol{v}'|}{|\boldsymbol{v}|}\sigma_s(t,\boldsymbol{r},\boldsymbol{\omega}' \to \boldsymbol{\omega}, E' \to E) \\ \tilde{f}(t,\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v}) & = f(t,\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v}) + m\dfrac{|\boldsymbol{v}'|}{|\boldsymbol{v}|}(1-\beta(t,\boldsymbol{r},E))\dfrac{\chi_p(t,\boldsymbol{r},E)}{4\pi}\nu\sigma_f(t,\boldsymbol{r},E'), \\ f_\ell(t,\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v}) & := m\dfrac{|\boldsymbol{v}'|}{|\boldsymbol{v}|}\beta_\ell(t,\boldsymbol{r},E')\chi_{d,\ell}(t,\boldsymbol{r},E)\nu\sigma_f(t,\boldsymbol{r},E'), \\ c_\ell(t,\boldsymbol{r},\boldsymbol{v}) & := \dfrac{m}{|\boldsymbol{v}|}\chi_{d,\ell}(t,\boldsymbol{r},E)C_\ell(t,\boldsymbol{r}), \\ \tilde{\sigma}_f(t,\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v}) & := \sum_{\ell=1}^{L}\lambda_\ell f_\ell(t,\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v}) + m\dfrac{|\boldsymbol{v}'|}{|\boldsymbol{v}|}(1-\beta(t,\boldsymbol{r},E))\chi_p(t,\boldsymbol{r},E)\nu\sigma_f(t,\boldsymbol{r},E'), \\ q(t,\boldsymbol{r},\boldsymbol{v}) & := \dfrac{1}{m}S(t,\boldsymbol{r},\boldsymbol{\omega},E) \end{cases} \tag{1.9}$$

From equation (1.1), we derive the following transport evolution equation for Ψ over $[0,T]\times\mathcal{R}\times\mathcal{V}$:

$$\begin{cases} \dfrac{\partial\Psi}{\partial t}(t,\boldsymbol{r},\boldsymbol{v}) + \boldsymbol{v}.\nabla\Psi(t,\boldsymbol{r},\boldsymbol{v}) + \sigma(t,\boldsymbol{r},\boldsymbol{v})\Psi(t,\boldsymbol{r},\boldsymbol{v}) \\ \qquad - \int_{\mathcal{V}}\tilde{f}(t,\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})\Psi(t,\boldsymbol{r},\boldsymbol{v}')d\boldsymbol{v}' - \sum_{\ell=1}^{L}\lambda_\ell c_\ell(t,\boldsymbol{r},\boldsymbol{v}) = q(t,\boldsymbol{r},\boldsymbol{v}) \\ \dfrac{\partial c_\ell}{\partial t}(t,\boldsymbol{r},\boldsymbol{v}) = -\lambda_\ell c_\ell(t,\boldsymbol{r},\boldsymbol{v}) + \int_{\mathcal{V}} f_\ell(t,\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})\Psi(t,\boldsymbol{r},\boldsymbol{v}')d\boldsymbol{v}', \end{cases} \tag{1.10}$$

where $d\boldsymbol{\omega}dE = \dfrac{m}{|\boldsymbol{v}|}d\boldsymbol{v}$.

Theorem 1.1.1 ensures the existence and uniqueness of the following Cauchy problem that comes from equation (1.10) when there are no delayed neutrons, the cross-sections do not vary in time and for vacuum boundary conditions, i.e. for:

$$\begin{cases} \dfrac{\partial\Psi}{\partial t}(t,\boldsymbol{r},\boldsymbol{v}) + \boldsymbol{v}.\nabla u(t,\boldsymbol{r},\boldsymbol{v}) + \sigma(\boldsymbol{r},\boldsymbol{v})\Psi(t,\boldsymbol{r},\boldsymbol{v}) \\ \qquad - \int_{\mathcal{V}}\tilde{f}(\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})\Psi(t,\boldsymbol{r},\boldsymbol{v}')d\boldsymbol{v}' = q(t,\boldsymbol{r},\boldsymbol{v}), \\ \Psi(t,.)|_{\Gamma^-} = 0, \\ \Psi(0,.) = \Psi^0 \end{cases} \tag{1.11}$$

The solution $\Psi(t,\boldsymbol{r},\boldsymbol{v})$ is sought as a function over time with values in $L^p(\mathcal{R}\times\mathcal{V})$, with $p \in [1,\infty[$. For a given $T > 0$, we define the space

$$\begin{aligned} \mathcal{W}_p = \quad & \{u \in L^p\left(]0,T[\times\mathcal{R}\times\mathcal{V}\right); \ \frac{\partial u}{\partial t} + \boldsymbol{v}.\nabla u \in L^p\left(]0,T[\times\mathcal{R}\times\mathcal{V}\right); u(0,.) \in L^p\left(\mathcal{R}\times\mathcal{V}\right); \\ & u|_{]0,T[\times\Gamma^-} \in L^p\left(]0,T[\times\Gamma^-\right) \text{ for the measure } |\boldsymbol{v}.\boldsymbol{n}|dtdsd\boldsymbol{v}\}, \end{aligned}$$

that is used in:

Theorem 1.1.1 (Chapter XXI of [33], section 1, paragraph 3, theorem 3)**.**
Assume that the functions involved in problem (1.11) are such that:

- $\sigma \in L^\infty(\mathcal{R}, \mathcal{V})$ *and* $\sigma \geq 0$
- $\tilde{f}$ *is positive and* $d\boldsymbol{v}$ *measurable for* $\boldsymbol{v}$ *and* $\boldsymbol{v}'$ *and there exists positive constants* M_a *and* M_b *such that:*
  i) $\int_{\mathcal{V}} \tilde{f}(\boldsymbol{r}, \boldsymbol{v}', \boldsymbol{v}) d\boldsymbol{v} \leq M_a, \ \forall (\boldsymbol{r}, \boldsymbol{v}') \in \mathcal{R} \times \mathcal{V}$
  ii) $\int_{\mathcal{V}} \tilde{f}(\boldsymbol{r}, \boldsymbol{v}', \boldsymbol{v}) d\boldsymbol{v}' \leq M_b, \ \forall (\boldsymbol{r}, \boldsymbol{v}) \in \mathcal{R} \times \mathcal{V}$
- $q \in L^p(]0, T[\times \mathcal{R} \times \mathcal{V}), \ p \in [1, \infty[$
- $\Psi^0 \in L^p(\mathcal{R} \times \mathcal{V})$

Then, problem (1.11) has a unique solution Ψ *in* $\mathcal{W}_p$ *in a weak sense and*

$$\Psi \in \mathcal{C}\left([0, T]; L^p(\mathcal{R} \times \mathcal{V})\right).$$

Furthermore, if Ψ^0 *is such that*

$$\boldsymbol{v}.\nabla \Psi^0 \in L^p(\mathcal{R} \times \mathcal{V}) \quad \textit{and} \quad \Psi^0|_{\Gamma^-} = 0$$

and q *such that*

$$q \in \mathcal{C}^1\left([0, T]; L^p(\mathcal{R} \times \mathcal{V})\right),$$

then Ψ *is a solution in a strong sense and verifies:*

$$\Psi \in \mathcal{C}^1\left([0, T]; L^p(\mathcal{R} \times \mathcal{V})\right) \quad \boldsymbol{v}.\nabla \Psi \in \mathcal{C}\left([0, T]; L^p(\mathcal{R} \times \mathcal{V})\right) \quad \Psi(t)|_{\Gamma^-} = 0, \ \forall t \in [0, T].$$

Finally, if $q \geq 0$*, then* $\Psi^0 \geq 0$ *implies* $\Psi \geq 0$*.*

Remark 1.1.2 (About the regularity of the solution)**.** *In order that the solution* $\psi(t, .)$ *at time* t *is regular in* $(\boldsymbol{r}, \boldsymbol{v})$*, it is necessary that the initial condition* ψ^0 *is regular. This condition is however not sufficient: if* $\psi^0 \in \mathcal{C}^\infty$ *but is not null in* Γ_-*, then* $\psi(t)$ *will not be continuous. Another source of discontinuity arises if the domain "has holes", i.e. if* $\mathbb{R}^3 \backslash \mathcal{R}$ *is not connected.*

1.2 The stationary case: resolution of a generalized eigenvalue problem

The initial conditions ψ^0 and $C_{\ell,0}$ depend on the situation under consideration. In the analysis of reactor cores, what one wishes in the end is to understand the connection between a stationary state and some transient state. One can first of all be interested in how the system can reach a steady state from a given transient (regardless of the events that have led to the given transient state). In that case, ψ^0 and $C_{\ell,0}$ would correspond to an unsteady state of the system. Although this first option is of the utmost importance for reactor safety, it is nowadays very difficult to have access to the knowledge of an initial condition corresponding to a given generic transient state. For this reason, the initial conditions in neutron reactor kinetics come from a steady state whose computation is well-known. In this section, we provide some results about the resolution of this stationary state.

1.2.1 The equation

In equilibrium, neither the flux nor the parameters of the system evolve in time and from equation (1.1) we easily derive an expression for the precursors:

$$C_{\ell,0}(\boldsymbol{r}) = \frac{1}{\lambda_\ell}\int_{E'=E_{\min}}^{E_{\max}} \beta_\ell(\boldsymbol{r},E')(\nu\sigma_f)(\boldsymbol{r},E')\phi(\boldsymbol{r},E')dE', \forall \ell \in \{1,\dots,L\}. \tag{1.12}$$

By inserting this formula in the stationary version of equation (1.1), we derive a PDE for the flux in $\mathcal{R}\times\mathbb{S}_2\times[E_{\min},E_{\max}]$:

$$\begin{cases} (L_0 - H_0 - F_0)\psi(\boldsymbol{r},\boldsymbol{\omega},E) = 0 \\ \psi = 0 \text{ in } \Gamma_-, \end{cases} \tag{1.13}$$

where, for simplicity, we have imposed vacuum boundary conditions and $S\equiv 0$. The notations are:

- $L_0\psi(\boldsymbol{r},\boldsymbol{\omega},E) = (\boldsymbol{\omega}.\nabla + \sigma_t(\boldsymbol{r},E))\,\psi(\boldsymbol{r},\boldsymbol{\omega},E)$,
- $H_0\psi(\boldsymbol{r},\boldsymbol{\omega},E) = \int_{\mathbb{S}_2}\int_{E'=E_{\min}}^{E_{\max}} \sigma_s(\boldsymbol{r},\boldsymbol{\omega'}\to\boldsymbol{\omega},E'\to E)\psi(\boldsymbol{r},\boldsymbol{\omega'},E')dE'd\boldsymbol{\omega'}$,
- $F_0\psi(\boldsymbol{r},\boldsymbol{\omega},E)$
  $= \dfrac{\chi_p(\boldsymbol{r},E)}{4\pi}\displaystyle\int_{E'=E_{\min}}^{E_{\max}} (1-\beta(\boldsymbol{r},E'))\,(\nu\sigma_f)(\boldsymbol{r},E')\phi(\boldsymbol{r},E')dE' + \sum_{\ell=1}^{L}\lambda_\ell\chi_{d,\ell}(\boldsymbol{r},E)C_\ell(t,\boldsymbol{r})$
  $= \displaystyle\int_{E'=E_{\min}}^{E_{\max}} \left(\frac{\chi_p(\boldsymbol{r},E)}{4\pi}(1-\beta(\boldsymbol{r},E')) + \sum_{\ell=1}^{L}\chi_{d,\ell}(\boldsymbol{r},E)\beta_\ell(\boldsymbol{r},E')\right)(\nu\sigma_f)(\boldsymbol{r},E')\phi(\boldsymbol{r},E')dE'$
  The use of (1.12) yields the final expression

$$F_0\psi(\boldsymbol{r},\boldsymbol{\omega},E) = \int_{E'=E_{\min}}^{E_{\max}} \chi(\boldsymbol{r},E)(\nu\sigma_f)(\boldsymbol{r},E')\phi(\boldsymbol{r},E')dE',$$

  where $\chi(\boldsymbol{r},E) := \left(\dfrac{\chi_p(\boldsymbol{r},E)}{4\pi}(1-\beta(\boldsymbol{r},E')) + \displaystyle\sum_{\ell=1}^{L}\chi_{d,\ell}(\boldsymbol{r},E)\beta_\ell(\boldsymbol{r},E')\right)$ is the so-called total spectrum that accounts for the global fission reaction rate regardless of the prompt or delayed origin of the fission.

Note that equation (1.13) is a homogeneous problem and that $\psi\equiv 0$ is a solution to it. For a given reactor geometry $\mathcal{R}$, the flux $\psi\equiv 0$ is in general the unique solution to this problem. Indeed, for $\mathcal{R}$ given, only very particular distributions of cross-sections will lead to non trivial values of ψ that satisfy (1.13). This seems to be in contradiction with the real physical situation in which the stationary reactor has a non zero flux, but we have to keep in mind that, in practice, the core evolves very slowly and never fully reaches stable conditions, i.e. the flux does not totally satisfy relation (1.13) and there is no contradiction.

We are therefore led to the search of a non zero flux that could be considered as a representation of the system under nearly steady conditions. For this purpose, problem (1.13) is "relaxed" into a generalized eigenvalue problem of the form:

$$\begin{cases} \text{Find } (\xi,\psi) \text{ such that:} \\ (L_0 - H_0)\psi(\boldsymbol{r},\boldsymbol{\omega},E) = \xi F_0\psi(\boldsymbol{r},\boldsymbol{\omega},E) \\ C_\ell(0,\boldsymbol{r}) = \dfrac{1}{\lambda_\ell}\displaystyle\int_{E'=E_{\min}}^{E_{\max}} \beta_\ell(0,\boldsymbol{r},E')(\nu\sigma_f)(0,\boldsymbol{r},E')\phi(\boldsymbol{r},E')dE', \forall \ell\in\{1,\dots,L\}, \\ \psi = 0 \text{ in } \Gamma_-, \end{cases} \tag{1.14}$$

where ξ is the generalized eigenvalue associated to the eigenvector ψ (that is non trivial, by definition of an eigenvector). Note that if 1 belongs to the spectrum of this problem, then the associated eigenvector ψ will be the stationary non trivial flux that we are originally looking for in equation (1.13). In this case, the problem is said to be *critical*. If $\xi\neq 1$, it means that the system cannot be stationary and the only information that we will obtain from problem (1.14) in this case is that

the system would have been critical if the fission term had been ξ times higher (we refer to [29] for a discussion about this issue). For this reason, the system under consideration is slightly modified by adjusting/rescaling the fission cross-section: if we consider exactly the same system but with a fission cross-section distribution

$$\widetilde{\nu\sigma_f} := \xi \nu\sigma_f, \tag{1.15}$$

then this system admits an exact steady state solution given by:

$$\begin{cases} (L_0 - H_0 - \tilde{F}_0)\psi(\boldsymbol{r}, \boldsymbol{\omega}, E) = 0 \\ \psi = 0 \text{ in } \Gamma_-, \end{cases} \tag{1.16}$$

where

$$\tilde{F}_0\psi(\boldsymbol{r}, \boldsymbol{\omega}, E) = \int_{E'=E_{\min}}^{E_{\max}} \chi(\boldsymbol{r}, E)(\widetilde{\nu\sigma_f})(\boldsymbol{r}, E')\phi(\boldsymbol{r}, E')dE'.$$

The evolution of this new system is given by:

$$\begin{cases} \dfrac{1}{|\boldsymbol{v}|}\partial_t\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) + (L - H - \tilde{F} - \tilde{Q})\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) = S(t, \boldsymbol{r}, \boldsymbol{\omega}, E) \\ \partial_t C_\ell(t, \boldsymbol{r}) = -\lambda_\ell C_\ell(t, \boldsymbol{r}) \\ \qquad\qquad + \displaystyle\int_{E'=E_{\min}}^{E_{\max}} \beta_\ell(t, \boldsymbol{r}, E')(\widetilde{\nu\sigma_f})(t, \boldsymbol{r}, E')\phi(t, \boldsymbol{r}, E')dE', \ \forall \ell \in \{1, \dots, L\}, \end{cases} \tag{1.17}$$

where the operators L, H, $\tilde{F}$, $\tilde{Q}$ are exactly the same ones as defined for equation (1.1) except that $\tilde{F}$ and $\tilde{Q}$ use $\widetilde{\nu\sigma_f}$ instead of $\nu\sigma_f$.

Since, in general, the resolution of (1.14) provides a value of ξ that is close to 1, it is commonly accepted that the system that incorporates the rescaling in the fission cross-section (equation (1.17)) is representative enough of the initial system under consideration (1.1). As a consequence, problem (1.17) is the one that is solved in practice because its initial condition corresponds to an exact, well defined stationary state.

Remark 1.2.1. ***The rescaling given in (1.15) has been performed in the MINARET solver and we therefore solve problem (1.17) with the initial conditions ψ^0 and $C_{\ell,0}$ given by (1.14). To simplify the notations of section 1.3 and following, $\widetilde{\nu\sigma_f}$ will be written without the tilde.***

But, what are the spectral properties of problem (1.14)? Does the spectrum lie in $\mathbb{C}$ or are all the eigenvalues real? What can be said about the associated eigenvectors ψ? Among all the eigenvectors, which one corresponds to the real, physical stationary flux? There exists many references in the literature that deal with this issue and the summary of the results go much beyond the scope of the present manuscript (we refer to [88] for an extensive study on this topic). However, as an example of what can be found in the literature, we recall in the following section a result that, under several hypothesis, provides an answer to some of these questions, especially the question corresponding to finding the stationary flux ψ^0 among all the possible eigenvectors. The main idea is that the solution of (1.1) is an angular flux that must be positive for obvious physical reasons. The theorem recalled in section 1.2.2 states that there exists a unique positive eigenvector of problem (1.14) and that it is associated to the greatest eigenvalue ξ_{eff} in modulus, which is single and positive. The stationary flux ψ^0 will be this eigenvector.

Remark 1.2.2. *In reactor physics, the inverse of ξ_{eff} is called the multiplication factor or k-effective:*

$$k_{eff} := \frac{1}{\xi_{eff}}.$$

1.2.2 Existence and uniqueness of the stationary flux

In order to recall the result about the existence and uniqueness of a unique positive eigenvector ψ^0 of problem (1.14), we need to derive an equivalent expression of (1.14) with the variables $(\boldsymbol{r}, \boldsymbol{v})$. We start from the time dependent equation (1.10) written in the phase space $(t, \boldsymbol{r}, \boldsymbol{v})$:

$$\begin{cases} \dfrac{\partial \Psi}{\partial t}(t,\boldsymbol{r},\boldsymbol{v}) + \boldsymbol{v}.\nabla\Psi(t,\boldsymbol{r},\boldsymbol{v}) + \sigma(t,\boldsymbol{r},\boldsymbol{v})\Psi(t,\boldsymbol{r},\boldsymbol{v}) \\ \qquad - \int_{\mathcal{V}} \tilde{f}(\boldsymbol{r},\boldsymbol{v}'.\boldsymbol{v})\Psi(t,\boldsymbol{r},\boldsymbol{v}')d\boldsymbol{v}' - \sum_{\ell=1}^{L} \lambda_\ell c_\ell(t,\boldsymbol{r},\boldsymbol{v}) = 0 \\ \dfrac{\partial c_\ell}{\partial t}(t,\boldsymbol{r},\boldsymbol{v}) = -\lambda_\ell c_\ell(t,\boldsymbol{r},\boldsymbol{v}) + \int_{\mathcal{V}} f_\ell(t,\boldsymbol{r},\boldsymbol{v},\boldsymbol{v}')\Psi(t,\boldsymbol{r},\boldsymbol{v}')d\boldsymbol{v}'. \end{cases}$$

Its stationary version reads:

$$\begin{cases} \boldsymbol{v}.\nabla\Psi(\boldsymbol{r},\boldsymbol{v}) + \sigma(\boldsymbol{r},\boldsymbol{v})\Psi(\boldsymbol{r},\boldsymbol{v}) = \\ \qquad \int_{\mathcal{V}} f(\boldsymbol{r},\boldsymbol{v}'.\boldsymbol{v})\Psi(\boldsymbol{r},\boldsymbol{v}')d\boldsymbol{v}' + \int_{\mathcal{V}} \tilde{\sigma}_f(\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})\Psi(\boldsymbol{r},\boldsymbol{v}')d\boldsymbol{v}', \end{cases} \tag{1.18}$$

where we have used the definitions of (1.9). Problem (1.14) can therefore be written as:

$$\begin{cases} \boldsymbol{v}.\nabla\Psi(\boldsymbol{r},\boldsymbol{v}) + \sigma(\boldsymbol{r},\boldsymbol{v})\Psi(\boldsymbol{r},\boldsymbol{v}) = \\ \qquad \int_{\mathcal{V}} f(\boldsymbol{r},\boldsymbol{v}'.\boldsymbol{v})\Psi(\boldsymbol{r},\boldsymbol{v}')d\boldsymbol{v}' + \xi \int_{\mathcal{V}} \tilde{\sigma}_f(\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})\Psi(\boldsymbol{r},\boldsymbol{v}')d\boldsymbol{v}', \\ \Psi = 0 \text{ over } \Gamma_-. \end{cases} \tag{1.19}$$

The following theorem ensures the existence of a unique positive flux solution Ψ^0 (called critical flux). This solution is associated with the greatest eigenvalue $\xi_{eff} = 1/k_{eff}$ in module and this eigenvalue is positive, real and simple.

Theorem 1.2.3 (Theorem 1.2.1 of [8])**.**
Assume that:

- $\sigma \in L^\infty(\mathcal{R}, \mathcal{V})$,
- $f(\boldsymbol{r}, \boldsymbol{v}', \boldsymbol{v})$ *and* $\tilde{\sigma}_f(\boldsymbol{r}, \boldsymbol{v}', \boldsymbol{v})$ *are real, positive, measurable over* $\boldsymbol{v}$ *and* $\boldsymbol{v}'$ *and there exits positive constants* M_a, M_b *such that:*

$$\begin{cases} \int_{\mathcal{V}} \tilde{\sigma_f}(\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})d\boldsymbol{v}' + \int_{\mathcal{V}} f(\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})d\boldsymbol{v}' \le M_a, & \forall(\boldsymbol{r},\boldsymbol{v}) \in \mathcal{R}\times\mathcal{V} \\ \int_{\mathcal{V}} \tilde{\sigma_f}(\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})d\boldsymbol{v} + \int_{\mathcal{V}} f(\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})d\boldsymbol{v} \le M_b, & \forall(\boldsymbol{r},\boldsymbol{v}') \in \mathcal{R}\times\mathcal{V} \end{cases}$$

- σ *and* f *verify almost everywhere in* $(\boldsymbol{r}, \boldsymbol{v}) \in \mathcal{R} \times \mathcal{V}$*:*

$$\begin{cases} \sigma(\boldsymbol{r},\boldsymbol{v}) - \int_{\mathcal{V}} f(\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})d\boldsymbol{v}' \ge \alpha, \\ \sigma(\boldsymbol{r},\boldsymbol{v}) - \int_{\mathcal{V}} f(\boldsymbol{r},\boldsymbol{v}',\boldsymbol{v})d\boldsymbol{v} \ge \alpha, \end{cases} \qquad \text{for a given } \alpha > 0.$$

Let $1 < p < \infty$*. Then, problem (1.19) has a countable number of eigenvalues and eigenvectors. The eigenvectors are elements of the Banach space*

$$W_p := \{u \in L^p(\mathcal{R}\times\mathcal{V}),\ \boldsymbol{v}.\nabla u \in L^p(\mathcal{R}\times\mathcal{V})\}.$$

Furthermore, if $\tilde{\sigma}_f$ *is strictly positive, then there exists a unique positive eigenvector of (1.19) that is associated with the greatest eigenvalue in modulus (denoted as* ξ_{eff}*) and this eigenvalue is single and positive.*

Remark 1.2.4. *The condition* $\tilde{\sigma}_f > 0$ *in theorem (1.2.3) is not entirely physical because* $\tilde{\sigma}_f = 0$ *in the reflector of the core. One can find elements in [88] that could lead to the statement of similar results but with the more realistic hypothesis* $\tilde{\sigma}_f \ge 0$*.*

1.3 Discretization of the time-dependent neutron transport equation

With the exception of some simple cases (see [107] for further references) where problem (1.1) can exactly be solved, the resolution of (1.1) needs to be numerically addressed and requires discretizations and approximations of the involved variables.

The aim of this section is to give a general overview of the usual **deterministic** techniques employed in the field of neutronics to address this issue. Special emphasis will be put on the methods that have been employed in practice in the MINARET solver.

Each approximation raises many interesting and fundamental questions. Among the more important ones stand:

- the well-posedness of the resulting equations once the variables have been discretized,
- the convergence to the original equation for an arbitrary high level of refinement.

Although the detailed answers to those questions go much beyond the scope of the present work and that some of the problems are even nowadays still the subject of active research, in the remaining of this section, a special effort will be done to highlight these issues.

Remark 1.3.1. *As already announced in remark 1.2.1 of section 1.2.1, we remind that, in the rest of this chapter, we will consider the kinetic transport equation that includes the "rescaling" of the fission term given by relation (1.15). To simplify the notations, we will omit the tilde in the quantity* $\widetilde{\nu\sigma_f}$.

1.3.1 Discretization of the time variable

Although any kind of time discretization method could be applied, the simple θ-scheme is the most commonly used technique for the resolution of equation (1.1). Since the aim of the present work has not been the exploration of innovative time discretization techniques, we will only focus on the traditional Euler backward scheme.

In the remaining, the index n will refer to time. We will denote as $[0,T] = \bigcup_{n=0}^{N-1}[t_n, t_{n+1}]$ the division of the full time interval and $\Delta T_{n+1} = t_{n+1} - t_n$.

1.3.2 Discretization of the energy variable

The discretization of the energy variable in the neutron transport equation is intimately linked to the problem of the efficient approximation of the cross-sections over the whole interval $[E_{\min}, E_{\max}]$, which is a challenging task due to the high oscillations of the cross-sections in the resonance domain. Such a task has traditionally been treated by homogenization of the cross-sections and leads to the multigroup approximation (see section 1.3.2.1). This is the approach that has been used in MINARET but we note that other alternatives exist and an overview of them is given in section 1.3.2.2.

1.3.2.1 The multigroup approximation

The most common discretization of the energy variable is the *multigroup approximation*. It is based on the division of the energy interval into G subintervals $[E_{\min}, E_{\max}] = [E_G, E_{G-1}] \cup \cdots \cup [E_1, E_0]$, with $E_{\min} = E_G < E_{G-1} < \cdots < E_0 = E_{\max}$. We denote $I_g = [E_g, E_{g-1}]$ for any $g \in \{1, \ldots, G\}$. On each interval, the angular flux is supposed to be the product of a function of energy $h^g(E)$ and a multigroup flux $\psi^g(t, \boldsymbol{r}, \boldsymbol{\omega})$ such that:

$$\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) \approx h^g(E)\psi^g(t, \boldsymbol{r}, \boldsymbol{\omega}), \ \forall E \in I_g, \tag{1.20}$$

where

$$\int_{I_g} h^g(E)dE = 1.$$

The knowledge of the weighting function $h^g(E)$ is in general not straightforward and it is a crucial step in order to obtain accurate and reliable results with the multigroup approximation (see, e.g., [91]). If I_g belongs to the the resonance interval of a given cross-section, the derivation of the weighting function $h^g(E)$ becomes particularly involved given the highly oscillatory behavior of the cross-sections in that interval (an example of this is given in figure 1.1). The methods dealing with this problem are called self-shielding techniques (see, e.g., [28]) and they are essentially based on homogenization strategies in a medium with strong spatial and geometrical simplifications.

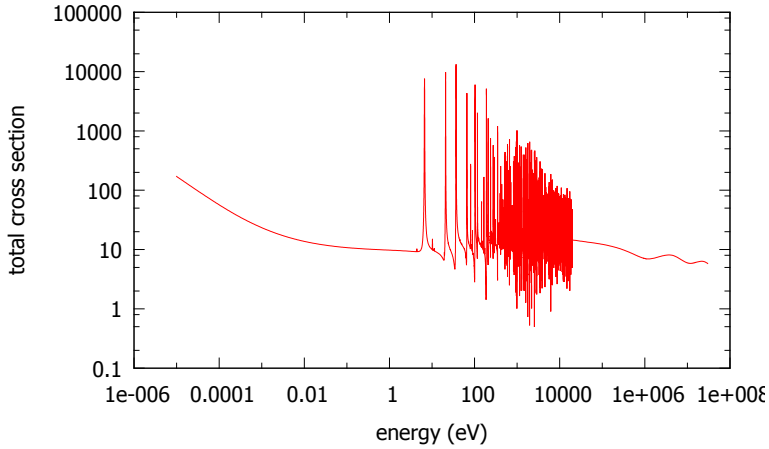


Figure 1.1: Total cross-section σ_t of the U_{92}^{238} atom as a function of the energy. Note the resonance interval for $E \in [1\ eV;\ 10^5\ eV]$.

In the present work, the output of these techniques (multigroup cross-sections and kinetic parameters) will be taken as a given parameter. They read:

$$\begin{cases} \sigma_t^g(t,\boldsymbol{r}) = \displaystyle\int_{I_g} h^g(E)\sigma_t(t,\boldsymbol{r},E)dE \\ (\nu\sigma_f)^g(t,\boldsymbol{r}) = \displaystyle\int_{I_g} h^g(E)(\nu\sigma_f)(t,\boldsymbol{r},E)dE \\ \sigma_s^{g'\to g}(t,\boldsymbol{r},\boldsymbol{\omega}'\to\boldsymbol{\omega}) = \displaystyle\int_{I_g} h^g(E)dE \int_{I_g} \sigma_s(t,\boldsymbol{r},\boldsymbol{\omega}'\to\boldsymbol{\omega},E'\to E)h^{g'}(E')dE' \\ \chi_p^g(t,\boldsymbol{r}) = \displaystyle\int_{I_g} \chi_p(t,\boldsymbol{r},E)h^g(E)dE \quad \text{and} \quad \chi_{d,l}^g(t,\boldsymbol{r}) = \int_{I_g} \chi_{d,l}(t,\boldsymbol{r},E)h^g(E)dE \\ \beta^g(t,\boldsymbol{r}) = \displaystyle\int_{I_g} \beta(t,\boldsymbol{r},E)h^g(E)dE \quad \text{and} \quad \beta_\ell^g(t,\boldsymbol{r}) = \int_{I_g} \beta_\ell(t,\boldsymbol{r},E)h^g(E)dE \end{cases}$$

This multigroup approximation combined with the Euler backward scheme yields the following set of source problems:

$$\begin{cases} \text{For } n \in \{0,\dots,N-1\}: \\ \text{given } \psi^{g,n}(\boldsymbol{r},\boldsymbol{\omega}) \text{ for any } g \in \{1,\dots,G\}, \\ \text{find over } \mathcal{R}\times\mathbb{S}_2 \text{ the angular flux } \psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) \text{ that is the solution of:} \\ \left(L^g - H^g - \tilde{F}^g - \tilde{Q}^g\right)\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \tilde{S}^{g,n}(\boldsymbol{r},\boldsymbol{\omega}), \quad \forall g \in \{1,\dots,G\}, \end{cases} \tag{1.21}$$

where $\psi^{g,n}(\boldsymbol{r},\boldsymbol{\omega})$ is the approximation of $\psi(t,\boldsymbol{r},\boldsymbol{\omega},E)$ at time $t=t_n$ and for $E \in I_g$. The following notations have been used:

- $\tilde{S}^{g,n}(\boldsymbol{r},\boldsymbol{\omega}) := \dfrac{\psi^{g,n}(\boldsymbol{r},\boldsymbol{\omega})}{V^g \Delta T_{n+1}}$, where V^g is the average velocity of the neutrons whose energy belong to the interval I_g. Note that for the computation of $\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega})$, the term $\tilde{S}^{g,n}(\boldsymbol{r},\boldsymbol{\omega})$ is known and is a source for the equation.
- $L^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \left(\boldsymbol{\omega}.\nabla + \left(\sigma_t^{g,n+1}(\boldsymbol{r}) + \dfrac{1}{V^g \Delta T_{n+1}}\right)\right)\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega})$
- $H^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \sum\limits_{g'=1}^{G} H^{g'\to g}\psi^{g',n+1}(\boldsymbol{r},\boldsymbol{\omega})$, with
  $H^{g'\to g}\psi^{g',n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \int_{\mathbb{S}_2} \sigma_s^{g'\to g,n+1}(\boldsymbol{r},\boldsymbol{\omega}'\to\boldsymbol{\omega})\psi^{g',n+1}(\boldsymbol{r},\boldsymbol{\omega'})d\boldsymbol{\omega'}$.
- $\tilde{F}^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \dfrac{\chi_p^{g,n+1}(\boldsymbol{r})}{4\pi}\sum\limits_{g'=1}^{G}\left(1-\beta^{g',n+1}(\boldsymbol{r})\right)(\nu\sigma_f)^{g',n+1}(\boldsymbol{r})\phi^{g',n+1}(\boldsymbol{r})$
- $\tilde{Q}^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \sum\limits_{\ell=1}^{L}\lambda_\ell\chi_{d,\ell}^{g,n+1}(\boldsymbol{r})C_\ell^{n+1}(\boldsymbol{r})$

We invoke theorem 4 of [33] (chapter XXI, section 2, paragraph 4) for the existence and uniqueness of a multigroup solution to equation (1.21) and also for the convergence of this equation to the continuous problem as the number G of energy groups increases.

Since we are considering an Euler backward scheme, for the precursors, we have for any $\ell \in \{1,\dots,L\}$:

$$C_\ell^{n+1}(\boldsymbol{r}) = \frac{1}{1+\lambda_\ell\Delta T_{n+1}}C_\ell^n(\boldsymbol{r}) + \frac{\Delta T_{n+1}}{1+\lambda_\ell\Delta T_{n+1}}\sum_{g'=1}^{G}\beta_\ell^{g',n+1}(\boldsymbol{r})(\nu\sigma_f)^{g',n+1}(\boldsymbol{r})\phi^{g',n+1}(\boldsymbol{r}). \tag{1.22}$$

If we insert relation (1.22) in equation (1.21), the set of source problems reads:

$$\begin{cases} \text{For } n\in\{0,\dots,N-1\}: \\ \text{given } \psi^{g,n}(\boldsymbol{r},\boldsymbol{\omega}) \text{ for any } g\in\{1,\dots,G\}, \\ \text{find over } \mathcal{R}\times\mathbb{S}_2 \text{ the angular flux } \psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) \text{ that is the solution of:} \\ \left(L^g - H^g - F^g\right)\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}), \quad \forall g\in\{1,\dots,G\}, \end{cases} \tag{1.23}$$

where:

- $F^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \sum\limits_{g'=1}^{G}F^{g',g}\psi^{g',n+1}$ and

$$F^{g',g}\psi^{g',n+1} = \left(\frac{\chi_p^{g,n+1}(\boldsymbol{r})}{4\pi}\left(1-\beta^{g',n+1}(\boldsymbol{r})\right) + \sum_{\ell=1}^{L}\frac{\lambda_\ell\beta_\ell^{g',n+1}(\boldsymbol{r})\chi_{d,\ell}^{g,n+1}(\boldsymbol{r})\Delta T_{n+1}}{1+\lambda_\ell\Delta T_{n+1}}\right)(\nu\sigma_f)^{g',n+1}(\boldsymbol{r})\phi^{g',n+1}(\boldsymbol{r}),$$

- $S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}) := \dfrac{\psi^{g,n}(\boldsymbol{r},\boldsymbol{\omega})}{V^g\Delta T_{n+1}} + \dfrac{1}{1+\lambda_\ell\Delta T_{n+1}}C_\ell^n(\boldsymbol{r})$.

For a given time t_{n+1}, the resulting set of equations is coupled with respect to the energy groups and problem (1.23) can be summarized by the following matrix system:

$$A_{G,G}\boldsymbol{\psi^{n+1}} = \boldsymbol{S^n}, \tag{1.24}$$

where

$$A_{G,G} = \begin{pmatrix} L^1 - H^{1\to1} - F^{1,1} & -H^{2\to1} - F^{2,1} & \dots & -H^{G\to1} - F^{G,1} \\ -H^{1\to2} - F^{1,2} & L^2 - H^{2\to2} - F^{2,2} & \dots & -H^{G\to2} - F^{G,2} \\ \vdots & \vdots & \ddots & \vdots \\ -H^{1\to G} - F^{1,G} & -H^{2\to G} - F^{2,G} & \dots & L^G - H^{G\to G} - F^{G,G} \end{pmatrix} \tag{1.25}$$

and

$$\boldsymbol{\psi^{n+1}} = \begin{pmatrix} \psi^{1,n+1} \\ \psi^{2,n+1} \\ \vdots \\ \psi^{G,n+1} \end{pmatrix} \quad ; \quad \boldsymbol{S^n} = \begin{pmatrix} S^{1,n} \\ S^{2,n} \\ \vdots \\ S^{G,n} \end{pmatrix}. \tag{1.26}$$

Numerical methods The inversion of the system (1.24) of equations is performed by iterative techniques. The most common one is the Gauss-Seidel numerical scheme. In MINARET, a slight modification of it (that we call "generalized Gauss-Seidel") has been implemented for numerical storage issues. Let $\psi_{(M)}^{g,n+1}$ be the approximation of $\psi^{g,n+1}$ at iteration number M with this scheme. Our implemented scheme reads:

$$M_{G,G}\boldsymbol{\psi_{(M+1)}^{n+1}} = N_{G,G}\boldsymbol{\psi_{(M)}^{n+1}} + \boldsymbol{S^n}, \tag{1.27}$$

where $A_{G,G} = M_{G,G} - N_{G,G}$, with

$$M_{G,G} = \begin{pmatrix} L^1 - H^{1\to 1} & 0 & \cdots & 0 \\ -H^{1\to 2} & L^2 - H^{2\to 2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -H^{1\to G} & -H^{2\to G} & \cdots & L^G - H^{G\to G} \end{pmatrix} \tag{1.28}$$

and

$$N_{G,G} = \begin{pmatrix} F^{1,1} & H^{2\to 1} + F^{2,1} & \cdots & H^{G\to 1} + F^{G,1} \\ F^{1,2} & F^{2,2} & \cdots & H^{G\to 2} + F^{G,2} \\ \vdots & \vdots & \ddots & \vdots \\ F^{1,G} & F^{2,G} & \cdots & F^{G,G} \end{pmatrix}. \tag{1.29}$$

Note that the difference between this scheme and a traditional Gauss-Seidel lies in the "explicit" treatment of the fission terms $F^{g',g}$ for $g' \leq g$. With this scheme, for a given energy group g, the problem to be solved reads:

$$\begin{aligned} &(L^g - H^{g\to g})\psi_{(M+1)}^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) \\ &\quad = \sum_{g'<g} H^{g'\to g}\psi_{(M+1)}^{g',n+1} + \sum_{g'>g} H^{g'\to g}\psi_{(M)}^{g',n+1} + \sum_{g'=1}^{G} F^{g',g}\psi_{(M)}^{g',n+1} + S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}). \end{aligned} \tag{1.30}$$

Note that equation (1.30) is a monoenergetic problem of the form:

$$\boldsymbol{\omega}.\nabla\psi(\boldsymbol{r},\boldsymbol{\omega}) + \sigma_t(\boldsymbol{r})\psi(\boldsymbol{r},\boldsymbol{\omega}) - \int_{\mathbb{S}_2} \sigma_s(\boldsymbol{r},\boldsymbol{\omega'}\to\boldsymbol{\omega})\psi(\boldsymbol{r},\boldsymbol{\omega'})d\boldsymbol{\omega'} = q(\boldsymbol{r},\boldsymbol{\omega}),\ \forall(\boldsymbol{r},\boldsymbol{\omega}) \in \mathcal{R}\times\mathbb{S}_2, \tag{1.31}$$

where the terms σ_t, σ_s, q must be understood as generic notations whose definition must be coherent with equation (1.30). Equation (1.31) is still a function of $\boldsymbol{\omega}$ and $\boldsymbol{r}$. In the following sections, we will discuss about the methods to discretize these variables but let us first outline other strategies for the discretization of the energy variable.

1.3.2.2 Other approaches

The most important drawback of the multigroup approximation lies in the fact that it considers the cross-sections and the flux to be constant within a group.

A first alternative to address this issue was suggested in [3]. It is based on a finite element approach for the energy variable and presents the additional advantage that it does not require the self-shielding stage [1]. Like in the multigroup approximation, the energy interval is divided into groups but a Galerkin projection of the angular flux ψ into an M_g dimensional polynomial basis for each interval $I_g = [E_g, E_{g-1}]$ is carried out. The approximation reads: $\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) \approx \sum_{i=1}^{M_g} \psi^i(t, \boldsymbol{r}, \boldsymbol{\omega}) f_i^g(E)$, where f_i^g is a polynomial of the finite element basis defined over I_g and $\psi^i(t, \boldsymbol{r}, \boldsymbol{\omega})$ is the i-th flux mode within a group. For each interval I_g, the resulting equations for the flux modes will be coupled and this is the main weakness of the method. Indeed, given the highly oscillatory behavior of the cross-sections in the resonance domain, high dimensional polynomial basis will be required for this region. As a result, the technique is very computationally costly because it leads to a coupled system whose global complexity is of order $\mathcal{O}(M_g^2 G)$ (the complexity in each interval being $\mathcal{O}(M_g^2)$). For this reason, this method is seldom used in practice.

As a remedy to this, [70] and [121] have recently proposed a wavelet-Galerkin method in which the traditional polynomial basis is replaced by compactly supported Daubechies wavelets [32]. Although both approximations would need the same dimension M_g to approximate the flux with the same level of accuracy, the main advantage of the wavelets is that the resulting system is sparse. We refer to [48] for an overview of the main results in neutronics in this respect.

A somehow intermediate energy discretization approach is the so-called probability table method (see [108]). After dividing the energy interval into G subintervals I_g, we consider the variations of a given cross section σ in I_g. Assuming that $\sigma \in [\sigma_{\max}, \sigma_{\min}]$ in I_g, let $[\sigma_{\max}, \sigma_{\min}] = \bigcup_{i=1}^{I} [\sigma_i, \sigma_{i+1}]$ be a division of this interval. The probability table method assigns a couple $\{p_i, \overline{\sigma}_i\}$ to each subinterval $[\sigma_i, \sigma_{i+1}]$ defined as

$$\overline{\sigma}_i = \frac{\int_{I_g} \sigma(E) \mathbb{1}_{[\sigma_i, \sigma_{i+1}]} dE}{E_{g-1} - E_g} \quad ; \quad p_i = \frac{\int_{I_g} \mathbb{1}_{[\sigma_i, \sigma_{i+1}]} dE}{E_{g-1} - E_g}$$

and a flux $\psi^i(t, \boldsymbol{r}, \boldsymbol{\omega})$ is sought for each $[\sigma_i, \sigma_{i+1}]$. This approximation of the flux is considered to be more accurate than the traditional multi-group approximation given the refinement in the cross-section variable. The mean value $\sum_{i=1}^{I} \psi^i(t, \boldsymbol{r}, \boldsymbol{\omega}) p_i$ of this method would therefore be the analogue of the multigroup flux $\psi^g(t, \boldsymbol{r}, \boldsymbol{\omega})$ for the interval I_g.

Remark 1.3.2. *To the best of the author's knowledge, the probability table method has only been applied in core calculations in a 1D transport code called SN1D ([74]).*

1.3.3 Discretization of the angular variable

We will explain hereafter the two major techniques for the discretization of the angular variable. The monoenergetic equation (1.31) will be taken as a starting point given that it is the problem that arises in the numerical resolution of the multigroup equations. We will work under the hypothesis of isotropic scattering, i.e., the scattering cross-section depends on the angular variable only by the cosine of the incidental and scattered directions [2] $\sigma_s(\boldsymbol{r}, \boldsymbol{\omega}' \to \boldsymbol{\omega}) \approx \sigma_s(\boldsymbol{r}, \boldsymbol{\omega}'.\boldsymbol{\omega})$. The equation becomes:

$$\boldsymbol{\omega}.\nabla\psi(\boldsymbol{r}, \boldsymbol{\omega}) + \sigma_t(\boldsymbol{r})\psi(\boldsymbol{r}, \boldsymbol{\omega}) - \int_{\mathbb{S}_2} \sigma_s(\boldsymbol{r}, \boldsymbol{\omega}'.\boldsymbol{\omega})\psi(\boldsymbol{r}, \boldsymbol{\omega}')d\boldsymbol{\omega}' = q(\boldsymbol{r}, \boldsymbol{\omega}), \ \forall (\boldsymbol{r}, \boldsymbol{\omega}) \in \mathcal{R} \times \mathbb{S}_2. \quad (1.32)$$

1. Since the self-shielding calculation is carried out under strong spatial and geometrical simplifications, it is an important source of errors in deterministic calculations.
2. The hypothesis of isotropic scattering physically means that neutrons are scattered with no preferred direction.

Although equation (1.32) could be taken as the starting point, it is standard to develop the scattering cross-section in Legendre polynomials (denoted as P_l) that are an $L^2([-1;1])$ Hilbert basis. The reason for doing this will appear clearer once the development will be introduced. We will have:

$$\sigma_s(\boldsymbol{r}, \boldsymbol{\omega'}.\boldsymbol{\omega}) = \frac{1}{4\pi}\sum_{l=0}^{\infty}(2l+1)\sigma_{s,l}(\boldsymbol{r})P_l(\boldsymbol{\omega'}.\boldsymbol{\omega}),$$

where

$$\sigma_{s,l}(\boldsymbol{r}) = 2\pi\int_{-1}^{1}\sigma_s(\boldsymbol{r},\mu)P_l(\mu)d\mu.$$

The scattering term becomes:

$$\int_{\mathbb{S}_2}\sigma_s(\boldsymbol{r},\boldsymbol{\omega}'.\boldsymbol{\omega})\psi(\boldsymbol{r},\boldsymbol{\omega'})d\boldsymbol{\omega'} = \sum_{l=0}^{\infty}\frac{2l+1}{4\pi}\sigma_{s,l}(\boldsymbol{r})\int_{\mathbb{S}_2}\psi(\boldsymbol{r},\boldsymbol{\omega'})P_l(\boldsymbol{\omega'}.\boldsymbol{\omega})d\boldsymbol{\omega'}.$$

The addition theorem $P_l(\boldsymbol{\omega'}.\boldsymbol{\omega}) = \dfrac{4\pi}{2l+1}\sum\limits_{m=-l}^{+l}Y_{lm}^*(\boldsymbol{\omega'})Y_{lm}(\boldsymbol{\omega})$ is then used to relate the Legendre polynomials with the spherical harmonics $Y_{l,m}(\boldsymbol{\omega})$. This is done with the aim of separating the directions $\boldsymbol{\omega}$ and $\boldsymbol{\omega'}$. We finally get:

$$\int_{\mathbb{S}_2}\sigma_s(\boldsymbol{r},\boldsymbol{\omega}'.\boldsymbol{\omega})\psi(\boldsymbol{r},\boldsymbol{\omega'})d\boldsymbol{\omega'} = \sum_{l=0}^{\infty}\sigma_{s,l}(\boldsymbol{r})\sum_{m=-l}^{+l}\phi_{l,m}(\boldsymbol{r})Y_{l,m}(\boldsymbol{\omega}), \tag{1.33}$$

where

$$\phi_{l,m}(\boldsymbol{r}) := \int_{\mathbb{S}_2}\psi(\boldsymbol{r},\boldsymbol{\omega'})Y_{l,m}(\boldsymbol{\omega'})d\boldsymbol{\omega'} \tag{1.34}$$

are the flux moments. In practical applications, the infinite sum over ℓ is truncated at a given order N, yielding to the so-called P_N expansion of the scattering term. Note from formula (1.33) that the P_N expansion involves scattering cross-section data that are independent of the directions $\boldsymbol{\omega'}$ and $\boldsymbol{\omega}$. This results in reduced and simplified storage data regarding this cross-section and is the main reason why this development has traditionally (and still nowadays) been used. The resulting equation reads:

$$\boldsymbol{\omega}.\nabla\psi(\boldsymbol{r},\boldsymbol{\omega}) + \sigma_t(\boldsymbol{r})\psi(\boldsymbol{r},\boldsymbol{\omega}) - \sum_{l=0}^{N}\sigma_{s,l}(\boldsymbol{r})\sum_{m=-l}^{+l}\phi_{l,m}(\boldsymbol{r})Y_{l,m}(\boldsymbol{\omega}) = q(\boldsymbol{r},\boldsymbol{\omega}),\ \forall(\boldsymbol{r},\boldsymbol{\omega})\in\mathcal{R}\times\mathbb{S}_2. \tag{1.35}$$

1.3.3.1 The discrete ordinates method (S_N method)

The discrete-ordinates version of equation (1.35) is obtained by solving the transport equation along discrete directions and by replacing the integrals over the unit sphere $\mathbb{S}_2$ by quadratures. The quadrature rule associated to the S_N approximation is composed of a set $Q_D = \{\boldsymbol{\omega}_d \in \mathbb{S}_2,\ 1 \leq d \leq D\}$ of $D = N(N+2)$ directions together with a set of associated weights $\{w_d,\ 1 \leq d \leq D\}$. The latter are chosen such that the following approximation is optimal in some sense:

$$\int_{\mathbb{S}_2} f(\boldsymbol{\omega})d\boldsymbol{\omega} \approx \sum_{d=1}^{D} w_d f(\boldsymbol{\omega}_d).$$

If we denote by ψ_d the approximation of ψ for the direction $\boldsymbol{\omega}_d$, the S_N approximation of equation (1.35) reads:

$$\begin{cases} \text{Find over } \mathcal{R} \text{ the angular flux } \psi_d(\boldsymbol{r}) \text{ that is the solution of:} \\ \boldsymbol{\omega}_d.\nabla\psi_d(\boldsymbol{r}) + \sigma_t(\boldsymbol{r})\psi_d(\boldsymbol{r}) \\ \qquad - \sum_{d'=1}^{D} \omega_{d'}\psi_{d'}(\boldsymbol{r}) \sum_{l=0}^{N} \sigma_{s,l}(\boldsymbol{r}) \sum_{m=-l}^{+l} Y_{l,m}(\boldsymbol{\omega}_{d'})Y_{l,m}(\boldsymbol{\omega}_d) = q_d(\boldsymbol{r}), \quad \forall d \in \{1,\dots,D\}, \end{cases} \tag{1.36}$$

where the flux moments are computed following the quadrature rule:

$$\phi_{l,m}(\boldsymbol{r}) \approx \sum_{d'=1}^{D} w_{d'}\psi_{d'}(\boldsymbol{r})Y_{l,m}(\boldsymbol{\omega}_{d'}).$$

Existence results and convergence rates One can find in the literature proofs about the well-posedness of equation (1.36). In these results, the choice of the quadrature rule is a critical point. We will mention here a result stated in [5] regarding a three-dimensional $L^2(\mathcal{R})$ error estimation between the scalar flux solution $\phi(\boldsymbol{r}) = \int_{\mathbb{S}_2} \psi(\boldsymbol{r},\boldsymbol{\omega'})d\boldsymbol{\omega'}$ of equation (1.32) and $\phi_D(\boldsymbol{r}) = \sum_{d=1}^{D} \omega_d\psi_d(\boldsymbol{r})$ of equation (1.36). Under the hypothesis that σ_s does not depend on the spatial variable and also supposing that the quadrature weights w_d are positive, we have:

$$\|\phi - \phi_D\|_{L^2(\mathcal{R})} \leq \frac{C}{\sqrt{D}} \left(\|\phi\|_{H^1(\mathcal{R})} + \|q\|_{H^1(\mathcal{R})} \right), \tag{1.37}$$

which ensures the convergence as the number of directions increases. Similar studies can be found for the supremum norm in, e.g., [119] and in [103] for $L^p(\mathcal{R})$, $1 \leq p < \infty$.

Some remarks about the quadrature rule The choice of the quadrature rule is a complex issue and is still nowadays an open question. Among the desirable properties of a quadrature formula stand:

- the use of positive weights $\omega_j > 0$ for stability and convergence issues,
- the ability to integrate as many spherical harmonics as possible,
- an even distribution of the directions Q_D,
- the rotational invariance under some symmetry group in the set of directions (this property is searched because there must be no favored direction).

To the best of the author's knowledge, there exists no quadrature rule that fulfills all these properties in $\mathbb{S}_2$. We nevertheless cite [2] for recent interesting advances on this topic where the authors are able to build quasi-uniformly distributed quadratures invariant under the icosahedral rotation group.

In the particular case of the MINARET solver, the so-called "Level Symmetric" quadrature is employed because the distribution of the angles is quite uniform and preserves planar symmetries and rotations between the axes. The weights are defined to integrate as many spherical harmonics as possible. However, as the number of directions increases, the positivity of the weights is no longer ensured (there exists counter-examples for $N > 16$, i.e. more than 288 directions) and other quadratures are required.

Negative weights can be avoided by using a product quadrature that combines two one-dimensional quadratures with positive weights. A very classical choice is a Gauss-Legendre quadrature for the cosine of the polar angle and a Chebyshev uniform quadrature for the azimuthal angle. A disadvantage of the product quadrature is that, for increasing values of the polar cosine, the quadrature directions concentrate around the polar axis and, as a consequence, the quadrature does not uniformly map $\mathbb{S}_2$. The solution could lie in the use of the quadratures of [2] for the resolution of our equation.

Some remarks about the S_N method The major drawback of the discrete ordinates method is the *ray-effect*, i.e. the presence of unphysical flux oscillations in certain situations when the number of directions is low. In general, this behavior occurs in problems with highly angular dependent sources and/or fluxes, or for spatially localized sources. In spite of this fact, discrete ordinates methods present the enormous advantage of leading to the resolution of a source iteration which is the iterative technique commonly employed to deal with the coupling with respect to the angular variable in the set of equations (1.36). If $\psi_{d,(m)}$ is the approximation of ψ_d at the m-th source iteration, then $\psi_{(m+1)}$ is the solution of:

$$\begin{aligned}\boldsymbol{\omega}_d.\nabla\psi_{d,(m+1)}(\boldsymbol{r}) + \sigma_t(\boldsymbol{r})\psi_{d,(m+1)}(\boldsymbol{r}) = \\ \sum_{d'=1}^{D}\omega_{d'}\psi_{d',(m)}(\boldsymbol{r})\sum_{l=0}^{N}\sigma_{s,l}(\boldsymbol{r})\sum_{k=-l}^{+l}Y_{l,k}(\boldsymbol{\omega}_{d'})Y_{l,k}(\boldsymbol{\omega}_d) + q_d(\boldsymbol{r})\end{aligned} \tag{1.38}$$

The major advantages of this numerical scheme are its easy implementation and that efficient sequential and parallel acceleration methods can be added, yielding to a rapid resolution of problem (1.36). More details about these acceleration techniques will be provided in section 1.6.

1.3.3.2 The spherical harmonics method (P_N method)

In this discretization, the angular component of $\psi(\boldsymbol{r},\boldsymbol{\omega})$ is expanded in spherical harmonics which is an orthonormal basis of $L^2(\mathbb{S}_2)$. The index N indicates the number of terms retained in the expansion, and, as N increases, the solution of the P_N equations converges to the solution of the transport equation. The expansion reads:

$$\psi(\boldsymbol{r},\boldsymbol{\omega}) \approx \sum_{\ell=0}^{N}\sum_{m=-\ell}^{+\ell}\phi_{\ell,m}(\boldsymbol{r})Y_{\ell,m}(\boldsymbol{\omega}),$$

where $\phi_{\ell,m}$ are the flux moments defined on equation (1.34). If we insert this formula in equation (1.32), we obtain:

$$\sum_{\ell=0}^{N}\sum_{m=-\ell}^{\ell}\left(\boldsymbol{\omega}.\nabla\phi_{l,m}(\boldsymbol{r}) + \sigma_t(\boldsymbol{r})\phi_{l,m}(\boldsymbol{r}) - \sigma_{s,\ell}(\boldsymbol{r})\phi_{l,m}(\boldsymbol{r}) - q_{l,m}(\boldsymbol{r})\right)Y_{\ell,m} = 0, \tag{1.39}$$

where we have expanded the source q and the scattering term into spherical harmonics following formula (1.33).

If we now project this equation onto each spherical harmonic component $Y_{l,m}$, we derive a set of $(N+1)^2$ coupled functions for the flux moments $\phi_{l,m}(\boldsymbol{r})$, $l \in \{0,\dots,N\}$, $m \in \{-l,\dots,+l\}$:

$$\sum_{k=0}^{N}\sum_{n=-\ell}^{\ell}\left(\int_{\mathbb{S}_2}\boldsymbol{\omega}Y_{k,n}(\boldsymbol{\omega})Y^*_{l,m}(\boldsymbol{\omega})d\boldsymbol{\omega}\right).\nabla\phi_{k,n}(\boldsymbol{r}) + \sigma_t(\boldsymbol{r})\phi_{l,m}(\boldsymbol{r}) - \sigma_{s,\ell}(\boldsymbol{r})\phi_{l,m}(\boldsymbol{r}) - q_{l,m}(\boldsymbol{r}) = 0,$$

where we have used the orthogonality of spherical harmonics:

$$\int_{\mathbb{S}_2}Y^*_{l,m}(\boldsymbol{\omega})Y_{k,n}(\boldsymbol{\omega})d\boldsymbol{\omega} = \delta_{l,k}\delta_{m,n}. \tag{1.40}$$

We are thus confronted to the evaluation of the integrals

$$\int_{\mathbb{S}_2}\boldsymbol{\omega}Y_{k,n}(\boldsymbol{\omega})Y^*_{l,m}(\boldsymbol{\omega})d\boldsymbol{\omega} = \begin{pmatrix}\displaystyle\int_{\mathbb{S}_2}\cos\varphi\sin\theta Y_{k,n}(\boldsymbol{\omega})Y^*_{l,m}(\boldsymbol{\omega})d\boldsymbol{\omega} \\ \displaystyle\int_{\mathbb{S}_2}\sin\varphi\sin\theta Y_{k,n}(\boldsymbol{\omega})Y^*_{l,m}(\boldsymbol{\omega})d\boldsymbol{\omega} \\ \displaystyle\int_{\mathbb{S}_2}\cos\theta Y_{k,n}(\boldsymbol{\omega})Y^*_{l,m}(\boldsymbol{\omega})d\boldsymbol{\omega}\end{pmatrix}, \tag{1.41}$$

where $\boldsymbol{\omega}$ has been expressed as a function of the polar angle $\theta \in [0, \pi]$ and the azimuthal angle $\varphi \in [0, 2\pi[$. By using the following recurrence formulas (whose derivation we omit in this manuscript) for the spherical harmonics in their complex formulation:

$$\begin{cases} \cos\theta Y_{l,m} = \sqrt{\dfrac{(l+m)(l-m)}{(2l-1)(2l+1)}} Y_{l-1,m} + \sqrt{\dfrac{(l+m+1)(l-m+1)}{(2l+1)(2l+3)}} Y_{l+1,m} \\ e^{i\varphi}\sin\theta Y_{l,m} = \sqrt{\dfrac{(l-m-1)(l-m)}{(2l-1)(2l+1)}} Y_{l-1,m+1} - \sqrt{\dfrac{(l+m+1)(l+m+2)}{(2l+1)(2l+3)}} Y_{l+1,m-1} \\ e^{-i\varphi}\sin\theta Y_{l,m} = -\sqrt{\dfrac{(l+m-1)(l+m)}{(2l-1)(2l+1)}} Y_{l-1,m-1} + \sqrt{\dfrac{(l-m+1)(l-m+2)}{(2l+1)(2l+3)}} Y_{l+1,m-1} \end{cases} \tag{1.42}$$

we derive the so-called P_N equations:

$$\left(\sum_{(k,n)\in\mathcal{P}_{l,m}} \boldsymbol{c}_{k,n}.\nabla\phi_{k,n}(\boldsymbol{r}) \right) + \sigma_t(\boldsymbol{r})\phi_{l,m}(\boldsymbol{r}) - \sigma_{s,\ell}(\boldsymbol{r})\phi_{l,m}(\boldsymbol{r}) - q_{l,m}(\boldsymbol{r}) = 0, \tag{1.43}$$

where $\mathcal{P}_{l,m} = \{(l \pm 1, m \pm 1); (l \pm 1, m)\}$. The exact form of the vector coefficients $\boldsymbol{c}_{k,n}$ will not be detailed here. The most important observation that follows from equation (1.43) is that the P_N equations are coupled but only with respect to the six "neighboring" moments given by $\mathcal{P}_{l,m}$. The most important drawback of this coupling in comparison with the resulting S_N equations is that the resolution of (1.43) is more involved than the source iteration. For this reason, and also because the formulation of boundary conditions is not straightforward and tedious in complex geometries, S_N methods are usually preferred in practical implementations.

1.3.3.3 Finite element discretization of the unit sphere

A somehow intermediate method between S_N and P_N would be to use a finite element discretization of the angular domain ($\mathbb{S}_2$). Like in the P_N method, this approach enables a continuous representation of the angular dependence and mitigates the ray-effects. On the other hand, it is a collocation technique like S_N.

In the context of neutron transport, this approach was explored several decades ago but, to the best of the author's knowledge, no current code uses this method. The method seems however to have a considerable interest among the radiative transfer community and we cite [15] for a recent paper on this topic.

1.3.4 Spatial discretization

System (1.38) – or, in a quite similar manner, system (1.43)– is a set of spatial problems whose form can be summarized by the following transport equation:

$$\boldsymbol{\omega}.\nabla\psi(\boldsymbol{r}) + \sigma_t(\boldsymbol{r})\psi(\boldsymbol{r}) = q(\boldsymbol{r}), \tag{1.44}$$

where the scattering term is included in the source q. We present hereafter two methods for its resolution that are traditionally employed in the field of neutronics.

1.3.4.1 The method of characteristics

This method is based on the exact integration of equation (1.44) along the trajectory given by the direction $\boldsymbol{\omega}$.

Let $\boldsymbol{r}_0 \in \partial\mathcal{R}$ and let $\gamma(t) = \boldsymbol{r}_0 + t\boldsymbol{\omega}$ for $t \in \mathbb{R}$. The set $\{(t, \gamma(t)) | t \in \mathbb{R}\}$ is a straight line in $\mathbb{R} \times \mathbb{R}^3$ called characteristic curve. It is straightforward to notice that

$$\boldsymbol{\omega}.\nabla\psi(\gamma(t)) = \frac{d}{dt}\psi(\gamma(t)).$$

It follows that equation (1.44) can be written as a first order ordinary differential equation:

$$\frac{d}{dt}\psi(\gamma(t)) + \sigma_t(\gamma(t))\psi(\gamma(t)) = q(\gamma(t))$$

Supposing that σ_t and q are constant, the exact solution is:

$$\psi(\boldsymbol{r}_0 + t\boldsymbol{\omega}) = \psi(\boldsymbol{r}_0)e^{-\sigma_t t} + \frac{1 - e^{-\sigma_t t}}{\sigma_t}\, q. \tag{1.45}$$

This result is the so-called "transmission formula". The rationale of the method of characteristics consists in defining a set of characteristic curves and, for each one of them, divide the spatial domain into constant regions where the flux can easily and exactly be propagated following equation (1.45). Thanks to this, the method is highly flexible since the major difficulty in the computation of the fluxes is the computation of the domain divisions. Paradoxically, this strenght represents also the major limitation of the method because complicated geometries require such a massive storage of data that the computation of a three-dimensional realistic core remains still nowadays an impossible task. We refer to [94] for recent works on this topic.

1.3.4.2 Finite elements

An alternative to the method of characteristics is the Galerkin projection of equation (1.44) and we will focus in the particular case of finite elements (i.e. the projections into piece-wise polynomial spaces). In the case of advection dominated problems like equation (1.44), continuous finite element methods suffer from instabilities (as illustrated in example 5.64 of [45]). An alternative technique would be finite volumes but they are poorly adapted if we need to increase the order of accuracy.

For these reasons, the discontinuous finite element method is preferred for the resolution of the neutron transport equation and this is the method that has been selected for the MINARET solver. The method was first proposed in [106] as a way to solve the advection equation (1.44) for neutronic calculations. The first analysis was presented in [69], showing a convergence rate of order $\mathcal{O}(h^k)$ on a general triangular grid and optimal convergence of order $\mathcal{O}(h^{k+1})$ on a cartesian grid of cell size h and with local polynomial approximation of order k. This result was later improved in [63] to $\mathcal{O}(h^{k+1/2})$-convergence on general grids and [102] proved the optimality of this convergence rate. These results assume smooth solutions and some developments for linear problems with non smooth solutions can be found in [26], [71].

The basic principles of the method are outlined in the remaining of this section, where we have followed the same structure and notations presented in [105] . The interested reader can refer to [60] for a broad overview of theoretical and practical aspects of discontinuous finite element methods.

Let $\mathcal{T}_h$ be an affine mesh [3] of the spatial domain $\mathcal{R}$. The subscript h has the usual sense as in the finite element litterature and denotes the mesh size. We define the approximation space V_h based on $\mathcal{T}_h$ such that functions are polynomials of degree at most k on each cell:

$$V_h = \{v \in L^2(\mathcal{R}) \mid \forall K \in \mathcal{T}_h,\ v|_K \in P_k\} \tag{1.46}$$

Note that this approximation is purely local and, thus, imposes no conditions on the grid structure. Besides, the locality confers a great flexibility to the method, easily allowing (h, p)-refinement.

3. In order to simplify the explanation, we will assume here that $\mathcal{R}$ is a polygon such that $\partial\mathcal{R} = \partial\mathcal{T}_h$.

The set of functions over each cell $K \in \mathcal{T}_h$ that can be mapped to polynomials of degree at most k over the reference element is denoted P_k. The local discontinuous Galerkin formulation of equation (1.44) consists in seeking $\psi_h \in V_h$ so that the following holds for all cells $K \in \mathcal{T}_h$ and for all test functions $v \in V_h$ supported on K:

$$-\int_K \psi_h \boldsymbol{\omega}.\nabla v dx + \int_{\partial K} F v dx = \int_K q v dx, \tag{1.47}$$

where the boundary conditions for the incoming flux are weakly enforced through the definition of F:

$$F = \begin{cases} \psi^- \boldsymbol{\omega}.\boldsymbol{n} & \text{if } \boldsymbol{\omega}.\boldsymbol{n} < 0 \\ \psi_h \boldsymbol{\omega}.\boldsymbol{n} & \text{otherwise} \end{cases}$$

$\boldsymbol{n}$ is the outward unit normal vector on ∂K and ψ^- is the value of the incoming value of the flux.

One of the advantages of this method is that the spatial problem can be solved cell after cell following an adequate order [4] without having to build a global matrix for the whole mesh. This results in a reduced storage requirement and a rapid numerical resolution.

Convergence rates of the fully discretized monoenergetic problem Sections 1.3.3 and 1.3.4 have dealt with the angular and spatial discretization of the monoenergetic problem given by equation (1.32). We remind that this problem arises after a Euler backward discretization of the time and a multigroup approximation of the energy variable.

In the case of MINARET, problem (1.32) has been discretized with an S_N technique and with discontinuous Galerkin finite elements for the space. In section 1.3.3 a discretization error for the semi-discretized problem in angle was presented. But, what kind of discretization error does one get in the fully discretized system? An answer to this problem can be found in [103] where the 1D case with constant scattering cross-sections is considered. Given a parameter $0 < \lambda < 1$, their problem reads:

$$\begin{cases} \mu \dfrac{\partial \psi}{\partial x}(x,\mu) + \psi(x,\mu) - \lambda \int_{-1}^{1} \psi(x,\eta) d\eta = q(x), \quad 0 < x < 1, \quad -1 \leq \mu \leq 1 \\ \psi(0,\mu) = 0, \text{ for } \mu > 0, \quad \psi(1,\mu) = 0, \text{ for } \mu < 0 \end{cases} \tag{1.48}$$

The result goes as follows:

Theorem 1.3.3. *Let $\psi \in L^2([0,1])$ be the solution to the source problem (1.48) and let ψ_N^h be the discrete ordinates approximation to the solution of (1.48), which is obtained using $2N$ Gauss-Legendre quadrature points on $[-1,1]$ with corresponding weights and the $\mathbb{P}_1$ discontinuous Galerkin finite element scheme for the spatial discretization. If the spatial mesh size $h = h(N) \to 0$ as $N \to \infty$, then, for N sufficiently large, the following estimate for the discretization error of the scalar flux holds:*

$$\|\phi - \phi_N^h\|_{L^2([0,1])} \leq C(\phi, q) \left(N^{-3/2} + 4h^2 N^{1/2} \right) \tag{1.49}$$

Proof. Apply theorem 4.4 of [103] with parameters $i = 1$, $k = 1$, $p = 2$ defined in the paper. □

Remark 1.3.4. *The above result is only one particular instance of the results of theorem 4.4 of [103]. In fact, one can find in this paper similar estimates for solutions ψ of any L^p regularity*

4. It was shown in [69] that, for any direction $\boldsymbol{\omega}$, there always exists a sorting of two-dimensional unstructured triangles so that the spatial problem can be fully solved. This is not the case in three-dimensional tetrahedral cells in which cycles can exist: we browse partially the whole mesh.

($1 \leq p \leq \infty$) and any order k of the polynomial of the discontinuous Galerkin method. What is of particular interest in this type of estimate is that one can find an optimal relation between the spatial mesh size and the S_N order. For instance, in the above result, the optimal relation between h and N is $h(N) = \mathcal{O}(N^{-1})$. With this choice, the convergence is at a rate of order $\mathcal{O}(N^{-3/2})$. From the numerical results presented in the same paper and also according to some references cited therein, the error estimates seem to be optimal but, to the best of our knowledge, no proof of optimality has ever been derived in this respect.

1.4 Approximations to the Boltzmann operator

1.4.1 The diffusion approximation

Under some given physical configurations, the search for the classical outputs of neutronic calculations (e.g. the power distribution) can be done through the computation of the angular mean value $\phi(t, \boldsymbol{r}, E) = \int_{\mathbb{S}_2} \psi(t, \boldsymbol{r}, \boldsymbol{\omega'}, E) d\boldsymbol{\omega'}$ instead of the angular flux $\psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E)$ without sacrificing much on the accuracy. In this section, we will see that $\phi(t, \boldsymbol{r}, E)$ is the solution of a diffusion equation that has the advantage of being much less computationally expensive than the transport equation from the memory storage and from the computational time point of view. For this reason and despite the loss of accuracy, the diffusion approximation of equation (1.1) has traditionally been preferred for the numerical analysis of reactor configurations. There exists a massive amount of literature dealing with the approximation of a transport equation by diffusion and here we will only give a brief overview of this topic. We will first justify the approximation by physical and formal arguments and after by the presentation of some more theoretical results that aim to show in what (mathematical) sense the diffusion is an asymptotic limit of transport.

1.4.1.1 A first simple approach

In section 1.3.3, two methods for the discretization of the angular variable have been explained in the context of the resolution of the monoenergetic source equation with isotropic scattering (see equation (1.32)). If we consider this problem and approximate its solution by its projection into spherical harmonics at order $N = 1$ (the so-called P_1 method), the approximation reads

$$\psi(\boldsymbol{r}, \boldsymbol{\omega}) \approx \phi_{0,0}(\boldsymbol{r}) Y_{0,0}(\boldsymbol{\omega}) + \phi_{1,-1}(\boldsymbol{r}) Y_{1,-1}(\boldsymbol{\omega}) + \phi_{1,0}(\boldsymbol{r}) Y_{1,0}(\boldsymbol{\omega}) + \phi_{1,1}(\boldsymbol{r}) Y_{1,1}(\boldsymbol{\omega}), \tag{1.50}$$

and the flux moments satisfy equation (1.39) with $N = 1$, i.e:

$$\sum_{\ell=0}^{N=1} \sum_{m=-\ell}^{\ell} \left(\boldsymbol{\omega}.\nabla \phi_{l,m}(\boldsymbol{r}) + \sigma_t(\boldsymbol{r}) \phi_{l,m}(\boldsymbol{r}) - \sigma_{s,\ell}(\boldsymbol{r}) \phi_{l,m}(\boldsymbol{r}) - q_{l,m}(\boldsymbol{r}) \right) Y_{\ell,m} = 0, \tag{1.51}$$

We are going to show that, under several hypothesis, this equation is in fact an approximation of the original transport problem by a diffusion equation. For this, we start by recalling that we can express the angular variable $\boldsymbol{\omega}$ as

$$\boldsymbol{\omega} = (\boldsymbol{\omega}_x, \boldsymbol{\omega}_y, \boldsymbol{\omega}_z)^T = (\cos\varphi \sin\theta, \sin\varphi \sin\theta, \cos\theta)^T \tag{1.52}$$

in spherical coordinates. Furthermore, the orthonormal spherical harmonics under consideration ($Y_{l,m}$, with $l \in \{0, 1\}$ and $m \in \{-l, +l\}$) have the explicit expressions:

$$Y_{0,0}(\theta, \varphi) = \frac{1}{2\sqrt{\pi}}, \quad Y_{1,0}(\theta, \varphi) = \frac{1}{2}\sqrt{\frac{3}{\pi}} \cos\theta, \quad Y_{1,\pm1}(\theta, \varphi) = \frac{1}{2}\sqrt{\frac{3}{\pi}} \sin\theta e^{\pm i\varphi}. \tag{1.53}$$

Thanks to equations (1.52) and (1.53), it is an easy matter to realize that

$$\boldsymbol{\omega}_x = \sqrt{\frac{\pi}{3}}(Y_{1,-1} + Y_{1,1}), \quad \boldsymbol{\omega}_y = i\sqrt{\frac{\pi}{3}}(Y_{1,-1} - Y_{1,1}), \quad \boldsymbol{\omega}_z = 2\sqrt{\frac{\pi}{3}}Y_{1,0}. \tag{1.54}$$

We now define the so-called current vector $\boldsymbol{J}(\boldsymbol{r}) = (J_x, J_y, J_z)^T$ as:

$$J_x = \frac{1}{8}\left(\frac{3}{\pi}\right)^{3/2}(\phi_{1,-1} + \phi_{1,1}), \quad J_y = i\frac{1}{8}\left(\frac{3}{\pi}\right)^{3/2}(\phi_{1,1} - \phi_{1,-1}), \quad J_z = i\frac{1}{8}\left(\frac{3}{\pi}\right)^{3/2}\phi_{1,0}. \tag{1.55}$$

By using equations (1.54) and (1.55), we can infer that relation (1.50) can equivalently be written in the form:

$$\psi(\boldsymbol{r}, \boldsymbol{\omega}) \approx \frac{\phi(\boldsymbol{r})}{4\pi} + \frac{3}{4\pi}\boldsymbol{\omega}.\boldsymbol{J}(\boldsymbol{r}). \tag{1.56}$$

This expression leads to interpret the P_1 projection of the flux as a first order Taylor expansion of ψ in the angular variable $\boldsymbol{\omega}$.

If we now project equation (1.51) into the four spherical harmonics $Y_{l,m}$, $l \in \{0, 1\}$ and $m \in \{-l, +l\}$, and express the resulting equations with the variables of relation (1.56), the P_1 equations read:

$$\begin{cases} \nabla.\boldsymbol{J}(\boldsymbol{r}) + \sigma_t(\boldsymbol{r})\phi(\boldsymbol{r}) = \sigma_{s,0}\phi(\boldsymbol{r}) + q_0(\boldsymbol{r}) \\ \frac{1}{3}\nabla\phi(\boldsymbol{r}) + \sigma_t(\boldsymbol{r})\boldsymbol{J}(\boldsymbol{r}) = \sigma_{s,1}\boldsymbol{J}(\boldsymbol{r}) + \boldsymbol{q_1}(\boldsymbol{r}). \end{cases} \tag{1.57}$$

At this point, we now present the two main classical reasonings that conclude the development. The first (and more physical one) consists in only considering the first equation and adding a closure relation between ϕ and $\boldsymbol{J}$ with physical arguments. The most simple and traditional one is:

$$\boldsymbol{J}(\boldsymbol{r}) = -D(\boldsymbol{r})\nabla\phi(\boldsymbol{r}), \quad D(\boldsymbol{r}) := \frac{1}{3\sigma_t(\boldsymbol{r})}, \tag{1.58}$$

which is called the Fick's law because it expresses the fact that the neutrons go from regions of high concentration to regions of low concentration, with a magnitude D that is proportional to the concentration gradient.

The second reasoning provides an improved expression of the Fick's law. If we come back to the equations of (1.57) and now assume that there is no anisotropy in the source term ($\boldsymbol{q_1} = \boldsymbol{0}$), then the second equation of (1.57) yields the relation:

$$\boldsymbol{J}(\boldsymbol{r}) = -D(\boldsymbol{r})\nabla\phi(\boldsymbol{r}), \quad D(\boldsymbol{r}) := \frac{1}{3\left(\sigma_t(\boldsymbol{r}) - \sigma_{s,1}(\boldsymbol{r})\right)}. \tag{1.59}$$

In both cases, if we use the Fick's law in the first equation of (1.57), we get:

$$-\nabla.\left(D(\boldsymbol{r})\phi(\boldsymbol{r})\right) + \sigma_t(\boldsymbol{r})\phi(\boldsymbol{r}) = \sigma_{s,0}\phi(\boldsymbol{r}) + q_0, \tag{1.60}$$

which is an elliptic problem for the scalar flux ϕ.

It is important to point out that the development above is not the only one that exists in order to show that, under several hypothesis, ϕ is the solution of an elliptic problem. Depending on the approach, different expressions of D can be derived and we refer to, e.g., [36] for an overview on this topic. In the following section, we will present a mathematical result from the literature as an illustration of the theoretical framework and hypothesis that show that diffusion can be seen as an asymptotic limit of transport.

1.4.1.2 Approximation for a monoenergetic model

We follow here the developments of chapter XXI, section 5.2 of [33]. Like in the previous section 1.4.1.1, the starting point is a monoenergetic equation with isotropic scattering as the one described in equation (1.32). In the present case, the time dependency is also considered and the authors analyze a monoenergetic transport equation of the form:

$$\frac{\partial \psi}{\partial t}(t,\boldsymbol{r},\boldsymbol{\omega}) + \boldsymbol{\omega}.\nabla\psi(t,\boldsymbol{r},\boldsymbol{\omega}) + \sigma_t(t,\boldsymbol{r})\psi(t,\boldsymbol{r},\boldsymbol{\omega}) = \int_{\mathbb{S}_2} \sigma_s(t,\boldsymbol{r},\boldsymbol{\omega'}.\boldsymbol{\omega})\psi(t,\boldsymbol{r},\boldsymbol{\omega'})d\boldsymbol{\omega'} + q(t,\boldsymbol{r},\boldsymbol{\omega}),\ \forall (t,\boldsymbol{r},\boldsymbol{\omega}) \in [0,T]\times\mathcal{R}\times\mathbb{S}_2. \tag{1.61}$$

In fact, the authors introduce some simplifications in the equation and their problem reads:

$$\begin{cases} \dfrac{\partial \psi}{\partial t}(t,\boldsymbol{r},\boldsymbol{\omega}) + \boldsymbol{\omega}.\nabla\psi(t,\boldsymbol{r},\boldsymbol{\omega}) + \tilde{\sigma}_t(\boldsymbol{r})\psi = \tilde{\sigma}_s(\boldsymbol{r})\displaystyle\int_{\mathbb{S}_2} f(\boldsymbol{\omega'},\boldsymbol{\omega})\psi(t,\boldsymbol{r},\boldsymbol{\omega'})d\boldsymbol{\omega'} \\ \psi(t,\boldsymbol{r},\boldsymbol{\omega}) = 0,\ \forall (\boldsymbol{r},\boldsymbol{\omega}) \in \Gamma_- \\ \psi(0,\boldsymbol{r},\boldsymbol{\omega}) = \psi^0(\boldsymbol{r},\boldsymbol{\omega}), \end{cases} \tag{1.62}$$

where $\tilde{\sigma}_t$ and $\tilde{\sigma}_s$ are bounded positive functions. In comparison with (1.61), problem (1.62) assumes that $q \equiv 0$ and that σ_t does not evolve in time:

$$\sigma_t(t,\boldsymbol{r}) = \sigma_t(\boldsymbol{r}) = \tilde{\sigma}_t(\boldsymbol{r}).$$

The scattering kernel $\sigma_s(t,\boldsymbol{r},\boldsymbol{\omega'}.\boldsymbol{\omega})$ does not evolve in time either and is approximated by

$$\sigma_s(t,\boldsymbol{r},\boldsymbol{\omega'}.\boldsymbol{\omega}) \approx \tilde{\sigma_s}(\boldsymbol{r})f(\boldsymbol{\omega},\boldsymbol{\omega'}),$$

where f is a positive function independent of $\boldsymbol{r}$, measurable and bounded over $\mathbb{S}_2\times\mathbb{S}_2$ such that

$$f(\boldsymbol{\omega},\boldsymbol{\omega'}) = f(\boldsymbol{\omega'},\boldsymbol{\omega}), \quad \forall (\boldsymbol{\omega},\boldsymbol{\omega'}) \ \in \mathbb{S}_2\times\mathbb{S}_2.$$

We now assume that the mean free path of the particles $1/\tilde{\sigma}_t$ is small compared to the dimensions of $\mathcal{R}$. This implies that $\tilde{\sigma}_t$ and $\tilde{\sigma}_s$ are large. Hence, we do the following change of variables:

$$\tilde{\sigma}_t = \frac{\sigma_t}{\varepsilon}; \qquad \tilde{\sigma}_s = \frac{\sigma_s}{\varepsilon},$$

with ε being a small parameter. We further assume that the time t under consideration is "far" from $t=0$, i.e. that t is large compared to the time scale of the transport phenomenon that is measured by $\dfrac{1}{\sigma_t|\boldsymbol{v}|}$. This justifies the change of variables $t' = t\varepsilon$.

Finally, we make the hypothesis that the absorption in the medium is very small, i.e., that the ratio $\dfrac{\sigma_s}{\sigma_t}$ is close to one. In particular, we assume that $\dfrac{\sigma_s}{\sigma_t} - 1 = \mathcal{O}(\varepsilon^2)$:

$$\sigma_t(\boldsymbol{r}) = \sigma_s(\boldsymbol{r}) - \varepsilon^2\sigma_a(\boldsymbol{r}),$$

where σ_a is a bounded function. The following theorem shows that diffusion is an asymptotic limit of transport under the L^∞ norm:

Theorem 1.4.1 (Chapter XXI of [33], section 5, paragraph 2, Theorem 1)**.** *Let $\mathcal{R}$ be a bounded open subset of $\mathbb{R}^3$ with a regular enough bound and let f, σ_t, σ_a be such that:*

- $f(\boldsymbol{\omega},\boldsymbol{\omega'}) = f(\boldsymbol{\omega'},\boldsymbol{\omega}), \quad \forall (\boldsymbol{\omega},\boldsymbol{\omega'}) \ \in \mathbb{S}_2\times\mathbb{S}_2,$
- $\displaystyle\int_{\mathbb{S}_2} f(\boldsymbol{\omega'},\boldsymbol{\omega})d\boldsymbol{\omega'} = 1,$

- $\exists \beta_0 > 0,\ \beta_1 > 0$, *such that* $\beta_0 \leq f(\boldsymbol{\omega}, \boldsymbol{\omega}') \leq \beta_1$ *and* $\beta_0 \leq \sigma_t(\boldsymbol{r}) \leq \beta_1$,
- $\exists \alpha \in]0,1[$ *such that* $\sigma_t \in \mathcal{C}^{3,\alpha}(\overline{\mathcal{R}})$ *and* $\sigma_a \in \mathcal{C}^{2,\alpha}(\overline{\mathcal{R}})$. [5]

Then, there exists a symmetric positive definite matrix $(a_{ij}(\boldsymbol{r}))$ *with the following property: For any function* $\psi^0 = \psi^0(\boldsymbol{r})$ *such that*

$$\psi^0 \in \mathcal{C}^{4,\alpha}(\overline{\mathcal{R}}),\ \text{with}\ \psi^0\Big|_{\partial\mathcal{R}} = 0,\ \sum_{i=1}^{3}\sum_{j=1}^{3} \frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial \psi^0}{\partial x_j}\right)\Bigg|_{\partial\mathcal{R}} = 0,$$

the strong solution ψ_ε *in* $\mathcal{C}\left([0,\infty[,\ L^\infty(\mathcal{R}\times\mathbb{S}_2)\right)$ *of the transport problem:*

$$\begin{cases} \dfrac{\partial \psi_\varepsilon}{\partial t}(t,\boldsymbol{r},\boldsymbol{\omega}) - \sigma_a(\boldsymbol{r})\psi_\varepsilon(t,\boldsymbol{r},\boldsymbol{\omega}) + \dfrac{1}{\varepsilon}\boldsymbol{\omega}.\nabla\psi_\varepsilon \\ \qquad + \dfrac{\sigma_s(\boldsymbol{r})}{\varepsilon^2}\left(\psi_\varepsilon(t,\boldsymbol{r},\boldsymbol{\omega}) - \displaystyle\int_{\mathbb{S}_2} f(\boldsymbol{\omega}',\boldsymbol{\omega})\psi_\varepsilon(t,\boldsymbol{r},\boldsymbol{\omega}')d\boldsymbol{\omega}'\right) = 0,\ \text{over}\ ]0,\infty[\times\mathcal{R}\times\mathbb{S}_2 \\ \psi_\varepsilon(t,\boldsymbol{r},\boldsymbol{\omega}) = 0,\ (\boldsymbol{r},\boldsymbol{\omega}) \in \Gamma_-,\ t>0, \\ \psi_\varepsilon(0,\boldsymbol{r},\boldsymbol{\omega}) = \psi^0(\boldsymbol{r}),\ (\boldsymbol{r},\boldsymbol{\omega}) \in \mathcal{R}\times\mathbb{S}_2, \end{cases} \tag{1.63}$$

and the strong solution ϕ *in* $\mathcal{C}\left([0,\infty[,\ L^\infty(\mathcal{R})\right)$ *of the diffusion problem:*

$$\begin{cases} \dfrac{\partial \phi}{\partial t} = \displaystyle\sum_{i=1}^{3}\sum_{j=1}^{3}\frac{\partial}{\partial x_i}\left(a_{ij}\frac{\partial\phi}{\partial x_j}\right) + \sigma_a(\boldsymbol{r})\phi(t,\boldsymbol{r},\boldsymbol{v})\ \text{over}\ ]0,\infty[\times\mathcal{R} \\ \phi(t,\boldsymbol{r}) = 0\ \boldsymbol{r}\in\partial\mathcal{R},\ t>0 \\ \phi(0,\boldsymbol{r}) = \psi^0(\boldsymbol{r}),\ \boldsymbol{r}\in\mathcal{R} \end{cases} \tag{1.64}$$

verify for any $t \geq 0$*:*

$$\|\psi_\varepsilon(t,.,.) - \phi(t,.)\|_{L^\infty(\mathcal{R}\times\mathbb{S}_2)} \leq C_{\psi^0}\varepsilon e^{\delta t}(1+t)\ \text{over}\ ]0,\infty[\times\mathcal{R} \tag{1.65}$$

where $\delta = \sup_{x\in\mathcal{R}} \sigma_a(x)$ *and* C_{ψ^0} *is a positive constant independent of* ε.

Remark 1.4.2. *The above result assumes that the initial condition* ψ^0 *does not depend on the velocity* $\boldsymbol{\omega}$*. One can find some elements in [33] that could lead to a similar result without this restriction. We also refer to the same authors for the presentation of analogue results in* L^p.

1.4.1.3 The diffusion equation for a reactor core

The exact diffusion equation that is usually employed for reactor core calculations is an extension of the monoenergetic case and it reads:

$$\begin{cases} \dfrac{1}{|\boldsymbol{v}|}\partial_t\phi(t,\boldsymbol{r},E) - \nabla.\left[D(t,\boldsymbol{r},E)\nabla\phi(t,\boldsymbol{r},E)\right] + \sigma_t(t,\boldsymbol{r},E)\phi(t,\boldsymbol{r},E) \\ \qquad - \displaystyle\int_{E'=E_{\min}}^{E_{\max}} \sigma_s(t,\boldsymbol{r},E'\to E)\phi(t,\boldsymbol{r},E')dE' \\ \qquad - \chi_p(t,\boldsymbol{r},E)\displaystyle\int_{E'=E_{\min}}^{E_{\max}} \left(1-\beta(t,\boldsymbol{r},E')\right)(\nu\sigma_f)(t,\boldsymbol{r},E')\phi(t,\boldsymbol{r},E')dE' \\ \qquad - \displaystyle\sum_{\ell=1}^{L}\lambda_\ell\chi_{d,\ell}(t,\boldsymbol{r},E)C_\ell(t,\boldsymbol{r}) = S(t,\boldsymbol{r},E) \\ \partial_t C_\ell(t,\boldsymbol{r}) = -\lambda_\ell C_\ell(t,\boldsymbol{r}) \\ \qquad + \displaystyle\int_{E'=E_{\min}}^{E_{\max}} \beta_\ell(t,\boldsymbol{r},E')(\nu\sigma_f)(t,\boldsymbol{r},E')\phi(t,\boldsymbol{r},E')dE',\ \forall \ell\in\{1,\dots,L\}, \end{cases} \tag{1.66}$$

5. $\mathcal{C}^{k,\alpha}(\overline{\mathcal{R}})$ denotes the set of functions u with continuous derivatives up to order k and the k^{th} partial derivatives are Hölder continuous with exponent α, i.e., $|f(x)-f(y)| \leq C_\alpha|x-y|^\alpha,\ \forall x\neq y\in\mathcal{R}$.

The resolution of this equation uses the same discretization techniques that have been outlined in section 1.3 for the transport case.

1.4.2 The Simplified P_N

This approximation has its roots in the following observation: the P_N equations have a very simple structure in 1D slab geometry (they are slightly coupled), and their number of unknowns is only $(N+1)$ (the form of these equations will explicitly be expressed further in this section). However, their extension to multidimensional geometries results in additional couplings and the original 1D simplicity is lost. Besides, the number of equations increases quadratically in N. To preserve the simple form of the 1D slab geometry case in three dimensions and also to deal with the large number and complexity of the P_N equations, Gelbard [51] proposed in the 1960s a simplification of the P_N equations which he called "simplified P_N" (SP_N). Like in the 1D case, in this approximation the number of SP_N equations equals $N+1$ instead of $(N+1)^2$ of the classical P_N.

The rationale of the approach consists in building an operator that generalizes the diffusion approximation and that takes into account a maximum number of transport effects. The main idea resides in assuming that the transport solution is locally nearly planar and that, as a consequence, we will locally have 1D slab problems: for a given spatial location $\boldsymbol{r}$, consider the local cartesian coordinates $(\boldsymbol{i_r}, \boldsymbol{j_r}, \boldsymbol{k_r})$ such that the flux has a planar symmetry with respect to $(\boldsymbol{i_r}, \boldsymbol{j_r})$. In this particular case, ψ can be found through the resolution of a one dimensional transport equation along the $\boldsymbol{k_r}$ axis (the $z_{\boldsymbol{r}}$ variable). Furthermore, because of the planar symmetry, the angular dependency $\boldsymbol{\omega_r} = (\theta_{\boldsymbol{r}}, \varphi_{\boldsymbol{r}})$ is simplified and we have $\boldsymbol{\omega_r} = (\theta_{\boldsymbol{r}}, \varphi_{\boldsymbol{r}}) = (\theta_{\boldsymbol{r}}, 0)$ for any $\varphi_{\boldsymbol{r}} \in [0, 2\pi]$. Since $\theta_{\boldsymbol{r}} \in [0, \pi[$, one can equivalently use the variable $\mu_{\boldsymbol{r}} = \cos\theta_{\boldsymbol{r}} \in [-1, 1]$ to describe the angular dependency. Hence, in a neighborhood of $\boldsymbol{r}$, the flux can be expressed as:

$$\psi(t, \boldsymbol{r}, \boldsymbol{\omega}) = \psi(t, z_{\boldsymbol{r}}, \mu_{\boldsymbol{r}}) = \sum_{\ell=0}^{\infty} \phi_\ell(t, z_{\boldsymbol{r}}) P_\ell(\mu_{\boldsymbol{r}}), \tag{1.67}$$

where P_ℓ are the Legendre polynomials. Thanks to this approximations, in a neighborhood of $\boldsymbol{r}$, the P_N equations correspond to a classical 1D slab problem:

$$\begin{cases} \dfrac{1}{v}\dfrac{\partial \phi_n}{\partial t}(t, z_{\boldsymbol{r}}) + \dfrac{\partial}{\partial z_{\boldsymbol{r}}}\left(\dfrac{n}{2n+1}\phi_{n-1}(t, z_{\boldsymbol{r}}) + \dfrac{n+1}{2n+1}\phi_{n+1}(t, z_{\boldsymbol{r}})\right) + \sigma_t(t, z_{\boldsymbol{r}})\phi_n(t, z_{\boldsymbol{r}}) \\ \qquad = \delta_{n,0}\left(\sigma_s(t, z_{\boldsymbol{r}})\phi_0(t, z_{\boldsymbol{r}}) + S(t, z_{\boldsymbol{r}})\right), \quad 0 \le n \le N \\ \phi_{-1} := 0, \ \phi_{N+1} := 0. \end{cases} \tag{1.68}$$

The SP_N approximation postulates that this local form of equations extended to three dimensions is a good approximation to the original transport equation. As a consequence, the SP_N equations are built as follows: the odd moments of the angular flux are replaced by vector functions $\boldsymbol{\phi}_{2\ell+1}$ and the even moments by scalar functions $\phi_{2\ell}$. The first order derivative is replaced either by a divergence operator for the vector functions or by a gradient operator for the scalar functions. This leads to the following system of $N+1$ equations:

$$\begin{cases} \dfrac{1}{v}\dfrac{\partial \phi_n}{\partial t}(t, \boldsymbol{r}) + \dfrac{n}{2n+1}\nabla.\boldsymbol{\phi}_{n-1} + \dfrac{n+1}{2n+1}\nabla.\boldsymbol{\phi}_{n+1} + \sigma_t(t, z)\phi_n(t, \boldsymbol{r}) \\ \qquad = \delta_{n,0}\left(\sigma_s(t, z)\phi_0(t, z) + S(t, z)\right), \ n \text{ even}, \\ \dfrac{1}{v}\dfrac{\partial \boldsymbol{\phi}_n}{\partial t}(t, \boldsymbol{r}) + \dfrac{n}{2n+1}\nabla\phi_{n-1} + \dfrac{n+1}{2n+1}\nabla\phi_{n+1} + \sigma_t(t, z)\boldsymbol{\phi}_n(t, \boldsymbol{r}) = 0, \ n \text{ odd}, \\ \phi_{-1} = 0, \ \phi_{N+1} = 0. \end{cases} \tag{1.69}$$

Note that problem (1.69) can be seen as $(N+1)/2$ coupled diffusion equations where the vector functions $\boldsymbol{\phi}_{2\ell+1}$ represent the current unknowns and the scalar functions $\phi_{2\ell}$ represent the flux unknowns.

To close the system, boundary conditions must be added (e.g. by imposing that even fluxes are zero on the boundary). In any event, it seems clear that, as the index N increases without bound, the solution of the SP_N equations does not converge to the solution of the transport equation. Thus, this infinite hierarchy of approximations presumably converges (this issue is nevertheless an open problem) but it will not converge in general to the solution of the transport equation. Despite this fact, the low order simplified P_N (SP_3 and SP_5) have numerically shown to give a substantial improvement in accuracy over P_1 (i.e. diffusive) results (see, e.g., [51]) and many existing codes run computations with this approximation on a regular basis (see, e.g. [12]).

1.4.3 Quasi-static methods

The name "quasi-static" gathers a whole family of multi-grid resolution methods for the computation of problem (1.1). The rationale of the approach is to split fast and slow variations of the flux in time following a factorization of the form

$$\psi(t,\boldsymbol{r},\boldsymbol{\omega},E) \approx a(t,\boldsymbol{r},\boldsymbol{\omega},E) f(t,\boldsymbol{r},\boldsymbol{\omega},E), \tag{1.70}$$

where a is the so-called *amplitude* function and represents fast variations and f is the *shape* function that slowly varies in time. There is a degree of arbitrariness in the choice of a and f and, depending on that choice, the method will be called differently. In any event, because of the factorization hypothesis, we could say that quasi-static methods are somehow on the fringe between acceleration techniques for the resolution of (1.1) and a simplification to the Boltzmann operator.

One of the most popular among these techniques is the so-called *Improved Quasi-Static* method in which the decomposition reads

$$\psi(t,\boldsymbol{r},\boldsymbol{\omega},E) \approx a(t) f(t,\boldsymbol{r},\boldsymbol{\omega},E), \tag{1.71}$$

i.e. the amplitude function depends only on time on the fast time scale. The introduction of this form of the flux into equation (1.1) leads to the following PDE equation for the shape function f:

$$\begin{aligned}\frac{1}{|\boldsymbol{v}|}\frac{\partial f}{\partial t}(t,\boldsymbol{r},\boldsymbol{\omega},E) + \left(L - H - F - \frac{Q}{a(t)}\right) f(t,\boldsymbol{r},\boldsymbol{\omega},E) = \\ \frac{S(t,\boldsymbol{r},\boldsymbol{\omega},E)}{a(t)} - \frac{1}{|\boldsymbol{v}|} f(t,\boldsymbol{r},\boldsymbol{\omega},E)\frac{1}{a(t)}\frac{da(t)}{dt}.\end{aligned} \tag{1.72}$$

The amplitude equation is obtained through a projection technique that involves the steady-state adjoint flux as a weighting function to ensure the uniqueness of the decomposition. After some algebra, one can derive a system of $(L+1)$ ODE's for the amplitude a and for the neutron precursors:

$$\begin{cases}\dfrac{da(t)}{dt} = \left(\dfrac{\rho(t)-\tilde{\beta}(t)}{\Lambda(t)}\right) a(t) + \sum\limits_{\ell=1}^{L} \lambda_\ell C_\ell(t) + \tilde{S}(t) \\ \dfrac{dC_\ell(t)}{dt} = -\lambda_\ell C_\ell(t) + \dfrac{\tilde{\beta}(t)}{\Lambda(t)} a(t), \quad \ell \in \{1,\dots,L\}.\end{cases} \tag{1.73}$$

This set is called the *point-kinetics* problem and describes the evolution of the amplitude function on the fast time scale. The kinetic parameters ρ, $\tilde{\beta}$, Λ depend on the shape function f. Note that equations (1.72) and (1.73) form a system of nonlinear equations given the dependence on a of equation (1.72) and the dependence on f of the kinetic parameters.

Two time steps are defined:

- The whole time interval $[0, T]$ is divided into large sub-intervals $[T_n, T_{n+1}]$. For each macro time step $\Delta T_{n+1} = T_{n+1} - T_n$, a and f are computed.
- Each macro interval $[T_n, T_{n+1}]$ is divided into fine time steps δt for which only a is derived.

The resolution of equations (1.72) and (1.73) is then iteratively performed on each macro-interval $[T_n, T_{n+1}]$: the shape equation is propagated with a macro time step ΔT_{n+1} and the amplitude equations with δt.

Remark 1.4.3. *Note that the method shares some similarities with the parareal in time algorithm (see section 1.6.2.2) in the sense that both are iterative techniques across the time domain. The main difference relies on the fact that parareal iterates on the whole time interval while quasi-static iterates on macro time steps. Beyond the similarity, both methods are also complementary because the quasi-static method could be employed as the coarse solver of the parareal in time method for the resolution of the time dependent transport equation (see section 1.6.2.2).*

The major advantages of quasi-static is that it can easily be coupled to any existing spatial solvers without major changes in the program. Besides, although no theoretical study of the error between the IQS solution and the direct solution of the Boltzmann equation has ever been derived, the approximation seems accurate enough according to the numerical results found in the literature (see, e.g. [23]). However, the speed-up performances of this method appear to be rather moderate to the best of the author's understanding: in [23], [52] the reported speed-ups are of order 2. An improvement of this method called *predictor-corrector quasi-static method* ([39], [40]) seems to be a promising track to improve the performances without sacrificing on the accuracy.

An interesting enhanced coupling of the quasi-static method with multi-grid techniques can be found in [23], where the decomposition reads:

$$\phi(t, \boldsymbol{r}, \boldsymbol{\omega}, E) \approx a(t, \boldsymbol{r}) f(t, \boldsymbol{r}, E), \tag{1.74}$$

where a is computed on a coarse spatial mesh, while f has a fine mesh.

1.5 State of the art of the existing 3-D time-dependent neutron transport solvers

There exists quite a large amount of industrial codes solving the steady-state multigroup neutron transport equation for reactor core calculations. However, the extension of these codes for time-dependent computations seems to have been seldom implemented so far, mainly because of excessive computing times. For this reason, the existing time-dependent codes have used the diffusion approximation to the Boltzmann operator that has been presented in section 1.4. To the best of the author's knowledge, only MINARET and TORT-TD [112] aim at solving the kinetic neutron transport equation (1.1) in three dimensional geometries. They are both multigroup S_N solvers that use a Euler backward in time discretization. They also use the same numerical schemes (generalized Gauss-Seidel and source iteration). The main difference relies in the spatial discretization: while TORT-TD works on cartesian grids, MINARET's mesh is "partially" unstructured in the sense that it is built by an extrusion of an initial two-dimensional unstructured mesh.

In the present work, a special effort has been made in order to provide solutions to the long computational times of such solvers and some sequential and parallel acceleration techniques have been explored. These will be outlined in section 1.6 in which we have also outlined several other existing methods that seem interesting to keep in mind for future works. Concrete performances will be presented in chapter 2.

1.6 About acceleration techniques for a time-dependent multigroup neutron transport S_N solver

We will focus here on the acceleration techniques for a solver built with:

- an Euler backward time discretization,
- a multigroup approximation for the energy,
- a discrete ordinates discretization for the directions.

This strategy is common in the field of neutronics as outlined in section 1.5 and it is also the one that has been followed in MINARET. For a given time step, we are led to the resolution of two embedded iterative algorithms (see algorithm 1.1):

1. a generalized Gauss-Seidel iterative algorithm for the resolution of the multigroup equations and
2. the computation of a source iteration problem for each energy group.

Given this scheme, the resolution can first of all be accelerated by sequential methods. In section 1.6.1, we will explain two of these methods that have been implemented in MINARET: the Chebyshev extrapolation and the Diffusion Synthetic Acceleration. These methods can be combined with parallel techniques. Section 1.6.2 will outline the two parallel techniques explored in MINARET and also several other methods that seem promising tracks to be considered for future works.

```
1: for t_n = ΔT to NΔT do
2:     While(not converge) do (generalized GS iterations – see equation (1.27))
3:     for g = 1 to G do
4:         Update fission operator
5:         Update scattering (except self-scattering)
6:         While(not converge) do (source iterations)
7:         for ω = ω_1 to ω_D do
8:             Update self-scattering
9:             Solve spatial problem (1.44) for ω
10:        end for
11:        Diffusion Synthetic Acceleration
12:        End While
13:    end for
14:    Chebyshev Extrapolation
15:    End While
16: end for
```

Algorithm 1.1: The iterative strategy implemented in MINARET

1.6.1 Sequential acceleration methods

1.6.1.1 Acceleration of the multigroup equations

In section 1.3.2.1, we saw that the resolution of the multigroup problem at each time step leads to the matrix system:

$$A_{G,G}\boldsymbol{\psi}^{\boldsymbol{n+1}} = \boldsymbol{S}^{\boldsymbol{n}}, \tag{1.75}$$

where we remind that:

$$A_{G,G} = \begin{pmatrix} L^1 - H^{1\to 1} - F^{1,1} & -H^{2\to 1} - F^{2,1} & \cdots & -H^{G\to 1} - F^{G,1} \\ -H^{1\to 2} - F^{1,2} & L^2 - H^{2\to 2} - F^{2,2} & \cdots & -H^{G\to 2} - F^{G,2} \\ \vdots & \vdots & \ddots & \vdots \\ -H^{1\to G} - F^{1,G} & -H^{2\to G} - F^{2,G} & \cdots & L^G - H^{G\to G} - F^{G,G} \end{pmatrix} \tag{1.76}$$

and

$$\boldsymbol{\psi^{n+1}} = \begin{pmatrix} \psi^{1,n+1} \\ \psi^{2,n+1} \\ \vdots \\ \psi^{G,n+1} \end{pmatrix} \quad ; \quad \boldsymbol{S^n} = \begin{pmatrix} S^{1,n} \\ S^{2,n} \\ \vdots \\ S^{G,n} \end{pmatrix}. \tag{1.77}$$

The inversion of the system (1.75) of equations is performed by a "generalized Gauss-Seidel" iterative method. We remind that, if $\psi^{g,n+1}_{(M)}$ is the approximation of $\psi^{g,n+1}$ at iteration number M with this scheme, our method reads:

$$M_{G,G}\boldsymbol{\psi^{n+1}_{(M+1)}} = N_{G,G}\boldsymbol{\psi^{n+1}_{(M)}} + \boldsymbol{S^n}, \tag{1.78}$$

where

$$M_{G,G} = \begin{pmatrix} L^1 - H^{1\to 1} & 0 & \cdots & 0 \\ -H^{1\to 2} & L^2 - H^{2\to 2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -H^{1\to G} & -H^{2\to G} & \cdots & L^G - H^{G\to G} \end{pmatrix} \tag{1.79}$$

and

$$N_{G,G} = \begin{pmatrix} F^{1,1} & H^{2\to 1} + F^{2,1} & \cdots & H^{G\to 1} + F^{G,1} \\ F^{1,2} & F^{2,2} & \cdots & H^{G\to 2} + F^{G,2} \\ \vdots & \vdots & \ddots & \vdots \\ F^{1,G} & F^{2,G} & \cdots & F^{G,G} \end{pmatrix}. \tag{1.80}$$

The purpose of this section is twofold: we will first show that the generalized Gauss-Seidel numerical scheme implemented in MINARET converges. After, we will briefly summarize the theoretical foundations of the classical Chebyshev acceleration that has been implemented in MINARET to speed-up its convergence. We start by proving

Theorem 1.6.1. *Assume that the quadrature rule used for the resolution of system (1.75) is composed of positive weights. Then, there exists a constant $\widetilde{\Delta T}_{n+1}$ such that, for any given time step $\Delta T_{n+1} < \widetilde{\Delta T}_{n+1}$, the generalized Gauss-Seidel iterative method described in (1.78) converges and its spectral radius is of order $\mathcal{O}(\Delta T_{n+1})$.*

Proof. After discretization of all the variables, the terms L^g, $H^{g'\to g}$, $F^{g',g}$ contained in $A_{G,G}$, $M_{G,G}$, $N_{G,G}$ become bloc matrices. Furthermore, the block matrices $H^{g'\to g}$, $F^{g',g}$ are composed of positive terms since, by assumption, we use a quadrature rule with positive weights. In the rest of the proof, we will work with the block matrices with the syntax employed for scalar values.

For any $g \in \{1, \ldots, G\}$, the term L^g (see equation (1.44)) contains a positive term $\sigma_t^{g,n+1} + \dfrac{1}{V^g \Delta T_{n+1}}$ that can become arbitrarily high by decreasing the time-step ΔT_{n+1}. It follows that there exists a limit value ΔT^*_{n+1} such that, if $\Delta T_{n+1} \leq \Delta T^*_{n+1}$, we will have that $A_{G,G}$ and $M_{G,G}$ are diagonally dominant matrices. Furthermore, there also exists a limit value ΔT^{**}_{n+1} such that, if $\Delta T_{n+1} \leq \Delta T^{**}_{n+1}$, we will have $L^g - H^{g\to g} - F^{g,g} > 0,\ \forall g \in \{1, \ldots, G\}$.

Let now ΔT_{n+1} be such that $\Delta T_{n+1} \leq \widetilde{\Delta T}_{n+1}$, where $\widetilde{\Delta T}_{n+1} := \min\left(\Delta T^*_{n+1}, \Delta T^{**}_{n+1}\right)$. Then, $A_{G,G}$ is diagonally dominant and:

$$|L^g - H^{g,g} - F^{g,g}| > \sum_{j \neq g} \left| H^{j \to g} + F^{g,j} \right|, \quad \forall g \in \{1, \ldots, G\}.$$

For any $\lambda \in \mathbb{C}$ such that $|\lambda| > 1$, we therefore have:

$$|L^g - H^{g,g} - F^{g,g}| > \sum_{j<g} \left| H^{j \to g} + \frac{1}{|\lambda|} F^{g,j} \right| + \frac{1}{|\lambda|} \sum_{j>g} \left| H^{j \to g} + F^{g,j} \right|, \quad \forall g \in \{1, \ldots, G\}. \tag{1.81}$$

We are now going to prove the convergence by showing that the spectral radius ρ of $M^{-1}_{G,G} N_{G,G}$ is lower than 1 ($M_{G,G}$ is invertible because it is diagonally dominant). Let $p_\lambda(M^{-1}_{G,G} N_{G,G})$ be the characteristic polynomial of $M^{-1}_{G,G} N_{G,G}$. If $I_{G,G}$ denotes the identity matrix of dimension G, then:

$$p_\lambda(M^{-1}_{G,G} N_{G,G}) = \det\left(M^{-1}_{G,G} N_{G,G} - \lambda I_{G,G}\right) = (-1)^G \det\left(M^{-1}_{G,G}\right) \det\left(M_{G,G} - \lambda N_{G,G}\right).$$

$p_\lambda(M^{-1}_{G,G} N_{G,G})$ is a polynomial of degree $G \geq 1$ and has at least one root. Given that $\det\left(M^{-1}_{G,G}\right) \neq 0$, it means that $\det\left(M_{G,G} - \lambda N_{G,G}\right)$ must have at least one root. We will now prove our claim by showing that any eigenvalue λ is such that $|\lambda| < 1$.

We will draw a contradiction: let us assume that $|\lambda| \geq 1$ and let us show that $M_{G,G} - \lambda N_{G,G}$ is invertible. We have:

$$\begin{aligned} |\lambda(L^g - H^{g \to g}) - F^{g,g}| &\geq |\lambda| \, |L^g - H^{g \to g}| - |F^{g,g}| \\ &\geq |\lambda| \left(|L^g - H^{g \to g}| - |F^{g,g}| \right) \\ &= |\lambda| \, |L^g - H^{g \to g} - F^{g,g}| \end{aligned} \tag{1.82}$$

where we have used the fact that $|\lambda| \geq 1$ and that $\left(|L^g - H^{g \to g}| - |F^{g,g}|\right) = L^g - H^{g \to g} - F^{g,g} > 0$ because $\Delta T_{n+1} < \widetilde{\Delta T}_{n+1}$. By using relations (1.81) and (1.82), we derive:

$$\begin{aligned} |\lambda(L^g - H^{g \to g}) - F^{g,g}| &\geq |\lambda| \, |L^g - H^{g \to g} - F^{g,g}| \\ &> \sum_{j<g} \left| \lambda H^{j \to g} + F^{g,j} \right| + \sum_{j>g} \left| H^{j \to g} + F^{g,j} \right|, \end{aligned}$$

which implies that $M_{G,G} - \lambda N_{G,G}$ is diagonally dominant, hence invertible and $\det(M_{G,G} - \lambda N_{G,G}) \neq 0$ for any $|\lambda| \geq 1$. Since p_λ has at least one root, we have drawn a contradiction. It follows that $|\lambda| < 1$ and the method converges.

Furthermore, if $|\lambda| < 1$, we can easily infer that the greatest eigenvalue in modulus of $M^{-1}_{G,G} N_{G,G}$ (i.e. $\rho(M^{-1}_{G,G} N_{G,G})$) must satisfy the condition:

$$\rho(M^{-1}_{G,G} N_{G,G}) \left(|L^g - H^{g \to g}| - \sum_{j<g} H^{j \to g} \right) \leq \sum_{j=1}^{G} F^{g,j} + \sum_{j>g} H^{j \to g}, \quad \forall g \in \{1, \ldots, G\}.$$

Hence the inequality:

$$\rho(M^{-1}_{G,G} N_{G,G}) \leq \frac{\max\limits_{g \in \{1,\ldots,G\}} \left(\sum_{j=1}^{G} F^{g,j} + \sum\limits_{j>g} H^{j \to g} \right)}{\min\limits_{g \in \{1,\ldots,G\}} \left(|L^g - H^{g \to g}| - \sum\limits_{j<g} H^{j \to g} \right)}. \tag{1.83}$$

The only dependence of the right hand side of (1.83) in ΔT_{n+1} is through the term L^g in the form of $1/\Delta T_{n+1}$. It thus follows that there exists a constant $C > 0$ such that

$$\rho(M_{G,G}^{-1} N_{G,G}) \leq C \Delta T_{n+1}, \tag{1.84}$$

which concludes the proof. □

The above result implies that the initial value for the iterate $\psi_{(M=0)}^{g,n+1}$ can be arbitrarily taken. However, as will be illustrated in chapter 2, the convergence of this method is very slow in practice. A first remedy to this is to choose an appropriate starting guess for $\psi_{(M=0)}^{g,n+1}$. The usual choice is $\psi_{(0)}^{g,n+1} = \psi_{(\infty)}^{g,n}$, where $\psi_{(\infty)}^{g,n}$ is the converged angular flux solution of the previous time. Although this option is reasonable because $\psi^{g,n+1}$ will presumably not differ very much from $\psi^{g,n}$ if ΔT_{n+1} is small, it will be shown in chapter 2 that there exists other alternatives that might be even more appropriate regarding the minimization of the number of iterations.

Other solutions for the slow convergence are acceleration techniques of the iterative scheme. Among the variety of methods that could be used in our case, we will focus on the very classical acceleration by Chebyshev polynomials because it is the one that has been implemented in the MINARET solver. For this, we follow the presentation made on chapter 5 of [120].

The Chebyshev acceleration belongs to the family of semi-iterative methods for which, given an iterative method of the form

$$\begin{cases} x_{(0)} \text{ given} \\ x_{(M)} = A x_{(M-1)} + g, \quad M \geq 1, \end{cases} \tag{1.85}$$

in which A is an n-dimensional square matrix and $x_{(M)}$ is the approximation of the unknown x at iteration number M, we consider a more general iterative procedure

$$y_{(M)} := \sum_{j=0}^{M} \nu_{j,M} x_{(j)}, \tag{1.86}$$

that algebraically combines the iterates $x_{(j)}$ given by the resolution of (1.85). The aim here is to determine the constants $\nu_{j,M}$ of (1.86) in such a way that the vectors $y_{(M)}$ tend rapidly to the unique solution x of the system

$$(I - A)x = g$$

in some sense.

Remark 1.6.2. *The link of these generic notations with the case of multigroup neutron transport is:* $A = M_{G,G}^{-1} N_{G,G}$ *and* $x_{(M)} = \boldsymbol{\psi_{(M)}^{n+1}}$.

If the initial approximation $x_{(0)}$ were equal to x, then it would follow from (1.85) that $x_{(M)} = x$ for any $M \geq 0$. In this case, $y_{(M)}$ should be equal to x, which imposes the condition

$$\sum_{j=0}^{M} \nu_{j,M} = 1, \quad M \geq 0. \tag{1.87}$$

Let

$$\begin{cases} \epsilon_{(M)} := x_{(M)} - x, \quad M \geq 0, \\ \tilde{\epsilon}_{(M)} := y_{(M)} - x, \quad M \geq 0, \end{cases} \tag{1.88}$$

be the error vectors associated with the vectors $x_{(M)}$ and $y_{(M)}$ respectively. From (1.87) and (1.88), we have:

$$\tilde{\epsilon}_{(M)} = \sum_{j=0}^{M} \nu_{j,M} x_{(j)} - \sum_{j=0}^{M} \nu_{j,M} x = \sum_{j=0}^{M} \nu_{j,M} \epsilon_{(j)} = \left(\sum_{j=0}^{M} \nu_{j,M} A^j \right) \epsilon_{(0)}, \tag{1.89}$$

where we have used that $\epsilon_{(j)} = A^j \epsilon_{(0)}$ in the last inequality. If we define the polynomial

$$p_M(x) := \sum_{j=0}^{M} \nu_{j,M} x^j, \quad M \geq 0, \tag{1.90}$$

we can thus write (1.89) in the form

$$\tilde{\epsilon}_{(M)} = p_M(A) \epsilon_{(0)}, \tag{1.91}$$

where $p_M(A)$ is a polynomial in the matrix A. If we assume that A is diagonalizable [6] and we express ϵ_0 in terms of the eigenbasis $\{u_i\}_{i=1}^n$, we can thus write

$$\epsilon_0 = \sum_{k=1}^{n} a_k u_k.$$

This yields

$$\tilde{\epsilon}_{(M)} = \sum_{k=1}^{n} a_k p_M(\lambda_k) u_k, \tag{1.92}$$

where λ_k is the eigenvalue associated to u_k. The Chebyshev extrapolation then restricts itself to the reduction of $\tilde{\epsilon}_{(M)}$ in the euclidean norm (that we will denote by $\|.\|$). From equation (1.92), it is clear that the minimization of $\|\tilde{\epsilon}_{(M)}\|$ requires to minimize the absolute value of the M dimensional polynomial p_M at the eigenvalues λ_k of A. We therefore have to look for the polynomial p_M^* such that

$$p_M^* = \underset{p_M(1)=1}{\arg\min} \left\{ \max_{x \in \mathrm{Sp}(A)} |p_M(x)| \right\}. \tag{1.93}$$

The resolution of this problem requires the knowledge of all the eigenvalues of A, which is a very difficult and time-consuming task and problem (1.93) is therefore replaced by the search of $\tilde{p}_M$ such that

$$\tilde{p}_M = \underset{p_M(1)=1}{\arg\min} \left\{ \max_{x \in \mathcal{E}} |p_M(x)| \right\},$$

where $\mathcal{E}$ is a compact set of the complex plane such that $\mathrm{Sp}(A) \subset \mathcal{E}$. Note that, in this new problem, the knowledge of the spectrum of the matrix is also required but in a much milder way. The solution to this problem is very well known and is given by:

$$\tilde{p}_M(z) = \frac{T_M(z)}{T_M(1/\mu)}, \quad \forall z \in \mathbb{C},\ M \in \mathbb{N} \tag{1.94}$$

where $T_M(z) = \cos\left(M \cos^{-1} z\right)$ is the Chebyshev polynomial of degree M and $\mu \in \mathcal{E}$ depends on $\mathcal{E}$. Once that the problem of minimizing $\|\tilde{\epsilon}_{(M)}\|$ has been solved thanks to formula (1.94), one could derive the coefficients $\nu_{j,M}$ and build $y_{(M)}$ using formula (1.86). However, this is not convenient

6. The interested reader can find in [117] a simpler development than the one that is presented here in the particular case in which the matrix A is hermitian and its eigenvalues are therefore real.

because one should store all the iterates $x_{(k)}$ ($k = \{0, \ldots, M\}$). To circumvent this difficulty, we may rewrite the original acceleration scheme given by (1.85)-(1.86) in the form

$$\begin{cases} x_{(M)} = Ax_{(M-1)} + g \\ y_{(M)} = b_{M,M}x_{(M)} + \sum\limits_{r=r_{start}}^{M-1} b_{r,M}y_{(r)}, \end{cases} \tag{1.95}$$

and retain as many terms in the summation as we wish by choosing $r_{start} = 0$ (we keep all the terms) or $0 < r_{start} \leq M-1$ (we keep fewer). The only requirement is that we recover the M degree Chebyshev polynomial in the residual. In the traditional Chebyshev extrapolation, we set $r_{start} = M-2$ which yields a scheme that can be written in the form:

$$\begin{cases} x_{(M)} = Ax_{(M-1)} + g \\ y_{(M)} = y_{(M-1)} + \alpha_n \left(x_{(M)} - y_{(M-1)}\right) + \beta_n \left(y_{(M-1)} - y_{(M-2)}\right). \end{cases} \tag{1.96}$$

The coefficients α_M and β_M are given by using the recursion formulae for the Chebyshev polynomials and by fixing $\mathcal{E}$. Under the assumption that $\mathrm{Sp}(A)$ is included in an ellipse with major axis on the real axis extending from a' to b' and minor axis of length $2e \leq (b'-a')$, we obtain:

$$\begin{cases} \alpha_1 = \dfrac{2a}{b'-a'}, \\ \alpha_M = \dfrac{2}{\left(1 - \frac{b'+a'}{2}\right)\rho} \dfrac{T_{M-1}(1/\rho)}{T_M(1/\rho)}, \ M > 1. \end{cases} \qquad \begin{cases} \beta_1 = 0, \\ \beta_M = \dfrac{T_{M-2}(1/\rho)}{T_M(1/\rho)}, \ M > 1. \end{cases} \tag{1.97}$$

The coefficients a, b and ρ are:

$$a = \frac{b'-a'}{2-(a'+b')}, \quad b = \frac{e}{1-\frac{a'+b'}{2}}, \quad \rho = \sqrt{a^2 - b^2}.$$

The parameters a', b' and e must be estimated for actual computation and their accuracy will have an impact on the speed of convergence of the scheme. In the case of neutron transport, the knowledge of the spectrum of the operator is still nowadays an open problem and the choice of these parameters cannot properly be done. In particular, for the implementation of this scheme in MINARET, we have made the assumption that the spectrum lies in the real axis ($e = 0$) and we have taken $b' = -a' = \rho(A)$, where $\rho(A) = \sup\limits_{\lambda \in \mathrm{Sp}(A)} |\lambda|$ is the spectral radius of A. Although this option might be simplistic, the main advantage is that it yields to a very simple form of the scheme:

$$\begin{cases} y_{(0)} = x_{(0)} \\ y_{(1)} = x_{(1)} \\ y_{(M+1)} = \omega_{M+1}\left(Ay_{(M)} + g - y_{(M-1)}\right) + y_{(M-1)}, \quad m \geq 1, \end{cases} \tag{1.98}$$

with

$$\begin{cases} \omega_1 := 1, \\ \omega_{M+1} := \dfrac{2C_M(1/\rho)}{\rho C_{M+1}(1/\rho)} = 1 + \dfrac{C_{M-1}(1/\rho)}{C_{M+1}(1/\rho)} \\ \qquad\quad = \dfrac{4/\rho^2}{4/\rho^2 - \omega_M}, \quad m \geq 1. \end{cases} \tag{1.99}$$

In this case, we only need an estimation of the spectral radius ρ.

1.6.1.2 Acceleration of the monoenergetic problem: DSA

The resolution of the multigroup equations with the (generalized) Gauss-Seidel iterative method leads to the resolution of monoenergetic problems of the form given in equation (1.32) that is recalled hereafter:

$$\boldsymbol{\omega}.\nabla\psi(\boldsymbol{r},\boldsymbol{\omega}) + \sigma_t(\boldsymbol{r})\psi(\boldsymbol{r},\boldsymbol{\omega}) - \int_{\mathbb{S}_2} \sigma_s(\boldsymbol{r},\boldsymbol{\omega'}.\boldsymbol{\omega})\psi(\boldsymbol{r},\boldsymbol{\omega'})d\boldsymbol{\omega'} = q(\boldsymbol{r},\boldsymbol{\omega}),\ \forall(\boldsymbol{r},\boldsymbol{\omega}) \in \mathcal{R}\times\mathbb{S}_2. \tag{1.100}$$

With a discrete ordinates discretization of the angular variable, we are confronted to the resolution of a set of coupled problems given by equation (1.36) that is also recalled here:

$$\begin{cases} \text{Find over } \mathcal{R} \text{ the angular flux } \psi_d(\boldsymbol{r}) \text{ that is the solution of:} \\ \boldsymbol{\omega}_d.\nabla\psi_d(\boldsymbol{r}) + \sigma_t(\boldsymbol{r})\psi_d(\boldsymbol{r}) - \sum\limits_{d'=1}^{D} \omega_{d'}\psi_{d'}(\boldsymbol{r}) \sum\limits_{l=0}^{N} \dfrac{2l+1}{4\pi}\sigma_{s,l}(\boldsymbol{r}) \sum\limits_{m=-l}^{+l} Y_{l,m}(\boldsymbol{\omega}_{d'})Y_{l,m}(\boldsymbol{\omega}_d) \\ \qquad = q_d(\boldsymbol{r}), \quad \forall d \in \{1,\dots,D\}. \end{cases} \tag{1.101}$$

(1.101) is a set of D spatial equations coupled with respect to the set of unknowns $\boldsymbol{\psi} = (\psi_d)_{d=1}^D$ and it can be written in the compact form:

$$(L - S)\,\boldsymbol{\psi} = Q, \tag{1.102}$$

where L denotes the transport operator, S the scattering source and Q the external source. After a spatial discretization, L, S and Q operate in a finite dimensional space and can be seen as matrices.

The most widespread technique for its resolution is the so-called source iteration that consists in iterating on the scattering source. If $\boldsymbol{\psi_{(m)}} = \left(\psi_{d,(m)}\right)_{d=1}^D$ is the approximation of $\boldsymbol{\psi}$ at the m-th iteration, then $\boldsymbol{\psi_{(m+1)}}$ is the solution of:

$$L\boldsymbol{\psi_{(m+1)}} = S\boldsymbol{\psi_{(m)}} + Q, \tag{1.103}$$

It is well-known from the literature that the above scheme converges (see, e.g., [1]), but at a very slow rate. By considering a Fourier expansion of the solution, it is nowadays well documented that the the Fourier modes with strong angular and spatial dependence rapidly converge whereas the modes with weak dependence on this variables converge at a much slower rate ([1]). With this observation as a starting point, the so-called *synthetic acceleration* methods aim at finding appropriate preconditioners for problem (1.101) by exploring methods that accelerate these slow converging modes.

Following the works of [89] for the steady-state solver in MINARET, in the present case of time-dependent transport, we have used the classical diffusion preconditioning, called *diffusion synthetic acceleration* (DSA), that aims at accelerating the convergence of the scalar flux ϕ (which is the first Fourier mode with respect to the angular variable). We follow the same guidelines as [1] to explain the idea:

For a given source iteration $m \geq 0$, the same transport sweep as (1.103) is performed:

$$L\boldsymbol{\psi_{(m+\frac{1}{2})}} = S\boldsymbol{\psi_{(m)}} + Q, \tag{1.104}$$

but note that now the index is $m+\frac{1}{2}$. The goal of synthetic acceleration is to formulate an equation for $\boldsymbol{\psi_{(m+1)}}$ that provides a significantly more accurate approximation to $\boldsymbol{\psi}$ than $\boldsymbol{\psi_{(m+\frac{1}{2})}}$ does. For this purpose, we substract equation (1.102) to (1.104), which gives:

$$L\left(\boldsymbol{\psi}-\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}\right)=S\left(\boldsymbol{\psi}-\boldsymbol{\psi}_{(\boldsymbol{m})}\right)=S\left(\boldsymbol{\psi}-\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}\right)+S\left(\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}-\boldsymbol{\psi}_{(\boldsymbol{m})}\right).$$

Therefore:

$$(L-S)\left(\boldsymbol{\psi}-\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}\right)=S\left(\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}-\boldsymbol{\psi}_{(\boldsymbol{m})}\right), \tag{1.105}$$

and we derive an expression of the exact solution $\boldsymbol{\psi}$ as a function of the iterates:

$$\boldsymbol{\psi}=\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}+(L-S)^{-1}S\left(\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}-\boldsymbol{\psi}_{(\boldsymbol{m})}\right). \tag{1.106}$$

Unfortunately, relation (1.106) requires the inversion of the full operator $(L-S)$ which is a problem as hard as the resolution of the inital problem (1.102). The idea of synthetic acceleration is to replace $(L-S)^{-1}$ by a "low-order" approximation

$$M^{-1}\approx(L-S)^{-1} \tag{1.107}$$

for which $M^{-1}S$ is easier to evaluate than $(L-S)^{-1}S$. Thanks to this, equation (1.106) provides an expression for $\boldsymbol{\psi}_{(\boldsymbol{m+1})}$:

$$\boldsymbol{\psi}_{(\boldsymbol{m+1})}=\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}+M^{-1}S\left(\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}-\boldsymbol{\psi}_{(\boldsymbol{m})}\right). \tag{1.108}$$

A particular choice for M involving a diffusion operator yields to the DSA scheme:

$$\begin{cases}\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}=L^{-1}S\boldsymbol{\psi}_{(\boldsymbol{m})}+L^{-1}Q,\\ \boldsymbol{\psi}_{(\boldsymbol{m+1})}=\boldsymbol{\psi}_{(\boldsymbol{m}+\frac{\boldsymbol{1}}{\boldsymbol{2}})}+e_{(m+1)},\end{cases} \tag{1.109}$$

where $e_{(m+1)}$ is the solution of the diffusion problem:

$$\begin{cases}-\operatorname{div}\left(\dfrac{1}{3\sigma_t(\boldsymbol{r})}\nabla e_{(m+1)}(\boldsymbol{r})\right)+(\sigma_t(\boldsymbol{r})-\sigma_s(\boldsymbol{r}))\,e_{(m+1)}(\boldsymbol{r})\\ \qquad=\sigma_s(\boldsymbol{r})\left(\phi_{(m+\frac{1}{2})}(\boldsymbol{r})-\phi_{(m)}(\boldsymbol{r})\right),\quad\forall\boldsymbol{r}\in\mathcal{R}\\ e_{(m+1)}(\boldsymbol{r})=0,\quad\forall\boldsymbol{r}\in\partial\mathcal{R}.\end{cases} \tag{1.110}$$

In the following, D will denote the operator

$$D(f)=-\operatorname{div}\left(\frac{1}{3\sigma_t(\boldsymbol{r})}\nabla f(\boldsymbol{r})\right)+(\sigma_t(\boldsymbol{r})-\sigma_s(\boldsymbol{r}))\,f(\boldsymbol{r})$$

In an effort to clarify the mathematical structure of this synthetic acceleration scheme, we will now show that the source iteration is a particular instance of the classical Richardson scheme and that the synthetic acceleration is a preconditioner. It first is straightforward to notice that system (1.102) is equivalent to

$$(I-L^{-1}S)\boldsymbol{\psi}=L^{-1}Q. \tag{1.111}$$

If we define $A:=I-L^{-1}S$, then, the source iteration scheme given in equation (1.103) is equivalent to

$$\boldsymbol{\psi}_{(\boldsymbol{m+1})}=(I-A)\boldsymbol{\psi}_{(\boldsymbol{m})}+L^{-1}Q, \tag{1.112}$$

which is a Richardson scheme. If we now consider the DSA scheme of (1.109) and rewrite it with the matrix A, we easily derive the relation:

$$\boldsymbol{\psi}_{(\boldsymbol{m+1})}=(I-PA)\,\boldsymbol{\psi}_{(\boldsymbol{m})}+L^{-1}Q, \tag{1.113}$$

where $P = I + D^{-1}S$. Hence, P is a preconditioner of the Richarson iterations. In the case where diffusion is a good approximation to transport, we will have $D \approx L - S$ and

$$P \approx I + (L-S)^{-1}S = (L-S)^{-1}[(L-S)+S] = (I - L^{-1}S)^{-1} = A^{-1}.$$

In other words, P will be a good approximation to A^{-1} provided that D is a good approximation to $L - S$.

A more quantitative proof for this can be found in [8] and we summarize here the main idea without providing the technical details. First, one can infer from equation (1.104) that:

$$\boldsymbol{\psi}_{(m+\frac{1}{2})} - \boldsymbol{\psi}_{(m-\frac{1}{2})} = L^{-1}S\left(\boldsymbol{\psi}_{(m)} - \boldsymbol{\psi}_{(m-1)}\right). \tag{1.114}$$

Then:

$$\begin{aligned}
\boldsymbol{\psi}_{(m)} - \boldsymbol{\psi}_{(m-1)} &= \boldsymbol{\psi}_{(m+\frac{1}{2})} - \boldsymbol{\psi}_{(m-\frac{1}{2})} + D^{-1}S\left(\left(\boldsymbol{\psi}_{(m+\frac{1}{2})} - \boldsymbol{\psi}_{(m)}\right) - \left(\boldsymbol{\psi}_{(m-\frac{1}{2})} - \boldsymbol{\psi}_{(m-1)}\right)\right) \\
&= \left((I + D^{-1}S)L^{-1}S - D^{-1}S\right)\left(\boldsymbol{\psi}_{(m)} - \boldsymbol{\psi}_{(m-1)}\right) \\
&= \left(L^{-1}S - D^{-1}S(I - L^{-1}S)\right)\left(\boldsymbol{\psi}_{(m-\frac{1}{2})} - \boldsymbol{\psi}_{(m-1)}\right) \\
&= \left(L^{-1}S(I - L^{-1}S)^{-1} - D^{-1}S\right)\left(I - L^{-1}S\right)\left(\boldsymbol{\psi}_{(m)} - \boldsymbol{\psi}_{(m-1)}\right).
\end{aligned}$$

It has been proven in theorems II.2.1.1 and IV.2.2.2 of [8] that $(I - L^{-1}S)^{-1}$ is bounded in $L^2(\mathcal{R})$ and $\|L^{-1}S(I - L^{-1}S)^{-1} - D^{-1}S\|_{\mathcal{L}(L^2(\mathcal{R}))}$ tends to zero as the physical problem tends to diffusive regimes, which is a conclusion that confirms the above discussion.

1.6.2 Parallel methods

In this section, we discuss about parallelization techniques to speed up the numerical scheme outlined in algorithm 1.1. The strategy underlying all of them is based on the idea of domain decomposition of each of the involved variables (time, energy, direction and space), i.e. the search for an expression of the global problem into smaller ones in which the subproblems can be concurrently treated on several processors. The most convenient situation occurs when one can find smaller problems that are decoupled from each other. The scaling in that case is then optimal. In our problem, this will be the case for the angular variable and, in some sense, also for the energy and spatial variables. However, in general, the subproblems are coupled and one needs to find an iterative strategy to rapidly converge to the global solution. As will be explained further, this is the case for the time variable.

Among all the techniques that are going to be presented, so far we have only included in MINARET the parallelization of the angular and temporal variables. However, other strategies can be found in the literature for the parallelization of the energy and spatial variables and they will also be outlined here since they seem interesting to keep in mind for future works.

1.6.2.1 Parallelization of the S_N directions

The numerical scheme outlined in algorithm 1.1 shows that, for a given energy group g and a given source iteration, the set of angular fluxes $\{\psi(\boldsymbol{r}, \boldsymbol{\omega}_d)\}_{d=1}^{D}$ is computed by a loop over the S_N directions (lines 7 to 10 of algorithm 1.1). In the case of vacuum boundary conditions (which is the one we have treated in our numerical applications), the spatial problem (1.44) is solved for each $\psi(\boldsymbol{r}, \boldsymbol{\omega}_d)$ and it is decoupled from the spatial problem of the other unknowns $\psi(\boldsymbol{r}, \boldsymbol{\omega}_{d'})$ $(d' \neq d)$. The loop of lines 7 to 10 is therefore an embarrassingly parallel task that can be performed concurrently on several processors by uniformly distributing the set of angular fluxes to be treated among the

different processors. For this reason, this type of parallelization is not only very easy to implement but it is also extremely efficient and we can find it in many steady state transport solvers like MINARET (see [62]) or DENOVO (see [34]). In the present work, we have extended this strategy to the time-dependent case in MINARET (the results are summarized in chapter 2).

In the case of reflective boundary conditions, the spatial problem for a given $\psi(\boldsymbol{r}, \boldsymbol{\omega}_d)$ is no longer decoupled from the other unknowns $\psi(\boldsymbol{r}, \boldsymbol{\omega}_{d'})$. The parallelization of the aforementioned loop is nevertheless still possible if each processor stores the boundary values of $\psi(\boldsymbol{r}, \boldsymbol{\omega}_d)$ for all $d \in \{1, \ldots, D\}$ (see [62]).

1.6.2.2 Parallelization of the time domain: the parareal in time algorithm

The excellent scalability properties of the parallelization of the angular variable are unfortunately limited by the number of processors that can be assigned for this task. This comes from the fact that the number of S_N angular directions is usually smaller than the classical number of available processors – because the number of directions is fixed in coherence with the accuracy of the rest of the variables. For this reason, if we have more processors at our disposal and wish additional speed-ups, the parallelization of other variables needs to be addressed. One can find in the literature techniques to parallelize the energy [34] and spatial variables [101]. However, regarding the parallelization of the time variable, the present work is (to the best of our knowledge) the first one that explores this type of parallelization in the neutron transport equation — preliminary works for neutron diffusion can be found in [14], [13] —.

Several approaches have been proposed over the years to decompose the time direction when solving a partial differential equation (see, e.g., [96], [22], [44]). Of these, the parareal in time algorithm (as for "parallel in real time"), whose performances we explore in this work, was first proposed a decade ago by [72] and has received an increasing amount of attention in the last years. During this time, the parareal method has been applied successfully to a number of applications (see, e.g., [7], [46], [110] among many others), demonstrating its versatility. Theoretical advances on this method include stability analysis ([10], [115], [9], [31]), its coupling with spatial domain decomposition methods ([86], [56]) and control problems ([83], [85]).

To see how the method works and how it has been applied to the neutron transport equation, we start by noting that equation (1.1), can be written in the following compact form:

$$\begin{cases} \dfrac{\partial y}{\partial t} + \mathcal{A}(t; y) = 0 \ , t \in [0, T]; \\ y(t = 0) = y_0, \end{cases} \tag{1.115}$$

where y will denote the unknown. In our case, $\mathcal{A}$ is the Boltzmann operator described in (1.1) and $y \equiv \psi(t, \boldsymbol{r}, \boldsymbol{\omega}, E)$. Let us assume that we have two propagators to solve (1.115):

- a fine one $\mathcal{F}_{\tau_0}^{\tau_1}(y(\tau_0))$ that, starting from time $\tau_0 \in [0, T]$ with the value $y(\tau_0)$, computes an approximation of the solution of (1.115) at time $\tau_1 \in [\tau_0, T]$ accurately but slowly,
- a coarse one $\mathcal{G}_{\tau_0}^{\tau_1}(y_0)$ that computes another approximation quickly but not so accurately (and not accurately enough).

The fine propagator $\mathcal{F}$ can, e. g., perform the propagation of the phenomenon from τ_0 to τ_1 with small time steps δt with very accurate physics described by $\mathcal{A}$. On the other hand, the coarse approximation $\mathcal{G}$ does not need to be as accurate as $\mathcal{F}$ and can be chosen much less expensive, e.g., by the use of a scheme with a much larger time step $\Delta T \gg \delta t$ or by treating "reduced physics" (i.e. by simplifying $\mathcal{A}$ into a less computer resources demanding operator). In neutronics:

- $\mathcal{F}$ will be the neutron transport propagations,
- any of the approximations to equation (1.1) explained in section 1.4 could be employed for $\mathcal{G}$.

In addition to these two propagators $\mathcal{F}$ and $\mathcal{G}$, the parareal in time algorithm is based on the division of the full interval $[0,T]$ into N sub-intervals $[0,T] = \bigcup_{n=0}^{N-1}[T_n, T_{n+1}]$ that will each be assigned to a processor P_n, assuming that we have N processors at our disposal. The parareal in time algorithm applied to (1.115) is an iterative technique where, at each iteration k, the value $y(T_n)$ is approximated by Y_n^k with an accuracy that tends to the one achieved by the fine solver when k increases. Y_n^k is obtained by the recurrence relation:

$$Y_{n+1}^{k+1} = \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^{k+1}) + \mathcal{F}_{T_n}^{T_{n+1}}(Y_n^k) - \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^k),\ n = 0, ..., N-1, \tag{1.116}$$

starting from $Y_{n+1}^0 = \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^0)$.

From formula (1.116), one can see by recursion that the method is exact after enough iterations. Indeed, for any $n > 0$, $Y_n^n = \mathcal{F}_0^{T_n}(y_0)$. However, convergence of Y_n^k to $\mathcal{F}_0^{T_n}(y_0)$ goes much faster than this as will be illustrated in our numerical example. Note also that the method can be cast into the category of predictor corrector algorithms, where the predictor is $\mathcal{G}_{T_n}^{T_{n+1}}(Y_n^{k+1})$ while the corrector is $\mathcal{F}_{T_n}^{T_{n+1}}(Y_n^k) - \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^k)$ (we refer to [49] for a detailed discussion about the several possible interpretations of the parareal method).

While the main results about the convergence properties of the method were studied in depth several years ago (see, e.g. [72], [7], [10]), more recent efforts ([87] [6] [43] [17]) focus on the algorithmics to implement it in order to improve the speed-up provided by the original algorithm suggested in [72]. This work continues this more recent trend: in chapter 2, we will discuss about a certain loss in the speed-up that we have observed in the case of neutron transport and that stems from the iterative numerical scheme employed to solve each time step (see algorithm 1.1). Chapter 3 will be devoted to the search for a solution to this problem.

1.6.2.3 Parallelization of the energy variable

We now discuss about potential ideas to parallelize the energy variable in the MINARET solver and we take the multigroup problem (1.23) as a starting point. In MINARET, its resolution has so far been performed by the generalized Gauss-Seidel iterative method that has been analyzed in section 1.6.1. We remind that this type of resolution is of the form (see equation (1.30)):

$$\begin{aligned}&(L^g - H^{g\to g})\psi_{(M+1)}^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega})\\ &\quad= \sum_{g'<g} H^{g'\to g}\psi_{(M+1)}^{g',n+1} + \sum_{g'>g} H^{g'\to g}\psi_{(M)}^{g',n+1} + \sum_{g'=1}^{G} F^{g',g}\psi_{(M)}^{g',n+1} + S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}).\end{aligned}$$

This iterative scheme is unfortunately poorly adapted if we wish to solve in parallel the unknowns $\boldsymbol{\psi^{n+1}} = (\psi^{g,n+1})_{g=1}^G$. The method should therefore be replaced by another one that could be better parallelized. The first idea would be to use a Jacobi scheme:

$$(L^g - H^{g\to g} - F^{g\to g})\psi_{(M+1)}^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \sum_{g'\neq g}\left(H^{g'\to g} + F^{g',g}\right)\psi_{(M)}^{g',n+1} + S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}), \tag{1.117}$$

for which the parallelization is easy. To the best of the author's knowledge, this possibility has never been tried out. The potential drawback is that the method might converge slowly but acceleration techniques such as the Chebyshev acceleration or DIIS (Direct inversion in the Iterative Subspace, [57]) exist and could reduce the number of iterations required.

Another option is to solve problem (1.23) with a Krylov method (e.g. GMRES). This idea has been explored in [34] for the resolution of the steady state case with good scalability results. We are going to slightly extend the scheme of [34] to describe how the method could be applied to the time dependent case:

Under the assumption that L^g is invertible in some sense, problem (1.23) is equivalent to:

$$\left(I - (L^g)^{-1}(H^g - F^g)\right)\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = (L^g)^{-1}S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}), \quad \forall g \in \{1,\dots,G\}, \tag{1.118}$$

where I is the identity operator. If we denote

$$\begin{cases} \tilde{A}^g = \left(I - (L^g)^{-1}(H^g - F^g)\right) & ; \quad q^g(\boldsymbol{r},\boldsymbol{\omega}) = (L^g)^{-1}S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}) \\ \tilde{A}_{G,G} = \begin{pmatrix} \tilde{A}^1 & 0 & \dots & 0 \\ 0 & \tilde{A}^2 & \dots & 0 \\ \vdots & \vdots & \ddots & 0 \\ 0 & 0 & \dots & \tilde{A}^G \end{pmatrix} & ; \quad \boldsymbol{q} = (q^g)_{g=1}^G, \end{cases}$$

then equation (1.118) can be written as:

$$\begin{aligned} & \tilde{A}^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = q^g(\boldsymbol{r},\boldsymbol{\omega}), \quad \forall g \in \{1,\dots,G\} \\ \Leftrightarrow \quad & \tilde{A}_{G,G}\boldsymbol{\psi^{n+1}} = \boldsymbol{q}. \end{aligned} \tag{1.119}$$

We now solve problem (1.119) with a Krylov method. If $\boldsymbol{\psi_{(0)}^{n+1}} = \left(\psi_{(0)}^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega})\right)_{g=1}^G$ is a starting guess solution for equation (1.119), let

$$\boldsymbol{r_{(0)}^{n+1}} = \tilde{A}_{G,G}\boldsymbol{\psi^{n+1}} - \boldsymbol{q} = \left(\tilde{A}^g\psi_{(0)}^{g,n+1} - q^g\right)_{g=1}^G$$

be its associated residual. Let $\mathcal{K}_p := \{\boldsymbol{r_{(0)}^{n+1}},\ \tilde{A}_{G,G}\boldsymbol{r_{(0)}^{n+1}},\dots,\tilde{A}_{G,G}^{p-1}\boldsymbol{r_{(0)}^{n+1}}\}$ be the Krylov subspace of order p associated to $\boldsymbol{r_{(0)}^{n+1}}$. It is very well known that the sequence of Krylov subspaces $(\mathcal{K}_p)$ is strictly monotonically increasing and that $\boldsymbol{\psi^{n+1}}$ belongs to the affine space $\boldsymbol{\psi_{(0)}^{n+1}} + \mathcal{K}_{p_{\max}}$ for a large enough dimension $p_{\max}$. Since $p_{\max}$ is a priori unknown and that it can be a very large number, the solution $\boldsymbol{\psi^{n+1}}$ is usually approximated by an element $\boldsymbol{\psi_{(p)}^{n+1}}$ of $\boldsymbol{\psi_{(0)}^{n+1}} + \mathcal{K}_p$ for $p < p_{\max}$ up to some accuracy. This requires the computation of:

1. the Krylov basis vectors $\tilde{A}_{G,G}^k\boldsymbol{r_{(0)}^{n+1}}$, $\forall k \in \{1,\dots,p-1\}$
2. the approximation of $\boldsymbol{\psi^{n+1}}$ by $\boldsymbol{\psi_{(p)}^{n+1}} \in \boldsymbol{\psi_{(0)}^{n+1}} + \mathcal{K}_p$.

A very common technique to do this is the GMRES method that works with an orthonomalized basis of $\mathcal{K}_p$ and finds $\boldsymbol{\psi_{(p)}^{n+1}}$ by the minimization of the euclidean norm of the residual $\boldsymbol{r_{(p)}^{n+1}} = \tilde{A}_{G,G}\boldsymbol{\psi_{(p)}^{n+1}} - \boldsymbol{q}$. The strategy is outlined in algorithm 1.2.

1: 1. $\boldsymbol{r_{(0)}^{n+1}} = \tilde{A}_{G,G}\boldsymbol{\psi_{(0)}^{n+1}} - \boldsymbol{q}$; $\boldsymbol{v_1} = \boldsymbol{r_{(0)}^{n+1}}/\|\boldsymbol{r_{(0)}^{n+1}}\|_2$; $\rho = \|\boldsymbol{r_{(0)}^{n+1}}\|_2$; $k = 0$
2: 2. While $\rho > \varepsilon$ and $k < k_{\max}$ do
3: (a) k←k+1
4: (b) $\boldsymbol{w_k} = \tilde{A}_{G,G}\boldsymbol{v_k}$
5: (c) For $j = 1,\dots,k$ do $h_{j,k} = (\boldsymbol{w_k},\boldsymbol{v_j})_2$
6: (d) $\boldsymbol{v_{k+1}} = \boldsymbol{w_k} - \sum_{j=1}^{k} h_{j,k}\boldsymbol{v_j}$
7: (e) $h_{k+1,k} = \|\boldsymbol{v_{k+1}}\|_2$
8: (f) $\boldsymbol{v_{k+1}} \leftarrow \boldsymbol{v_{k+1}}/\|\boldsymbol{v_{k+1}}\|_2$
9: (g) $\boldsymbol{e_1} = (1,0,\dots,0) \in \mathbb{R}^{k+1}$; $\rho = \|\boldsymbol{r_k^{n+1}}\|_2 = \min_{\boldsymbol{y}\in\mathbb{R}^{k+1}} \|\beta\boldsymbol{e_1} - H_{k+1,k}\boldsymbol{y}\|_2$
10: End while
11: 3. $\boldsymbol{\psi_{(k)}^{n+1}} = \boldsymbol{\psi_{(0)}^{n+1}} + \sum_{j=1}^{k} y_j\boldsymbol{v_j}$

Algorithm 1.2: Parallel resolution of the multigroup equations with a GMRES scheme

The most time consuming part in algorithm 1.2 is the matrix-vector multiplication $\boldsymbol{w_k} = \tilde{A}_{G,G}\boldsymbol{v_k} = \sum_{g=1}^{G} \tilde{A}^g v_k^g$ (line b) but the key is that this product can be very easily parallelized. If we assign an energy group to a processor, the operation $\tilde{A}^g v_k^g$ can be done in parallel for all $g \in \{1, \ldots, G\}$. Indeed:

$$\tilde{A}^g v_k^g = \left(I - (L^g)^{-1}(H^g - F^g)\right) v_k^g = v_k^g - \tilde{v}_k^g, \tag{1.120}$$

where $\tilde{v}_k^g$ is the solution of the following problem:

$$\begin{cases} L^g \tilde{v}_k^g(\boldsymbol{r}, \boldsymbol{\omega}) = (H^g + F^g)\, v_k^g(\boldsymbol{r}, \boldsymbol{\omega}) \\ \qquad\qquad = \sum_{g'=1}^{G} \left(H^{g' \to g} v_k^{g'}(\boldsymbol{r}, \boldsymbol{\omega}) + F^{g',g} v_k^{g'}(\boldsymbol{r}, \boldsymbol{\omega})\right) \\ \tilde{v}_k^g(\boldsymbol{r}, \boldsymbol{\omega}) = 0 \text{ over } \Gamma_-. \end{cases} \tag{1.121}$$

The right hand side of equation (1.121) couples the energy groups but the term $H^{g' \to g} v_k^{g'}(\boldsymbol{r}, \boldsymbol{\omega}) + F^{g',g} v_k^{g'}(\boldsymbol{r}, \boldsymbol{\omega})$ can be computed by processor g' and be provided to processor g by efficient reduce operations. Thanks to this observation, steps (b) to (f) can efficiently be treated by parallel computations in the g variable and this is the key to speed up the multigroup equations in this case. Furthermore, one can couple this strategy with the parallelization of the angular and/or spatial variable to solve the advection equation (1.121).

Last but not least, it is worth to mention that this method requires a relatively bigger storage effort than the generalized Gauss-Seidel scheme because one must store all the Krylov basis functions. Another potential difficulty is that the convergence can be slow and, in that case, preconditioners need to be included. We refer to [64] for more details about the Krylov methods and the implementation of GMRES.

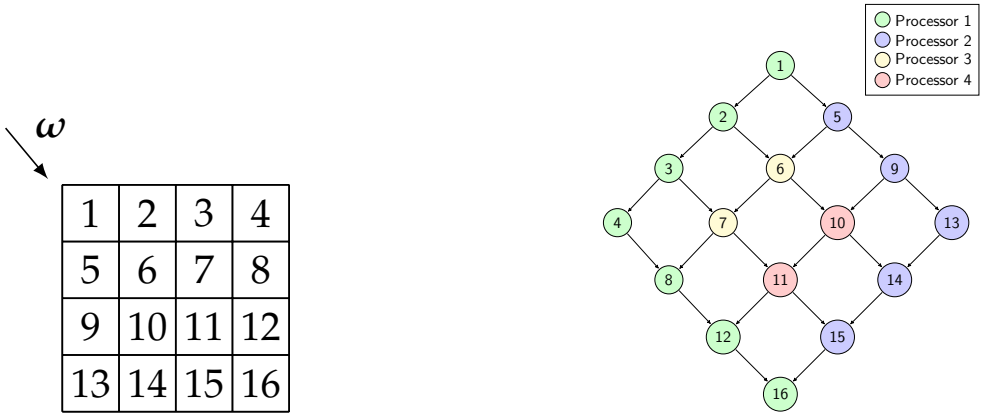
Remark 1.6.3. *We emphasize that this method has not been implemented yet in the MINARET solver.*

1.6.2.4 Parallelization of the spatial domain

In the most internal loop of the numerical scheme given in algorithm 1.1, one has to solve a set of advection spatial problems of the form of (1.44) for D directions $\boldsymbol{\omega}_d$. Let us say that we solve this problem with a finite element approach like the one used in MINARET.

A first option to parallelize this equation is to use spatial domain decomposition methods. However, because of the advective nature of the PDE, the efficient techniques existing for elliptic problems (like the ones explored in [61] for the steady state neutron SP_N equations) cannot be applied and the decomposition of the spatial domain in the transport equation remains still nowadays an open problem. We nevertheless mention the works of [50], [54] in this direction.

An alternative to this is to analyze the resolution of (1.44) from an algorithmic point of view. Since (1.44) is an advective equation, in the case of a resolution with finite elements, the problem is locally solved cell after cell by a sweeping technique. The order depends on the direction $\boldsymbol{\omega}$ because a cell cannot be solved for a particular direction until its "upstream" neighbors have been solved. In other words, a given cell can be computed provided that the incoming fluxes for this cell are known, i.e. the fluxes along cell faces for which $\boldsymbol{\omega}.\boldsymbol{n}$ is negative. As an example, consider the situation of the simple mesh of figure 1.2(a): cell number 2 cannot be solved until we have not solved cell 1. This leads to the dependency graph of figure 1.2(b). Each vertex represents a cell and the arrows are the dependencies between cells. A vertex cannot be solved until its predecessors have not been computed.



(a) A structured mesh... **(b)** ... and its data flow diagram.

Figure 1.2: Scheduling sweeping technique: an example

It is thus clear that the efficient computation in parallel of this type of problem is very difficult because one has to schedule the tasks by taking into account the constraints on the sweeping ordering. The key is to assign a set of cells to each processor so that it does not spend much time waiting for upstream tasks to be performed.

Figure 1.2(b) shows one possible solution to parallelize the situation of our simple example. By the use of 4 processors, each cell can be solved as soon as its predecessors have been determined. The computation of the 16 cells is therefore computed in 7 steps, yielding a speed-up of 16/7.

This idea has been explored in [101] where the authors have developed a task scheduling algorithm to perform spatial parallel sweeps on unstructured meshes .

Chapter 2

MINARET: Towards a parallel 3D time-dependent neutron transport solver

This is a submitted paper with J.-J. Lautard and Y. Maday. Its reference in the manuscript is:

[68] MINARET: Towards a parallel 3D time-dependent neutron transport solver. Submitted, 2014.

2.1 Introduction

In the framework of numerical simulations, the advances of computing architectures in the last decades have resulted in a progressive replacement of traditional coarse models by ever-increasing finer ones. Historically speaking, this trend has been possible on a first stage thanks to the advances of computer power capabilities per machine.

In the field of nuclear core calculations and, more particularly, regarding the deterministic resolution of the time-dependent neutron transport equation, the enhanced processor capabilities have led to significant advances. Indeed, the quite recent developments of the 2D time-dependent transport code DORT-TD [100] and its posterior extended three dimensional version TORT-TD (and even its coupling with thermal-hydraulics codes [112]) confirm this fact and have proven that the traditional diffusion [24], improved quasi-static [52] or point kinetics [107] traditional approximations can nowadays start to be overcome. While this represents a significant progress for security assessment, the long computing times of neutron transport remain still nowadays the main obstacle to face in order to definitely use transport on a regular basis. In this framework, the present work is a contribution to show that the resolution of the time dependent neutron transport equation in realistic 3D geometries is feasible in a reasonable amount of time by the use of modern parallel computer architectures together with innovative numerical schemes.

For this purpose, we start by presenting in sections 2.2 and 2.3 the newly developed time dependent 3D multigroup discrete ordinates neutron transport code that has been included in the MINARET solver (which is a tool developed at CEA in the framework of the APOLLO3® [53] project). In particular, in section 2.2 some properties and notations of the equation (that depends on time, energy, angular and space variables) will be recalled and section 2.3 will be devoted to the set of discretizations applied to each of the variables together with the numerical iterative schemes that have been implemented.

We will after come to the sequential and parallel acceleration techniques that we have explored to accelerate the computations. Our numerical results will be related to a 3D test case described in section 2.4. In section 2.5, we will present the two traditional sequential accelerations that have

been included (the Chebyshev extrapolation and the Diffusion Synthetic Acceleration) and provide some numerical evidences of their efficiency. We will also explain how extra speed-ups can be obtained by the choice of an appropriate starting guess in the outer iterations.

The remaining sections will be devoted to the main contribution of the present work which consists in the acceleration of MINARET by the parallelization of some of the present variables. Section 2.6 deals with the technique implemented to parallelize the angular variable (following the previous work of [62]). In section 2.7, a study about the additional speed-up that can be obtained if this angular parallelization is coupled with the parallelization of the time variable will be presented. This task has been performed by a domain decomposition method called the parareal in time algorithm (see [72] [7] [10] for its theoretical foundations) and has been implemented following the preliminary analysis of [14] [13] where this technique was successfully applied to the time-dependent neutron diffusion equation. We will propose an extension of the classical theoretical expected speed-up formulae (summarized in, e.g. [6]) in order to take into account the coupling of parareal with other iterative methods in a more realistic way. The comparison of these formulae with the results obtained with MINARET in a numerical test case will highlight the potentialities of parareal to accelerate long time transport computations thanks to a negligible impact of the communication time between processors.

The methods used to parallelize the angular and temporal variables are not the only possibility to exploit concurrency: the parallelization of the energy variable seems to be a promising field to explore to accelerate calculations as outlined in [113]. Regarding the spatial variables however, because of the hyperbolic nature of the space in the transport equation, the spatial domain decomposition techniques existing for elliptic problems (like the ones explored in [61] for the steady-state neutron SP_n equations) cannot be applied and the efficient decomposition of the spatial domain in the transport equation represents still nowadays a difficult problem. It will nevertheless be discussed in section 2.7.4 how the parareal in time algorithm could be used for the parallelization of the spatial variables in transport equations (as an alternative to other domain decomposition methods explored for this kind of problem like the works of [50] [54]).

2.2 The time-dependent neutron transport equation

The evolution of the angular flux ψ of neutrons in a reactor core $\mathcal{R}$ is governed by a linear Boltzmann equation whose terms physically express a balance between the free neutrons that are created and that disappear in the core. We will consider here the three-dimensional case ($\mathcal{R} \subset \mathbb{R}^3$) where ψ depends on 7 variables, namely the time $t \in [0, T]$, the position within the reactor denoted as $\boldsymbol{r} \in \mathcal{R}$, the velocity of the neutrons $\boldsymbol{v} = \sqrt{2E/m}\ \boldsymbol{\omega}$ where $E \in [E_{\min}, E_{\max}]$ stands for the energy of the neutron, $\boldsymbol{\omega} = \dfrac{\boldsymbol{v}}{|\boldsymbol{v}|} \in \mathbb{S}_2$ stands for the direction of the velocity and m is the mass of the neutron. We will have $\boldsymbol{v} \in \mathcal{V}$, where $\mathcal{V} = \mathbb{S}_2 \times [E_{\min}, E_{\max}]$ is a compact subset of $\mathbb{R}^3$. The fission chain reaction that takes place inside the core leads to the presence of some radioactive isotopes that emit neutrons with a given delay (we refer to them as precursors of delayed neutrons). This phenomenon must be taken into account in our balance equation, hence the coupling of the Boltzmann equation with a set of first order ODE's expressing the evolution in $\mathcal{R}$ of the precursors' concentration that will be denoted as $\mathbf{C} = \{C_\ell\}_{\ell \in \{1,\dots,L\}}$.

The set $(\psi, \mathbf{C})$ is thus the solution to the following initial value problem over the domain

$\mathcal{D} = \{(t, \boldsymbol{r}, \boldsymbol{\omega}, E) \in [0,T] \times \mathcal{R} \times \mathbb{S}_2 \times [E_{\min}, E_{\max}]\}$:

$$\begin{cases} \frac{1}{|\boldsymbol{v}|}\partial_t \psi(t,\boldsymbol{r},\boldsymbol{\omega},E) + (L - H - F - Q)\psi(t,\boldsymbol{r},\boldsymbol{\omega},E) = 0 \\ \partial_t C_\ell(t,\boldsymbol{r}) = -\lambda_\ell C_\ell(t,\boldsymbol{r}) \\ \qquad\qquad + \int_{E'=E_{\min}}^{E_{\max}} \beta_\ell(t,\boldsymbol{r},E')(\nu\sigma_f)(t,\boldsymbol{r},E')\phi(t,\boldsymbol{r},E')dE', \ \forall \ell \in \{1,\dots,L\}, \end{cases} \tag{2.1}$$

where $\phi(t,\boldsymbol{r},E) = \int_{\mathbb{S}_2} \psi(t,\boldsymbol{r},\boldsymbol{\omega'},E)d\boldsymbol{\omega'}$ is the scalar flux and the following operator notations have been used:

- $L\psi(t,\boldsymbol{r},\boldsymbol{\omega},E) = (\boldsymbol{\omega}.\nabla + \sigma_t(t,\boldsymbol{r},E))\,\psi(t,\boldsymbol{r},\boldsymbol{\omega},E)$ is the advection operator,
- $H\psi(t,\boldsymbol{r},\boldsymbol{\omega},E) = \int_{\mathbb{S}_2}\int_{E'=E_{\min}}^{E_{\max}} \sigma_s(t,\boldsymbol{r},\boldsymbol{\omega'} \to \boldsymbol{\omega}, E' \to E)\psi(t,\boldsymbol{r},\boldsymbol{\omega'},E')dE'd\boldsymbol{\omega'}$ is the scattering operator,
- $F\psi(t,\boldsymbol{r},\boldsymbol{\omega},E) = \frac{\chi_p(t,\boldsymbol{r},E)}{4\pi}\int_{E'=E_{\min}}^{E_{\max}} (1-\beta(t,\boldsymbol{r},E'))\,(\nu\sigma_f)(t,\boldsymbol{r},E')\phi(t,\boldsymbol{r},E')dE'$ is the prompt fission operator,
- $Q\psi(t,\boldsymbol{r},\boldsymbol{\omega},E) = \sum_{\ell=1}^{L} \lambda_\ell \chi_{d,\ell}(t,\boldsymbol{r},E)C_\ell(t,\boldsymbol{r})$ is the delayed fission source.

In the enlisted terms, $\sigma_t(t,\boldsymbol{r},E)$ denotes the total cross-section and $\sigma_s(t,\boldsymbol{r},\boldsymbol{\omega'} \to \boldsymbol{\omega}, E' \to E)$ is the scattering cross-section from energy E' and direction $\boldsymbol{\omega'}$ to energy E and direction $\boldsymbol{\omega}$. $\sigma_f(t,\boldsymbol{r},E)$ is the fission cross-section. $\nu(t,\boldsymbol{r},E)$ is the average number of neutrons emitted and $\chi_p(t,\boldsymbol{r},E)$ and $\chi_{d,\ell}(t,\boldsymbol{r},E)$ are respectively the prompt spectrum and the delayed spectrum of precursor ℓ. λ_ℓ and $\beta_\ell(t,\boldsymbol{r},E)$ are the decay constant and the delayed neutron fraction of precursor ℓ respectively and $\beta(t,\boldsymbol{r},E) = \sum_{\ell=1}^{L} \beta_\ell(t,\boldsymbol{r},E)$.

We will work with initial conditions ψ^0 and $C_{\ell,0}$ at $t=0$ and vacuum boundary conditions over $\partial\mathcal{R}$, i.e.

$$\begin{cases} \psi = 0, \ \text{on } [0,T] \times \partial\mathcal{R}_- \times \mathbb{S}_2 \times \mathbb{R}_+ \\ \psi(0,.) = \psi^0(.), \ \text{on } \mathcal{R} \times \mathbb{S}_2 \times \mathbb{R}_+ \\ C_\ell(0,.) = C_{\ell,0}(.) \ \text{on } \mathcal{R}, \end{cases}$$

where $\partial\mathcal{R}_- := \{\boldsymbol{r} \in \partial\mathcal{R} \mid \boldsymbol{\omega}.\overrightarrow{n} < 0\}$ denotes the part of the boundary where the angular flux is incoming. The knowledge of the initial conditions ψ^0 and $C_{\ell,0}$ is a complex issue in itself. In nuclear safety computations like the ones we are interested in, it is of special interest to analyze transients starting from a stable state of the core. The derivation of this state is related to an eigenvalue problem whose foundations are very well established. We refer to [107] and [33] for physical and mathematical aspects of it and to [89] for numerical details about its computation in the MINARET solver.

Remark 2.2.1 (A diffusion problem as an approximation to the transport equation (2.1))**.** *Under some given physical hypothesis (see, e.g. [107] and [33]), the angular mean value* $\phi(t,\boldsymbol{r},E) = \int_{\mathbb{S}_2} \psi(t,\boldsymbol{r},\boldsymbol{\omega'},E)d\boldsymbol{\omega'}$ *of the angular flux* $\psi(t,\boldsymbol{r},\boldsymbol{\omega},E)$ *satisfies a diffusion equation that has the advantage of being much less computationally expensive than the transport equation from the memory storage and from the computational time point of view. Although the present work deals with the resolution of the transport equation (2.1), the existence of such surrogate approximation will be used in our case in some acceleration techniques.*

2.3 Discretization and implementation in the MINARET solver

With the exception of some simple cases (see [107] for further references) where problem (2.1) can exactly be solved, the resolution of (2.1) needs to be numerically addressed and requires discretizations and approximations of the involved variables. The MINARET solver uses traditional discretization techniques and this section briefly explains them by putting special stress on the iterative numerical schemes that have been implemented.

We start by discretizing the energy variable and deriving the multigroup version of equation (2.1). The strategy is based on the division of the energy interval into G subintervals: $[E_{\min}, E_{\max}] = [E_G, E_{G-1}] \cup \cdots \cup [E_1, E_0]$. For $1 \leq g \leq G$, we denote by ψ^g the approximation of ψ in the subinterval $[E_g, E_{g-1}]$. Further, let $[0,T] = \bigcup_{n=0}^{N-1}[t_n, t_{n+1}]$ be a division of the full time interval and $\Delta T_{n+1} = t_{n+1} - t_n$. An Euler-backward scheme for the time variable is then applied. Let $\psi^{g,n}(\boldsymbol{r},\boldsymbol{\omega})$ be the approximation of $\psi(t,\boldsymbol{r},\boldsymbol{\omega},E)$ at time $t = t_n$ and for $E \in [E_g, E_{g-1}]$. Given $\{\psi^{g,n}(\boldsymbol{r},\boldsymbol{\omega})\}_{g=1}^{G}$, the set of unknowns $\{\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega})\}_{g=1}^{G}$ for the time t_{n+1} is the solution of the following set of coupled source problems:

$$\begin{cases} \text{Find over } \mathcal{R}\times\mathbb{S}_2 \text{ the angular flux } \psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) \text{ that is the solution of:} \\ \left(L^g - H^g - \tilde{F}^g - \tilde{Q}^g\right)\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \tilde{S}^{g,n}(\boldsymbol{r},\boldsymbol{\omega}), \quad \forall g \in \{1,\ldots,G\}. \end{cases} \tag{2.2}$$

The following notations have been used:

- $\tilde{S}^{g,n}(\boldsymbol{r},\boldsymbol{\omega}) := \dfrac{\psi^{g,n}(\boldsymbol{r},\boldsymbol{\omega})}{V^g \Delta T_{n+1}}$, where V^g is the average velocity of the neutrons whose energy belong to the interval $[E_g, E_{g-1}]$. Note that for the computation of $\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega})$, the term $\tilde{S}^{g,n}(\boldsymbol{r},\boldsymbol{\omega})$ is known and is a source for the equation.
- $L^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \left(\boldsymbol{\omega}.\nabla + \left(\sigma_t^{g,n+1}(\boldsymbol{r}) + \dfrac{1}{V^g\Delta T_{n+1}}\right)\right)\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega})$
- $H^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \sum\limits_{g'=1}^{G} H^{g'\to g}\psi^{g',n+1}(\boldsymbol{r},\boldsymbol{\omega})$, with
  $H^{g'\to g}\psi^{g',n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \displaystyle\int_{\mathbb{S}_2} \sigma_s^{g'\to g,n+1}(\boldsymbol{r},\boldsymbol{\omega}'\to\boldsymbol{\omega})\psi^{g',n+1}(\boldsymbol{r},\boldsymbol{\omega}')d\boldsymbol{\omega}'$.
- $\tilde{F}^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \dfrac{\chi_p^{g,n+1}(\boldsymbol{r})}{4\pi}\sum\limits_{g'=1}^{G}\left(1-\beta^{g',n+1}(\boldsymbol{r})\right)(\nu\sigma_f)^{g',n+1}(\boldsymbol{r})\phi^{g',n+1}(\boldsymbol{r})$, where $\phi^{g,n+1}(\boldsymbol{r}) = \displaystyle\int_{\mathbb{S}_2}\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega})d\boldsymbol{\omega}'$, $\forall g \in \{1,\ldots,G\}$.
- $\tilde{Q}^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \sum\limits_{\ell=1}^{L}\lambda_\ell\chi_{d,\ell}^{g,n+1}(\boldsymbol{r})C_\ell^{n+1}(\boldsymbol{r})$

The coefficients $\sigma_t^{g,n+1}(\boldsymbol{r})$, $\sigma_s^{g'\to g,n+1}(\boldsymbol{r},\boldsymbol{\omega}'\to\boldsymbol{\omega})$, $(\nu\sigma_f)^{g,n+1}$, $\chi_p^{g,n+1}(\boldsymbol{r})$ and $\chi_{d,\ell}^{g,n+1}(\boldsymbol{r})$ correspond to energy average values in $[E_g, E_{g-1}]$ at time $t = t_n$ of the coefficients $\sigma_t(t,\boldsymbol{r},E)$, $\sigma_s(t,\boldsymbol{r},\boldsymbol{\omega}'\to\boldsymbol{\omega},E'\to E)$, $\sigma_f(t,\boldsymbol{r},E)$, $\nu(t,\boldsymbol{r},E)$, $\chi_p(t,\boldsymbol{r},E)$ and $\chi_{d,\ell}(t,\boldsymbol{r},E)$. We also have $\beta^{g,n+1}(\boldsymbol{r}) = \sum\limits_{\ell=1}^{L}\beta_\ell^{g,n+1}(\boldsymbol{r})$. The Euler backward scheme applied to the precursors' equation provides $C_\ell^{n+1}(\boldsymbol{r})$ for any $\ell \in \{1,\ldots,L\}$:

$$C_\ell^{n+1}(\boldsymbol{r}) = \frac{1}{1+\lambda_\ell\Delta T_{n+1}}C_\ell^n(\boldsymbol{r}) + \frac{\Delta T_{n+1}}{1+\lambda_\ell\Delta T_{n+1}}\sum_{g'=1}^{G}\beta_\ell^{g',n+1}(\boldsymbol{r})(\nu\sigma_f)^{g',n+1}(\boldsymbol{r})\phi^{g',n+1}(\boldsymbol{r}). \tag{2.3}$$

The insertion of (2.3) into (2.2) finally yields the set of source problems:

$$\begin{cases} \text{Find over } \mathcal{R}\times\mathbb{S}_2 \text{ and } \forall g \text{ the angular flux } \psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) \text{ that is the solution of:} \\ \left(L^g - H^g - F^g\right)\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}), \end{cases} \tag{2.4}$$

where:

- $F^g\psi^{g,n+1}(\boldsymbol{r},\boldsymbol{\omega}) = \sum_{g'=1}^{G} F^{g',g}\psi^{g',n+1}$ and
  $F^{g',g}\psi^{g',n+1} = \left(\frac{\chi_p^{g,n+1}(\boldsymbol{r})}{4\pi}\left(1-\beta^{g',n+1}(\boldsymbol{r})\right) + \sum_{\ell=1}^{L}\frac{\lambda_\ell\beta_\ell^{g'}\chi_{d,\ell}^{g}\Delta T_{n+1}}{1+\lambda_\ell\Delta T_{n+1}}\right)(\nu\sigma_f)^{g',n+1}(\boldsymbol{r})\phi^{g',n+1}(\boldsymbol{r}),$
- $S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}) := \frac{\psi^{g,n}(\boldsymbol{r},\boldsymbol{\omega})}{V^g\Delta T_{n+1}} + \frac{1}{1+\lambda_\ell\Delta T_{n+1}}C_\ell^n(\boldsymbol{r}).$

Because of the coupling in the energy variable, system (2.4) is iteratively solved with a numerical method that we will call "generalized Gauss-Seidel scheme" (these are the so called "outer iterations" in neutronics). The scheme goes as follows: let $\psi_{(M)}^{g,n+1}$ be the approximation of $\psi^{g,n+1}$ at iteration number M. If we denote

$$\boldsymbol{\psi^{n+1}} = \begin{pmatrix}\psi^{1,n+1}\\ \psi^{2,n+1}\\ \vdots\\ \psi^{G,n+1}\end{pmatrix} \quad ; \quad \boldsymbol{S^n} = \begin{pmatrix}S^{1,n}\\ S^{2,n}\\ \vdots\\ S^{G,n}\end{pmatrix}, \tag{2.5}$$

then the scheme reads:

$$\begin{cases} M_{G,G}\boldsymbol{\psi_{(M+1)}^{n+1}} = N_{G,G}\boldsymbol{\psi_{(M)}^{n+1}} + \boldsymbol{S^n} \\ \boldsymbol{\phi_{(M=0)}^{n+1}} \text{ given,} \end{cases} \tag{2.6}$$

where $A_{G,G} = M_{G,G} - N_{G,G}$, with

$$M_{G,G} = \begin{pmatrix} L^1 - H^{1\to1} & 0 & \cdots & 0 \\ -H^{1\to2} & L^2 - H^{2\to2} & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ -H^{1\to G} & -H^{2\to G} & \cdots & L^G - H^{G\to G} \end{pmatrix} \tag{2.7}$$

and

$$N_{G,G} = \begin{pmatrix} F^{1,1} & H^{2\to1} + F^{2,1} & \cdots & H^{G\to1} + F^{G,1} \\ F^{1,2} & F^{2,2} & \cdots & H^{G\to2} + F^{G,2} \\ \vdots & \vdots & \ddots & \vdots \\ F^{1,G} & F^{2,G} & \cdots & F^{G,G} \end{pmatrix}. \tag{2.8}$$

Note that the difference between this scheme and a traditional Gauss-Seidel lies in the "explicit" treatment of the fission terms $F^{g',g}$ for $g' \leq g$.

Remark 2.3.1. *Despite the fact that we look for angular fluxes, in (2.6) the initial guess* $\boldsymbol{\phi_{(M=0)}^{n+1}}$ *corresponds to flux moments. This is due to the fact that our iterative scheme is built such that one does not need to give initial angular flux guesses but only flux moments.*

It has been proven in chapter 1 of [92] (theorem 1.6.1) that this scheme converges for small enough time steps. In MINARET, the iterations are performed until the average error in the scalar flux

$$e_{outer}^{n+1}(M+1) := \frac{\sum_{g=1}^{G}\int_{\mathcal{R}}|\phi_{(M+1)}^{g,n+1} - \phi_{(M)}^{g,n+1}|\phi_{(M+1)}^{g,n+1}d\boldsymbol{r'}}{\sum_{g=1}^{G}\int_{\mathcal{R}}\phi_{(M+1)}^{g,n+1}\phi_{(M+1)}^{g,n+1}d\boldsymbol{r'}} \tag{2.9}$$

goes below a given convergence threshold ε_{outer}. The convergence property implies that the choice of $\boldsymbol{\phi_{(M=0)}^{n+1}}$ will have an impact on the number of iterations required to achieve a given tolerance ε_{outer} but not in the convergence itself. The most usual choice is to take $\boldsymbol{\phi_{(M=0)}^{n+1}} = \boldsymbol{\phi_{(\infty)}^{n}}$, where

the subscript ∞ denotes converged values of the flux. As will be outlined in section 2.5, there might be cleverer choices than $\boldsymbol{\phi}^{n}_{(\infty)}$ to minimize the number of iterations required to converge.

For a given iteration M and a given energy group g, the problem to be solved reads:

$$\begin{aligned}(L^g - H^{g\to g})\psi^{g,n+1}_{(M+1)}(\boldsymbol{r},\boldsymbol{\omega}) \\ = \sum_{g'<g} H^{g'\to g}\psi^{g',n+1}_{(M+1)} + \sum_{g'>g} H^{g'\to g}\psi^{g',n+1}_{(M)} + \sum_{g'=1}^{G} F^{g',g}\psi^{g',n+1}_{(M)} + S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}),\end{aligned} \quad (2.10)$$

which is a monoenergetic problem of the form:

$$\boldsymbol{\omega}.\nabla\psi(\boldsymbol{r},\boldsymbol{\omega}) + \sigma_t(\boldsymbol{r})\psi(\boldsymbol{r},\boldsymbol{\omega}) - \int_{\mathbb{S}_2} \sigma_s(\boldsymbol{r},\boldsymbol{\omega'}\to\boldsymbol{\omega})\psi(\boldsymbol{r},\boldsymbol{\omega'})d\boldsymbol{\omega'} = q(\boldsymbol{r},\boldsymbol{\omega}),\ \forall(\boldsymbol{r},\boldsymbol{\omega})\in\mathcal{R}\times\mathbb{S}_2, \quad (2.11)$$

where the terms σ_t, σ_s, q must be understood as generic notations whose definition must be coherent with equation (2.10). Since equation (2.11) is integral in the angular variable and differential in space, a second numerical scheme is performed ("inner or source iterations"). If $\psi^{g,n+1}_{(M,m)}$ is the approximation of $\psi^{g,n+1}_{(M)}$ at the $m-th$ inner iteration, then $\psi^{g,n+1}_{(M,m+1)}$ is the solution of:

$$L^g\psi^{g,n+1}_{(M,m+1)}(\boldsymbol{r},\boldsymbol{\omega}) = H^{g\to g}\psi^{g,n+1}_{(M,m)}(\boldsymbol{r},\boldsymbol{\omega}) + \tilde{S}(\boldsymbol{r},\boldsymbol{\omega}), \quad (2.12)$$

with

$$\tilde{S}(\boldsymbol{r},\boldsymbol{\omega}) = \sum_{g'<g} H^{g'\to g}\psi^{g',n+1}_{(M+1)} + \sum_{g'>g} H^{g'\to g}\psi^{g',n+1}_{(M)} + \sum_{g'=1}^{G} F^{g',g}\psi^{g',n+1}_{(M)} + S^{g,n}(\boldsymbol{r},\boldsymbol{\omega}).$$

It has been shown in [92] (section 1.6.1.2 of chapter 1) that this strategy is equivalent to a Richardson scheme. The iterations are performed until the relative error

$$e^{g,n+1}_{inner}(m+1) := \frac{\|\phi^{g,n+1}_{(M,m+1)} - \phi^{g,n+1}_{(M,m)}\|_{L^2(\mathcal{R})}}{\|\phi^{g,n+1}_{(M,1)} - \phi^{g,n+1}_{(M,0)}\|_{L^2(\mathcal{R})}} \quad (2.13)$$

goes below a given convergence threshold ε_{inner}.

The angular discretization of equation (2.11) has been performed with the discretes ordinates of order n technique (S_n), i.e., problem (2.11) is solved for a discrete number of directions $\{\boldsymbol{\omega}_d\}_{d=1}^{D}$, where $D = n(n+2)$. The scattering operator is computed in practice by a standard expansion in Legendre polynomials of arbitrary order and the integrals in the angular variable are effectively computed by a quadrature formula involving the points and weights of the level-symmetric rule.

The space variables are treated with discontinuous Galerkin finite elements of arbitrary order and the order can be spatially adapted. The three-dimensional spatial mesh is "partially unstructured" in the sense that it is built by extrusion of an initial two-dimensional unstructured mesh (we refer to [90] for further details on MINARET's mesh generator).

Figure 2.1 summarizes the described two-stage nested iterative strategy implemented in MINARET.

```
for t_n = ΔT to NΔT do
    While(not converge) do (generalized GS iterations – see equation (2.6))
    for g = 1 to G do
        Update fission operator
        Update scattering (except self-scattering)
        While(not converge) do (source iterations)
        for ω = ω_1 to ω_D do
            Update self-scattering
            Solve spatial problem (2.12) for ω
        end for
        Diffusion Synthetic Acceleration
        End While
    end for
    Chebyshev Extrapolation
    End While
end for
```

Algorithm 2.1: The iterative strategy implemented in MINARET

2.4 Definition of the numerical test cases

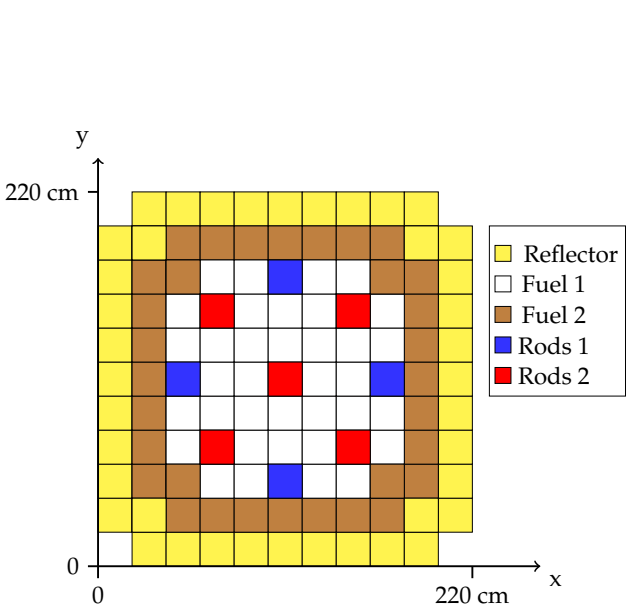
We briefly explain in this section the two test cases that will be used to illustrate the numerical performances of the acceleration methods that are going to be discussed in the remaining of this paper.

The first test case (denoted below as "case A") corresponds to the so called TWIGL benchmark and it represents a rod withdrawal (see [67]). The geometry of the core is three-dimensional and the domain is $\mathcal{R} = \{(x, y, z) \in \mathbb{R}^3 |\ 0 \leq x \leq 220\ cm;\ 0 \leq y \leq 220\ cm;\ 0 \leq z \leq 200\ cm;\ \}$. A cross-sectional view at the height $z = 180\ cm$ is specified in table 2.1a. The first group of rods (blue) is withdrawn from $t = 0$ ($z = 100\ cm$ measured starting from below) until $t = 26.6\ s.$ ($z = 180\ cm$) at a constant speed. The second group of rods (red) is inserted from $t = 7.5\ s.$ ($z = 180\ cm$) until $t = 47.7\ s.$ ($z = 60\ cm$) and the simulated interval of time is $[0, T]$ with $T = 70\ s$ (see table 2.1b). The evolution of the power is also represented in table 2.1b.

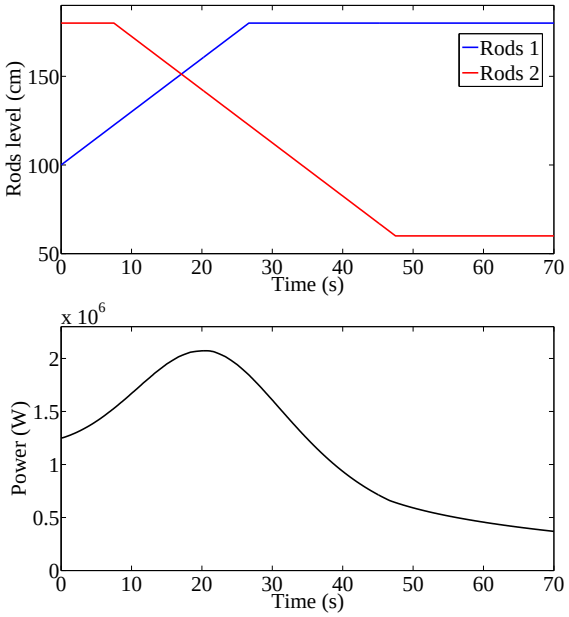
The second test case (denoted below as "case B") uses the same geometry as the TWIGL benchmark but an oscillatory sequence of motion of the rods has been devised so that power fluctuations are produced. The simulated interval of time is $[0, T]$ with $T = 250\ s$ (see table 2.2 for the details).

Both tests have been carried out with $G = 2$ energy groups, $L = 6$ precursors and vacuum boundary conditions. All the computations that will be presented hereafter have been obtained in a cluster of 38 nodes of 16 Gb memory, each one composed of 8 cores of 2814 MHz speed.

Remark 2.4.1. *In the TWIGL benchmark from the literature ([67]), calculations are done in a quarter of a core with reflective boundary conditions in the inner parts of the core. In our case, the full geometry has been computed in order to be coherent with case B that has no spatial symmetries.*

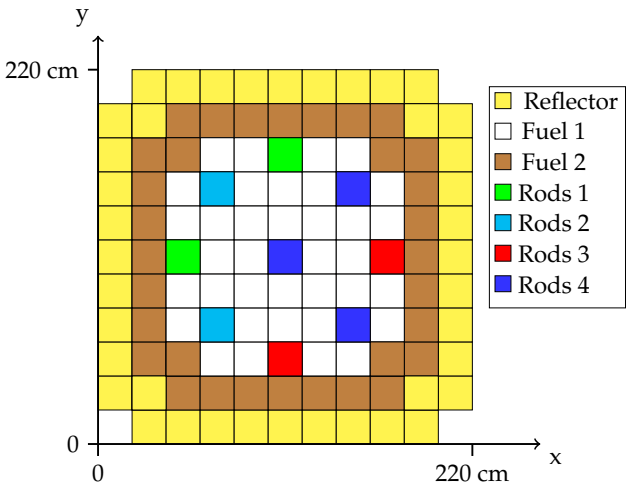


(a) Cross-sectional view of the core ($z = 180\ cm$).

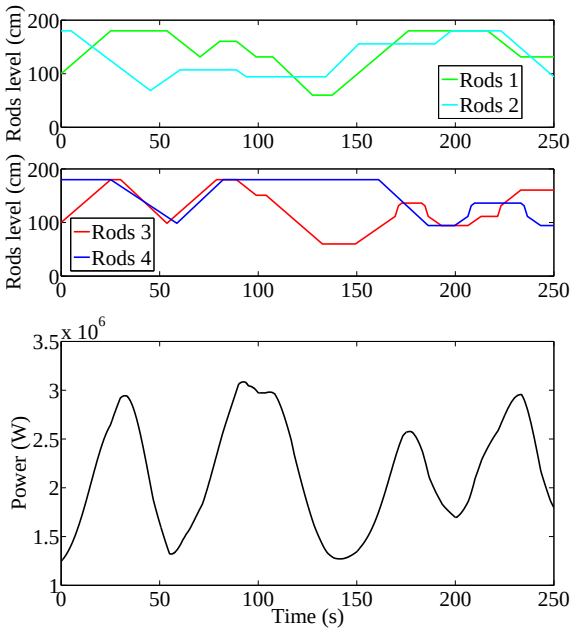


(b) Rods' vertical motion and power evolution.

Table 2.1: Case A (TWIGL benchmark).



(a) Cross-sectional view of the core ($z = 180\ cm$).



(b) Rods' vertical motion and power evolution.

Table 2.2: Case B.

2.5 Sequential acceleration techniques

The convergence of the iterative resolution of the multigroup problem given in (2.6) is often extremely slow and acceleration methods are required in order to obtain reasonable computing times.

Two traditional sequential accelerations have been included in MINARET. The first one is the Chebychev extrapolation in the outer iterations. It consists on adding a linear combination of the fluxes after each Gauss-Seidel iteration:

$$\begin{cases} M_{G,G}\boldsymbol{\psi}^{\boldsymbol{n+1}}_{\boldsymbol{(M+1/2)}} = N_{G,G}\boldsymbol{\psi}^{\boldsymbol{n+1}}_{\boldsymbol{(M)}} + \boldsymbol{S^n} \\ \boldsymbol{\phi}^{\boldsymbol{n+1}}_{\boldsymbol{(M+1)}} = \alpha_{M+1}\left(\boldsymbol{\phi}^{\boldsymbol{n+1}}_{\boldsymbol{(M+1/2)}} - \boldsymbol{\phi}^{\boldsymbol{n+1}}_{\boldsymbol{(M-1)}}\right) + \boldsymbol{\phi}^{\boldsymbol{n+1}}_{\boldsymbol{(M-1)}}, \ M \geq 1 \\ \boldsymbol{\phi}^{\boldsymbol{n+1}}_{\boldsymbol{(M=0)}} \text{ given.} \end{cases} \tag{2.14}$$

We refer to [117] for the exact form of the coefficients (α_M) and the theoretical foundations of this acceleration scheme.

Remark 2.5.1. *Note that the scheme (2.14) is well-defined thanks to the fact that* $N_{G,G}\boldsymbol{\psi}^{\boldsymbol{n+1}}_{\boldsymbol{(M)}}$ *requires the only knowledge of* $\boldsymbol{\phi}^{\boldsymbol{n+1}}_{\boldsymbol{(M+1)}}$ *(see the definition of* $N_{G,G}$ *of equation (2.8)).*

The second acceleration scheme is the so-called diffusion synthetic acceleration (DSA) that has been added for the convergence of the inner iterations. It reads:

$$\begin{cases} L^g\psi^{g,n+1}_{(M,m+1/2)}(\boldsymbol{r},\boldsymbol{\omega}) = H^{g\to g}\psi^{g,n+1}_{(M,m)}(\boldsymbol{r},\boldsymbol{\omega}) + \tilde{S}(\boldsymbol{r},\boldsymbol{\omega}), \forall g \in \{1,\dots,G\} \\ \psi^{g,n+1}_{(M,m+1)}(\boldsymbol{r},\boldsymbol{\omega}) = \psi^{g,n+1}_{(M,m+1/2)}(\boldsymbol{r},\boldsymbol{\omega}) + e^{g,n+1}_{(M,m+1)}(\boldsymbol{r}), \end{cases} \tag{2.15}$$

where $e^{g,n+1}_{(M,m+1)}$ is the solution of the diffusion problem:

$$\begin{cases} -\operatorname{div}\left(\dfrac{1}{3\left(\sigma_t^g(\boldsymbol{r}) + \frac{1}{V^g\Delta T_{n+1}}\right)}\nabla e^{g,n+1}_{(M,m+1)}(\boldsymbol{r})\right) \\ \qquad + \left(\sigma_t^g(\boldsymbol{r}) + \frac{1}{V^g\Delta T_{n+1}} - \sigma_s^{g,g}(\boldsymbol{r})\right) e^{g,n+1}_{(M,m+1)}(\boldsymbol{r}) \\ \qquad = \sigma_s(\boldsymbol{r})\left(\phi^{g,n+1}_{(M,m+\frac{1}{2})}(\boldsymbol{r}) - \phi^{g,n+1}_{(M,m)}(\boldsymbol{r})\right), \quad \forall \boldsymbol{r} \in \mathcal{R} \\ e^g_{(m+1)}(\boldsymbol{r}) = 0, \quad \forall \boldsymbol{r} \in \partial\mathcal{R}. \end{cases} \tag{2.16}$$

DSA is an acceleration scheme because it acts as a preconditioner of transport to solve equation (2.12). We refer to [1] (sections I.D and II.B) and [89] for more theoretical details about this method. In MINARET, the spatial resolution of the DSA problem (2.16) is discretized with discontinuous Galerkin finite elements of the same order than the ones employed in problem (2.12). The discretized DSA problem is iteratively solved with a conjugate gradient method preconditioned by SSOR. If r_i denotes the residual at the i-th iteration, the DSA iterations are performed until the ratio

$$\frac{\|r_i\|_{L^2(\mathcal{R})}}{\|r_0\|_{L^2(\mathcal{R})}} \tag{2.17}$$

goes below a given convergence threshold ε_{DSA}.

We illustrate the performances in MINARET of both acceleration methods through some numerical results obtained for the test case A.

To begin with, table 2.3 lists the number of outer iterations M_{outer}, inner iterations N_{inner} and DSA iterations N_{DSA} required to perform a propagation from time 0 to time 5/3 s. in an S_4 calculation. We also provide the exact computing times obtained in our cluster. The convergence criteria associated to the errors (2.9), (2.13) and (2.17) have been fixed to $\varepsilon_{outer} = 10^{-5}$, $\varepsilon_{inner} = 10^{-1}$ and $\varepsilon_{DSA} = 10^{-2}$. The product $M_{outer}N_{inner}$ is also given as an estimation of the complexity of the resolution (the complexity added by the DSA can be neglected in a first approach in a sequential calculation). As the first case of table 2.3 shows, it is clear that the solver needs acceleration

techniques in order to converge in a reasonable time. While the inclusion of the Chebyshev extrapolation (case 2) already represents a dramatic improvement in the computing time (by reducing about 10 times the number of outer iterations), this performance can still be improved by another factor of about 10 if the Chebyshev extrapolation is coupled with DSA in the inner iterations (case 3). This is achieved thanks to the reduction of the number of inner iterations.

Case	Chebyshev	DSA	M_{outer}	N_{inner}	N_{DSA}	$M_{outer}N_{inner}$	Computing time (s)
1	No	No	678	29784	0	$\approx 2000.10^4$	7510
2	Yes	No	67	2900	0	$\approx 19.10^4$	736.5
3	Yes	Yes	59	345	1557	$\approx 2.10^4$	87.67

Table 2.3: An illustration of the impact on the speed-up performances of the Chebyshev extrapolation and the DSA.

Another factor of about 3 can further be obtained if the initial guess $\boldsymbol{\phi^{n+1}_{(M=0,N=0)}}$ of the outer iterations is well chosen. The classical choice is

$$\phi^{g,n+1}_{(M=0,N=0)} = \phi^{g,n}_{(\infty)}, \quad \forall g \in \{1,\dots,G\}. \tag{2.18}$$

This option is reasonable because for small time steps one could conjecture that the system does not change very much from t_n to t_{n+1}. Other possibilities that exploit the information of the previous time steps have been explored (these are at the cost of storing additional information). One can first try a linear extrapolation of the flux:

$$\phi^{g,n+1}_{(M=0,N=0)} = \phi^{g,n}_{(\infty)} + \frac{t_{n+1}-t_n}{t_n - t_{n-1}}\left(\phi^{g,n}_{(\infty)} - \phi^{g,n-1}_{(\infty)}\right), \quad \forall g \in \{1,\dots,G\}. \tag{2.19}$$

However, according to the point kinetics approximation, the behavior of the flux is rather exponential and another idea would be an exponential extrapolation:

$$\phi^{g,n+1}_{(M=0,N=0)} = \phi^{g,n}_{(\infty)} \exp^{\frac{t_{n+1}-t_n}{t_n-t_{n-1}} \ln\left(\frac{\phi^{g,n}_{(\infty)}}{\phi^{g,n-1}_{(\infty)}}\right)}, \quad \forall g \in \{1,\dots,G\}. \tag{2.20}$$

Another interesting option is to use the diffusion approximation to build a two-level propagation scheme. The idea goes as follows: the computation of the solution with the diffusion approximation can be obtained very quickly in comparison with the transport solution. For a given time t_{n+1}, we can therefore compute the solution at t_{n+1} coming from the diffusion (denoted here as $\tilde{\phi}^{g,n+1}_{(\infty)}$) and use it as a starting guess to compute $\phi^{g,n+1}$:

$$\phi^{g,n+1}_{(M=0,N=0)} = \tilde{\phi}^{g,n+1}_{(\infty)}, \quad \forall g \in \{1,\dots,G\}.. \tag{2.21}$$

As will be illustrated in the numerical results, this is a bad choice whose main problem is that the diffusion solution has a different orbit than the transport one, hence the degraded computing times (the transport solver needs to correct the orbit and converge to the transport solution). However, since the diffusion approximation seems to present the good trend, one can conjecture that

$$\frac{\phi^{g,n+1} - \phi^{g,n}}{t_{n+1}-t_n} \approx \frac{\tilde{\phi}^{g,n+1} - \tilde{\phi}^{g,n}}{t_{n+1}-t_n}.$$

In this case, we can try:

$$\phi^{g,n+1}_{(M=0,N=0)} = \phi^{g,n}_{(\infty)} + \tilde{\phi}^{g,n+1}_{(\infty)} - \tilde{\phi}^{g,n}_{(\infty)}, \quad \forall g \in \{1,\dots,G\}. \tag{2.22}$$

The numerical results will show that this is a good starting guess. Furthermore, if we suppose that the trend is exponential as point kinetics suggests, an interesting initial guess could be:

$$\phi^{g,n+1}_{(M=0,N=0)} = \phi^{g,n}_{(\infty)} \exp^{\frac{t_{n+1}-t_n}{t_n-t_{n-1}} \ln\left(\frac{\tilde{\phi}^{g,n+1}_{(\infty)}}{\tilde{\phi}^{g,n}_{(\infty)}}\right)}, \quad \forall g \in \{1,\dots,G\}. \tag{2.23}$$

Starting guess	Formula
A (traditional)	$\phi^{g,n+1}_{(M=0,N=0)} = \phi^{g,n}_{(\infty)}$
B (linear extrapolation)	$\phi^{g,n+1}_{(M=0,N=0)} = \phi^{g,n}_{(\infty)} + \frac{t_{n+1}-t_n}{t_n-t_{n-1}}\left(\phi^{g,n}_{(\infty)} - \phi^{g,n-1}_{(\infty)}\right)$
C (exponential extrapolation)	$\phi^{g,n+1}_{(M=0,N=0)} = \phi^{g,n}_{(\infty)} \exp^{\frac{t_{n+1}-t_n}{t_n-t_{n-1}} \ln\left(\frac{\phi^{g,n}_{(\infty)}}{\phi^{g,n-1}_{(\infty)}}\right)}$
D (plain multilevel)	$\phi^{g,n+1}_{(M=0,N=0)} = \tilde{\phi}^{g,n+1}_{(\infty)}$
E (multilevel linear)	$\phi^{g,n+1}_{(M=0,N=0)} = \phi^{g,n}_{(\infty)} + \tilde{\phi}^{g,n+1}_{(\infty)} - \tilde{\phi}^{g,n}_{(\infty)}$
F (multilevel exponential)	$\phi^{g,n+1}_{(M=0,N=0)} = \phi^{g,n}_{(\infty)} \exp^{\frac{t_{n+1}-t_n}{t_n-t_{n-1}} \ln\left(\frac{\tilde{\phi}^{g,n+1}_{(\infty)}}{\tilde{\phi}^{g,n}_{(\infty)}}\right)}$

Table 2.4: List of the explored starting guesses.

We summarize all the options in table 2.4. Their perfomances have been tested in "case A" with a constant time-step of 5/3 s. In figure 2.1 we plot the computing times per time step as well as the cumulative ones. We also plot M_{outer} and $M_{outer}N_{inner}N_{DSA}$. From these figures, it seems thus clear that the use of a multilevel scheme outperforms the rest of the approaches provided that we do a linear or exponential extrapolation. The computing times are reduced by a factor of about 3 with this strategy. Options B and C provide a more moderate gain compared to the traditional case A. As it can be observed from the figures, the speed up comes from the reduction of the number of outer iterations M_{outer}, which results in a dramatic reduction of the total number of iterations $M_{outer}N_{inner}N_{DSA}$.

Once these sequential acceleration techniques have been implemented, very few gain in the speed-up can be obtained by adding other sequential techniques to the code and, if additional speed-ups are required, it is necessary to explore efficient parallelization techniques. We therefore devote the rest of the paper to the analysis of the parallelization of the angular and the temporal variables.

2.6 Parallelization of the angular variable

The numerical scheme outlined in algorithm 2.1 shows that, for a given energy group g and a given inner iteration m, the set of angular fluxes $\{\psi(\boldsymbol{r},\boldsymbol{\omega}_d)\}_{d=1}^{D}$ is computed by a loop over the S_n directions (lines 7 to 10 of algorithm 2.1). For each $\psi(\boldsymbol{r},\boldsymbol{\omega}_d)$, the spatial problem 2.12 is solved and it is decoupled from the spatial problem of the other unknowns $\psi(\boldsymbol{r},\boldsymbol{\omega}_{d'})$ $(d' \neq d)$. The loop of lines 7 to 10 is therefore an embarrassingly parallel task that can be performed concurrently on several processors by uniformly distributing the set of angular fluxes to be treated among the different processors.

From an implementation point of view, the distribution of the tasks is performed in MINARET in a master-slave fashion with the MPI library. This implementation strategy has the important advantage of alleviating the memory storage per processor in comparison with a sequential implementation because each processor stores only the angular fluxes $\psi(\boldsymbol{r},\boldsymbol{\omega}_d)$ of its assigned directions

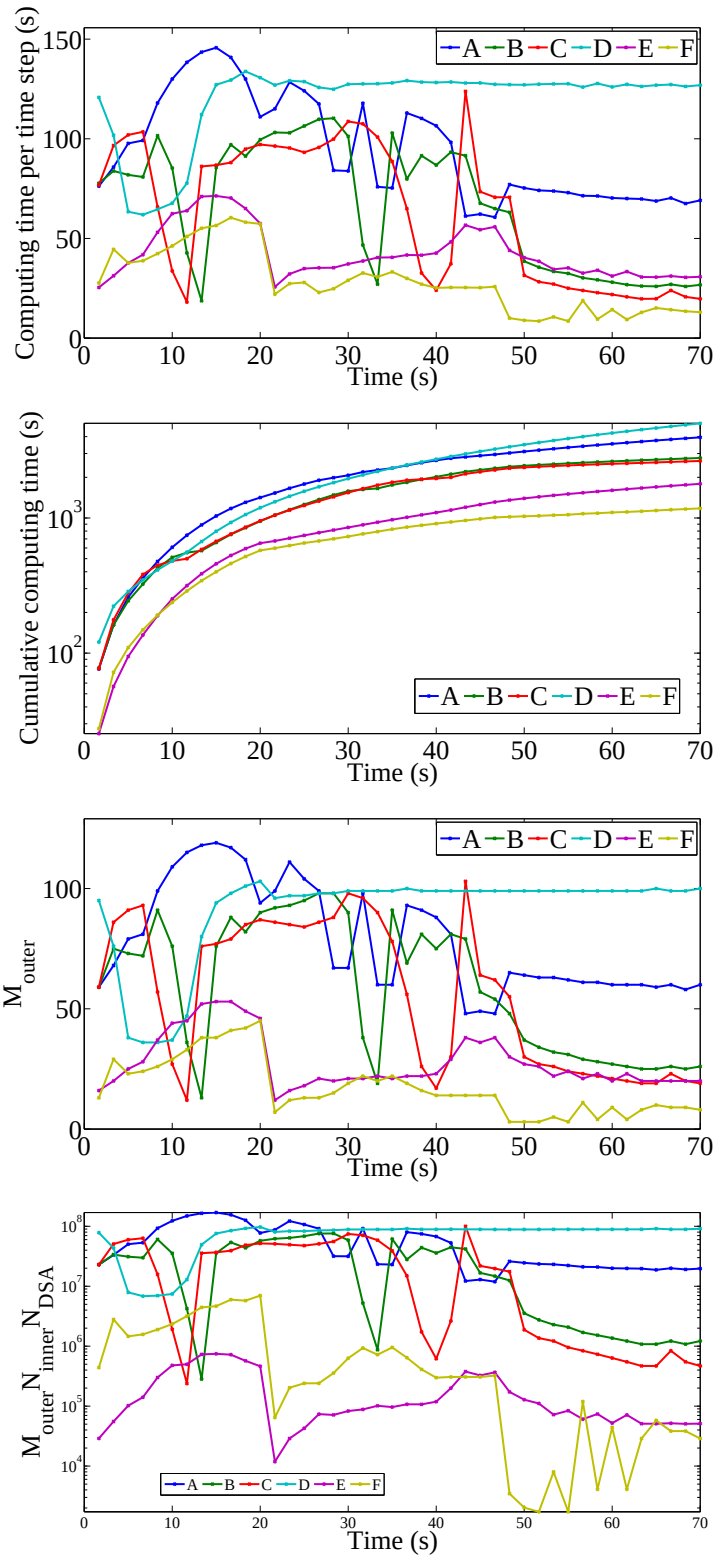


Figure 2.1: Performances of the initial guesses.

(and the moments of the flux are only stored by the master). Thanks to this fact, MINARET can address time-depend calculations involving a large number of directions and leading to HPC problems (we refer to [62] for similar results on this topic but for the steady state case).

At the end of each inner iteration, the master gathers all the angular fluxes $\{\psi(\boldsymbol{r}, \boldsymbol{\omega}_d)\}_{d=1}^{D}$. After computing the scalar flux $\phi(\boldsymbol{r})$, it performs the diffusion synthetic acceleration.

Table 2.5 and figure 2.2a show the numerical performances of this implementation in a strong scaling test regarding the angular variable: the test case A has been performed for a fixed number of directions $D = 24$ with an increasing number of processors N that treat the loop over the directions. There is a trade-off between:

- the number of directions assigned to each processor
- the spatial complexity for the calculation of an unknown $\psi(\boldsymbol{r}, \boldsymbol{\omega}_d)$ (resolution of problem 2.12)
- the computing time required to perform the DSA step (that is run sequentially)

For a reduced number of processors, the algorithm has excellent scalability properties ($N \leq 8$). The behavior is degraded for larger values of N because the amount of work assigned to each processor decreases. The time to perform the loop on the angular directions is therefore reduced whereas the time to do the DSA remains constant because it is not parallelized: the DSA becomes a bottleneck. This issue could be overcome by its parallelization with domain decomposition methods or multigrid techniques like in the works of [4] and [93] respectively.

As a consequence of all this factors, in order not to lose much efficiency, there is a minimum number of directions $\boldsymbol{\omega}_d$ that need to be treated by each processor. In the present case, the most reasonable choice according to this criterion seems to be to assign $N/D = 4$ directions per processor (see table 2.5).

It is also desirable that the number of processors N is a divisor of the total number of directions D in order to have an uniform distribution of the tasks between processors. This is indeed a source of inefficiency as illustrated in table 2.5 for the case $N = 10$ (some processors will treat 3 directions and others only 2).

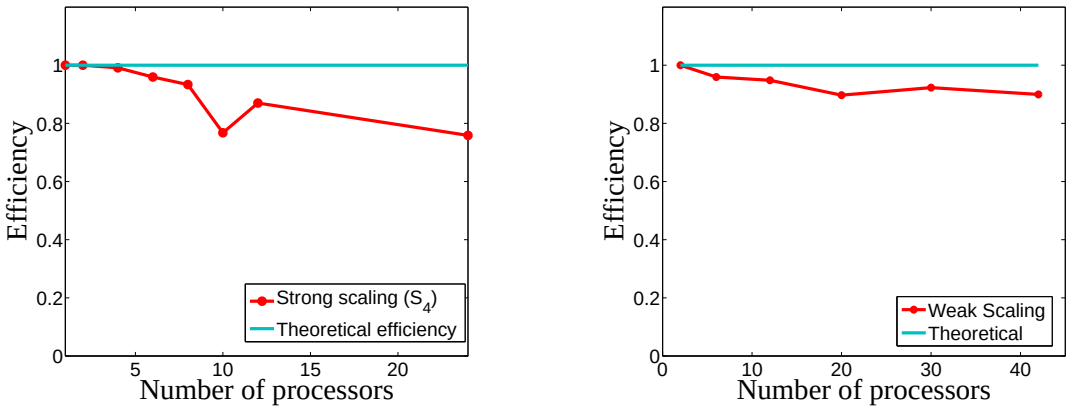
With the "optimal" number of $N/D = 4$ directions per processor being fixed, a weak scaling test has been performed where the angular S_n approximation is increased (D increases) by incrementing the number N of processors. The results are summarized in table 2.6 and figure 2.2b where it can be noticed that the efficiency is almost not degraded as N increases. This is a numerical proof that shows that, provided that we have enough processors at our disposal, extremely precise S_n approximations can be performed without increasing the total computing time in comparison with lower S_n approximations.

D	24	24	24	24	24	24	24	24
N	1	2	4	6	8	10	12	24
D/N_{proc}	24	12	6	4	3	2 or 3	2	1
Efficiency	1	1	0.99	0.96	0.933	0.77	0.87	0.76

Table 2.5: Efficiency in the strong scaling test for the angular variable (case A)

S_n approx	2	4	6	8	10	12
N_{dir}	8	24	48	80	120	168
N_{proc}	2	6	12	20	30	42
N_{dir}/proc	4	4	4	4	4	4
Efficiency	1	0.96	0.95	0.90	0.922	0.90

Table 2.6: Efficiency in the weak scaling test for the angular variable (case A).



(a) Strong scaling. **(b)** Weak scaling.

Figure 2.2: Efficiency in the parallelization of the angular variable (case A)

2.7 Parallelization of the time variable

As has been outlined in the previous section, an efficient technique for the acceleration of the resolution of the time dependent neutron transport equation is the parallelization of the angular variable. Its performances seem to be only slightly degraded in weak scaling cases, which implies that arbitrary high S_n orders can be addressed in a reasonable time. The most usual case, however, is to fix the S_n angular accuracy in coherence with the accuracy fixed for other variables (like, e.g., the spatial variable in which the accuracy is given by the finite element polynomial approximation). For this reason, the number of allocated processors to efficiently accelerate a given calculation is upper bounded and, if we have more processors at our disposal and wish additional speed-ups, the parallelization of other variables needs to be addressed. In this context, it is interesting to consider the extra speed-up that can bring the parallelization of the temporal variable. In the present case, this task has been adressed by a domain decomposition technique: the parareal in time algorithm. This section is organized as follows: after a brief recall of the basics of the parareal in time algorithm, an extension of the traditional theoretical speed-up formula will be proposed in order to properly take into account our particular case in which parareal is coupled with other iterative techniques at each time propagation. Finally, an analysis of the performances of the method for the resolution of transport transients with MINARET will be presented. The implemented results consider the parallelization of the time without coupling it with the parallelization of the angle. They are nevertheless representative enough of the accelerations that could be obtained in addition to the ones provided by the angular parallelization.

2.7.1 The parareal in time algorithm

The unsteady problem (2.1) can be written in a more compact form:

$$\frac{\partial y}{\partial t} + \mathcal{A}(t; y) = 0 \ , t \in [0, T]; \tag{2.24}$$

it is complemented with initial conditions: $y(t = 0) = y_0$.

We assume that we have two propagators to solve (2.24): a fine one $\mathcal{F}_{\tau_0}^{\tau_1}(y(\tau_0))$ that, starting from time $\tau_0 \in [0, T]$ with the value $y(\tau_0)$, computes an approximation of the solution of (2.24) at time $\tau_1 \in [\tau_0, T]$ accurately but slowly, and a coarse one $\mathcal{G}_{\tau_0}^{\tau_1}(y_0)$ that computes an other approximation quickly but not so accurately (and not accurately enough). The fine propagator $\mathcal{F}$ can, e.

g., perform the propagation of the phenomenon from τ_0 to τ_1 with small time steps δt with very accurate physics described by $\mathcal{A}$. On the other hand, the coarse approximation $\mathcal{G}$ does not need to be as accurate as $\mathcal{F}$ and can be chosen much less expensive e.g. by the use of a scheme with a much larger time step $\Delta T \gg \delta t$ or by treating "reduced physics" (i.e. by simplifying $\mathcal{A}$ into a less computer resources demanding operator).

In addition to these two propagators $\mathcal{F}$ and $\mathcal{G}$, the parareal in time algorithm is based on the division of the full interval $[0,T]$ into N sub-intervals $[0,T] = \bigcup_{n=0}^{N-1}[T_n, T_{n+1}]$ that will each be assigned to a processor P_n, assuming that we have N processors at our disposal. The parareal in time algorithm applied to (2.24) is an iterative technique where, at each iteration k, the value $y(T_n)$ is approximated by Y_n^k with an accuracy that tends to the one achieved by the fine solver when k increases. Y_n^k is obtained by the recurrence relation:

$$Y_{n+1}^{k+1} = \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^{k+1}) + \mathcal{F}_{T_n}^{T_{n+1}}(Y_n^k) - \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^k),\ n = 0, ..., N-1 \tag{2.25}$$

starting from $Y_{n+1}^0 = \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^0)$.

From formula (2.25), it can first of all be seen by recursion that the method is exact after enough iterations. Indeed, for any $n > 0$, $Y_n^n = \mathcal{F}_0^{T_n}(y_0)$. However, convergence of Y_n^k to $\mathcal{F}_0^{T_n}(y_0)$ goes much faster than this as will be illustrated in our numerical example. Second, by the recurrence formula (2.25), the parareal in time algorithm can be cast in the category of predictor corrector algorithms, where the predictor is $\mathcal{G}_{T_n}^{T_{n+1}}(Y_n^{k+1})$ while the corrector is $\mathcal{F}_{T_n}^{T_{n+1}}(Y_n^k) - \mathcal{G}_{T_n}^{T_{n+1}}(Y_n^k)$ (we refer to [49] for a detailed discussion about the several possible interpretrations of the parareal method).

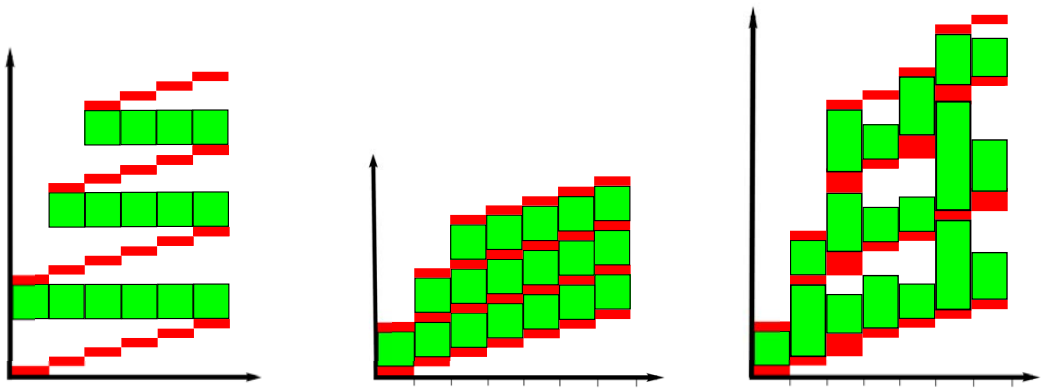
2.7.2 Algorithmics and theoretical speed-up

While the main results about the convergence properties of the method were studied in depth a decade ago (see, e.g. [72] [7] [10]), more recent efforts ([87] [6] [43] [17]) focus on the algorithmics to implement it in order to improve the speed-up provided by the original algorithm suggested in [72]. It consisted on a master-slave type of implementation where the master carried out the coarse propagation in the whole time interval $[0;T]$, each slave being in charge of the fine propagations over its assigned time slice and sending $\mathcal{F}_{T_n}^{T_{n+1}}(Y_n^k)$ to the master so that the master computed the parareal corrections of equation 2.25, $\forall n$. This original algorithm gives rise to two main computing drawbacks: the coarse propagation by the master is a bottleneck in the computation and the memory requirement in the master processor scales linearly with the number of slaves.

A remedy to both drawbacks is a distributed algorithm that was suggested in [6]: for each processor P_n, the fine and the coarse solvers are propagated over $[T_n, T_{n+1}]$ and the parareal correction Y_{n+1}^{k+1} is carried out. The process is repeated until convergence, i.e. $\|Y_n^{k+1} - Y_n^k\| < \eta$, $\forall n$, where η is a given tolerance. A graphical description of the master-slave and distributed algorithms is shown in figures 2.3a and 2.3b in the ideal case where each processor is identical and the communication time is negligible.

It is easy to realize that the distributed implementation does not change the number of iterations in order the parareal algorithm to converge but it provides better speed-ups than the original master-slave version (see formula (2.26) below). This is the reason why the distributed algorithm has been implemented in this study.

To the best of the authors knowledge, the theoretical analysis for the maximum attainable speed-up provided by the parareal algorithm in different types of algorithms has always been made under the assumptions that the computational cost of the fine and the coarse solvers is identical from one processor to another and that the communication time is negligible. Under these hypothesis, the maximum speed-up for the master-slave (S_{MS}) and distributed algorithms (S_D) are respectively (see [6]):



(a) A balanced master-slave scheme.

(b) A balanced distributed scheme.

(c) A distributed scheme with task imbalance.

Figure 2.3: Two different algorithms to implement the parareal in time method (a-b) and an illustration of the imbalance in the tasks (c) observed when the parareal algorithm is coupled with other iterative schemes for each time step propagation (in the example, $k^* = 3$ and $N = 7$ processors).

$$S_{MS} = \frac{T_{seq}}{T_{para,MS}} = \frac{N}{Nr(1+k^*)+k^*} \quad ; \quad S_D = \frac{T_{seq}}{T_{para,D}} = \frac{N}{Nr+k^*(1+r)}, \tag{2.26}$$

where k^* is the number of parareal iterations needed in order to converge and $r = \dfrac{T_{\mathcal{G}}}{T_{\mathcal{F}}}$, $T_{\mathcal{G}}$ and $T_{\mathcal{F}}$ are the computational costs of the coarse and fine propagators per processor. Note that $S_D > S_{MS}$ for any k^*, r and $N > 1$.

In the case that the fine and the coarse propagators solve each time step with an iterative numerical method, it is possible that the cost of the fine and the coarse solvers dramatically vary form one processor to another depending on the numerical complexity of the events that take place in each time slice ΔT (and this complexity cannot be predicted a priori). Figure 2.3c illustrates this fact. Formulae (2.26) need therefore to be extended to the broader case in which the computational costs $T_{\mathcal{G}} = T_{\mathcal{G}}(k,p)$ and $T_{\mathcal{F}} = T_{\mathcal{F}}(k,p)$ depend on the processor p and the parareal iteration k. It is easy to show that a more adequate formula for the speed-up in this case is:

$$\begin{cases} \tilde{S}_D &= \dfrac{T_{seq}}{\tilde{T}_{para,D}} \\ &= \dfrac{T_{seq}}{\sum\limits_{p=0}^{N-1} T_{\mathcal{G}}(0,p) + \sum\limits_{k=1}^{k^*} \max\limits_{p\in\{0,\dots,N-1\}} (T_{\mathcal{G}}(k,p) + T_{\mathcal{F}}(k,p))}, \\ \tilde{S}_{MS} &= \dfrac{T_{seq}}{\tilde{T}_{para,MS}} \\ &= \dfrac{T_{seq}}{\sum\limits_{p=0}^{N-1} T_{\mathcal{G}}(0,p) + \sum\limits_{k=1}^{k^*} \left(\sum\limits_{p=0}^{N-1} T_{\mathcal{G}}(k,p) + \max\limits_{p\in\{0,\dots,N-1\}} T_{\mathcal{F}}(k,p) \right)}, \end{cases} \tag{2.27}$$

where the communication time between processors has been neglected. Note that in the generalized formulae (2.27), we also find that $\tilde{S}_D > \tilde{S}_{MS}$ since we have $\tilde{T}_{para,MS} - \tilde{T}_{para,D} \geq \sum\limits_{k=1}^{k^*} \sum\limits_{p\neq p^*(k)} T_{\mathcal{G}}(k,p) > 0$, where $T_{\mathcal{G}}(k,p^*) := \max\limits_{p\in\{0,\dots,N-1\}} T_{\mathcal{G}}(k,p)$.

Remark 2.7.1. *Slightly better speed-ups than the ones provided by the distributed algorithm can be achieved with the event-based parareal algorithm suggested by [17] which, in turn, represents a major improvement from the processor utilization point of view. The algorithm exploits the fact that the coarse and fine propagations can be considered as a collection of tasks that can be treated by a processor as soon as their initial conditions are fulfilled. Once the task is performed, the processor treats the following task, if any, leading to an optimization of the processor utilization. However, since the present work focuses essentially on the feasibility and attainable speed-ups of parareal applied to equation (2.4), the distributed algorithm has been selected for its simpler implementation.*

2.7.3 Numerical application

The parareal algorithm has been applied to the resolution of the test cases A and B. An S_4 transport propagator has been used as the fine solver whereas two coarse solvers have been tried out:

- an S_4 transport propagator (the only difference with the fine solver is the size of the time steps used: δt for $\mathcal{F}$ and $\Delta t = T_{n+1} - T_n > \delta t$ for $\mathcal{G}$),
- a diffusion propagator.

All calculations have been evaluated with a convergence test (for the parareal iterations) in which the tolerance η has been fixed to the precision of the numerical scheme (i. e. $\eta \sim 10^{-3}$). The tolerance in the convergence for the outer and inner iterations has been fixed to $\varepsilon_{outer} = 10^{-5}$ and $\varepsilon_{inner} = 10^{-1}$. With this thresholds, parareal convergence has been achieved after only $k^* = 2$, 3 or at most 4 iterations of the parareal in time algorithm.

In the following subsections, after giving a numerical proof of the convergence of the parareal algorithm in our case of study, some results about measured speed-ups will be presented.

2.7.3.1 A numerical proof of the convergence

Figure 2.4 illustrates that parareal effectively converges in the particular case where both propagators use S_4 transport to solve test case A. The fine solver has a time step $\delta t = 5/3$ $s.$ and the coarse one $\Delta t = 4\delta t$.

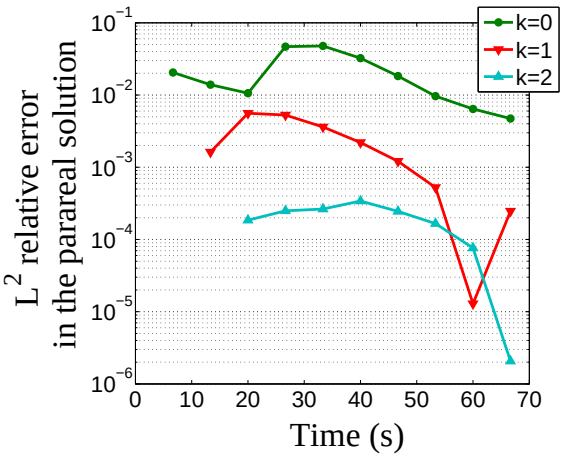


Figure 2.4: An example of the numerical convergence of the parareal algorithm in our neutron transport case.

The points represent the errors

$$e^k(T_n) = \frac{\|\Phi_n^k - \mathcal{F}_0^{T_n}(\Phi_0)\|_{L^2}}{\|\mathcal{F}_0^{T_n}(\Phi_0)\|_{L^2}}, \ \forall n \in \{0, 1, ...N\}, \ k \in \{0, 1, 2\} \tag{2.28}$$

between the parareal scalar flux Φ_n^k and the sequential fine solution $\mathcal{F}_0^{T_n}(\Phi_0)$.

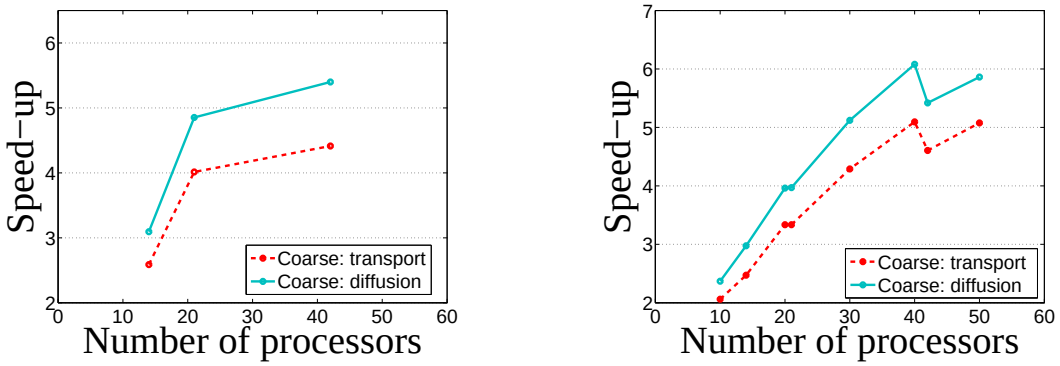
2.7.3.2 Speed-up performances

In the following strong and weak scaling tests, the fine solver has a fixed time step of $\delta t = 1/12$ s.

Strong scaling results: For the strong scaling analysis, the test case A has been solved with MINARET on an increasing number N of processors. The size of each sub-interval $T_{n+1} - T_n$ is constant for all n and equal to the time step of the coarse solver Δt. In order to increase the number of processors solving the transient in the fixed time interval $[0; 70\ s.]$, the coarse time step has been reduced from $\Delta t = 60\delta t$ to $\Delta t = 20\delta t$.

The measured speed-ups are plotted in figure 2.5a and are in perfect agreement with the theoretical formula $\tilde{S}_D$. It can therefore be infered that the communication time between processors is negligible in our case and the obtained results are optimal (regarding the fixed convergence thresholds $\eta = 10^{-3}$, $\varepsilon_{outer} = 10^{-5}$ and $\varepsilon_{inner} = 10^{-1}$). Another interesting element to note is that one gets better speed-ups with a coarse diffusion propagator. This result seems reasonable because diffusion propagations are faster than transport ones.

We also observe that for a reduced number of processors, the speed-up increases linearly until it reaches a plateau for more than 21 processors. This is due to the fact that, for large values of N, the size of the sub-intervals $\Delta t = T_{n+1} - T_n$ decreases. As a result, the size of the problem addressed by each processor decreases and we reach a point in which the addition of more processors does not improve any longer the performances.



(a) Strong scaling results (test case A). **(b)** Weak scaling results (test case B).

Figure 2.5: Parareal scaling results.

Weak scaling results For this alternative evaluation of the scaling, we will focus on the test case B. We now consider the case in which the time step of the coarse solver Δt is fixed to $60\delta t$ and the transient has a variable length $T(N) = N\Delta t$ (i.e. the size of the problem linearly increases with the number N of processors). As an example, for $N = 14$, transient B will be solved in the time interval $[0, 70\ s.]$, whereas when $N = 42$ the time interval will be $[0; 210\ s.]$.

The measured speed-ups are plotted in figure 2.5b and, like in the strong scaling case, they are in perfect agreement with the theoretical formula $\tilde{S}_D$. The most important result here is that the distributed algorithm can effectively speed-up long time calculations: the global trend for the speed-up is to increase linearly with the number of processors. The discontinuity in the trend observed between a number of processors $N = 41$ and 42 comes from the fact that, due to the increasing size of the interval $[0, T(N)]$, the number of parareal iterations k^* raises from 3 to 4 at this stage.

2.7.4 A parareal in space and energy algorithm?

In this part, we will discuss about the possibility to use the parareal algorithm to parallelize the space and energy variables.

The method was originally suggested for the time variable but it is quite straightforward to realize that the variable t of equation 2.24 is "dummy" in the sense that it could also represent a spatial variable: parareal provides also a method to parallelize 1D advection equations. The extension to 3D spatial advective problems like the current one (see equation 2.12) seems therefore theoretically possible: for each angular unknown flux $\psi(\boldsymbol{r}, \boldsymbol{\omega}_d)$, the spatial mesh could be divided in a manner that is coherent with the direction of propagation $\boldsymbol{\omega}_d$. Each part of the mesh could be assigned to a different processor that would perform the fine propagation (i.e. the transport propagation of $\psi(\boldsymbol{r}, \boldsymbol{\omega}_d)$). The coarse solver could consist in a diffusion approximation of the original equation 2.1. This idea is, however, not the first attempt to parallelize hyperbolic spatial problems. There exists indeed several references on this topic and we refer to, e.g., [50] [54] for interesting developments on this issue.

If we now observe the multigroup problem (equation 2.4) or the outline of the resolution of a transient in algorithm 2.1, it can be seen that, for a given time step, the energy groups are solved through a loop that could in turn be also parallelized by the parareal in time algorithm: the coarse solver would propagate a reduced number of energy groups while the fine solver would propagate the problem for all the energy groups.

Conclusion

The developments presented in this paper have shown on a first stage how the MINARET solver has been extended to address time dependent problems. Such computations usually involve extremely large numbers of unknowns and acceleration techniques are required in order to run the calculations in a reasonable time. To address this issue, several sequential and parallel acceleration methods have been explored:

The two sequential accelerations included in MINARET are classical (the Chebyshev extrapolation and the diffusion synthetic acceleration) but it has been shown by a concrete example that they are essential in making the outer and inner iterative schemes converge in a reasonable time (the computing times are reduced by a factor of about 100 from the initial one). It has further been noted that one can still reduce the computing time by the use of a multilevel scheme that involves diffusion propagations and an exponential extrapolation formula.

Regarding parallel accelerations, it has first been explained how the parallelization of the angular directions can efficiently speed-up calculations. Its excellent scalability for a reduced number N of processors is degraded as N grows. This is due to the sequential computation of the DSA: since its computing time remains constant with N, its contribution to the global computing time becomes more and more significant as N raises because the time to perform the loop on the angular directions decreases with N. This problem could be solved by parallelizing the DSA by domain decomposition techniques.

Provided that we have enough processors at our disposal, the parallelization in the angular directions could be coupled with the parallelization of the time variable by the parareal in time algorithm. The efficiency of this method is much lower than the performances provided by other parallelization techniques, but this is due to the difficult task of parallelizing a variable that is sequential by nature. It has nevertheless been illustrated that the method can provide additional speed-ups for the computation of –long time– neutron transport transients. Two types of coarse solvers have been explored: one that does not degrade the original S_n transport model and another that uses the diffusion approximation. The results obtained with the diffusion coarse solver are slightly higher. This is due to the fact that diffusion propagations are performed much faster than the transport ones and because the number of required parareal iterations is not degraded in comparison with the other case.

A loss in the performances of the parareal algorithm has been detected because it has been coupled with a generalized Gauss-Seidel iterative techniques in the propagation of each time step. In the same spirit as the works of Maday and Turinici in [86] or Minion in [87], [43] where the parareal in time algorithm has already been coupled with spatial domain decomposition and spectral deferred corrections, a way to improve the present results could consist in enhancing the coupling between the parareal in time algorithm and the outer iterations of the multigroup problem 2.4.

Acknowledgements

The authors would like to thank A.M. Baudron for fruitful discussions.

This work was supported in part by the joint research program MANON between CEA-Saclay and University Pierre et Marie Curie-Paris 6. The authors are also indebted to AREVA-NP and EDF for their financial support.

Chapter 3

A coupled parareal reduced basis scheme

This is an ongoing work with Y. Maday and K. Riahi.

3.1 Introduction

The parareal in time algorithm allows to use parallelism in the time direction over an interval $[0, T]$. Let us consider an evolution problem that reads: Find a time dependent function $\underline{\mathrm{u}}(t) \in \mathbb{R}^{\mathcal{N}}$ solution to the following problem

$$\begin{cases} \mathcal{M}\frac{d\underline{\mathrm{u}}}{dt} + \mathcal{A}\underline{\mathrm{u}} = \underline{\mathrm{f}} \\ \underline{\mathrm{u}}(0) = \underline{\mathrm{u}}_0 \end{cases} \tag{3.1}$$

where $\underline{\mathrm{f}}$ is a given time dependent vector. This problem may come from the spatial discretization of a parabolic linear problem: $\mathcal{M}$ is then the mass matrix and $\mathcal{A}$ the stiffness matrix.

Any discretization in time of (3.1), based on a given time step provides a discrete propagator that allows to transport any given "initial" condition at time t to the associated discrete solution to this differential equation on a time range of size τ, for any given τ. If the time step is small enough the approximation will be accurate. We denote $\mathcal{F}_\tau^t$ such a fine flow. If the time step is larger, this provides a less accurate discrete flow that is generally less expensive to implement; let us denote by $\mathcal{G}_\tau^t$ such a coarse flow. Let be given a decomposition of the solution time interval $[0, T]$ into $\underline{\mathrm{N}}$ time intervals $[T_N, T_{N+1}]$, $N = 0, \ldots, \underline{\mathrm{N}} - 1$ of — say — uniform size : $\Delta T = T_{N+1} - T_N$. The parareal in time algorithm to solve (3.1) — in its plain version — is a predictor-corrector method that proposes a series of approximated solutions $(U_k^N)_k$ to $\underline{\mathrm{u}}(T_N)$ that converge as k goes to infinity to its fine approximation given by $\mathcal{F}$. The algorithm reads

$$U_{k+1}^{N+1} = \mathcal{G}_{\Delta T}^{T_N}(U_{k+1}^N) + \mathcal{F}_{\Delta T}^{T_N}(U_k^N) - \mathcal{G}_{\Delta T}^{T_N}(U_k^N) \tag{3.2}$$

where $\mathcal{G}_{\Delta T}^{T_N}(U_k^N)$ proposes the approximate propagation with the coarse solver over a time range of size ΔT from the initial value U_k^N and similarly $\mathcal{F}_{\Delta T}^{T_N}(U_k^N)$ is the associated propagation with the fine solver.

In the ideal case where:

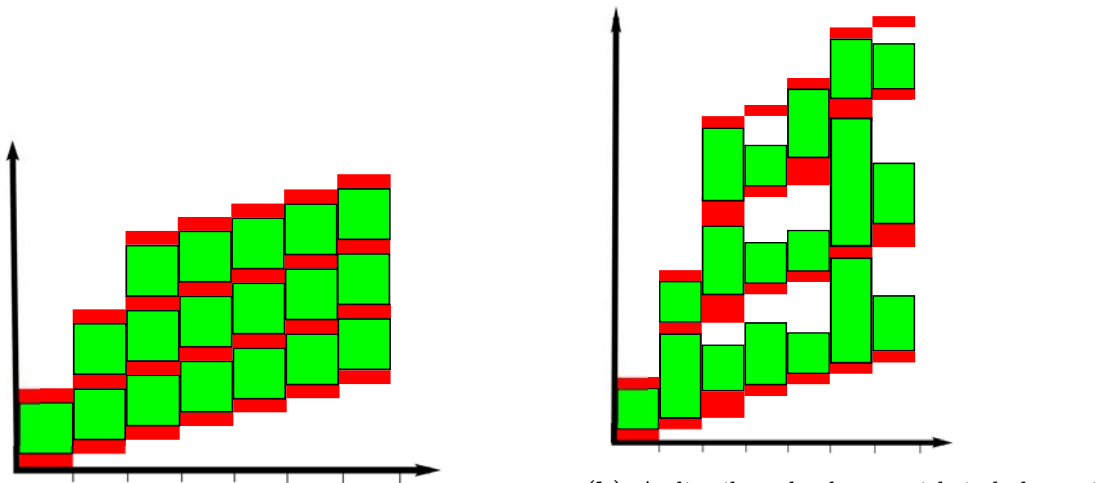
- the work to perform the fine simulation over a time window of size τ — thus between some time t and time $t + \tau$ — is independent of t and scales linearly in τ,
- the coarse solver is almost free

then the time to perform one parareal iteration (devoting each interval $[T_N, T_{N+1}], 1 \leq N \leq \underline{N}$, to a processor) is equal to the time to perform the sequential fine solution divided by the number of processors. Since at least 2 iterations need to be performed (and generally 4 to 6) in order to get convergence, the speed up can never be close to optimal. As an example, we have seen in chapter 2 that, when applied to the problem of time-dependent neutron transport, the plain parareal in time algorithm can accelerate by about a factor of five the sequential computation of a transient with 40 processors.

In addition to this "intrinsic" lack of optimality, the example of neutron transport also shows that, in general, the cost of the propagations of the fine solver (and also of the coarse one) can vary from one time t to another time t' and this leads to a loss in the performances that parareal can theoretically provide. In neutron transport, this is due to the fact that the resolution of each time step is performed by iterative techniques (notice, by the way, that this situation is very general and can occur in the resolution of other PDE's). As a result, for a given fixed convergence criterion in the internal iterations, the cost of the fine propagations can dramatically vary from one processor to another depending on the numerical complexity that takes place in each time slice $[T_n; T_{n+1}]$ to which processor P_n is assigned. Figure 3.1 illustrates this situation and compares it with the standard case existing in the literature where this imbalance is not taken into account.

Note that all this is in opposition to what occurs in the (spatial) domain decomposition procedure since, in this case, most of the codes have a complexity that is super linear in the subdomain's size. Hence splitting a domain in, e.g., 4 subdomains, diminishes the complexity of the subdomain resolution by a factor larger than 4 (and this is why, even if implemented on a serial machine, the domain decomposition method, may be interesting in global complexity and can be considered as a serial iterative scheme).

In order to match this feature in the parareal context, it has been proposed in [86], [75], [87] to diminish the cost of the fine solver, and take benefit of the iterative process in order to improve the realization of the fine solver. In [86] for example (see also [56]) it has been proposed to use a domain decomposition algorithm to compute the fine solver and to limit the number of (domain decomposition) iterations during each (parareal) iterations and to resume the iterations by using the previous state as an initial guess in the further domain decomposition iterations. This idea can actually be extended to any type of other iterative procedures like optimal control [86], high order time stepping [87] or any linear or nonlinear fixed point procedure.



(a) A balanced distributed scheme.

(b) A distributed scheme with imbalance in the tasks.

Figure 3.1: Illustration of the imbalance in the tasks observed in MINARET.

In this chapter, we adapt this strategy in building a scheme in which the internal iterations are truncated and the convergence is obtained across the parareal iterations. After a convergence analysis of the proposed parareal scheme, we will present some numerical results together with strategies that allow to diminish the complexity of the implementation in storage. The study will be carried out in a slightly simplified framework: instead of addressing the case of neutron transport, some numerical results will be presented in the case of the diffusion approximation. In order to limit the storage that this procedure involves, we propose to use a greedy reduced basis approach to largely diminish the memory requirement.

3.2 Convergence analysis of the parareal scheme with truncated internal iterations

The so called "*fine*" time discretization to solve problem (3.1) will, for example, be an Euler backward method that involves a time step δt with $\underline{\underline{n}}\delta t = \Delta T$ and $\underline{\underline{n}}\delta t = T$ (and $\underline{\underline{N}}\ \underline{\underline{n}} = \underline{\underline{n}}$). The approximations of $\underline{u}(t^n)$ for $t^n = n\delta t$ will be denoted as $\underline{u}^n$. This yields: given $\underline{u}^n \in \mathbb{R}^{\mathcal{N}}$, find $\underline{u}^{n+1} \in \mathbb{R}^{\mathcal{N}}$ such that

$$\begin{cases} A\underline{u}^{n+1} = B\underline{u}^n + \underline{f}^n, \quad n \in \{0, 1, \dots, \underline{\underline{n}} - 1\}, \\ \underline{u}^0 = \underline{u}_0 \end{cases} \tag{3.3}$$

where $\underline{f}^n \in \mathbb{R}^{\mathcal{N}}$ is a given right hand side (an approximation of $\underline{f}$ at time t^n). As is standard in the parareal literature , we denote by $\mathcal{F}$ the associated discrete propagator.

In (3.3), $A = \frac{\mathcal{M}}{\delta t} + \mathcal{A}$ and $B = \frac{\mathcal{M}}{\delta t}$. The solution procedure to get $(\underline{u}^n)_n$ from (3.3) involves a Jacobi or a Gauss Seidel algorithm that leads to

$$\forall n, 0 \le n \le \underline{\underline{n}} - 1, \quad \forall j, 1 \le j \le J^* - 1, \quad D\underline{u}^{n+1,j} = (D - A)\underline{u}^{n+1,j-1} + B\underline{u}^{n,J^*} + \underline{f}^n \tag{3.4}$$

starting from $\underline{u}^{n+1,0} = \underline{u}^{n,J^*}$, with a maximum of J^* iterations, sufficiently large to guarantee convergence and $\underline{u}^{0,J^*} = \underline{u}_0$.

In the above algorithm, D may be $D = \mathrm{diag}[A] = \mathrm{diag}[\frac{\mathcal{M}}{\delta t} + \mathcal{A}]$ or $D = \frac{\mathcal{M}}{\delta t} + \mathrm{diag}[\mathcal{A}]$ or even $D = \frac{\mathcal{M}}{\delta t}$ in the Jacobi case. Equation (3.4) is equivalent to

$$\underline{u}^{n+1,j} = (Id - D^{-1}A)\underline{u}^{n+1,j-1} + D^{-1}B\underline{u}^{n,J^*} + D^{-1}\underline{f}^n \tag{3.5}$$

and the convergence of the iterative scheme is obtained assuming that the norm of the matrix $Id - D^{-1}A$ verifies $\rho \equiv \|(Id - D^{-1}A)\| < 1$.

Let us now combine this Jacobi (or Gauss Seidel) iterative procedure with the parareal algorithm (3.2) used with the times $T_N = N\underline{\underline{n}}\delta t$. As detailed above, the fine solver involves some internal iterations based on Jacobi (or Gauss Seidel). If J^* iterations are used, the fine solver is exact and (3.2) is fully implemented. In order to save time, the proposed alternative is to perform only few iterations J with J smaller than (the recommended) J^*. This yields a non converged version $\widetilde{\mathcal{F}}_J$ of $\mathcal{F}$, and we are going to analyze the hypothesis under which the scheme

$$U^{N+1}_{k+1,J} = \mathcal{G}^{T_N}_{\Delta T}(U^N_{k+1,J}) + \widetilde{\mathcal{F}}^{T_N}_{J,\Delta T}(U^N_{k,J}) - \mathcal{G}^{T_N}_{\Delta T}(U^N_{k,J}) \tag{3.6}$$

converges similarly to (3.2) as is explained in the next theorem.

For this, let us first explain more in detail the solution procedure over each interval $[T_N, T_{N+1}]$ that allows to define what we have denoted as $\widetilde{\mathcal{F}}_J$. The following notations will be important to easily switch from the global framework over $[0, T]$ to the local one over each $[T_N, T_{N+1}]$: we denote by $t^N_n \equiv t_N + n\delta t$, $n = 0, \dots, \underline{n}$ the local time steps, where $t^N_0 = T_N$ and $t^N_{\underline{n}} = T_{N+1}$.

The approximated fine solution then consists in solving (3.3) for each fine time step t_n^N with initial condition $U_{k,J}^N$, i.e. $\underline{u}_k^{0,J} = U_{k,J}^N$

$$D\underline{u}_k^{n+1,j} = (D-A)\underline{u}_k^{n+1,j-1} + B\underline{u}_k^{n,J} + \underline{f}^n, \quad j = 1,\dots,J, \quad n = 0,\dots,\underline{n}-1 \tag{3.7}$$

with a Jacobi initialization (denoted below as "*first case*")

$$\underline{u}_k^{n+1,0} = \underline{u}_{k-1}^{n+1,J}, \tag{3.8}$$

or (denoted below as "*second case*")

$$\underline{u}_k^{n+1,0} = \underline{u}_{k-1}^{n+1,J} + \underline{u}_k^{n,J} - \underline{u}_{k-1}^{n,J}. \tag{3.9}$$

The approximation of the solution at time T_{N+1} is provided by the solution $\underline{u}_k^{\underline{n},J}$ to (3.7) and we set $\widetilde{\mathcal{F}}_{J,\Delta T}^{T_N}(U_{k,J}^N) = \underline{u}_k^{\underline{n},J}$.

Let us now turn to the convergence analysis of this new parareal scheme (3.6). Theorem 3.2.1 will show that, under reasonable hypothesis, the error $\mathcal{E}_{k,J}^N = \|U_{k,J}^N - U^N\|$ between the parareal solution $U_{k,J}^N$ and the sequential fine solution $U^N = \underline{u}^{N,\underline{n}}$ tends to zero for all N as the parareal iterations k tend to infinity.

Theorem 3.2.1. *Assume that we have the following classical stability hypothesis on $\mathcal{F}_\tau$, $\mathcal{G}_\tau$ and ε accuracy over $\delta\mathcal{G}_\tau := \mathcal{F}_\tau - \mathcal{G}_\tau$:*

$$|\mathcal{F}_\tau(t,x) - \mathcal{F}_\tau(t,y)| \le (1+C\tau)|x-y|, \tag{3.10}$$
$$|\mathcal{G}_\tau(t,x) - \mathcal{G}_\tau(t,y)| \le (1+C\tau)|x-y|, \tag{3.11}$$
$$|\delta\mathcal{G}_\tau(t,x) - \delta\mathcal{G}_\tau(t,y)| \le C\tau\varepsilon|x-y| \tag{3.12}$$

Assume in addition that $\|Id - A^{-1}B\| \le C\delta t$ and that $\rho^J \le C\delta t\varepsilon^2$. Then, there exists a constant $C > 0$ such that

$$\max_N \mathcal{E}_{k,J}^N \le C\varepsilon^k.$$

Remark 3.2.2. *The hypothesis $\|Id - A^{-1}B\| \le C\delta t$ made in theorem 3.2.1 is classical in the numerical analysis of the fine solver, it allows to prove that the fine propagator is δt accurate. In addition note that it implies that*

$$\|A^{-1}B\| \le 1 + C\delta t \tag{3.13}$$

that will be used in the sequel and can actually be proven directly for instance if (3.1) is a system of differential equations and the constant C then depends on some norms of $\mathcal{M}$ and $\mathcal{A}$. Property (3.13) can actually also be improved in the case where the system comes from the spatial discretization of a partial differential equation like the heat equation where $\mathcal{A}$ is symmetric positive definite since in this case $A^{-1}B$ is a contraction, i.e.,

$$\|A^{-1}B\| < 1. \tag{3.14}$$

Proof. From (3.6), we derive that

$$\begin{aligned} U_{k+1,J}^{N+1} - U^{N+1} &= \mathcal{G}_{\Delta T}^{T_N}(U_{k+1,J}^N) - \mathcal{G}_{\Delta T}^{T_N}(U^N) - \mathcal{G}_{\Delta T}^{T_N}(U_{k,J}^N) + \mathcal{G}_{\Delta T}^{T_N}(U^N) \\ &\quad + \widetilde{\mathcal{F}}_{J,\Delta T}^{T_N}(U_{k,J}^N) - \mathcal{F}_{\Delta T}^{T_N}(U^N) \\ &= \mathcal{G}_{\Delta T}^{T_N}(U_{k+1,J}^N) - \mathcal{G}_{\Delta T}^{T_N}(U^N) - [\mathcal{G}_{\Delta T}^{T_N} - \mathcal{F}_{\Delta T}^{T_N}](U_{k,J}^N) + [\mathcal{G}_{\Delta T}^{T_N} - \mathcal{F}_{\Delta T}^{T_N}](U^N) \\ &\quad + \widetilde{\mathcal{F}}_{J,\Delta T}^{T_N}(U_{k,J}^N) - \mathcal{F}_{\Delta T}^{T_N}(U_{k,J}^N) \end{aligned} \tag{3.15}$$

By taking norms in (3.15) and using (3.11) and (3.12), we derive

$$\mathcal{E}_{k+1,J}^{N+1} \leq [1 + C\Delta T]\mathcal{E}_{k+1,J}^{N} + C\varepsilon\Delta T\mathcal{E}_{k,J}^{N} + \|\widetilde{\mathcal{F}}_{\Delta T}(U_{k,J}^{N}) - \mathcal{F}_{\Delta T}(U_{k,J}^{N})\|. \tag{3.16}$$

Let us now examine the new term $\|\widetilde{\mathcal{F}}_{J,\Delta T}^{T_N}(U_{k,J}^{N}) - \mathcal{F}_{\Delta T}^{T_N}(U_{k,J}^{N})\|$ resulting from the incomplete iteration procedure. Note that $\mathcal{F}_{\Delta T}^{T_N}(U_{k,J}^{N})$ is the solution $\underline{\mathrm{u}}_k^{\mathrm{n}}$ to the following exact fine solver given by

$$A\underline{\mathrm{u}}_k^{n+1} = B\underline{\mathrm{u}}_k^{n} + \underline{\mathrm{f}}^{n}, \tag{3.17}$$

with the same initial condition $\underline{\mathrm{u}}_k^0 = U_{k,J}^N$ as $\widetilde{\mathcal{F}}_{\Delta T}^{T_N}(U_{k,J}^{N})$. Let us now introduce $\underline{\tilde{\mathrm{u}}}_k^{n+1}$ given by

$$A\underline{\tilde{\mathrm{u}}}_k^{n+1} = B\underline{\mathrm{u}}_k^{n,J} + \underline{\mathrm{f}}^{n}. \tag{3.18}$$

By subtracting (3.17) and (3.18) and setting $e_k^{n,j} = \underline{\mathrm{u}}_k^{n,j} - \underline{\mathrm{u}}_k^{n}$, we get

$$\underline{\tilde{\mathrm{u}}}_k^{n+1} - \underline{\mathrm{u}}_k^{n+1} = A^{-1}Be_k^{n,J}. \tag{3.19}$$

By taking norms in the last relation and using the asumption $\|A^{-1}B\| \leq 1 + C\delta t$, we obtain that

$$\|\underline{\tilde{\mathrm{u}}}_k^{n+1} - \underline{\mathrm{u}}_k^{n+1}\| \leq (1 + C\delta t)\,\|e^{n,J}\|. \tag{3.20}$$

If we now set, $\tilde{e}_k^{n+1,j} = \underline{\mathrm{u}}_k^{n+1,j} - \underline{\tilde{\mathrm{u}}}_k^{n+1}$, we can derive from (3.7) and (3.18) that

$$D\tilde{e}_k^{n+1,j} = (D - A)\,\tilde{e}_k^{n+1,j-1}, \tag{3.21}$$

which, by a bootstrap argument, produces

$$\|\tilde{e}_k^{n+1,J}\| \leq \rho^J \|\tilde{e}_k^{n+1,0}\|. \tag{3.22}$$

We can thus write

$$\begin{aligned}
\|e_k^{n+1,J}\| &= \|\underline{\mathrm{u}}_k^{n+1,J} - \underline{\tilde{\mathrm{u}}}_k^{n+1} + \underline{\tilde{\mathrm{u}}}_k^{n+1} - \underline{\mathrm{u}}_k^{n+1}\| \\
&\leq \|\tilde{e}_k^{n+1,J}\| + \|\underline{\tilde{\mathrm{u}}}_k^{n+1} - \underline{\mathrm{u}}_k^{n+1}\| \\
&\leq \rho^J \|\tilde{e}_k^{n+1,0}\| + (1 + C\delta t)\,\|e^{n,J}\|,
\end{aligned} \tag{3.23}$$

where we have first used the triangle inequality and then relations (3.20) and (3.22). Given that $\|\tilde{e}_k^{n+1,0}\| \leq \|e_k^{n+1,0}\| + \|\underline{\tilde{\mathrm{u}}}_k^{n+1} - \underline{\mathrm{u}}_k^{n+1}\| \leq \|e_k^{n+1,0}\| + (1 + C\delta t)\|e_k^{n,J}\|$, we derive

$$\|e_k^{n+1,J}\| \leq \rho^J \|e_k^{n+1,0}\| + (1 + C\delta t)\,(1 + \rho^J)\|e^{n,J}\|. \tag{3.24}$$

If we focus on the first case, using (3.8), we can write that

$$\begin{aligned}
\|e_k^{n+1,0}\| &= \|\underline{\mathrm{u}}_k^{n+1,0} - \underline{\mathrm{u}}_k^{n+1}\| \\
&\leq \|e_{k-1}^{n+1,J}\| + \|\underline{\mathrm{u}}_{k-1}^{n+1} - \underline{\mathrm{u}}_k^{n+1}\|,
\end{aligned} \tag{3.25}$$

and we can bound $\|\underline{\mathrm{u}}_{k-1}^{n+1} - \underline{\mathrm{u}}_k^{n+1}\|$ as

$$\begin{aligned}
\|\underline{\mathrm{u}}_{k-1}^{n} - \underline{\mathrm{u}}_k^{n}\| &= \|\mathcal{F}_{n\delta t}^{t_n^N}\left(U_{k-1,J}^N\right) - \mathcal{F}_{n\delta t}^{t_n^N}\left(U_{k,J}^N\right)\| \\
&\leq (1 + Cn\delta t)\,\|U_{k-1,J}^N - U_{k,J}^N\| \\
&\leq (1 + Cn\delta t)\,[\|U_{k-1,J}^N - U^N\| + \|U_{k,J}^N - U^N\|] \\
&\leq (1 + C\Delta T)\,[\mathcal{E}_{k,J}^N + \mathcal{E}_{k-1,J}^N],
\end{aligned} \tag{3.26}$$

where we have used property (3.10) to derive the first inequality. If we now set $\theta = (1+C\delta t)(1+\rho^J)$ and insert inequalities (3.25) and (3.26) in (3.23), we derive

$$\|e_k^{n+1,J}\| \leq \rho^J \|e_{k-1}^{n+1,J}\| + \rho^J(1+C\Delta T)[\mathcal{E}_{k,J}^N + \mathcal{E}_{k-1,J}^N] + \theta\|e_k^{n,J}\|. \tag{3.27}$$

By another bootstrapping argument over n in $e_k^{n,J}$, we get

$$\|e_k^{n+1,J}\| \leq \frac{1-\theta^n}{1-\theta}\rho^J \left(\max_{m,0\leq m\leq n+1} \|e_{k-1}^{m,J}\| + (1+C\Delta T)[\mathcal{E}_{k,J}^N + \mathcal{E}_{k-1,J}^N] \right).$$

Since $\theta > 1$, the term $\dfrac{1-\theta^n}{1-\theta}$ can be bounded by $n\theta^n$. Moreover, δt and ρ^J are small quantities so we can bound $n\theta^n$ by $Cn(1+n\delta t)(1+n\rho^J)$ with a moderate constant C. Hence,

$$\|e_k^{n+1,J}\| \leq Cn(1+n\delta t)(1+n\rho^J)\rho^J \left(\max_{m,0\leq m\leq n+1} \|e_{k-1}^{m,J}\| + (1+C\Delta T)[\mathcal{E}_{k,J}^N + \mathcal{E}_{k-1,J}^N] \right). \tag{3.28}$$

By a bootstrapping argument over k on inequality (3.28), we infer that

$$\begin{aligned} \max_{m,0\leq m\leq \underline{n}} \|e_k^{m,J}\| \quad \leq \quad & \left(C\underline{n}(1+\Delta T)(1+\underline{n}\rho^J)\rho^J\right)^k \max_{m,0\leq m\leq \underline{n}} \|e_0^{m,J}\| \\ & + (1+C\Delta T)\sum_{\ell=1}^{k} [C\underline{n}(1+\Delta T)(1+\underline{n}\rho^J)\rho^J]^\ell [\mathcal{E}_{k-\ell,J}^N + \mathcal{E}_{k-\ell+1,J}^N]. \end{aligned}$$

From the hypothesis made on ρ, we have $1+\underline{n}\rho^J \leq 1 + C\underline{n}\delta t\varepsilon^2 = 1 + C\Delta T\varepsilon^2$. Also, $(1+\Delta T)^k = \mathcal{O}(1+k\Delta T)$ and $\max_{m,0\leq m\leq \underline{n}} \|e_0^{m,J}\| = \mathcal{O}(\Delta T)$ so we can write the last inequality in the form

$$\max_{m,0\leq m\leq \underline{n}} \|e_k^{m,J}\| \leq C\Delta T[\underline{n}\rho^J]^k + c\sum_{\ell=1}^{k} [\underline{n}\rho^J]^\ell [\mathcal{E}_{k-\ell,J}^N + \mathcal{E}_{k-\ell+1,J}^N], \tag{3.29}$$

We infer from inequality (3.29) that

$$\|\widetilde{\mathcal{F}}_{J,\Delta T}^{T_N}(U_{k,J}^N) - \mathcal{F}_{\Delta T}^{T_N}(U_{k,J}^N)\| \leq C\Delta T[\underline{n}\rho^J]^k + c\sum_{\ell=1}^{k} [\underline{n}\rho^J]^\ell [\mathcal{E}_{k-\ell,J}^N + \mathcal{E}_{k-\ell+1,J}^N] \tag{3.30}$$

and therefore, using (3.30) in (3.16), we obtain

$$\begin{aligned} \mathcal{E}_{k+1,J}^{N+1} \quad \leq \quad & [1+C\Delta T]\mathcal{E}_{k+1,J}^N + C\varepsilon\Delta T\mathcal{E}_{k,J}^N \\ & + C\Delta T[\underline{n}\rho^J]^k + c\sum_{\ell=1}^{k} [\underline{n}\rho^J]^\ell [\mathcal{E}_{k-\ell,J}^N + \mathcal{E}_{k-\ell+1,J}^N]. \end{aligned} \tag{3.31}$$

We are now going to prove by induction over k that

$$\forall k \geq 1, \quad \max_N \mathcal{E}_{k,J}^N \leq C\varepsilon^k. \tag{3.32}$$

Since the statement is valid for $k = 1$ because the accuracy between the fine and the coarse solvers is given by ε, we only need to verify the induction step. If we use the induction hypothesis and also the hypothesis made over ρ^J, we can easily infer from (3.31) that

$$\mathcal{E}_{k+1,J}^{N+1} \leq [1+C\Delta T]\mathcal{E}_{k+1,J}^N + C\Delta T\varepsilon^{k+1}.$$

A bootstrap over N gives

$$\mathcal{E}_{k+1,J}^{N+1} \leq C\varepsilon^{k+1}, \tag{3.33}$$

for $0 \leq N \leq \underline{N} - 1$. □

Remark 3.2.3. *From the details of the previous proof, it can be expected that the second type of initialisation of the internal iterations (3.9) presents a faster convergence in comparison with the first case (3.8). Indeed, we expect (verified by numerical simulations) that a term ΔT in front of the the term $\sum_{\ell=1}^{k}[\underline{n}\rho^J]^\ell[\mathcal{E}^N_{k-\ell,J}+\mathcal{E}^N_{k-\ell+1,J}]$ appears in the expression of $\mathcal{E}^{N+1}_{k+1,J}$ given in formula (3.26) to provide a bound that looks like*

$$\|\underline{u}^{n,J}_{k-1}-\underline{u}^{n,J}_{k}+U^N_{k,J}-U^N_{k-1,J}\|\leq C\Delta T[\mathcal{E}^N_{k,J}+\mathcal{E}^N_{k-1,J}], \tag{3.34}$$

which is ΔT better than what arises in the first case (see (3.31)) and allows to get the same convergence for the parareal iterations under a less stringent condition over ρ^J. This observation will be illustrated in the numerical example of the following section.

3.3 An application to the kinetic neutron diffusion equation

3.3.1 The model

As a numerical example, we consider the resolution of the time dependent multigroup neutron diffusion equation in a reactor core $\mathcal{R}$ over the time interval $[0,T]$ and with vacuum boundary conditions. This equation has been introduced in chapter 1 section 1.4.1. We recall here that the problem in its continuous form consists in finding for all $(t,\boldsymbol{r})\in[0,T]\times\mathcal{R}$ the set of multigroup fluxes $\boldsymbol{\phi(t,r)}=\left(\phi^1(t,\boldsymbol{r}),\dots,\phi^G(t,\boldsymbol{r})\right)^T$ and the set of precursors' concentrations $\boldsymbol{C(t,r)}=(C_1(t,\boldsymbol{r}),\dots,C_L(t,\boldsymbol{r}))^T$ that are the solution of:

$$\begin{cases}
\dfrac{1}{V^g}\partial_t\phi^g(t,\boldsymbol{r})-\nabla.\left[D^g(t,\boldsymbol{r})\nabla\phi^g(t,\boldsymbol{r})\right]+\sigma_t^g(t,\boldsymbol{r})\phi^g(t,\boldsymbol{r})\\
\qquad-\displaystyle\sum_{g'=1}^{G}\sigma_s^{g'\to g}(t,\boldsymbol{r})\phi^{g'}(t,\boldsymbol{r})\\
\qquad-\chi_p^g(t,\boldsymbol{r})\displaystyle\sum_{g'=1}^{G}\left(1-\beta^{g'}(t,\boldsymbol{r})\right)(\nu\sigma_f)^{g'}(t,\boldsymbol{r})\phi^{g'}(t,\boldsymbol{r})\\
\qquad-\displaystyle\sum_{\ell=1}^{L}\lambda_\ell\chi^g_{d,\ell}(t,\boldsymbol{r})C_\ell(t,\boldsymbol{r})=0,\ \forall g\in\{1,\dots,G\}\\
\partial_tC_\ell(t,\boldsymbol{r})=-\lambda_\ell C_\ell(t,\boldsymbol{r})\\
\qquad+\displaystyle\sum_{g'=1}^{G}\beta_\ell^{g'}(t,\boldsymbol{r})(\nu\sigma_f)^{g'}(t,\boldsymbol{r})\phi^{g'}(t,\boldsymbol{r}),\ \forall\ell\in\{1,\dots,L\}.\\
\phi^g(t,\boldsymbol{r})=0,\ \forall(t,\boldsymbol{r})\in[0,T]\times\partial\mathcal{R}.
\end{cases} \tag{3.35}$$

The initial conditions $\boldsymbol{\phi(0,r)}$ and $\boldsymbol{C(0,r)}$ are given by the resolution of the stationary diffusion equation (the analogue of the stationary transport equation described in section 1.2.1 of chapter 1). Although the notations have already been introduced in chapter 1, we recall that the coefficient V^g is the neutron velocity, D^g is the diffusion coefficient, σ_t^g is the total cross-section and $\sigma_s^{g'\to g}$ is the scattering cross-section from energy group g' to energy group g. The coefficients χ_p^g and $\chi^g_{d,\ell}$ are respectively the prompt spectrum in energy group g and the delayed spectrum of precursor ℓ in energy g. Finally, the terms λ_ℓ and β_ℓ^g are respectively the decay constant and the delayed neutron fraction of precursor ℓ.

Let $[0,T]=\bigcup\limits_{n=0}^{\underline{\underline{n}}-1}[t^n;t^{n+1}]$ be a fine division of the full time interval as described in section 3.2. For the resolution of (3.35), we define a fine solver $\mathcal{F}$ that is built by applying an Euler backward discretization. We denote by $\boldsymbol{\phi^n(r)}=\left(\phi^{1,n}(\boldsymbol{r}),\dots,\phi^{G,n}(\boldsymbol{r})\right)^T$ and $\boldsymbol{C^n(r)}=(C_1^n(\boldsymbol{r}),\dots,C_L^n(\boldsymbol{r}))^T$

the approximation of $\boldsymbol{\phi(t,r)}$ and $\boldsymbol{C(t,r)}$ at time $t = t^n$ with this fine solver, $n = 0, \dots, \underline{\underline{n}}$. At each time step, we are led to the resolution of a system that can be written in a block form:

$$\begin{gathered}\text{Given } \boldsymbol{\phi^n} \text{ and } \boldsymbol{C^n}, \text{ find } \boldsymbol{\phi^{n+1}} \text{ and } \boldsymbol{C^{n+1}} \text{ such that:} \\ \begin{pmatrix} A_{1,1} & A_{1,2} \\ A_{2,1} & A_{2,2} \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi^{n+1}} \\ \boldsymbol{C^{n+1}} \end{pmatrix} = \frac{1}{\delta t} \begin{pmatrix} \boldsymbol{\phi^n} \\ \boldsymbol{C^n} \end{pmatrix}, \quad n = 0, \dots, \underline{\underline{n}} - 1. \end{gathered} \tag{3.36}$$

The matrix $A_{1,1} \in \mathbb{R}^{G \times G}$ accounts for the coupling between multi-group fluxes. The matrix $A_{1,2}$ and $A_{2,1}$ represent the coupling between the fluxes and the precursors' concentrations. Finally, $A_{2,2}$ is diagonal since there is no coupling between precursors. System (3.36) of equations is solved with a block Jacobi numerical scheme. If $\boldsymbol{\phi^{n,j}(r)}$ and $\boldsymbol{C^{n,j}(r)}$ are the approximations of $\boldsymbol{\phi^n(r)}$ and $\boldsymbol{C^n(r)}$ at the j-th Jacobi iteration, the considered scheme reads:

$$\begin{pmatrix} A_{1,1} & 0 \\ 0 & A_{2,2} \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi^{n+1,j}} \\ \boldsymbol{C^{n+1,j}} \end{pmatrix} = - \begin{pmatrix} 0 & A_{1,2} \\ A_{2,1} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi^{n+1,j-1}} \\ \boldsymbol{C^{n+1,j-1}} \end{pmatrix} + \frac{1}{\delta t} \begin{pmatrix} \boldsymbol{\phi^{n,J^*}} \\ \boldsymbol{C^{n,J^*}} \end{pmatrix}, \; j = 1, \dots, J^*. \tag{3.37}$$

We will assume that J^* iterations are enough to reach the convergence according to some well-chosen criterion. As outlined in 3.2, if we solve sequentially the fine problem over $[0, T]$, the most classical choice in the initial guess for the Jacobi iterations is to take the solution at the previous time:

$$(\boldsymbol{\phi^{n+1,0}}, \boldsymbol{C^{n+1,0}})^T = (\boldsymbol{\phi^{n,J^*}}, \boldsymbol{C^{n,J^*}})^T.$$

However, we are interested here in the resolution of problem (3.35) with the parareal scheme that uses a reduced number $J < J^*$ of Jacobi iterations in the fine solver. In other words, we are going to numerically study the convergence of the scheme introduced in equation (3.6) and that we remind here:

$$U^{N+1}_{k+1,J} = \mathcal{G}^{T_N}_{\Delta T}(U^N_{k+1,J}) + \widetilde{\mathcal{F}}^{T_N}_{J,\Delta T}(U^N_{k,J}) - \mathcal{G}^{T_N}_{\Delta T}(U^N_{k,J})$$

where $U^N_{k,J}$ is, in our case, the parareal flux and precursors' concentrations solution at iteration k and at time T_N. With the notations of section 3.2, $\widetilde{\mathcal{F}}^{T_N}_{J,\Delta T}(U^N_{k,J})$ starts at time $T_N = t^N_0$ and reaches $T_{N+1} = t^N_{\underline{\underline{n}}}$ by performing $\underline{\underline{n}}$ propagations with a time step of δt. The solution $\widetilde{\mathcal{F}}^{T_N}_{J,\Delta T}(U^N_{k,J})$ can therefore be written in our case as:

$$\widetilde{\mathcal{F}}^{T_N}_{J,\Delta T}(U^N_{k,J}) = \begin{pmatrix} \boldsymbol{\phi^{\underline{\underline{n}}}_{k,J}} \\ \boldsymbol{C^{\underline{\underline{n}}}_{k,J}} \end{pmatrix}. \tag{3.38}$$

The intermediate states that are performed by $\widetilde{\mathcal{F}}^{T_N}_{J,\Delta T}$ to reach the state given in (3.38) are the following: when, $n = 0$, the initial condition is $U^N_{k,J}$ and $\widetilde{\mathcal{F}}^{T_N}_{J,\Delta T}$ will solve the Jacobi iterations:

$$\begin{pmatrix} A_{1,1} & 0 \\ 0 & A_{2,2} \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi^1_{k,j}} \\ \boldsymbol{C^1_{k,j}} \end{pmatrix} = - \begin{pmatrix} 0 & A_{1,2} \\ A_{2,1} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi^1_{k,j-1}} \\ \boldsymbol{C^1_{k,,j-1}} \end{pmatrix} + \frac{1}{\delta t} X^N_k, \quad j = 1, \dots, J. \tag{3.39}$$

For $n = 1, \dots, \underline{\underline{n}} - 1$, the solver $\widetilde{\mathcal{F}}$ solves at each fine time step:

$$\begin{pmatrix} A_{1,1} & 0 \\ 0 & A_{2,2} \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi^{n+1}_{k,j}} \\ \boldsymbol{C^{n+1}_{k,j}} \end{pmatrix} = - \begin{pmatrix} 0 & A_{1,2} \\ A_{2,1} & 0 \end{pmatrix} \begin{pmatrix} \boldsymbol{\phi^{n+1}_{k,j-1}} \\ \boldsymbol{C^{n+1}_{k,j-1}} \end{pmatrix} + \frac{1}{\delta t} \begin{pmatrix} \boldsymbol{\phi^n_{k,J}} \\ \boldsymbol{C^n_{k,J}} \end{pmatrix}, \quad j = 1, \dots, J. \tag{3.40}$$

The two options for the Jacobi initialization read in our case:

$$\begin{pmatrix} \boldsymbol{\phi}_{k,0}^{n+1} \\ \boldsymbol{C}_{k,0}^{n+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\phi}_{k-1,J}^{n+1} \\ \boldsymbol{C}_{k-1,J}^{n+1} \end{pmatrix} \tag{3.41}$$

and

$$\begin{pmatrix} \boldsymbol{\phi}_{k,0}^{n+1} \\ \boldsymbol{C}_{k,0}^{n+1} \end{pmatrix} = \begin{pmatrix} \boldsymbol{\phi}_{k-1,J}^{n+1} \\ \boldsymbol{C}_{k-1,J}^{n+1} \end{pmatrix} + \begin{pmatrix} \boldsymbol{\phi}_{k,J}^{n} \\ \boldsymbol{C}_{k,J}^{n} \end{pmatrix} - \begin{pmatrix} \boldsymbol{\phi}_{k-1,J}^{n} \\ \boldsymbol{C}_{k-1,J}^{n} \end{pmatrix} \tag{3.42}$$

Note that, for each interval $[T_N, T_{N+1}]$, these initializations require the knowledge of the fine states at all times t_n^N ($n = 0, \dots, \underline{\mathrm{n}}$) at the previous iteration $k-1$. From an implementation point of view, this point can easily become an important issue to address given the high memory demand that the storage of all these states can imply (recall that, in the case of neutron transport in a realistic 3D reactor core, each one of the fine solutions can involve $\mathcal{O}(10^{12})$ elements). For this reason, we have explored in our numerical application whether we can alleviate this storage by using in the Jacobi initialization surrogates

$$\pi_M \left[\begin{pmatrix} \boldsymbol{\phi}_{k-1,J}^{n} \\ \boldsymbol{C}_{k-1,J}^{n} \end{pmatrix} \right], \quad n = 0, \dots, \underline{\mathrm{n}} \tag{3.43}$$

obtained by projection of the elements

$$\begin{pmatrix} \boldsymbol{\phi}_{k-1,J}^{n} \\ \boldsymbol{C}_{k-1,J}^{n} \end{pmatrix}, \quad n = 0, \dots, \underline{\mathrm{n}}. \tag{3.44}$$

over an appropriate reduced basis space X_M of dimension M much smaller than $\underline{\mathrm{n}}$. But this raises the questions:

- Are the convergence properties of our scheme degraded with the use of surrogates in the Jacobi initialization? To what extend?
- What is the smallest dimension M that allows to have good convergence properties for our standards? And further: is this dimension compatible with our storage limitations?
- How to build the reduced basis X_M without storing all the fine elements?

Although the theory to answer to the first point is still under development and can be handled through a further approximation in $\widetilde{\mathcal{F}}$, we will provide some elements of answer in section 3.3.2 through some numerical results.

Regarding the second and third points, we are currently exploring the following idea: assume that the dimension M is fixed *a priori* by our memory limitations. If we perform a Proper Orthogonal Decomposition (POD) over the set of fine solutions given in (3.44), let X_M be the reduced basis spanned by the M eigenvectors associated to the M largest eigenvalues of the POD correlation matrix. Since the computation of X_M requires the knowledge of all the fine solutions (3.44), we propose to build a "moving-window POD reduced space", i.e. a reduced space that is updated as the index n increases. There are several possibilities to address this and we explain the idea through an example: assume that $M = 10$ and that we can store a maximum number of $M_{\max} = 20$ elements. Let us fix $\underline{\mathrm{n}} = 100$ (then $n = 0, \dots, 100$). We start by storing the first 20 fine solutions ($n = 0, \dots, 19$) and extract out of these 20 modes a first POD basis of $X_{10}^{(1)}$ of dimension 10. We now have in memory 10 elements so we continue by storing the next 10 fine solutions ($n = 20, \dots, 29$). We now perform a second POD with the 10 basis functions of $X_{10}^{(1)}$ and the 10 new fine solutions. This gives $X_{10}^{(2)}$. And we continue the process.

In the following section, we present some first numerical results related to the model described in this section.

3.3.2 Some first results

We have applied the new parareal in time scheme for the resolution of equation (3.35) in the context of the TWIGL benchmark transient (see chapter 2 section 2.4). The considered time interval is $[0, T]$ with $T = 40\ s$. The computations have been performed with a solver implemented in FreeFem++ ([58]) based on a $\mathbb{P}_1$ spatial discretization over tetrahedral meshes. Its main foundations have been established during previous studies on the application of the plain parareal scheme to the neutron diffusion equation (see [14]).

The sequential fine solver $\mathcal{F}_{\delta t}$ has been built with an Euler backward time discretization with a time step of $\delta t = 0.1\ s$. The resolution of each time step involves a Jacobi scheme as described in (3.37). The accuracy of $\mathcal{F}_{\delta t}$ has been estimated to be of order $\varepsilon_{\mathcal{F}} = \mathcal{O}(10^{-2})$ by comparing its solution with the solution computed with a solver using an ultra-fine time step that has been taken as an extremely good approximation of the exact solution.

Since $T = 40\ s$ and $\delta t = 0.1\ s$, we are doing $\underline{\underline{n}} = 400$ fine propagations in this example and, in order not to degrade the accuracy of $\mathcal{F}_{\delta t}$ along these 400 propagations, we propose to converge the internal Jacobi iterations until the error of the residual between two Jacobi iterations

$$\frac{\| \left(\boldsymbol{\phi_k^{n,j}}, \boldsymbol{C_k^{n,j}}\right)^T - \left(\boldsymbol{\phi^{n,j-1}}, \boldsymbol{C^{n,j-1}}\right)^T \|_{\ell_2(L^2(\mathcal{R}))}}{\| \left(\boldsymbol{\phi^{n,j}}, \boldsymbol{C^{n,j}}\right)^T \|_{\ell_2(L^2(\mathcal{R}))}} \tag{3.45}$$

goes below $\varepsilon_{\mathcal{F}}/\underline{\underline{n}} \approx 10^{-5}$ at every time t_n. This is achieved by taking $J^* = 4$ in the present case (note that the block diagonal Jacobi is extremely fast due to the fact that D is close to A). The coarse solver $\mathcal{G}_{\Delta T}$ is built in the same way as the fine solver $\mathcal{F}_{\delta t}$ but with a much larger time step $\Delta T = 5\ s$. Regarding the degraded fine solver $\widetilde{\mathcal{F}}_{J,\Delta T}$, it makes propagations over intervals of size ΔT by performing $\underline{n} = 50$ propagations of size δt with Jacobi internal iterations. The considered values for J of incomplete Jacobi iterations are $J \in \{1, 2, 3\}$ and the two types of Jacobi initializations (3.41) and (3.42) have been explored. We will also present some results concerning the use of a reduced basis to limit the storage.

The convergence of our scheme will be analyzed through the study of the errors of the degraded fine propagations with respect to the fine sequential resolution, i.e. we are interested in the following errors:

$$e_k^n = \frac{\| \left(\boldsymbol{\phi_k^{n,J}}, \boldsymbol{C_k^{n,J}}\right)^T - \left(\boldsymbol{\phi^{n,J^*}}, \boldsymbol{C^{n,J^*}}\right)^T \|_{\ell_2(L^2(\mathcal{R}))}}{\| \left(\boldsymbol{\phi^{n,J^*}}, \boldsymbol{C^{n,J^*}}\right)^T \|_{\ell_2(L^2(\mathcal{R}))}}, \quad n = 0, \ldots, \underline{\underline{n}}. \tag{3.46}$$

Note that the errors given in (3.46) cannot be used in practice during a calculation as a stopping criterion because the sequential fine resolution will not be carried out. An online computable estimator of the convergence would involve some residual error in the parareal solutions, like, e.g.,

$$r_k = \max_{1 \leq N \leq \underline{N}} \frac{\| \left(\boldsymbol{\phi_k^{N,J}}, \boldsymbol{C_k^{N,J}}\right)^T - \left(\boldsymbol{\phi_{k-1}^{N,J}}, \boldsymbol{C_{k-1}^{N,J}}\right)^T \|_{\ell_2(L^2(\mathcal{R}))}}{\| \left(\boldsymbol{\phi_k^{N,J}}, \boldsymbol{C_k^{N,J}}\right)^T \|_{\ell_2(L^2(\mathcal{R}))}}, \quad k \geq 2. \tag{3.47}$$

However, in this study, we are placing ourselves in an *a posteriori* validation to carry out the convergence study as accurately as possible and we will therefore analyze the convergence in the parareal iterations through the errors e_k^n. Convergence will be achieved when $\sup_{0 \leq n \leq \underline{\underline{n}}} e_k^n < \varepsilon_{tol}$, where ε_{tol} is our error tolerance. We will consider that a value $\varepsilon_{tol} = \varepsilon_{\mathcal{F}}/10 = 10^{-3}$ is a tight enough choice that preserves in our parareal solutions the original accuracy $\varepsilon_{\mathcal{F}}$ of the sequential fine solver.

As a starting point, let us consider a degraded fine solver with only $J = 1$ and with the first type of starting guess (3.41) (see figure 3.2). As it can be observed, the convergence across the parareal

iterations is extremely slow and it has still not been achieved after $k = 6$ parareal iterations. In fact, the errors stagnate for $k > 6$ and this first example illustrates that there exists a limit in the degradation of $\tilde{\mathcal{F}}_{J,\Delta T}$ with respect to $\mathcal{F}$. With this first starting guess, we actually need $J = 3$ internal iterations to converge in a reasonable number ($k = 4$) of parareal iterations (see figure 3.2). However, this case with $J = 3$ is very close to the original number $J^* = 4$ of internal iterations and we are here in a case that does not provide a significant improvement in the efficiency with respect to the traditional parareal scheme.

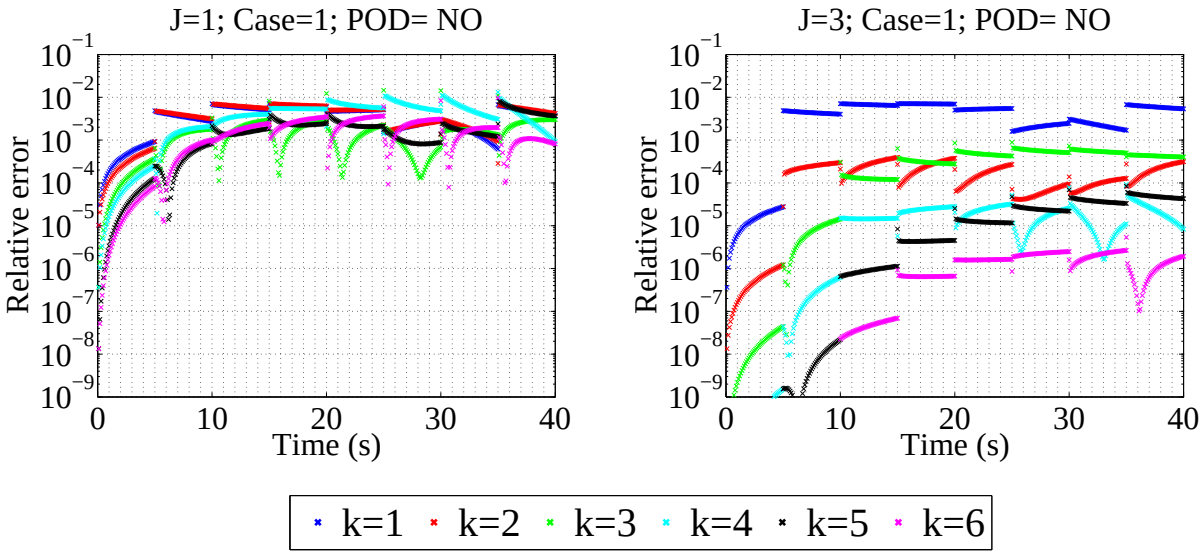


Figure 3.2: First type of Jacobi initialization with $J = 1$ and $J = 3$. The legend will also apply for the rest of the plots.

On the other hand, if we now turn to analyze the second type of initialization proposed in (3.42), the gain in efficiency is much larger as convergence is achieved after $k = 3$ parareal iterations with only $J = 1$ internal iterations (see figure 3.3, left plot). Indeed, if we neglect the cost of the coarse solver and of the communication time, an estimation of the efficiency is given by J^*/Jk, which, in this case, is above 1. This implies that, if we implemented the proposed parareal scheme in only one processor, we would be computing faster (and with the same accuracy) an approximation of the solution at every time t_n than the sequential resolution with the fine solver. We nevertheless point out at this stage that the study of the actual performances of the method should be carried out with computations of the residual (3.47) rather than with the errors (3.46) as in the present discussion. Indeed, we usually expect one additional parareal iteration to detect convergence with residuals like (3.47) and, in this case, the efficiency would be smaller than one, but close to it. Note also that, as was illustrated in the more involved MINARET solver, we generally need more iterations than $J^* = 4$.

It is now important to analyze whether the convergence properties of the scheme are degraded if the initial values of the internal iterations are replaced by using surrogates coming from a reduced basis X_M of small dimension M as explained in (3.43). Figure 3.3 (right plot) illustrates that the use of the proposed moving-window POD is feasible (if tuned in a proper way) and does not degrade the convergence properties (at least with the convergence criterion of $\varepsilon_{tol} = 10^{-3}$ that we have fixed). The plots represent calculations with the second type of Jacobi initialization and where we have used a reduced basis of dimension $M = 2$ that has been built with a size of the moving window of $M_{\max} = 5$.

Remark 3.3.1. *Note that the computations of the POD basis X_M are going to slightly degrade*

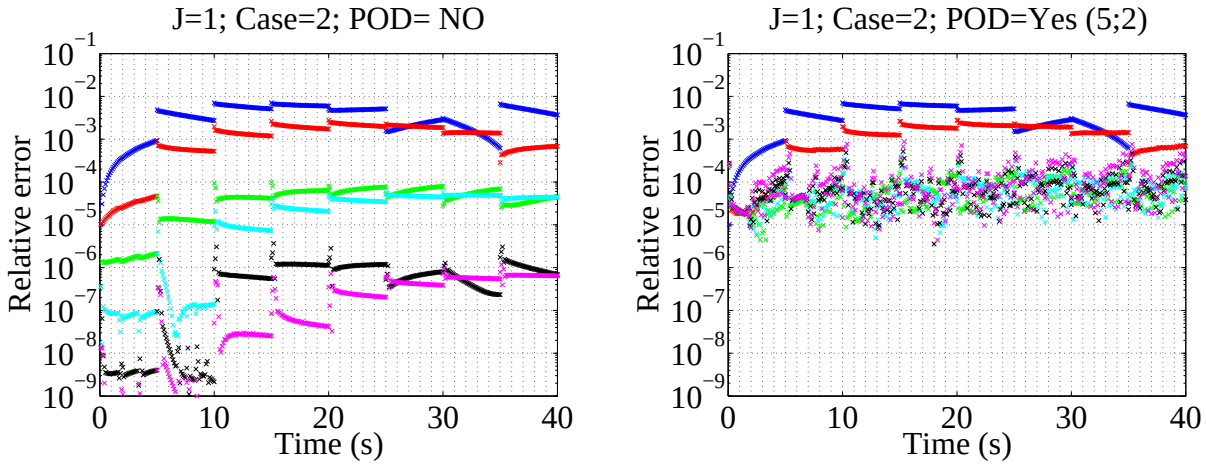


Figure 3.3: Second type of Jacobi initialization with $J = 1$.

the parallel efficiency of the scheme and a more precise study about its impact on the acceleration performances and also on how to determine their dimension M on the run should be analyzed in the future.

Conclusion

In this preliminary analysis, we have illustrated the possibility of maintaining the convergence rate of the original parareal in time algorithm while degrading the correction stage by using a non converged iterative solver $\widetilde{\mathcal{F}}$. This is possible at the price of starting the next parareal iterations in $\widetilde{\mathcal{F}}$ with a good enough initialization. However, these good initilizations require the storage of a lot of information that, for real size problems, would kill the effectiveness and put into question the feasibility of the degraded parareal scheme. We have proposed to store these informations in a compact way by using a POD basis computed on the run. The resulting scheme is, at least on the toy problem we consider, accurate and with parallel efficiency much more interesting than the plain original version. In addition, the behavior of this new scheme is supported by numerical analysis.

Part II

Numerical models for the real-time monitoring of physical processes

Chapter 4

A generalized empirical interpolation method : application of reduced basis techniques to data assimilation

This is a joint work with Y. Maday that has been published with the reference:

[76] Y. Maday and O. Mula. A generalized empirical interpolation method: application of reduced basis techniques to data assimilation. Analysis and Numerics of Partial Differential Equations, XIII:221–236, 2013.

4.1 Introduction

The representation of some physical or mechanical quantities, representing a scalar or vectorial function that depends on space, time or both, can be elaborated through at least two – possibly – complementary approaches: the first one, called explicit hereafter, is based on the measurement of some instances of the quantity of interest that consists in getting its value at some points from which, by interpolation or extrapolation, the quantity is approximated in other points than where the measurements have been performed. The second approach, called implicit hereafter, is more elaborated. It is based on a model, constructed by expertise, that implicitly characterizes the quantity as a solution to some problem fed with input data. The model can e.g. be a parameter dependent partial differential equation, the simulation of which allows to get an approximation of the quantity of interest, and, actually, many more outputs than the sole value of the quantity of interest. This second approach, when available, is more attractive since it allows to have a better understanding of the working behavior of the phenomenon that is under consideration. In turn, it facilitates optimization, control or decision making.

Nevertheless for still a large number of problems, the numerical simulation of this model is indeed possible — though far too expensive to be performed in a reasonable enough time. The combined efforts of numerical analysts, specialists of algorithms and computer scientists, together with the increase of the performances of the computers allow to increase every day the domains of application where numerical simulation can be used, to such an extent that it is possible now to rigorously adapt the approximation, degrade the models, degrade the simulation, or both in an intelligent way without sacrificing the quality of the approximation where it is required.

Among the various ways to reduce the problem's complexity stand approaches that use the smallness of the Kolmogorov n-width [66] of the manifold of all solutions considered when the parameters varies continuously in some range. This idea, combined with the Galerkin method is at the basis of the reduced basis method and the Proper Orthogonal Decomposition (POD) methods

to solve parameter dependent partial differential equations. These approximation methods allow to build the solution to the model associated to some parameter as a linear combination of some precomputed solutions associated to some well chosen parameters. The precomputations can be lengthy but are performed off-line, the online computation has a very small complexity, based on the smallness of the Kolmogorov n-width. We refer to [99] [104] for an introduction to these approaches.

Another possibility, rooted on the same idea, is the empirical interpolation method (EIM) that allows, from values of the quantity at some interpolating points, to build a linear combination of again preliminary fully determined quantities associated to few well chosen instances of the parameter. The linear combination is determined in such a way that it takes the same values at the interpolating points as the quantity we want to represent. This concept generalizes the classical – e.g. polynomial or radial basis – interpolation procedure and is recalled in the next section. The main difference is that the interpolating function are not a priori known but depend on the quantity we want to represent.

In this paper we first aim at generalizing further this EIM concept by replacing the pointwise evaluations of the quantity by more general measures, mathematically defined as linear forms defined on a superspace of the manifold of appropriate functions. We consider that this generalization, named Generalized Empirical Interpolation Method (GEIM), represents already an improvement with respect to classical interpolation reconstructions.

As a first practical application of this GEIM, we propose a coupled approach based on the domain decomposition of the computational domain into two parts : one small domain Ω_1 where the Kolmogorov n-width of the manifold is not small and where the parametrized PDE will be simulated and the other subdomain Ω_2 , much larger but with a small Kolmogorov n-width because for instance the solution is driven over Ω_2 by the behavior of the solution over Ω_1. The idea is then to first construct (an approximation of) the solution from the measurements using the GEIM. In turn this reconstruction, up to the interface between Ω_1 and Ω_2, provides the necessary boundary conditions for solving the model over Ω_1.

This is not the first attempt to use the small Kolmogorov width for another aim than the POD or reduced basis technique which are both based on a Galerkin approach. In [21] e.g. the smallness of the Kolmogorov width is used to post-process a coarse finite element approximation and get an improved accuracy.

The problems we want to address with this coupled approach, stem from, e.g., actual industrial process or operations that work on a day-to-day basis; they can be observed with experimental sensors that provide sound data and are able to characterize part of their working behavior. We think that the numerical simulation and data mining approaches for analyzing real life systems are not enough merged in order to (i) complement their strength and (ii) cope for their weaknesses. This paper is a contribution in this direction.

In the last section, we evoke the problem of uncertainty and noises in the acquisition of the data, since indeed, the data are most often polluted by noises. Due to this, statistical data acquisition methods are used to filter out the source signals so that an improved knowledge is accessible. In many cases though, and this is more and more the case now, the data are far too numerous to all be taken into account, most of them are thus neglected because people do not know how to analyze them, in particular when the measures that are recorded are not directly related to some directly understandable quantity.

4.2 Generalized Empirical Interpolation Method

The rationale of all our approach relies on the possibility to approximately represent a given set, portion of a regular manifold (here the set of solution to some PDE), as a linear combination of

very few computable elements. This is linked to the notion of n-width following Kolmogorov [66]:

Definition 4.2.1. *Let F be a subset of some Banach space $\mathcal{X}$ and Y_n be a generic n-dimensional subspace of $\mathcal{X}$. The deviation between F and Y_n is*

$$E(F;Y_n) := \sup_{x\in F} \inf_{y\in Y_n} \|x-y\|_{\mathcal{X}}.$$

The Kolmogorov n-width of F in $\mathcal{X}$ is given by

$$\begin{aligned} d_n(F,\mathcal{X}) &:= \inf\{E(F;Y_n) : Y_n \text{ a } n\text{-dimensional subspace of } X\} \\ &= \inf_{Y_n} \sup_{x\in F} \inf_{y\in Y_n} \|x-y\|_{\mathcal{X}} \ . \end{aligned} \tag{4.1}$$

The n-width of F thus measures to what extent the set F can be approximated by an n-dimensional subspace of $\mathcal{X}$.

We assume from now on that F and $\mathcal{X}$ are composed of functions defined over a domain $\Omega \subset \mathbb{R}^d$, where $d = 1, 2, 3$ and that F is a compact set of $\mathcal{X}$.

4.2.1 Recall of the Empirical Interpolation Method

We begin by describing the construction of the empirical interpolation method ([11], [55], [80]) that allows us to define simultaneously the set of generating functions recursively chosen in F together with the associated interpolation points. It is based on a greedy selection procedure as outlined in [95, 104, 118]. With $\mathcal{M}$ being some given large number, we assume that the dimension of the vectorial space spanned by F : span(F) is of dimension $\geq \mathcal{M}$.

The first generating function is $\varphi_1 = \arg\max_{\varphi\in F} \|\varphi(\cdot)\|_{L^\infty(\Omega)}$, the associated interpolation point satisfies $x_1 = \arg\max_{x\in\overline{\Omega}} |\varphi_1(x)|$, we then set $q_1 = \varphi_1(\cdot)/\varphi_1(x_1)$ and $B^1_{11} = 1$. We now construct, by induction, the nested sets of interpolation points $\Xi_M = \{x_1, \ldots, x_M\}, 1 \leq M \leq M_{\max}$, and the nested sets of basis functions $\{q_1, \ldots, q_M\}$, where $M_{\max} \leq \mathcal{M}$ is some given upper bound fixed *a priori*. For $M = 2, \ldots, M_{\max}$, we first solve the interpolation problem : Find

$$\mathcal{I}_{M-1}[\varphi(\cdot)] = \sum_{j=1}^{M-1} \alpha_{M-1,j}[\varphi] q_j \ , \tag{4.2}$$

such that

$$\mathcal{I}_{M-1}[\varphi(\cdot)](x_i) = \varphi(x_i), \quad i = 1, \ldots, M-1 \ , \tag{4.3}$$

that allows to define the $\alpha_{M-1,j}[\varphi], 1 \leq j \leq M-1$, as it can be proven indeed that the $(M-1)\times(M-1)$ matrix of running entry $q_j(x_i)$ is invertible, actually it is lower triangular with unity diagonal.

We then set

$$\forall \varphi \in F, \quad \varepsilon_{M-1}(\varphi) = \|\varphi - \mathcal{I}_{M-1}[\varphi]\|_{L^\infty(\Omega)} \ , \tag{4.4}$$

and define

$$\varphi_M = \arg\max_{\varphi\in F} \varepsilon_{M-1}(\varphi) \ , \tag{4.5}$$

and

$$x_M = \arg\max_{x\in\overline{\Omega}} |\varphi_M(x) - \mathcal{J}_{M-1}[\varphi_M](x)| \ , \tag{4.6}$$

we finally set $r_M(x) = \varphi_M(x) - \mathcal{J}_{M-1}[\varphi_M(x)]$, $q_M = r_M / r_M(x_M)$ and $B^M_{ij} = q_j(x_i), 1 \leq i, j \leq M$.

The Lagrangian functions — that can be used to build the interpolation operator $\mathcal{I}_M$ in $X_M = \text{span}\{\varphi_i, 1 \leq i \leq M\} = \text{span}\{q_i, 1 \leq i \leq M\}$ over the set of points $\Xi_M = \{x_i, 1 \leq i \leq M\}$ — verify

for any given M, $\mathcal{I}_M[u(\,\cdot\,)] = \sum_{i=1}^M u(x_i) h_i^M(\,\cdot\,)$, where $h_i^M(\,\cdot\,) = \sum_{j=1}^M q_j(\,\cdot\,)[B^M]^{-1}_{ji}$ (note indeed that $h_i^M(x_j) = \delta_{ij}$).

The error analysis of the interpolation procedure classically involves the Lebesgue constant $\Lambda_M = \sup_{x\in\Omega} \sum_{i=1}^M |h_i^M(x)|$.

Lemma 4.2.2. *For any $\varphi \in F$, the interpolation error satisfies*

$$\|\varphi - \mathcal{I}_M[\varphi]\|_{L^\infty(\Omega)} \leq (1+\Lambda_M) \inf_{\psi_M \in X_M} \|\varphi - \psi_M\|_{L^\infty(\Omega)}. \tag{4.7}$$

The last term in the right hand side of the above inequality is known as the best fit of φ by elements in X_M.

4.2.2 The generalization

Let us assume now that we do not have access to the values of $\varphi \in F$ at points in Ω easily, but, on the contrary, that we have a dictionary of linear forms $\sigma \in \Sigma$ — assumed to be continuous in some sense, e.g. in $L^2(\Omega)$ with norm 1 — the application of which over each $\varphi \in F$ is easy. Our extension consists in defining $\tilde{\varphi}_1$, $\tilde{\varphi}_2$,..., $\tilde{\varphi}_M$ and a family of associated linear forms σ_1, σ_2,..., σ_M such that the following generalized interpolation process (our GEIM) is well defined :

$$\mathcal{J}_M[\varphi] = \sum_{j=1}^M \beta_j \tilde{\varphi}_j, \text{ such that } \quad \forall i = 1,\dots,M, \quad \sigma_i(\mathcal{J}_M[\varphi]) = \sigma_i(\varphi) \tag{4.8}$$

Note that the GEIM reduces to the EIM when the dictionary is composed of dirac masses, defined in the dual space of $\mathcal{C}^0(\Omega)$.

As explained in the introduction, our generalization is motivated by the fact that, in practice, measurements provide outputs from function φ that are some averages — or some moments — of φ over the actual size of the mechanical device that takes the measurement.

Among the questions raised by GEIM:

- is there an optimal selection for the linear forms σ_i within the dictionary Σ ?
- is there a constructive optimal selection for the functions $\tilde{\varphi}_i$?
- given a set of linearly independent functions $\{\tilde{\varphi}_i\}_{i\in[1,M]}$ and a set of continuous linear forms $\{\sigma_i\}_{i\in[1,M]}$, does the interpolant (in the sense of (4.8)) exist?
- is the interpolant unique?
- how does the interpolation process compares with other approximations (in particular orthogonal projections)?
- Under what hypothesis can we expect the GEIM approximation to converge rapidly to φ?

In what follows, we provide answers to these questions either with rigorous proofs or with numerical evidences.

The construction of the generalized interpolation functions and linear forms is done recursively, following the same procedure as in the previous subsection, based on a greedy approach, both for the construction of the interpolation linear forms $\tilde{\varphi}_i$ and the associated forms selected in the dictionary Σ : The first interpolating function is, e.g.:

$$\tilde{\varphi}_1 = \arg\sup_{\varphi\in F} \|\varphi\|_{L^2(\Omega)},$$

the first interpolating linear form is:

$$\sigma_1 = \arg\sup_{\sigma\in\Sigma} |\sigma(\tilde{\varphi}_1)|.$$

We then define the first basis function as: $\tilde{q}_1 = \dfrac{\tilde{\varphi}_1}{\sigma_1(\tilde{\varphi}_1)}$. The second interpolating function is:

$$\tilde{\varphi}_2 = \arg\sup_{\varphi\in F} \|\varphi - \sigma_1(\varphi)\tilde{q}_1\|_{L^2(\Omega)}.$$

The second interpolating linear form is:

$$\sigma_2 = \arg\sup_{\sigma\in\Sigma} |\sigma(\tilde{\varphi}_2 - \sigma_1(\tilde{\varphi}_2)\tilde{q}_1)|,$$

and the second basis function is defined as:

$$\tilde{q}_2 = \frac{\tilde{\varphi}_2 - \sigma_1(\tilde{\varphi}_2)\tilde{q}_1}{\sigma_2(\tilde{\varphi}_2 - \sigma_1(\tilde{\varphi}_2)\tilde{q}_1)},$$

and we proceed by induction : assuming that we have built the set of interpolating functions $\{\tilde{q}_1, \tilde{q}_2, \ldots, \tilde{q}_{M-1}\}$ and the set of associated interpolating linear forms $\{\sigma_1, \sigma_2, \ldots, \sigma_{M-1}\}$, for $M > 2$, we first solve the interpolation problem : find $\{\widetilde{\alpha_j^{M-1}(\varphi)}\}_j$ such that

$$\forall i = 1, \ldots, M-1, \quad \sigma_i(\varphi) = \sum_{j=1}^{M-1} \widetilde{\alpha_j^{M-1}}(\varphi)\sigma_i(\tilde{q}_j),$$

and then compute:

$$\mathcal{J}_{M-1}[\varphi] = \sum_{j=1}^{M-1} \widetilde{\alpha_j^{M-1}}(\varphi)\tilde{q}_j$$

We then evaluate

$$\forall \varphi \in F, \quad \varepsilon_M(\varphi) = \|\varphi - \mathcal{J}_{M-1}[\varphi]\|_{L^2(\Omega)},$$

and define:

$$\tilde{\varphi}_M = \arg\sup_{\varphi\in F} \varepsilon_{M-1}(\varphi)$$

and: $\sigma_M = \arg\sup_{\sigma\in\Sigma} |\sigma(\tilde{\varphi}_M - \mathcal{J}_{M-1}[\tilde{\varphi}_M])|$ The next basis function is then

$$\tilde{q}_M = \frac{\tilde{\varphi}_M - \mathcal{J}_{M-1}[\tilde{\varphi}_M]}{\sigma_M(\tilde{\varphi}_M - \mathcal{J}_{M-1}[\tilde{\varphi}_M])}.$$

We finally define the matrix $\widetilde{B^M}$ such that $\tilde{B}^M_{ij} = \sigma_i(\tilde{q}_j)$, and set $\tilde{X}_M \equiv \text{span}\{\tilde{q}_j,\ j \in [1, M]\} = \text{span}\{\tilde{\varphi}_j,\ j \in [1, M]\}$. It can be proven as in [95, 104, 118].

Lemma 4.2.3. *For any $M \leq M_{max}$, the set $\{\tilde{q}_j,\ j \in [1, M]\}$ is linearly independent and $\tilde{X}_M$ is of dimension M. The matrix B^M is lower triangular with unity diagonal (hence invertible) with other entries in the interval $[-1, 1]$. The generalized empirical interpolation procedure is well-posed in $L^2(\Omega)$.*

In order to quantify the error of the interpolation procedure, like in the standard interpolation procedure, we introduce the Lebesgue constant in the L^2 norm: $\Lambda_M = \sup_{\varphi\in L^2(\Omega)} \dfrac{\|\mathcal{J}_M[\varphi]\|_{L^2(\Omega)}}{\|\varphi\|_{L^2(\Omega)}}$ i.e. the L^2–norm of $\mathcal{J}_M$. A similar result as in the previous subsection holds.

Lemma 4.2.4. *$\forall \varphi \in F$, the interpolation error satisfies:*

$$\|\varphi - \mathcal{J}_M[\varphi]\|_{L^2(\Omega)} \leq (1 + \Lambda_M) \inf_{\psi_M \in \tilde{X}_M} \|\varphi - \psi_M\|_{L^2(\Omega)}.$$

A (very pessimistic) upper-bound for Λ_M is:

$$\Lambda_M \leq 2^{M-1} \max_{i\in[1,M]} \|\tilde{q}_i\|_{L^2(\Omega)}.$$

Proof. The first part is standard and relies on the fact that, for any $\psi \in \tilde{X}_M$ then $\mathcal{J}_M(\psi_M) = \psi_M$. It follows that

$$\forall \psi_M \in \tilde{X}_M, \quad \|\varphi - \mathcal{J}_M[\varphi]\|_{L^2(\Omega)} = \|[\varphi - \psi_M] - \mathcal{J}_M[\varphi - \psi_M]\|_{L^2(\Omega)} \leq (1 + \Lambda_M)\|\varphi - \psi_M\|_{L^2(\Omega)}.$$

Let us now consider a given $\varphi \in F$ and its interpolant $\mathcal{J}_M[\varphi] = \sum_{i=1}^{M} \widetilde{\alpha_i^M}(\varphi)\tilde{q}_i$ in dimension M. The constants $\widetilde{\alpha_i^M}(\varphi)$ come from the generalized interpolation problem: $\forall j \in [1, M]$, $\sigma_j(\varphi) = \sum_{i=1}^{j-1} \widetilde{\alpha_i^M}(\varphi)\sigma_j(\tilde{q}_i) + \widetilde{\alpha_j^M}(\varphi)\sigma_j(\tilde{q}_j)$. We infer the recurrence relation for the constants:

$$\forall j \in [1, M], \ \widetilde{\alpha_j^M}(\varphi) = \sigma_j(\varphi) - \sum_{i=1}^{j-1} \alpha_i(\varphi)\sigma_j(\tilde{q}_i).$$

Based on the properties of the entries in matrix $\tilde{B}^M$ stated in lemma 4.2.3, we can obtain, by recurrence, an upper bound for each $\widetilde{\alpha_j^M}(\varphi)$: $\forall j \in [1, M]$, $|\widetilde{\alpha_j^M}(\varphi)| \leq (2^{j-1})\,\|\varphi\|_{L^2(\Omega)}$. Then, $\forall \varphi \in F$, $\forall M \leq M_{\max}$: $\| \mathcal{J}_M(\varphi)\|_{L^2(\Omega)} \leq \left[\sum_{i=1}^{M} (2^{j-1})\,\|\tilde{q}_i\|_{L^2(\Omega)}\right] \|\varphi\|_{L^2(\Omega)}$. Therefore: $\Lambda_M \leq 2^{M-1} \max_{i\in[1,M]} \|\tilde{q}_i\|_{L^2(\Omega)}$. Note that the norms of the rectified basis function $\tilde{q}_i$ verify $\|\tilde{q}_i\|_{L^2(\Omega)} \geq 1$ from the hypothesis done on the norm of the σ_i.

□

4.2.3 Numerical results

The results that we present here to illustrate the GEIM are based on data acquired in silico using the finite element code Freefem++ [58] on the domain represented on figure 4.1.

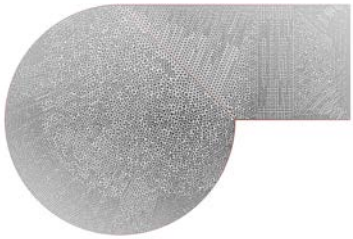


Figure 4.1: The domain Ω and its mesh.

We consider over the domain $\Omega \in \mathbb{R}^2$ the Laplace problem :

$$\begin{aligned} -\Delta\varphi &= f, \quad \text{in } \Omega \\ f &= 1 + (\alpha \sin(x) + \beta \cos(\gamma\pi y))\chi_1(x, y) \end{aligned} \tag{4.9}$$

complemented with homogeneous Dirichlet boundary conditions. Here α, β and γ are 3 parameters freely chosen in given intervals in $\mathbb{R}$ that modulate the forcing term on the right hand side. We assume that the forcing term only acts on a part of Ω named Ω_1 ($\Omega_1 = \text{support}(\chi_1)$) and we denote as Ω_2 the remaining part $\Omega_2 = \Omega \setminus \overline{\Omega}_1$.

The easy observation is that the solution φ, depends on the parameters α, β, γ : we plot in figure 4.2 one of the possible solutions

We also note that the restriction $\varphi_{|\Omega_2}$ to Ω_2 is indirectly dependent on these coefficients and thus is a candidate for building a set (when the parameters vary) of small Kolmogorov width. This

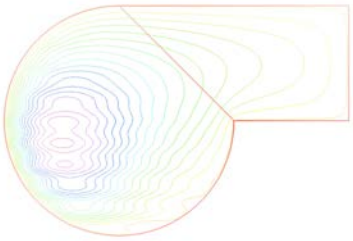


Figure 4.2: One of the solutions, we note that the effect of the forcing is mainly visible on domain Ω_1 on the left hand side.

Figure 4.3: Three generic solutions restricted on the sub-domain Ω_2.

can be guessed if we look at the numerical simulations obtained for three representative choices for α, β, γ (see figure 4.3).

For the GEIM, we use moments computed from the restriction of the solution $\varphi(\alpha, \beta, \gamma)$ over Ω_2 multiplied by localized functions with small compact support over Ω_2. The reconstructed solutions with the GEIM based on only 5 interpolating functions is 10^{14} times better than the reconstructed function with 1 interpolating function illustrating the high order of the reconstruction's convergence.

In the next example, we choose a similar problem but the shape of domain Ω_2 is a further parameter (see figure 4.4).

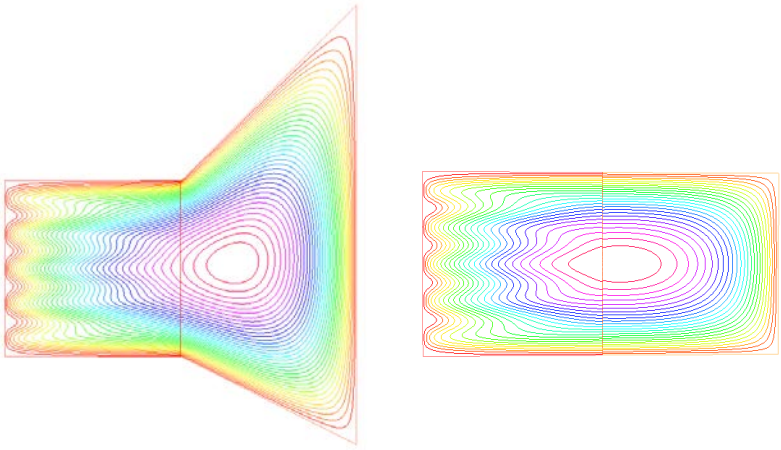


Figure 4.4: Two generic solutions when shape of the sub-domain Ω_2 varies.

In order to get an idea of the Kolmogorov width of the set $\{\varphi_{|\Omega_2}(\alpha, \beta, \gamma, \Omega_2)\}$, we perform two Singular Value Decompositions (one in L^2, the other in H^1) over 256 values (approximated again with Freefem++) and plot the decay rate of the eigenvalues ranked in decreasing order: the results are shown on figure 4.5.

We note that after about 9 eigenvalues, the finite element error dominates the decay rate of the true eigenvalues. The GEIM is built up again with captors represented as local weighted averages

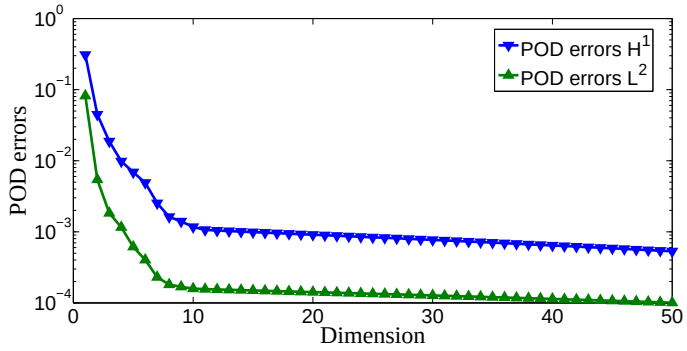


Figure 4.5: Two SVD (in L^2 and in H^1) of the set of solutions over Ω_2.

over Ω_2. The interpolation error is presented on the next figure (figure 4.6) and we note that the decay rate, measured both in L^2 and H^1 is again quite fast. In order to compare with the best fit represented by the projection, in L^2 or in H^1, we use the SVD eigenvectors associated with the first M eigenvalues and compare it with $\mathcal{J}_M$, for various values of M. This is represented on figure 4.7.

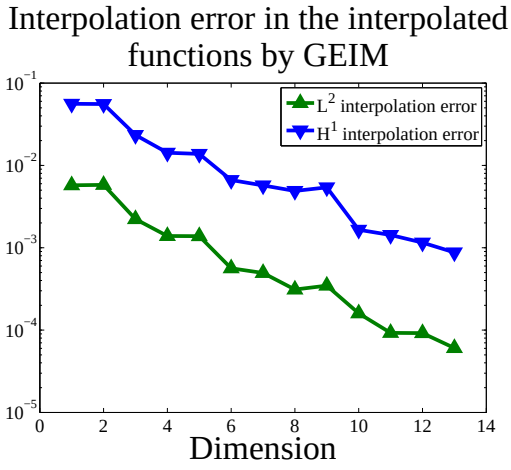


Figure 4.6: The worse GEIM error with respect to M .

The very good comparison allows to expect that the Lebesgue constant is much better than what is announced in lemma 4.2.4. A computational estimation (represented in Fig. 4.8) of Λ_M has been carried out:

$$\widetilde{\Lambda_M} = \max_{i\in[1,256]} \frac{\|\mathcal{I}_M[u_i]\|_{L^2(\Omega)}}{\|u_i\|_{L^2(\Omega)}}.$$

4.2.4 The framework

Imagine that we want to supervise a process in **real-time** for which we have a parameter dependent PDE. Assume that the computation of the solution over the full domain Ω is too expensive but we are in a situation where the domain Ω can be decomposed, as before, into two non overlapping subdomains Ω_1 and Ω_2 such that:

- Ω_1 is a small subdomain but the set of the restriction of the parameter dependent solutions has a large Kolmogorov width.

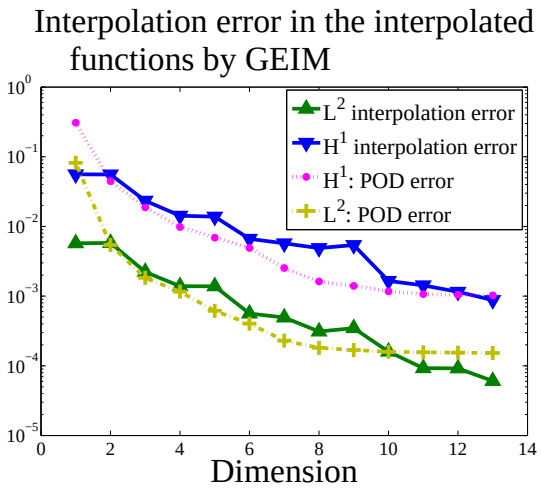


Figure 4.7: Evolution of the GEIM error versus the best fit error, both in L^2 and in H^1-norms.

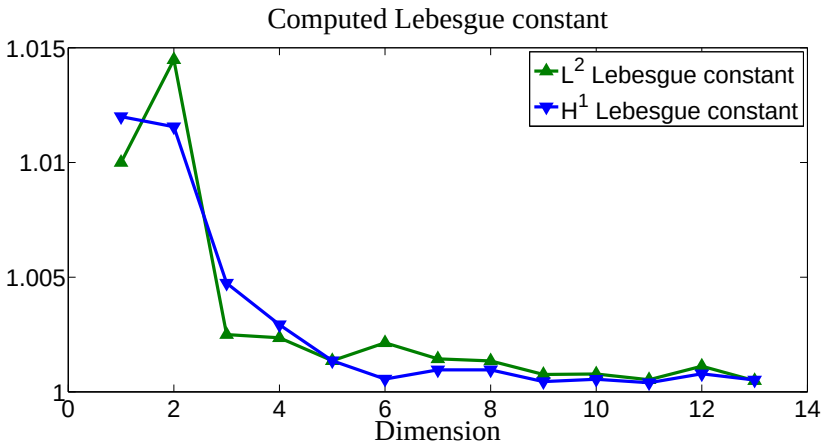


Figure 4.8: Evolution of the Lebesgue constant, i.e. the norm of the GEIM operator, both in L^2 and in H^1.

- Ω_2 is a big subdomain but the set of the restriction of the parameter dependent solutions has a small Kolmogorov n-width

In addition, we assume that it is possible to get outputs from sensors based in Ω_2. The GEIM allows to accurately reconstruct the current solution associated to some parameters over Ω_2 and thus is able to build the boundary condition necessary over the interface between Ω_1 and Ω_2. This boundary condition complemented with the initially given boundary condition over $\partial\Omega$ provides the necessary boundary condition over $\partial\Omega_1$. This defines the original PDE set now over Ω_1 and not over the whole Ω as is illustrated in figures 4.9 and 4.10.

4.2.5 The combined approach – numerical results

We take over the numerical frame of the previous section and go further. We want to apply the GEIM to have a knowledge of the solution $\varphi_{|\Omega_2}$ and want to use the trace of the reconstruction on the interface to provide the boundary condition, over $\partial\Omega_1$ to the problem

$$-\Delta\varphi = f, \ in \ \Omega_1$$
$$f = 1 + (\alpha \sin(x) + \beta \cos(\gamma\pi y))\chi_1(x, y)$$

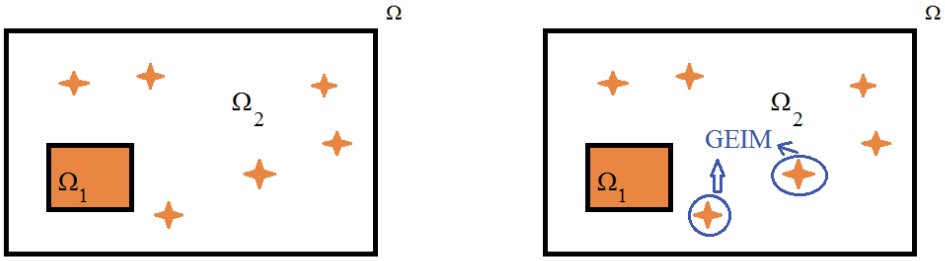


Figure 4.9: Schematic representation of the reconstruction over Ω_2.

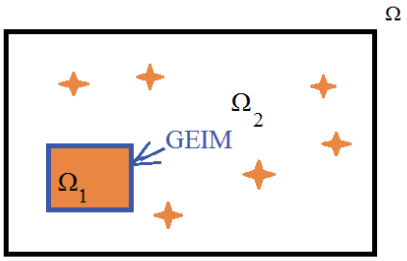


Figure 4.10: Schematic representation of the recovery over Ω_1 thanks to the knowledge of the interface condition.

derived from (4.9).

The results are presented in figure 4.11 where both the H^1 error on $\varphi_{|\Omega_1}$ and $\varphi_{|\Omega_2}$ are presented as a function of M being the number of interpolation data that are used to reconstruct $\varphi_{|\Omega_2}$. This illustrates that the use of the small Kolmogorov width of the set $\{\varphi_{|\Omega_2}\}$ as the parameters vary (including the shape of Ω_2) can help in determining the value of the full φ all over Ω.

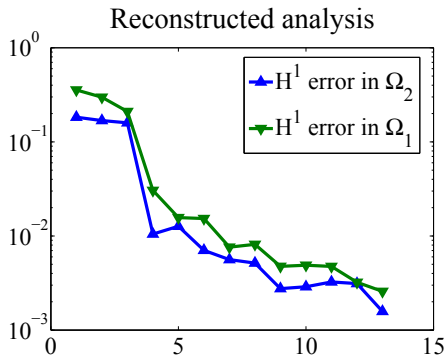


Figure 4.11: Reconstructed analysis — error in H^1-norm over Ω_1 and Ω_2.

4.3 About noisy data

In practical applications, data are measured with an intrinsic noise due to physical limitations of the sensors. In some sense, the noisy data acquired from the sensors are exact acquisitions

from a noisy function that we consider to be a Markovian random field with spatial values locally dependent (on the support of the sensor) and globally independent (from one sensor to the others). An extension of the previous development needs therefore to be done in order to take this fact under consideration.

Let us assume that all the sensors are subject to the same noise, i.e. provide averages — or some moments — computed, not from φ, but from a random process $\varphi_\varepsilon \simeq \mathcal{N}(\varphi, \varepsilon^2)$. The norm of the GEIM operator being equal to Λ_M the GEIM-reconstruction forms a random process $\mathcal{J}_M[\varphi_\varepsilon] \simeq \mathcal{N}(\mathcal{J}_M[\varphi], \Lambda_M^2\varepsilon^2)$ due to linearity.

Even though the Lebesgue constant seems to be small in practice, we would like to use all the data that are available in order to get a better knowledge of φ. For the definition of $\mathcal{J}_M$ we indeed only use M data selected out of a large set of all data. For this purpose, let us consider that, with some greedy approaches, we have determined P independent series of M different captors $\{\sigma_1^{(p)}, \sigma_2^{(p)}, \ldots, \sigma_M^{(p)}\}$, $\forall 1 \leq p \leq P$. For each of these series, the GEIM applied to φ is noisy and each application provides $\mathcal{J}_M^p[\varphi_\varepsilon] \simeq \mathcal{N}(\mathcal{J}_M^p[\varphi], {\Lambda_M^p}^2\varepsilon^2)$. We shall use these P reconstructions by averaging them and expect to improve the variance of the reconstruction.

Let $\lambda^{-1} = \frac{1}{P}\sum_{p=1}^{P}\frac{1}{\Lambda_N^p}$. Since the P realizations : $\{\mathcal{J}_M^p[\varphi_\varepsilon]\}_p$ are independent, then the random variable $\overline{\mathcal{J}_M^P}(\varepsilon) = \frac{\lambda}{P}\sum_{p=1}^{P}\frac{\mathcal{J}_M^p[\varphi_\varepsilon]}{\Lambda_N^{(p)}}$ follows a Gaussian Markov random field of parameters $\mathcal{N}(\mathcal{J}_N(\varphi), \frac{\epsilon^2\lambda^2}{P})$. A realization of this random process could be chosen for an improved estimate of $\mathcal{J}_M(\varphi)$. Indeed, the law of the error follows $\mathcal{N}(0, \frac{\epsilon^2\lambda^2}{P})$ and its variance can be less than the size of the initial noise on the captors (ϵ) provided that $\Lambda_N^{(p)} < \sqrt{P}, \forall 1 \leq p \leq P$, which, from the numerical experiments, seems to be the case.

4.4 Conclusions

We have presented a generalization of the Empirical Interpolation Method, based on ad'hoc interpolating functions and data acquired from sensors of the functions to be represented as those that can arise from data assimilation. We think that the GEIM is already interesting per se as it allows to select in a greedy way the most informative sensors one after the other. It can also propose, in case this is feasible, to build better sensors in order to complement a given family of existing ones and/or detect in which sense some of them are useless because redundant. Finally we also explain how noise on the data can be filtered out.

The coupled use of GEIM with reduced domain simulation is also proposed based on domain decomposition techniques leading to a small portion where numerical simulation is performed and a larger one based on data assimilation.

We think that the frame presented here can be used as an alternative to classical Bayesian or frequentistic statistic where the knowledge developed on the side for building mathematical models and their simulations can be fully used for data mining (we refer also to [98] and [109] for recent contributions in this direction).

Acknowledgements

This work was supported in part by the joint research program MANON between CEA-Saclay and University Pierre et Marie Curie-Paris 6. We want to thank G. Biot from LSTA and G. Pagès from LPMA for constructive discussions on the subject.

Chapter 5

The generalized empirical interpolation method: stability theory on Hilbert spaces and an application to the Stokes equation

This is a submitted paper with Y. Maday, A.T. Patera and M. Yano. Its reference in the manuscript is:

[77] Y. Maday, O. Mula, A.T. Patera and M. Yano. The generalized Empirical Interpolation Method: stability theory on Hilbert spaces with an application to the Stokes equation. Submitted, 2014.

Abstract

The Generalized Empirical Interpolation Method (GEIM) is an extension first presented in [76] of the classical empirical interpolation method (see [11], [55], [80]) that replaces the evaluation at interpolating points by interpolating continuous linear functionals on a class of Banach spaces. As outlined in [76], this allows to relax the continuity constraint in the target functions and expand the application domain. A special effort has been made in this paper to understand the concept of stability condition of the generalized interpolant (the Lebesgue constant) by relating it to an inf-sup problem in the case of Hilbert spaces. On a second part, it will be explained how GEIM can be employed to monitor in real time physical experiments by combining the acquisition of measurements from the process with mathematical models (parameter dependent PDE's). This idea will be illustrated through a parameter dependent Stokes problem in which it will be shown that the pressure and velocity fields can efficiently be reconstructed with a relatively low dimension of the interpolating spaces.

Introduction

Let $\mathcal{X}$ be a Banach space of functions defined over a domain $\overline{\Omega} \in \mathbb{R}^d$ (or $\mathbb{C}^d$), let $(X_n)_n$, $X_n \subset \mathcal{X}$, be a family of finite dimensional spaces, $\dim X_n = n$, and let $(S_n)_n$ be an associated family of sets of points: $S_n = \{x_i^n\}_{i=1}^n$, with $x_i^n \in \overline{\Omega}$. The problem of interpolating any function $f \in \mathcal{X}$ has traditionally been stated as:

$$\text{"Find } f_n \in X_n \text{ such that } f_n(x_i^n) = f(x_i^n),\ \forall i \in \{1, \dots, n\}\text{"}, \tag{5.1}$$

where we note that it is implicitly required that $\mathcal{X}$ is a Banach space of continuous functions. The most usual approximation in this sense is the Lagrangian interpolation, where the interpolating spaces X_n are of polynomial nature (spanned by plain polynomials, rational functions, Fourier series...) and the question on how to appropriately select the interpolating points in this case has broadly been explored. Although there exists still nowadays open issues on Lagrangian interpolation (see, e.g. [25]), it is also interesting to look for extensions of this procedure in which the interpolating spaces X_n are not necessarily of polynomial nature. The search for new interpolating spaces X_n is therefore linked with the question on how to optimally select the interpolating points in this case and how to obtain a process that is at least stable and close to the best approximation in some sense.

Although several procedures have been explored in this direction (we refer to [114], [47] and also to the kriging studies in the stochastic community such as [65]), of particular interest for the present work is the Empirical Interpolation Method (EIM, [11], [55], [80]) that has been developed in the broad framework where the functions f to approximate belong to a compact set F of continuous functions ($\mathcal{X} = \mathcal{C}(\Omega)$). The structure of F is supposed to make any $f \in F$ be approximable by finite expansions of small size. This is quantified by the Kolmogorov $n-$width $d_n(F, \mathcal{X})$ of F in $\mathcal{X}$ (see definition 5.2 below) whose smallness measures the extent to which F can be approximated by some finite dimensional space $X_n \subset \mathcal{X}$ of dimension n. Unfortunately, in general, the best approximation n-dimensional space is not known and, in this context, the Empirical Interpolation Method aims at building a family of suitable enough interpolating spaces X_n together with sets of interpolating points S_n such that the interpolation is well posed. This is done by a greedy algorithm on both the interpolating points and the interpolating selected functions φ_i (see [11]). This procedure has the main advantage of being constructive, i.e. the sequence of interpolating spaces (X_n) and interpolating points (S_n) are hierarchically defined and the procedure can easily be implemented by recursion.

A recent extension of this interpolation process consists in generalizing the evaluation at interpolating points by application of a class of interpolating continuous linear functionals chosen in a given dictionary $\Sigma \subset \mathcal{L}(\mathcal{X})$. This gives rise to the so-called Generalized Empirical Interpolation Method (GEIM). In this new framework, the particular case where the space $\mathcal{X} = L^2(\Omega)$ was first studied in [76]. We also mention the preliminary works of [38] in which the authors introduced the use of linear functionals in EIM in a finite dimensional framework. In the present paper, we will start by revisiting the foundations of the theory in order to show that GEIM holds for Banach spaces $\mathcal{X}$ (section 5.1). The concept of stability condition (Lebesgue constant, Λ_n) of the generalized interpolant will also be introduced.

In the particular case where $\mathcal{X}$ is a Hilbert space, we will provide an interpretation of the generalized interpolant of a function as an oblique projection. This will shed some light in the understanding of GEIM from an approximation theory perspective (section 5.2.1). This point of view will be the key to show that the Lebesgue constant is related to an inf-sup problem (section 5.2.2) that can be easily computed (section 5.3). The derived formula can be seen as an extension of the classical formula for Lagrangian interpolation to Hilbert spaces. It will also be shown that the Greedy algorithm aims at minimizing the Lebesgue constant in a sense that will be made precise in section 5.2.3. Furthermore, the inf-sup formula that will be introduced will explicitly show that there exists an interaction between the dictionary Σ of linear functionals and the Lebesgue constant. Although it has so far not been possible to derive a general theory about the impact of Σ on the behavior of the Lebesgue constant, we present in section 5.4 a first simple example in which this influence is analyzed.

The last part of the paper (section 5.5) will deal with the potential applications of the method. In particular, we will explain how GEIM can be used to build a tool for the real-time monitoring of a physical or industrial process. This will be done by combining measurements collected from

the process itself with a mathematical model (a parameter dependent PDE) that represents our physical understanding of the process under consideration. This idea will be illustrated through a parameter dependent Stokes problem for $\mathcal{X} = (H^1(\Omega))^2 \times L^2(\Omega)$.

Taking advantage of this idea, we will outline in the conclusion how the method could be used to build an adaptive tool for the supervision of experiments that could distinguish between normal and accidental conditions. We believe that this tool could help in taking real-time decisions regarding the security of processes.

5.1 The Generalized Empirical Interpolation Method

Let $\mathcal{X}$ be a Banach space of functions defined over a domain $\Omega \subset \mathbb{R}^d$, where $d = 1, 2, 3$. Its norm is denoted by $\|.\|_{\mathcal{X}}$. Let F be a compact set of $\mathcal{X}$. With $\mathcal{M}$ being some given large number, we assume that the dimension of the vectorial space spanned by F (denoted as $\mathcal{F} = \text{span}\{F\}$) is of dimension larger than $\mathcal{M}$. Our goal is to build a family of n-dimensional subspaces of $\mathcal{X}$ that approximate well enough any element of F. The rationale of this approach is linked to the notion of n-width following Kolmogorov [66]:

Definition 5.1.1. *Let F be a subset of some Banach space $\mathcal{X}$ and Y_n be a generic n-dimensional subspace of $\mathcal{X}$. The deviation between F and Y_n is*

$$E(F; Y_n) := \sup_{x \in F} \inf_{y \in Y_n} \|x - y\|_{\mathcal{X}} .$$

The Kolmogorov n-width of F in $\mathcal{X}$ is given by

$$\begin{aligned} d_n(F, \mathcal{X}) &:= \inf\{E(F; Y_n) : Y_n \text{ a } n\text{-dimensional subspace of } X\} \\ &= \inf_{Y_n} \sup_{x \in F} \inf_{y \in Y_n} \|x - y\|_{\mathcal{X}} . \end{aligned} \tag{5.2}$$

The smallness of the n-width of F thus measures to what extent the set F can be approximated by an n-dimensional subspace of $\mathcal{X}$. Several reasons can account for a rapid decrease of $d_n(F, \mathcal{X})$: if F is a set of functions defined over a domain, we can refer to regularity, or even to analyticity, of these functions with respect to the domain variable (as analyzed in the example in [66]). Another possibility — that will actually be used in our numerical application— is when $F = \{u(\mu, .),\ \mu \in D\}$, where D is a compact set of $\mathbb{R}^p$ and $u(\mu, .)$ is the solution of a PDE parametrized by μ. The approximation of any element $u(\mu, .) \in F$ by finite expansions is a classical problem addressed by, among others, reduced basis methods and the regularity of u in μ can also be a reason for having a small n-width as the results of [82] and [27] show.

Finally, let us also assume that we have at our disposal a dictionary of linear functionals $\Sigma \subset \mathcal{L}(\mathcal{X})$ with the following properties:

P1: $\forall \sigma \in \Sigma,\ \|\sigma\|_{\mathcal{L}(\mathcal{X})} = 1$.

P2: *Unisolvence property*: If $\varphi \in \text{span}\{F\}$ is such that $\sigma(\varphi) = 0,\ \forall \sigma \in \Sigma$, then $\varphi = 0$.

Given this setting, GEIM aims at building M-dimensional interpolating spaces X_M spanned by suitably chosen functions $\{\varphi_1, \varphi_2, \ldots, \varphi_M\}$ of F together with sets of M selected linear functionals $\{\sigma_1,\ \sigma_2, \ldots,\ \sigma_M\}$ coming from Σ such that any $\varphi \in F$ is well approximated by its generalized interpolant $\mathcal{J}_M[\varphi] \in X_M$ defined by the following interpolation property:

$$\forall \varphi \in \mathcal{X}, \quad \mathcal{J}_M[\varphi] \in X_M \text{ such that } \sigma_i(\mathcal{J}_M[\varphi]) = \sigma_i(\varphi),\ \forall i = 1, \ldots, M. \tag{5.3}$$

The definition of GEIM in the sense of (5.3) raises several questions:

- is there an optimal selection for the linear functionals σ_i within the dictionary Σ ?
- is there a constructive optimal selection for the functions $\varphi_i \in F$?

- given a set of linearly independent functions $\{\varphi_i\}_{i\in[1,M]}$ and a set of continuous linear functionals $\{\sigma_i\}_{i\in[1,M]}$, does the interpolant exist in the sense of (5.3)?
- is the interpolant unique?
- Under what hypothesis can we expect the GEIM approximation to converge rapidly to φ?

In what follows, we provide answers to these questions either with rigorous proofs or with numerical evidences.

The construction of the generalized interpolation spaces X_M and the selection of the suitable associated linear functionals is recursively performed by following a greedy procedure very similar to the one of the classical EIM. The first selected function is, e.g.,

$$\varphi_1 = \arg\sup_{\varphi\in F} \|\varphi\|_{\mathcal{X}},$$

that defines $X_1 = \text{span}\{\varphi_1\}$. The first interpolating linear functional is

$$\sigma_1 = \arg\sup_{\sigma\in\Sigma} |\sigma(\varphi_1)|.$$

The interpolation operator $\mathcal{J}_1 : \mathcal{X} \mapsto X_1$ is defined such that (5.3) is true for $M = 1$, i.e. $\sigma_1(\mathcal{J}_1[\varphi]) = \sigma_1(\varphi)$, for any $\varphi \in \mathcal{X}$. To facilitate the practical computation of the generalized interpolant, we express it in terms of

$$q_1 = \frac{\varphi_1}{\sigma_1(\varphi_1)},$$

which will be the basis function that will be employed for X_1. In this basis, the interpolant reads

$$\mathcal{J}_1[\varphi] = \sigma_1(\varphi)q_1, \quad \forall\varphi \in \mathcal{X}.$$

We then proceed by induction. With $M_{\max} < \mathcal{M}$ being an upper bound fixed *a priori*, assume that, for a given $1 \le M < M_{\max}$, we have selected a set of functions $\{\varphi_1, \varphi_2, \dots, \varphi_M\}$ associated to the basis functions $\{q_1, q_2, \dots, q_M\}$ and the interpolating linear functionals $\{\sigma_1, \sigma_2, \dots, \sigma_M\}$. The generalized interpolant is assumed to be well defined by (5.3), i.e.,

$$\mathcal{J}_M[\varphi] = \sum_{j=1}^{M} \alpha_j^M(\varphi) q_j, \quad \varphi \in \mathcal{X},$$

where the coefficients $\alpha_j^M(\varphi),\ j = 1, \dots, M$ are given by the interpolation problem

$$\begin{cases} \text{Find } \{\alpha_j^M(\varphi)\}_{j=1}^M \text{ such that:} \\ \sum\limits_{j=1}^{M} \alpha_j^M(\varphi) B_{i,j}^M = \sigma_i(\varphi),\ \forall i = 1, \dots, M. \end{cases}$$

where $B_{i,j}^M$ are the coefficients of the $M \times M$ matrix $B^M := (\sigma_i(q_j))_{1\le i,j\le M}$. We now define

$$\forall\varphi \in F, \quad \varepsilon_M(\varphi) = \|\varphi - \mathcal{J}_M[\varphi]\|_{\mathcal{X}}.$$

At the $M+1$-th stage of the greedy algorithm, we choose φ_{M+1} such that

$$\varphi_{M+1} = \arg\sup_{\varphi\in F} \varepsilon_M(\varphi) \tag{5.4}$$

and

$$\sigma_{M+1} = \arg\sup_{\sigma\in\Sigma} |\sigma(\varphi_{M+1} - \mathcal{J}_M[\varphi_{M+1}])|. \tag{5.5}$$

The next basis function is then:

$$q_{M+1} = \frac{\varphi_{M+1} - \mathcal{J}_M[\varphi_{M+1}]}{\sigma_{M+1}(\varphi_{M+1} - \mathcal{J}_M[\varphi_{M+1}])} .$$

We finally set $X_{M+1} \equiv \text{span}\{\varphi_j,\ 1 \leq j \leq M+1\} = \text{span}\{q_j,\ 1 \leq j \leq M+1\}$. The interpolation operator $\mathcal{J}_{M+1} : \mathcal{X} \mapsto X_{M+1}$ is given by

$$\mathcal{J}_{M+1}[\varphi] = \sum_{j=1}^{M+1} \alpha_j^{M+1}(\varphi) q_j, \quad \forall \varphi \in \mathcal{X},$$

so as to satisfy (5.3). The coefficients $\alpha_j^{M+1}(\varphi),\ j = 1, \ldots, M+1$, are therefore given by the interpolation problem

$$\begin{cases} \text{Find } \{\alpha_j^{M+1}(\varphi)\}_{j=1}^{M+1} \text{ such that:} \\ \sum_{j=1}^{M+1} \alpha_j^{M+1}(\varphi) B_{i,j}^{M+1} = \sigma_i(\varphi),\ \forall i = 1, \ldots, M+1, \end{cases}$$

where $B^{M+1} = (\sigma_i(q_j))_{1 \leq i,j \leq M+1}$.

By following exactly the same guidelines as in [76] where the particular case $\mathcal{X} = L^2(\Omega)$ was addressed, it can be proven that, in the general case where $\mathcal{X}$ is a Banach space, the generalized interpolation is well-posed: for any $1 \leq M < \mathcal{M}$, the set of functions $\{q_j,\ j \in [1, M]\}$ is linearly independent and therefore the space X_M is of dimension M. Furthermore, the matrix B^M is lower triangular with unity diagonal (hence invertible) with off-diagonal entries in $[-1, 1]$.

Note that GEIM reduces to EIM if $\mathcal{X} = \mathcal{C}^0(\Omega)$ and Σ is composed of Dirac masses. Also, if the cardinality $\#F$ of F is finite, then the Greedy algorithm is exact in the sense that $F \subset X_{\#F}$. This type of property does not hold in traditional Lagrangian interpolation due to the fact that the interpolating polynomial spaces are used to interpolate continuous functions that are not necessarily of polynomial nature. Finally, note also that the approach can be shortcut if the basis functions are available, in which case the interpolating linear functionals/points are the only output of GEIM/EIM.

It is also important to point out that the current extension of EIM presents two major advantages: first, it allows the interpolation of functions of weaker regularity than $\mathcal{C}^0(\Omega)$. The second interest is related to the potential applications of GEIM: the use of linear functionals can model in a more faithful manner real sensors involved in physical experiments (indeed, these are in practice no point evaluations as it is usually supposed but rather local averages of some quantity of interest). The potentialities of these two aspects will be illustrated in the numerical application presented in section 5.5.

We now state a first result about the interpolation error of GEIM.

Theorem 5.1.2 (Interpolation error on a Banach space)**.** *$\forall \varphi \in \mathcal{X}$, the interpolation error satisfies:*

$$\|\varphi - \mathcal{J}_M[\varphi]\|_{\mathcal{X}} \leq (1 + \Lambda_M) \inf_{\psi_M \in X_M} \|\varphi - \psi_M\|_{\mathcal{X}}, \tag{5.6}$$

where

$$\Lambda_M := \|\mathcal{J}_M\|_{\mathcal{L}(\mathcal{X})} = \sup_{\varphi \in \mathcal{X}} \frac{\|\mathcal{J}_M[\varphi]\|_{\mathcal{X}}}{\|\varphi\|_{\mathcal{X}}} \tag{5.7}$$

is the Lebesgue constant in the $\mathcal{X}$ norm.

Proof. The desired result easily follows since for any $\varphi \in \mathcal{X}$ and any $\psi_M \in X_M$ we have:

$$\begin{aligned} \|\varphi - \mathcal{J}_M[\varphi]\|_{\mathcal{X}} &= \|[\varphi - \psi_M] - \mathcal{J}_M[\varphi - \psi_M]\|_{\mathcal{X}} \\ &\leq \|I_F - \mathcal{J}_M\|_{\mathcal{L}(\mathcal{X})}\|\varphi - \psi_M\|_{\mathcal{X}} \\ &\leq (1 + \|\mathcal{J}_M\|_{\mathcal{L}(\mathcal{X})})\|\varphi - \psi_M\|_{\mathcal{X}}, \end{aligned}$$

which yields the desired inequality. □

The last term in the right hand side of equation (5.6) is known as the best fit of φ by elements in the space X_M. However, X_M does not in general coincide with the optimal M-dimensional space in the sense that $X_M \neq X_M^{\text{opt}}$, with $X_M^{\text{opt}} = \underset{\substack{Y_M \subset \mathcal{X} \\ \dim(Y_M)=M}}{\arg\inf} E(F, Y_M)$. This raises the question of the quality of the finite dimensional subspaces X_M provided by the Greedy selection procedure. It has been proven first in [78] in the case of $\mathcal{X} = L^2(\Omega)$ and then in [79] in a general Banach space that the interpolating spaces X_M coming from the Greedy selection procedure of GEIM are quite optimal and that the lack of optimality comes from the Lebesgue constant. The main results are the following (see [79]):

Theorem 5.1.3 (See corollary 6.3.13 of [79])**.**

i) If $d_M(F, \mathcal{X}) \leq C_0 M^{-\alpha}$, $M = 1, 2, \ldots$ and that $(1+\Lambda_M) \leq C_\zeta M^\zeta$, for any $M = 1, 2, \ldots$, then the interpolation error satisfies for any $\varphi \in F$ and any $\beta > \frac{1}{2}$ the inequality $\|\varphi - \mathcal{J}_M[\varphi]\|_{\mathcal{X}} \leq C_\zeta C_1 M^{-\alpha+2\zeta+\beta}$, where

$$C_1 := \max\left\{ C_0 2^{\frac{2\alpha^2}{\zeta}} \left(\frac{\zeta+\beta}{\beta-\frac{1}{2}}\right)^\alpha \max\left(1; C_\zeta^{\frac{\zeta+\beta}{\zeta}}\right); \max_{M=1,\ldots,2\lfloor 2(\zeta+\beta)\rfloor+1} M^{\alpha-\zeta-\beta} \right\}.$$

ii) If (Λ_M) is a monotonically increasing sequence and if $d_M(F, \mathcal{X}) \leq C_0 e^{-c_1 M^\alpha}$ for any $M \geq 1$, then, for any $\varphi \in F$, the interpolation error can be bounded as

$$\|\varphi - \mathcal{J}_M[\varphi]\|_{\mathcal{X}} \leq \begin{cases} 4C_0(1+\Lambda_1), & \text{if } M = 1. \\ \sqrt{2C_0}(1+\Lambda_M)^2\sqrt{M}e^{-c_2 M^\alpha}, & \text{if } M \geq 2. \end{cases}$$

As a consequence of this result, the interpolation error of GEIM will converge if the Lebesgue constant "does not increase too fast" in the sense that it allows that the previous upper bounds tend to zero as the dimension M increases. By following the same lines as in [76], it can be proven that when $\mathcal{X}$ is a Banach space, the Lebesgue constant has the exponential upper-bound

$$\Lambda_M \leq 2^{M-1} \max_{i\in[1,M]} \|q_i\|_{\mathcal{X}}, \tag{5.8}$$

which implies that the decay of $d_M(F, \mathcal{X})$ should be exponential in order to converge. However, the behavior of $(\Lambda_M)_M$ observed in numerical applications (see section 5.5) is rather linear and leads us to expect that the upper bound of (5.8) is far from being optimal in a class of set F of small Kolmogorov n-width.

5.2 Further results in the case of a Hilbert space

In this section $\mathcal{X}$ is a Hilbert space of functions where the norm $\|.\|_{\mathcal{X}}$ is induced by the inner product $(\cdot,\cdot)_{\mathcal{X}}$. We will see that in this case the generalized interpolant can been seen as an oblique projection. It will also be proven that we can derive a sharp interpolation error bound in this case. An explicit (and easily computable) formula for the Lebesgue constant will also be obtained and this formula will be used to show that the Greedy algorithm aims at minimizing the Lebesgue constant.

5.2.1 Interpretation of GEIM as an oblique projection

For $1 \leq j \leq M$, if σ_j is the j^{th}-linear functional selected by the greedy algorithm, let w_j be its Riesz representation in $\mathcal{X}$, i.e. w_j is such that

$$\forall \varphi \in \mathcal{X}, \quad \sigma_j(\varphi) = (w_j, \varphi)_{\mathcal{X}}. \tag{5.9}$$

It follows from the well posedness of the generalized interpolation that $\{\sigma_1, \dots, \sigma_M\}$ are linearly independent and therefore $\{w_1, \dots, w_M\}$ are also linearly independent. With these notations, we can provide the following interpretation of the generalized interpolant of a function (see figure 5.1 for a schematic representation):

Lemma 5.2.1. *$\forall f \in \mathcal{X}$, $\mathcal{J}_M[f]$ is an oblique projection onto the space X_M orthogonal to the M-dimensional space $W_M = \text{span}\{w_1, \dots, w_M\}$, i.e.*

$$(\mathcal{J}_M(f) - f, w)_{\mathcal{X}} = 0, \ \forall w \in W_M. \tag{5.10}$$

Proof. For any $f \in \mathcal{X}$, the interpolation property reads $\sigma_j(f) = \sigma_j(\mathcal{J}_M[f])$, for $1 \leq j \leq M$. It is then clear that $(w_j, f)_{\mathcal{X}} = (w_j, \mathcal{J}_M[f])_{\mathcal{X}}$ and the result easily follows from the fact that $\{w_1, \dots, w_M\}$ are a basis of W_M. □

A direct consequence of lemma 5.2.1 is the following result:

Corollary 5.2.2. *If $\Sigma = \left(\text{span}\{F\}^{\perp}\right)^{\circ}$, then $W_M = X_M$ and the resulting generalized interpolant is the orthogonal projection of f onto the space X_M.*

Proof. If $\Sigma = \left(\text{span}\{F\}^{\perp}\right)^{\circ}$, then, from the arg max definition of σ_k in the greedy algorithm, the Riesz representation of σ_k is the function $w_k = \varphi_k - \mathcal{J}_{k-1}(\varphi_k)$ for $k \geq 2$ and $w_1 = \varphi_1$ if $k = 1$. The interpolation property $\sigma_k(f - \mathcal{J}_M(f)) = 0$ implies in this case that $(w_k, f - \mathcal{J}_{k-1}(f))_{\mathcal{X}}$ for any $k \in \{1, \dots, M\}$. But since the family $\{w_1, \dots, w_M\}$ is a basis of X_M in this particular case, it follows that $(f - \mathcal{J}_M(f), w)_{\mathcal{X}} = 0$ for all $w \in X_M$. □

Remark 5.2.3. *The case $\Sigma = \left(\text{span}\{F\}^{\perp}\right)^{\circ}$ is a theoretical situation that does not usually hold in practical applications. Corollary 5.2.2 is however a first step towards the theoretical understanding of the impact of the dictionary Σ on the interpolation procedure.*

From lemma 5.2.1, note that $\mathcal{J}_M[f]$ can also be seen as a particular Petrov-Galerkin approximation of the function f in the case where the approximation space is X_M and the trial space is W_M. Indeed, the search for the generalized interpolant can be stated as:

$$\begin{cases} \text{Given } f \in \mathcal{X}, \text{ find } \mathcal{J}_M[f] \in X_M \text{ such that} \\ (\mathcal{J}_M[f], w)_{\mathcal{X}} = (f, w)_{\mathcal{X}}, \quad \forall w \in W_M. \end{cases} \tag{5.11}$$

This formulation leads to the classical error estimation:

$$\|f - \mathcal{J}_M[f]\|_{\mathcal{X}} \leq \left(1 + \frac{1}{\beta_M}\right) \inf_{\psi_M \in X_M} \|f - \psi_M\|_{\mathcal{X}}, \tag{5.12}$$

where β_M is the inf-sup constant

$$\beta_M := \inf_{x \in X_M} \sup_{w \in W_M} \frac{(x, w)_{\mathcal{X}}}{\|x\|_{\mathcal{X}} \|w\|_{\mathcal{X}}}. \tag{5.13}$$

It will be proven in the next section that the parameter $1/\beta_M$, which is, in fact, equal to the Lebesgue constant Λ_M. We will also see that the error bound provided in relation (5.12) is slightly suboptimal due to the presence of the coefficient 1 before the parameter $1/\beta_M$.

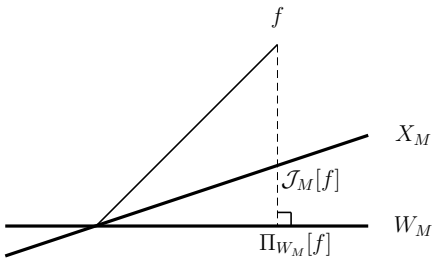


Figure 5.1: Interpretation of $\mathcal{J}_M[f]$ as an oblique projection.

5.2.2 Interpolation error

The interpretation of the generalized interpolant as an oblique projection is useful to derive the following result about the interpolation error:

Theorem 5.2.4 (Interpolation error on a Hilbert space)**.** *$\forall \varphi \in \mathcal{X}$, the interpolation error satisfies the sharp upper bound:*

$$\|\varphi - \mathcal{J}_M[\varphi]\|_{\mathcal{X}} \leq \Lambda_M \inf_{\psi_M \in X_M} \|\varphi - \psi_M\|_{\mathcal{X}} \tag{5.14}$$

where $\Lambda_M := \|\mathcal{J}_M\|_{\mathcal{L}(\mathcal{X})} = \sup_{\varphi \in \mathcal{X}} \dfrac{\|\mathcal{J}_M[\varphi]\|_{\mathcal{X}}}{\|\varphi\|_{\mathcal{X}}}$ *is the Lebesgue constant in the $\mathcal{X}$ norm. Furthermore,* $\Lambda_M = \dfrac{1}{\beta_M}$, *where*

$$\beta_M := \inf_{x \in X_M} \sup_{w \in W_M} \frac{(x, w)_{\mathcal{X}}}{\|x\|_{\mathcal{X}} \|w\|_{\mathcal{X}}}. \tag{5.15}$$

Proof. Let $\nu_M := \inf\limits_{w^\perp \in W_M^\perp} \sup\limits_{x^\perp \in X_M^\perp} \dfrac{(w^\perp, x^\perp)_{\mathcal{X}}}{\|w^\perp\|_{\mathcal{X}} \|x^\perp\|_{\mathcal{X}}}$. It is immediate that

$$\forall w^\perp \in W_M^\perp, \quad \nu_M \|w^\perp\|_{\mathcal{X}} \leq \sup_{x^\perp \in X_M^\perp} \frac{(w^\perp, x^\perp)_{\mathcal{X}}}{\|x^\perp\|_{\mathcal{X}}}.$$

Furthermore, for any $\varphi \in \mathcal{X}$, it follows from lemma 5.2.1 that $\varphi - \mathcal{J}_M[\varphi] \in W_M^\perp$. Then:

$$\nu_M \|\varphi - \mathcal{J}_M[\varphi]\|_{\mathcal{X}} \leq \sup_{x^\perp \in X_M^\perp} \frac{(\varphi - \mathcal{J}_M[\varphi], x^\perp)_{\mathcal{X}}}{\|x^\perp\|_{\mathcal{X}}}. \tag{5.16}$$

Besides, for any $x \in X_M$ and any $x^\perp \in X_M^\perp$:

$$(\varphi - x, x^\perp)_{\mathcal{X}} = (\varphi - \mathcal{J}_M[\varphi], x^\perp)_{\mathcal{X}} \ . \tag{5.17}$$

The Cauchy-Schwarz inequality applied to (5.16) combined with relation (5.17) yields:

$$\nu_M \|\varphi - \mathcal{J}_M[\varphi]\|_{\mathcal{X}} \leq \inf_{x \in X_M} \|\varphi - x\|_{\mathcal{X}}. \tag{5.18}$$

Next, it can be proven (see corollary 5.6.1 in appendix) that $\nu_M = \beta_M$, which yields the inequality

$$\|\varphi - \mathcal{J}_M[\varphi]\|_{\mathcal{X}} \leq \frac{1}{\beta_M} \inf_{\psi_M \in X_M} \|\varphi - \psi_M\|_{\mathcal{X}}. \tag{5.19}$$

The end of the proof consists in showing that $\frac{1}{\beta_M} = \Lambda_M = \sup_{\varphi \in \mathcal{X}} \frac{\|\mathcal{J}_M[\varphi]\|_{\mathcal{X}}}{\|\varphi\|_{\mathcal{X}}}$. This is done by noting first of all that formula (5.15) implies that

$$\forall \varphi \in \mathcal{X}, \quad \beta_M \|\mathcal{J}_M[\varphi]\|_{\mathcal{X}} \leq \sup_{w \in W_M} \frac{(\mathcal{J}_M[\varphi], w)_{\mathcal{X}}}{x} \leq \|\mathcal{J}_M[\varphi]\|_{\mathcal{X}},$$

where we have used the fact that $(\mathcal{J}_M[\varphi], w)_{\mathcal{X}} = (\varphi, w)_{\mathcal{X}}$ for all $w \in W_M$ and the Cauchy-Schwarz inequality. Therefore,

$$\forall \varphi \in \mathcal{X}, \ \|\mathcal{J}_M[\varphi]\|_{\mathcal{X}} \leq \frac{1}{\beta_M} \|\varphi\|_{\mathcal{X}},$$

which implies that $\Lambda_M \leq \frac{1}{\beta_M}$.

Let us now denote by v_M an element of X_M with norm $\|v_M\|_{\mathcal{X}} = 1$ such that

$$\sup_{w_M \in W_M} \frac{(v_M, w_M)_{\mathcal{X}}}{\|w_M\|_{\mathcal{X}}} = \beta_M.$$

If we call w_M^* the $\mathcal{X}$ projection of v_M over W_M, then

$$v_M = w_M^* + \tilde{w}_M^*,$$

with $w_M^* \in W_M$ and $\tilde{w}_M^* \in W_M^{\perp}$, so that $(w_M^*, \tilde{w}_M^*) = 0$. We have $\mathcal{J}_M(w_M^*) = v_M$. Indeed, by definition $\mathcal{J}_M[w_M^*] \in X_M$ and $\forall w_M \in W_M$, $(\mathcal{J}_M[w_M^*], w_M)_{\mathcal{X}} = (w_M^*, w_M)_{\mathcal{X}}$, which is exactly what v_M satisfies. In addition, $\sup_{w_M \in W_M} \frac{(v_M, w_M)_{\mathcal{X}}}{\|w_M\|_{\mathcal{X}}}$ is achieved for $w_M = w_M^*$ so that $\|w_M^*\|_{\mathcal{X}} = \beta_M$. This ends the proof that

$$1 = \|v_M\|_{\mathcal{X}} = \|\mathcal{J}_M[w_M^*]\|_{\mathcal{X}} = \frac{1}{\beta_M} \|w_M^*\|_{\mathcal{X}}.$$

Since the above result implies that $\frac{1}{\beta_M} \leq \Lambda_M$, we conclude that $\frac{1}{\beta_M} = \Lambda_M$.

□

Remark 5.2.5. *The link between the Lebesgue constant Λ_M and the inf-sup quantity β_M introduced in theorem 5.2.4 shows that Λ_M depends on the dictionary of linear functionals Σ and also on the interpolating space X_M. Although no theoretical analysis of the impact of these elements has been possible so far, we present in section 5.4 a numerical study about the influence of the dictionary Σ in Λ_M.*

Remark 5.2.6. *Note that, since theorem 5.2.4 holds only in Hilbert spaces, formula (5.15) does not apply to the Lebesgue constant of the classical EIM given that it is defined in the $L^\infty(\Omega)$ norm. The Hilbertian framework allows nevertheless to consider Dirac masses as linear functionals like in EIM if we place ourselves, e.g., in $H^2(\Omega)$.*

5.2.3 The Greedy algorithm aims at optimizing the Lebesgue constant

If we look in detail at the steps followed by the Greedy algorithm, once X_{M-1} and W_{M-1} have been derived, the construction of X_M and W_M starts by adding an element φ to X_{M-1}. In the Greedy process, this is done following formula (5.4), but let us analyze what happens when we add any $\varphi_M \in F$. The first consequence of its addition is that the resulting inf-sup constant becomes zero:

$$\inf_{\varphi \in \text{span}\{X_{M-1}, \varphi_M\}} \sup_{w \in W_{M-1}} \frac{(\varphi, w)_{\mathcal{X}}}{\|\varphi\|_{\mathcal{X}} \|w\|_{\mathcal{X}}} = 0. \tag{5.20}$$

Indeed, the addition of φ_M to the interpolating basis functions has the consequence of adding the element $\tilde{\varphi}_M = \varphi_M - \mathcal{J}_{M-1}[\varphi_M]$ that, by definition, is such that $(\tilde{\varphi}_M, w)_{\mathcal{X}} = 0,\ \forall w \in W_{M-1}$. We thus need to add an element to W_{M-1} in order to stabilize the inf-sup condition.

Let us denote by W the set of Riesz representations in $\mathcal{X}$ of the elements of our dictionary Σ. Since

$$\inf_{\varphi \in X_{M-1}+\varphi_M} \sup_{w \in W_{M-1}} \frac{(\varphi, w)_{\mathcal{X}}}{\|\varphi\|_{\mathcal{X}} \|w\|_{\mathcal{X}}}$$

is reached by $\tilde{\varphi}_M$, the aim is to add an element w_M of W that maximizes

$$\max_{w \in W} \frac{(\tilde{\varphi}_M, w)_{\mathcal{X}}}{\|w\|_{\mathcal{X}}}. \tag{5.21}$$

Since the elements of the dictionary are of norm 1 (see property P1 above), this corresponds exactly to one of the steps performed by the Greedy algorithm (see equation (5.5)). Furthermore, from the unisolvence property of our dictionary, the application

$$\begin{aligned} \|.\|_* : \mathcal{X} &\mapsto \mathbb{R} \\ \varphi &\mapsto \max_{w \in W} (\varphi, w)_{\mathcal{X}} \end{aligned}$$

defines a norm in $\mathcal{X}$. Then, formula (5.21) reads:

$$\max_{w \in W} \frac{(\tilde{\varphi}_M, w)_{\mathcal{X}}}{\|w\|_{\mathcal{X}}} = \|\varphi_M - \mathcal{J}_{M-1}[\varphi_M]\|_*.$$

It is thus clear that the choice of φ_M that maximizes the value of β_M is the one that maximizes $\varphi_M - \mathcal{J}_{M-1}[\varphi_M]$ in the $\|.\|_*$ norm. However, since in practice we do not have access to the entire knowledge of this norm, $\|.\|_*$ is replaced by the ambient norm $\|.\|_{\mathcal{X}}$:

$$\varphi_M = \arg\max_{\varphi \in F} \|\varphi - \mathcal{J}_{M-1}[\varphi]\|_* \sim \arg\max_{\varphi \in F} \|\varphi - \mathcal{J}_{M-1}[\varphi]\|_{\mathcal{X}}, \tag{5.22}$$

which is exactly what the Greedy algorithm does (see (5.4)). Hence, as a conclusion, with the practical tools that can be implemented, the choice of φ_M aims at minimizing the Lebesgue constant with the approximation explained in (5.22).

5.3 Practical implementation of the Greedy algorithm and the Lebesgue constant

In the present section, we discuss about some practical issues regarding the implementation of the Greedy algorithm and the Lebesgue constant Λ_M.

Since the cardinality of F is usually infinite, the practical implementation of the Greedy algorithm is carried out in a large enough sample subset $\mathcal{S}_F$ of finite cardinality $\#\mathcal{S}_F$ much larger than the dimension of the discrete spaces X_M and W_M we plan to use. For example, if $F = \{u(\mu, .), \mu \in \mathcal{D}\}$, we choose $\mathcal{S}_F = \{u(\mu, .), \mu \in \Xi_\mu \subset \mathcal{D}\}$ and Ξ_μ consists of $\#\mathcal{S}_F$ parameter sample points μ. We assume that this sample subset is representative enough of the entire set F in the sense that $\sup_{x \in F} \left\{ \inf_{y \in \text{span}\{\mathcal{S}_F\}} \|x - y\|_{\mathcal{X}} \right\}$ is much smaller than the accuracy we envision through the interpolation process. This assumption is valid for small dimension of F, or, more precisely, for small dimension of the parameter set $\mathcal{D}$. In case it cannot be implemented directly, we can follow two strategies that have been introduced on greedy approaches for reduced basis approximations

either based on (parameter) domain decomposition like in [41] or [42] based on an adaptive construction of the sample subset, starting from a very coarse definition as in [84]. These approaches have not been implemented here but we do not foresee any difficulty in adopting them to the GEIM framework.

The following lemma shows that the generalized interpolant can be recursively computed.

Lemma 5.3.1. *For any function $f \in \mathcal{X}$, we have the following recursion for $M \geq 1$*

$$\begin{cases} \mathcal{J}_M[f] = \mathcal{J}_{M-1}[f] + \sigma_M(f - \mathcal{J}_{M-1}[f])q_M \\ \mathcal{J}_0[f] = 0 \end{cases} \tag{5.23}$$

and the generalized interpolant of f can be recursively computed.

Proof. Using the fact that the spaces X_M are hierarchically defined, both hand sides of (5.23) belong to X_M. Using the fact that $\sigma_i(q_M) = 0$ for $i < M$ and the definition of $\mathcal{J}_M$ and $\mathcal{J}_{M-1}$, we infer that

$$\sigma_i\left(\mathcal{J}_M[f]\right) = \sigma_i\left(\mathcal{J}_{M-1}[f] + \sigma_M\left(f - \mathcal{J}_{M-1}[f]\right)q_M\right), \quad \forall i < M.$$

Finally, it is clear that the right and left and sides have the same image trough σ_M. The equality holds by uniqueness of the generalized interpolation procedure.

□

Remark 5.3.2. *This result also holds for the classical EIM case.*

The greedy algorithm is in practice a very time-consuming task whose computing time could significantly be reduced by the use of parallel architectures and the use of formula (5.23) as is outlined in algorithm 5.1.

Once X_M and W_M have been constructed thanks to algorithm 5.1, the Lebesgue constant can be computed by the resolution of an eigenvalue problem as is explained in

Lemma 5.3.3. *If $\{\tilde{q}_1, \ldots, \tilde{q}_M\}$ and $\{\tilde{w}_1, \ldots, \tilde{w}_M\}$ are orthonormal basis of X_M and W_M respectively, then*

$$\beta_M = 1/\Lambda_M = \sqrt{\lambda_{\min}(A^T A)}, \tag{5.24}$$

where A is the $M \times M$ matrix whose entries are $A_{i,j} = (\tilde{w}_i, \tilde{q}_j)_{\mathcal{X}}$ and $\lambda_{\min}(A^T A)$ denotes the minimum eigenvalue of the positive definite matrix $A^T A$.

Proof. Since

$$\beta_M = \inf_{x \in X_M} \sup_{w \in W_M} \frac{(x, w)_{\mathcal{X}}}{\|x\|_{\mathcal{X}} \|w\|_{\mathcal{X}}} = \inf_{x \in \mathbb{R}^M} \sup_{w \in \mathbb{R}^M} \frac{(Ax, w)_2}{\|x\|_2 \|w\|_2} = \inf_{x \in \mathbb{R}^M} \frac{\|Ax\|_2}{\|x\|_2},$$

the result easily follows because $\dfrac{\|Ax\|_2^2}{\|x\|_2^2}$ is the Rayleigh quotient of $A^T A$ whose infimum is achieved by $\lambda_{\min}(A^T A)$. □

Remark 5.3.4. *Note that β_M corresponds to the minimum singular value of the matrix A, which is a matrix of small size $M \times M$. Its computation can be easily performed by, e.g., the inverse power method.*

Input: Σ, $\mathcal{S}_F = \{f_k \in F\}_{k=1}^{\#\mathcal{S}_F}$, ε_{tol}, $M_{\max}, M = 0$
Assign a set of functions $\{f_{k_{p,start}}, \dots, f_{k_{p,stop}}\}$ to each processor p.
repeat
  $M \leftarrow M + 1$
  $\varepsilon_{p,\max} = 0$ ▷ *parallel*
  for $k = \{k_{p,start}, \dots, k_{p,stop}\}$ **do**
    $f = f_k$
    Compute and store $\sigma_M(f - \mathcal{J}_M(f))$.
    Assemble $\mathcal{J}_{M+1}(f)$ following formula (5.23)
    Compute $\varepsilon_{M+1} = \|f - \mathcal{J}_{M+1}(f)\|_{\mathcal{X}}$
    if $\varepsilon_{M+1} > \varepsilon_{p,\max}$ **then**
      $k_{p,\max} = k$ and $\varepsilon_{p,\max} = \varepsilon_{M+1}$
    end if
  end for ▷ *end parallel*
  Gather $\{(\varepsilon_{p,\max}, k_{p,\max})\}_{p=1}^{N_{proc}}$ and find $(\varepsilon_{\max}, k_{\max}) = \underset{p\in\{1,\dots,N_{proc}\}}{\arg\max} (\varepsilon_{p,\max}, k_{p,\max})$.
  $r_{M+1} = f_{k_{\max}} - \mathcal{J}_M(f_{k_{\max}})$
  $\tilde{\varepsilon}_{p,\max} = 0$ ▷ *parallel*
  for $j = \{j_{p,start}, \dots, j_{p,stop}\}$ **do**
    $\sigma = \sigma_j$
    Compute $\tilde{\varepsilon}_{M+1} = |\sigma(r_{M+1})|$
    if $\tilde{\varepsilon}_{M+1} > \tilde{\varepsilon}_{p,\max}$ **then**
      $j_{p,\max} = j$ and $\tilde{\varepsilon}_{p,\max} = \tilde{\varepsilon}_{M+1}$
    end if
  end for ▷ *end parallel*
  Gather $\{(\tilde{\varepsilon}_{p,\max}, j_{p,\max})\}_{p=1}^{N_{proc}}$ and find $(\tilde{\varepsilon}_{\max}, j_{\max}) = \underset{p\in\{1,\dots,N_{proc}\}}{\arg\max} (\tilde{\varepsilon}_{p,\max}, j_{p,\max})$.
  Compute and store $q_{M+1} = \dfrac{r_{M+1}}{\sigma_{j_{\max}}(r_{M+1})}$.
  Store $\sigma_{M+1} = \sigma_{j_{\max}}$.
  Compute and store w_{M+1} (Riesz representation of σ_{M+1}).
until $\varepsilon_{\max} < \varepsilon_{tol}$ or $M > M_{\max}$
Output: $\{\sigma_1, \dots, \sigma_{M+1}\}$, $W_{M+1} = \text{span}\{w_1, \dots, w_{M+1}\}$, $X_{M+1} = \text{span}\{q_1, \dots, q_{M+1}\}$.

Algorithm 5.1: Practical implementation of the Greedy procedure

5.4 A numerical study about the impact of the dictionary Σ of linear functionals in the Lebesgue constant

As outlined in remark 5.2.5, the explicit expression of the Lebesgue constant presented in formula (5.15) shows that Λ_M is intimately linked to the dictionary of linear functionals Σ that is used in the Greedy algorithm to build the interpolation process. With the exception of the trivial case considered in corollary 5.2.2, no theoretical analysis of the impact of Σ in the behavior of the Lebesgue constant has been possible so far. For this reason, we present here some numerical results on this issue as a first illustration of this connection. The same computations will also let us numerically validate the formula (5.15) for Λ_M, whose original definition is given by (5.7).

We place ourselves in $\Omega = [0, 1]$ and consider the numerical approximation in $L^2(\Omega)$ or $H^1(\Omega)$ of the following compact set:

$$F = \{f(., \mu_1, \mu_2) \mid (\mu_1, \mu_2) \in [0.01, 24.9] \times [0, 15]\}, \tag{5.25}$$

where

$$f(x, \mu_1, \mu_2) = \frac{1}{\sqrt{1 + (25 + \mu_1 \cos(\mu_2 x))x^2}}, \quad \forall x \in \Omega.$$

We remind that $L^2(\Omega) = \{f \mid \|f\|_{L^2(\Omega)} < \infty\}$, where the norm $\|\cdot\|_{L^2(\Omega)}$ is induced by the inner product $(w,v)_{L^2(\Omega)} = \int_\Omega w(x)v(x)dx$. Also, $H^1(\Omega) = \{f \mid \|f\|_{H^1(\Omega)} < \infty\}$, where the norm $\|\cdot\|_{H^1(\Omega)}$ is induced by the inner product $(w,v)_{H^1(\Omega)} = \int_\Omega w(x)v(x)dx + \int_\Omega \nabla w(x).\nabla v(x)dx$.

Any $f \in F$ will be approximated by its generalized interpolant at dimension M. For this purpose, the practical construction of the interpolating space X_M and the selection of the linear functionals is done through the Greedy algorithm described in section 5.3. The following dictionary of linear functionals has been employed:

$$\Sigma = \{\sigma_k \in \mathcal{L}(\mathcal{X}),\ k \in \{1,\dots,N_{sensor}\}\}, \tag{5.26}$$

where $N_{sensor} = 150$, and

$$\sigma_k(\varphi) = \int_{x\in\Omega} c_{k,s}(x)\varphi(x)dx, \quad \forall \varphi \in \mathcal{X}. \tag{5.27}$$

The function $c_{k,s}$ reads:

$$c_{k,s}(x) = \frac{m_{k,s}(x)}{\|m_{k,s}(x)\|_{L^1(\Omega)}}, \quad \forall x \in \Omega,$$

where

$$m_{k,s}(x) := e^{-(x-x_k)^2/(2s^2)}, \quad \forall x \in \Omega$$

and $x_k \in \Omega$. We will explore the variation of the coefficient $s \in \mathbb{R}_+$ in order to understand the influence of the dictionary Σ on Λ_M.

5.4.1 Validation of the inf-sup formula

We will first start by fixing s to a value of 0.005 and by numerically validating formula (5.15) of the Lebesgue constant by comparing it to the value given by the original formula (5.7).

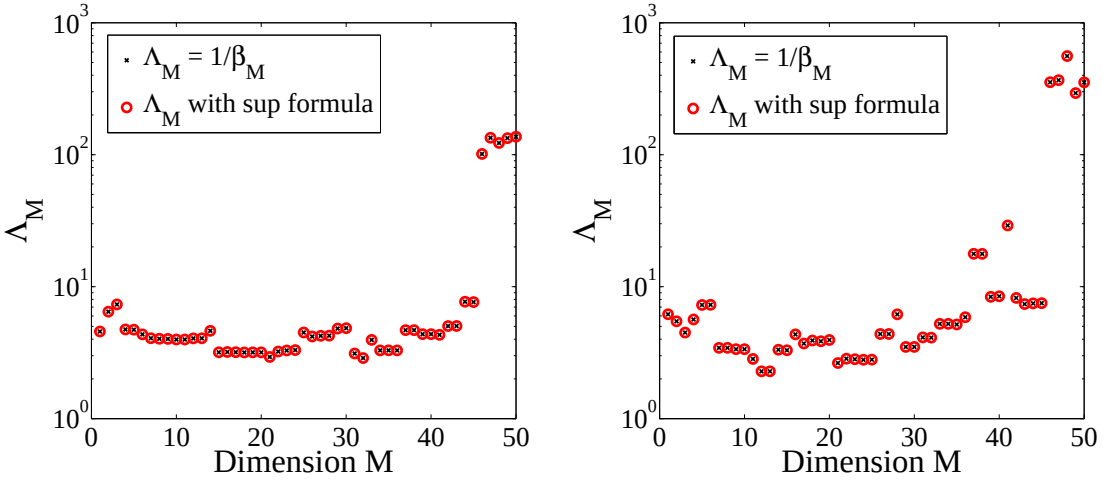
Regarding the computation of (5.15), the quantity β_M has been derived using formula (5.24) of lemma 5.3.3. It suffices to evaluate the scalar products of the matrix A defined in that lemma and obtain the minimum eigenvalue of A^TA. For the practical computations, a $\mathbb{P}_1$ finite element approximation of the functions $\tilde{q}_i$ and $\tilde{w}_i$ has been used in order to simplify the scalar product evaluation in the L^2 and H^1 spaces. For the same reason and as a matter of global coherence, the computation of the original formula of the Lebesgue constant $\sup_{\varphi\in\mathcal{X}} \frac{\|\mathcal{J}_M[\varphi]\|_\mathcal{X}}{\|\varphi\|_\mathcal{X}}$ has also involved the same $\mathbb{P}_1$ finite element approximation of the elements of $\mathcal{X}$. This approach leads to the computation of a discrete Raleigh quotient, whose derivation is explained in detail in appendix B.

The results of the computation are given in figure 5.2 and show an excellent agreement between both values in L^2 and H^1. The same agreement holds for any value of the parameter s of the linear functionals, but, as will be presented in the next section, the behavior of Λ_M varies depending on this parameter.

In the particular case presented here, the behavior of the Lebesgue constant does not significantly change if we place us in L^2 or in H^1 and Λ_M remains constant (the degradation in the behavior for $M \geq 44$ is due to numerical round-off errors).

5.4.2 Impact of the dictionary of linear functionals

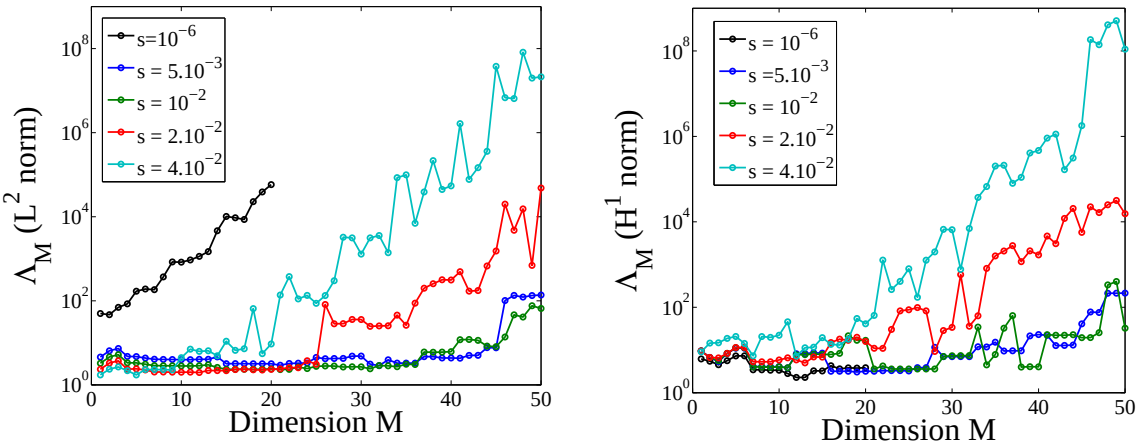
We now study the impact of s on the evolution of the Lebesgue constant through our example in one dimension. For this purpose, we present in figures 5.3a and 5.3b the behavior in L^2 and in H^1 of Λ_M for different values of s.



(a) $\mathcal{X} = L^2([0;1])$ **(b)** $\mathcal{X} = H^1([0;1])$

Figure 5.2: Numerical validation of the inf-sup formula: comparison between formulae (5.7) and (5.15).

To begin with, we will focus on the behavior for sufficiently large values of s and analyze the range $s \geq 5.10^{-3}$. It can be observed that, as s increases, the behavior of the Lebesgue constant is progressively degraded in both norms. The sequence (Λ_M) starts to diverge at dimensions that are lower and lower as s increases (compare, e.g., the behaviors between the case $s = 2.10^{-2}$ and $s = 4.10^{-2}$). An intuitive manner to interpret this observation is as follows: the dictionary under consideration in this example (see formula (5.26)) consists on local averages operations whose "range" is controlled by s. As s increases, the range increases and a limit will be reached in which the addition of more linear functionals will result in a redundant addition of information because of an overlap of the domains where the local averages are acting. As a result, the larger s, the sooner this redundancy will appear and the more unstable the process.



(a) L^2 norm. **(b)** H^1 norm.

Figure 5.3: Impact on Λ_M of the parameter s of the linear functionals.

It is also important to understand the behavior when the parameter s tends to zero. In this case, the linear functionals tend to Dirac masses, that are elements of H^{-1} but not of L^2. Hence,

in the limit $s = 0$, the definition of the space W_M will be possible in H^1 but not in L^2 because the problem:

$$\begin{cases} \text{Find } w_i \in \mathcal{X} \text{ such that:} \\ \sigma_i(\varphi) = (w_i, \varphi)_{\mathcal{X}} = \delta_{x_i}(\varphi), \quad \forall \varphi \in \mathcal{X} \end{cases} \tag{5.28}$$

is well-defined in H^1 and not in L^2. This observation helps to understand first of all why Λ_1 remains roughly constant in H^1 as s decreases whereas it behaves as $s^{-1/2}$ in the L^2 norm (see figure 5.4). Indeed, in the H^1 case, we have the inequality

$$\frac{\|\mathcal{J}_1[\varphi]\|_{H^1(\Omega)}}{\|\varphi\|_{H^1(\Omega)}} = |\sigma_1(\varphi)| \frac{\|q_1\|_{H^1(\Omega)}}{\|\varphi\|_{H^1(\Omega)}} \leq \|\varphi\|_{L^\infty(\Omega)} \frac{\|q_1\|_{H^1(\Omega)}}{\|\varphi\|_{H^1(\Omega)}}, \quad \forall \varphi \in H^1(\Omega),$$

which is bounded for any $s \in \mathbb{R}_+$. However, in the case of $L^2(\Omega)$, it can be inferred that

$$\frac{\|\mathcal{J}_1[\varphi]\|_{L^2(\Omega)}}{\|\varphi\|_{L^2(\Omega)}} = |\sigma_1(\varphi)| \frac{\|q_1\|_{L^2(\Omega)}}{\|\varphi\|_{L^2(\Omega)}} \leq \frac{\|m_{1,s}\|_{L^2(\Omega)}}{\|m_{1,s}\|_{L^1(\Omega)}} \|q_1\|_{L^2(\Omega)}, \quad \forall \varphi \in H^1(\Omega)$$

where we have applied the Cauchy-Schwarz inequality to $|\sigma_1(\varphi)|$. A simple change of variable $u = \dfrac{x - x_1}{s}$ in the evaluation of $\dfrac{\|m_{1,s}\|_{L^2(\Omega)}}{\|m_{1,s}\|_{L^1(\Omega)}}$ leads to the bound

$$\frac{\|\mathcal{J}_1[\varphi]\|_{L^2(\Omega)}}{\|\varphi\|_{L^2(\Omega)}} \leq C \|q_1\|_{L^2(\Omega)} s^{-1/2}, \quad \forall \varphi \in L^2(\Omega), \tag{5.29}$$

where

$$C = \frac{\int\limits_{\Omega} e^{-u^2} du}{\int\limits_{\Omega} e^{-u^2/2} du}.$$

In figure 5.4, note that for values $s \leq 10^{-4}$, the behavior of Λ_1 no longer follows $s^{-1/2}$ but this is due to computer limitations. Indeed, the computations have been carried out with a maximum number of 10^4 degrees of freedom in the $\mathbb{P}_1$ approximation because of memory storage issues. As a result, for $s \leq 10^{-4}$, we no longer capture enough information with this finite element precision.

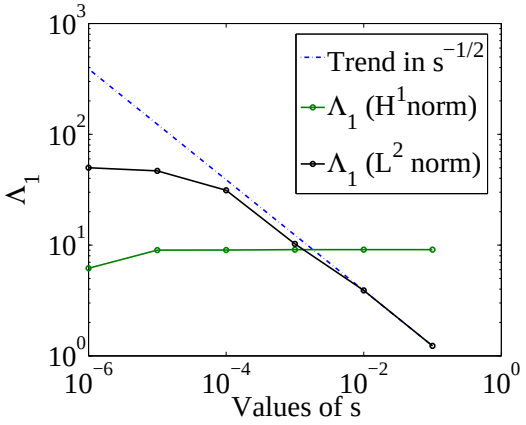


Figure 5.4: Behavior of Λ_1 as a function of s (H^1 and L^2 norms). Remark: the scale of the figure is log-log.

As a consequence of the diverging behavior of Λ_1 in L^2 as the parameter s decreases, it is reasonable to expect that the sequence (Λ_M) quickly diverges as $s \to 0$ in L^2 but that it remains

bounded in H^1. This behavior is indeed illustrated in figures 5.3a and 5.3b through the example of $s = 10^{-6}$, in which it is possible to observe the phenomenon.

5.5 Application of GEIM to the real-time monitoring of a physical experiment

The main purpose of this section is to illustrate that GEIM can be used as a tool for the real-time monitoring of a physical or industrial process. This will be done by combining mathematical models (a parameter dependent PDE) with measurements from the experiment.

5.5.1 The general method

Let us assume that we want to monitor in real time a field u_{true} appearing as an input for some quantities of interest in a given experiment that involves sensor measurements. We assume that the conditions of the experiment are described by a vector of parameters $\mu_{\text{true}} \in E$, where E is a compact set of $\mathbb{R}^p$, and that u_{true} is the solution of a parameter dependent PDE

$$D_\mu u = g_\mu \quad \mu \in E \tag{5.30}$$

when $\mu = \mu_{\text{true}}$ (in other words $u_{\text{true}} = u_{\mu_{\text{true}}}$). The vector μ_{true} will be unknown in general so the computation of u_{true} cannot be done by traditional discretization techniques like finite elements. Besides, even if μ_{true} was known, its computation could not be performed in real-time with classical techniques. For all these reasons, we propose to compute the generalized interpolant $\mathcal{J}_M[u_{\text{true}}]$ as an approximation of u_{true} that can be derived in real time and that does not sacrifice much on the accuracy of the approximation.

Such an approximation requires that the set of solutions $\{u_\mu, \forall \mu \in E\}$ is included in some compact set F of $\mathcal{X}$ that is of small Kolmogorov n-width in $\mathcal{X}$ ([27]). A dictionary $\Sigma \subset \mathcal{L}(\mathcal{X})$ is also required, but note that the sensors of the experiment can mathematically be modelled by elements of $\mathcal{L}(\mathcal{X})$. We will therefore assume that we have a dictionary composed of the linear functionals representing each one of the sensors.

Since we need to define the generalized interpolating spaces $X_M = \text{span}\{q_1, \dots, q_M\}$ together with the suitable interpolating linear functionals $\{\sigma_1, \dots, \sigma_M\}$, a greedy algorithm has to be performed beforehand and therefore the computation of $\mathcal{J}_M[u_{\text{true}}]$ is divided into two steps:

- In an *offline phase* (i.e. before the experiment takes place):
  - We define a finite subset $\mathcal{S}_F = \{u(\mu, .), \mu \in \Xi_\mu \subset E\} \subset F$ and solve (5.30) for each element of $\mathcal{S}_F$ with an accurate enough discretization strategy. This can be done with traditional approximation tools like, e.g., finite elements or a reduced basis strategy.
  - Following the steps of algorithm 5.1, a greedy algorithm over the set $\mathcal{S}_F$ is performed to build an M-dimensional reduced basis $X_M = \text{span}\{q_j \in F, j \in [1, M]\}$ together with the suitable linear functionals $\{\sigma_1, \dots, \sigma_M\}$. The selection of the linear functionals means that, among all the sensors in the experiment that constitute our dictionary Σ, we select the M most suitable according to the greedy criterion.
- In an *online phase* (i.e. when the experiment is running), we collect in real time the measurements

  $$\{\sigma_1(u_{\mu_{\text{true}}}), \dots, \sigma_M(u_{\mu_{\text{true}}})\}$$

  from the M selected sensors. The generalized interpolant $\mathcal{J}_M[u_{\mu_{\text{true}}}]$ can then be computed following formula (5.3). It has been observed so far (see the numerical example below and [76]) that the interpolation error decreases very quickly as the dimension M increases and therefore relatively small values of M are required to reach a good accuracy in the approximation of

$u_{\mu_{\text{true}}}$ by $\mathcal{J}_M[u_{\mu_{\text{true}}}]$. Thanks to this, the computation of $\mathcal{J}_M[u_{\mu_{\text{true}}}]$ can be performed in real-time (or almost).

Remark 5.5.1. *Note that our strategy supposes that the physical experiment u_{true} is perfectly described by the solution u_μ of (5.30) when $\mu = \mu_{true}$. This is a very strong hypothesis because the model might not perfectly describe the experiment under consideration. Besides, it is here assumed that there is no noise in the measurements, which is also a strong assumption. In [76], some preliminary analysis has been presented to take into account the presence of noise in the measurements. Regarding the model bias, in the recent works of [81, 122], the authors are able to take it into account under several hypothesis in the so called "Parametrized-Background Data-Weak Formulation" for variational data assimilation. In fact, GEIM is a particular instance of this method for the case (with the notations of [81]) $N = M$ and this latter choice is appropriate for situations in which the bias is small.*

Remark 5.5.2. *In the strategy proposed in this section, sensor measurements are incorporated in the interpolation procedure through the space W_M (which is spanned by the Riesz representations of the linear functionals of the sensors). In the reference [16], one can find an early work in oceanography in which data assimilation is also incorporated through the construction of the space W_M. However, in the case of [16], no a priori error analysis was provided in the computational procedure that was proposed.*

5.5.2 A numerical application involving the Stokes equation

We are going to illustrate the procedure in the case where the experiment corresponds to a lid-driven cavity problem that takes place in the spatial domain $\Omega = [0;1] \times [0;1] \subset \mathbb{R}^2$. We consider two parameters $\boldsymbol{\mu} = (\mu_1, \mu_2) \in [1;8] \times [1;8]$ such that, for a given μ, the parametrized PDE reads:

$$\begin{cases} \text{Find the solution } (\boldsymbol{u}_\mu, p_\mu) \in \left(H^1(\Omega)\right)^2 \times L^2(\Omega) \text{ of :} \\ -\Delta \boldsymbol{u}_\mu + \mathbf{grad}(p_\mu) = \mathbf{f}_\mu, \text{ in } \Omega \\ \text{div}(\mathbf{u}_\mu) = 0, \text{ in } \Omega \\ \mathbf{u}_\mu = \begin{pmatrix} x(1-x) \\ 0 \end{pmatrix}, \text{ on } \Gamma_1 \\ \mathbf{u}_\mu = \mathbf{0}, \text{ on } \partial\Omega \setminus \Gamma_1 \end{cases} \tag{5.31}$$

where the forcing term $\mathbf{f}_\mu = \begin{pmatrix} 100\sin(\mu_1 \Pi y) \\ -100\sin\left(\mu_2 \Pi \frac{1-x}{2}\right) \end{pmatrix}$ and $\Gamma_1 = \{x \in [0;1],\ y = 1\}$. Two examples of solutions are provided on figures 5.5 and 5.6.

We assume that:

- The set of solutions $\{(\mathbf{u}, p)(\mu),\ \forall \mu\} \subset F$ and F is of small Kolmogorov n-width in $\left(H^1(\Omega)\right)^2 \times L^2(\Omega)$. This assumption is made *a priori* and will be verified *a posteriori* in a convergence study of the interpolation errors.
- we have velocity and pressure sensors at our disposal which mathematically means that we have:
  - a dictionary for the velocity: $\Sigma^{\mathbf{u}} = \{\sigma^{\mathbf{u}}\} \subset \mathcal{L}(H^1(\Omega)^2)$
  - a dictionary for the pressure: $\Sigma^p = \{\sigma^p\} \subset \mathcal{L}(L^2(\Omega))$

In our numerical example, the linear functionals that have been used consist on local averages of the same form as (5.26) and (5.27) but adapted to the 2D case. The parameter s has been fixed to $s = 10^{-3}$ and we will have $N_{sensor} = 100$ sensors for the pressure and other $N_{sensor} = 100$ sensors for the velocity. The centers of these local averages are located on a 10×10 equispaced grid of Ω.

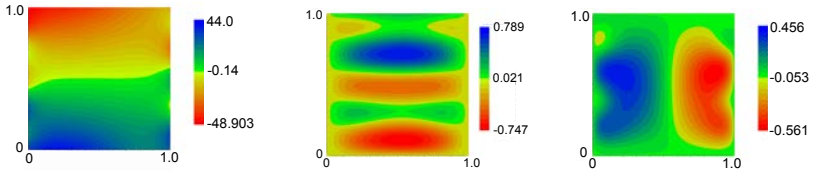


Figure 5.5: From left to right: pressure, horizontal and vertical velocity solutions for the parameter $\boldsymbol{\mu} = (5; 1)$.

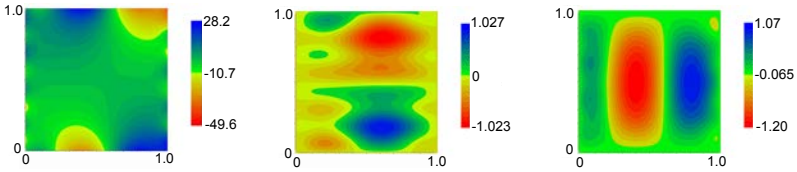


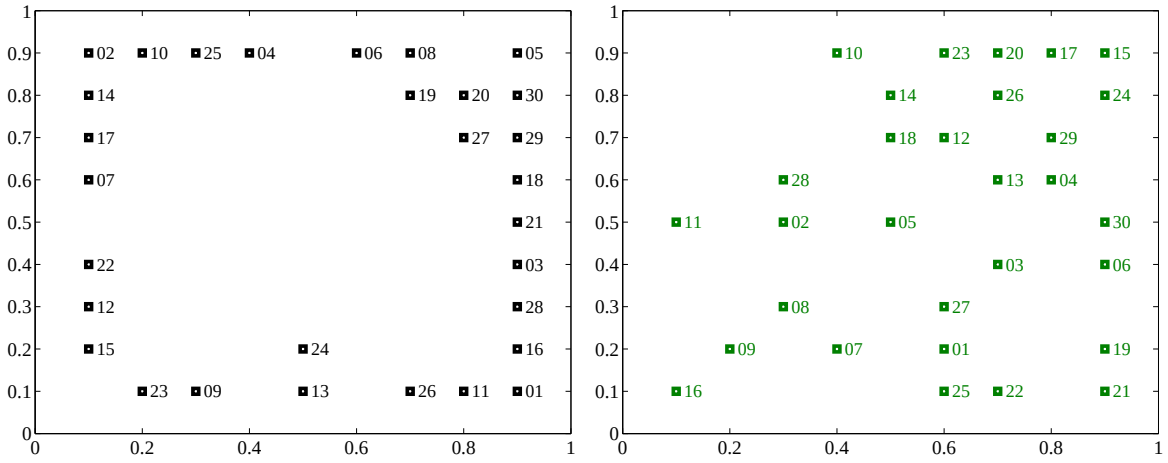
Figure 5.6: From left to right: pressure, horizontal and vertical velocity solutions for the parameter $\boldsymbol{\mu} = (8; 5)$.

Given an experiment corresponding to the vector of parameters μ_{exp}, we are going to — quickly and accurately— approximate in Ω the vectorial field $(\mathbf{u}, p)(\mu_{\text{exp}})$ by its generalized interpolant $\mathcal{J}_M\left[(\mathbf{u}, p)(\mu_{\text{exp}})\right]$ thanks to the only knowledge of measurements from sensors. Because we are facing here the reconstruction of a vectorial field, several potential input from $(\mathbf{u}, p)(\mu)$ can be proposed. In the present paper, three classes of them will be considered. They will all fulfill the divergence-free condition for the velocity interpolant $\text{div}\left(\mathcal{J}_M\left[\mathbf{u}(\mu)\right]\right) = 0$.

Reconstruction 1: Independent treatment of $u(\mu)$ and $p(\mu)$.
The first possibility consists in considering $(\mathbf{u}, p)(\mu)$ not as a vectorial field but as two independent fields $\mathbf{u}(\mu)$ and $p(\mu)$ to interpolate independently with velocity measurements for $\mathbf{u}(\mu)$ and pressure measurements for $p(\mu)$. In other words, the generalized interpolant is defined in this case as $\mathcal{J}_{M_\mathbf{u},M_p}\left[(\mathbf{u}, p)(\mu)\right] = \left(\mathcal{J}^{\mathbf{u}}_{M_\mathbf{u}}[\mathbf{u}(\mu)]; \mathcal{J}^{p}_{M_p}[p(\mu)]\right)$. This requires the offline computation of two greedy algorithms: one for the velocity and another for the pressure. Each one respectively provides:

- a velocity basis $\{\mathbf{u}(\mu_i)\}_{i=1}^{M_\mathbf{u}}$ and a set of $M_\mathbf{u}$ velocity sensors $\{\sigma^{\mathbf{u}}_i\}_{i=1}^{M_\mathbf{u}}$ chosen among the dictionary $\Sigma^{\mathbf{u}}$. The interpolant for the velocity will be $\mathcal{J}^{\mathbf{u}}_{M_\mathbf{u}}[\mathbf{u}(\mu)] = \sum_{i=1}^{M_\mathbf{u}} \alpha_i \mathbf{u}(\mu_i)$ where the α_i are given by the interpolating conditions $\sigma^{\mathbf{u}}_i\Big(\mathcal{J}^{\mathbf{u}}_{M_\mathbf{u}}[\mathbf{u}(\mu)]\Big) = \sigma^{\mathbf{u}}_i\Big(\mathbf{u}(\mu)\Big),\ \forall i \in \{1, \ldots, M_\mathbf{u}\}$.
- a pressure basis $\{p(\mu_j)\}_{j=1}^{M_p}$ and a set of pressure sensors $\{\sigma^{p}_j\}_{j=1}^{M_p}$ chosen among Σ^p. The interpolant for the pressure will be $\mathcal{J}^{p}_{M}[p(\mu)] = \sum_{j=1}^{M_p} \gamma_j p(\mu_j)$ where the γ_j are given by the interpolating conditions $\sigma^{p}_j\Big(\mathcal{J}^{p}_{M_p}[p(\mu)]\Big) = \sigma^{p}_j\Big(p(\mu)\Big),\ \forall j \in \{1, \ldots, M_p\}$.

Note that in this approximation, the construction of $\mathcal{J}_{M_\mathbf{u},M_p}[(\mathbf{u}, p)(\mu)]$ involves M_p pressure sensors and $M_\mathbf{u}$ velocity sensors, i.e. $M_p + M_\mathbf{u}$ coefficients. In figure 5.7, we have represented the locations of the sensors in the order given by the greedy algorithm.



(a) Pressure sensors **(b)** Velocity sensors

Figure 5.7: Locations of the sensors for reconstruction 1.

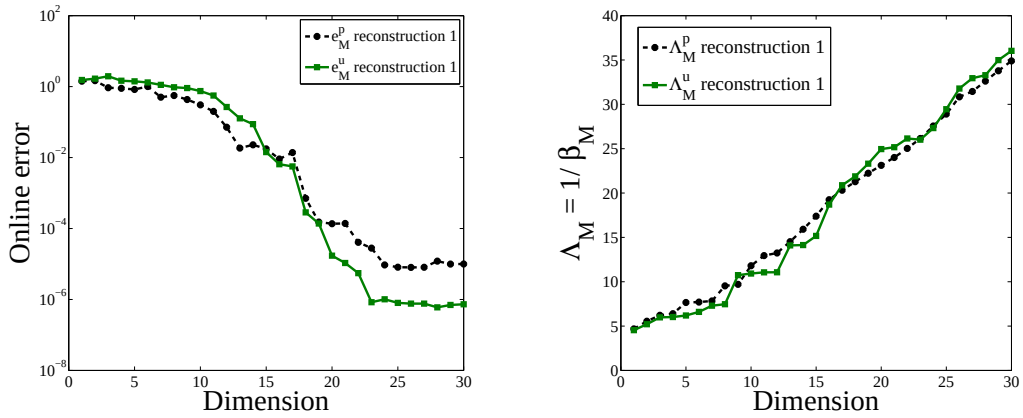


Figure 5.8: Reconstruction 1: A numerical estimation of the behavior of the interpolation error (left) and the Lebesgue constant (right) as a function of the dimension of the interpolating spaces X_M

The performances of the method are plotted in figure 5.8 where a numerical estimation of the behavior of the interpolating errors for the reconstruction of u and p have been represented. These values have been obtained by the interpolation of 196 configurations coming from different parameter values μ_i following formula:

$$\begin{cases} e^p_{M_p} = \max\limits_{i\in\{1,\dots,196\}} \dfrac{\|p(\mu_i) - \mathcal{J}^p_{M_p}[p(\mu_i)]\|_{L^2(\Omega)}}{\|p(\mu_i)\|_{L^2(\Omega)}} \\ e^{\mathbf{u}}_{M_{\mathbf{u}}} = \max\limits_{i\in\{1,\dots,196\}} \dfrac{\|\mathbf{u}(\mu_i) - \mathcal{J}^{\mathbf{u}}_{M_{\mathbf{u}}}[\mathbf{u}(\mu_i)]\|_{H^1(\Omega)^2}}{\|\mathbf{u}(\mu_i)\|_{H^1(\Omega)^2}}. \end{cases} \tag{5.32}$$

In this figure, we can observe the convergence of the interpolation errors for both the velocity and pressure fields. After a preasymptotic stage for interpolating spaces of small dimension, an exponential convergence of the error is observed. After about dimension $M = 25$, the error stagnates

due to the fact that we have reached the finite element accuracy used for the computation of the offline snapshots. The computation of the Lebesgue constants

$$\Lambda^p_{M_p} := \sup_{p\in L^2(\Omega)} \frac{\|\mathcal{J}^p_{M_p}(p)\|_{L^2(\Omega)}}{\|p\|_{L^2(\Omega)}}, \quad \Lambda^{\mathbf{u}}_{M_{\mathbf{u}}} = \sup_{\boldsymbol{u}\in H^1(\Omega)^2} \frac{\|\mathcal{J}^{\mathbf{u}}_{M_{\mathbf{u}}}(\mathbf{u})\|_{H^1(\Omega)^2}}{\|\mathbf{u}\|_{H^1(\Omega)^2}} \tag{5.33}$$

has also been performed following formula (5.15). Its behavior seems linear with the dimension of interpolation and is therefore far from the crude theoretical upper bound given in formula (5.8). From the results presented in section 5.4, an idea to improve the behavior of the Lebesgue constant could be to consider a smaller value for s. However, in the present context, we have not sought the optimization of (Λ_M) as a function of the parameter s because, in a real case, the linear functionals are fixed by the filter characteristics of the sensors involved in the experiment.

Reconstructions 2 and 3: Vectorial treatment for $\mathbf{u}(\mu)$ and $p(\mu)$.
An alternative to the first reconstruction is to consider $(\mathbf{u},p)(\mu)$ as a vectorial field and define its generalized interpolant as $\mathcal{J}_M[(\mathbf{u},p)(\mu)] := \sum_{i=1}^{M} \gamma_i(\mathbf{u},p)(\mu_i)$, where now only M coefficients γ_i are involved. The joint basis $\{(\mathbf{u},p)(\mu_i)\}_{i=1}^M$ is provided by a greedy algorithm in the online stage together with a set of M linear functionals $\{\sigma_i^{(\mathbf{u},p)}\}_{i=1}^M$. Each of these linear functionals involve pressure and velocity measurements at a given spatial location and are defined as $\sigma_i^{(\mathbf{u},p)} := \sigma_i^{\mathbf{u}}(\mathbf{u}) + \sigma_i^p(p)$. The interpolating conditions for the inference of the coefficients γ_i are now the following:

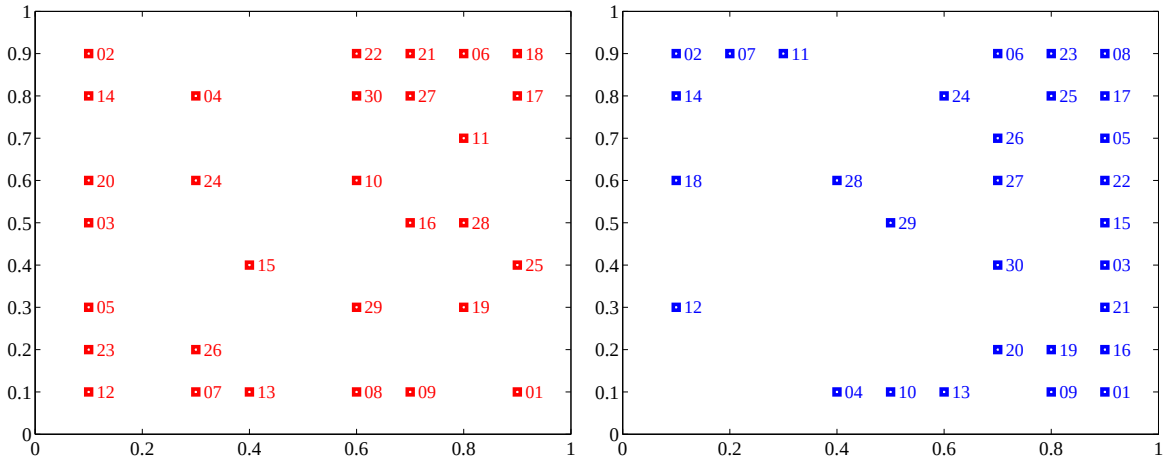
$$\sigma_i^{(\mathbf{u},p)}\Big((\mathbf{u},p)(\mu)\Big) = \sigma_i^{(\mathbf{u},p)}\Big(\mathcal{J}_M[(\mathbf{u},p)(\mu)]\Big) = \sum_{j=1}^{M}\gamma_j\sigma_i^{(\mathbf{u},p)}\Big((\mathbf{u},p)(\mu_j)\Big),\ \forall i\in\{1,\dots,M\}, \tag{5.34}$$

Notice that this definition of the linear functionals $\sigma^{(\mathbf{u},p)}$ can involve both velocity and pressure measurements or can take into account velocity or pressure measurements only by setting $\sigma^{\mathbf{u}} = 0$ or $\sigma^p = 0$. We have explored this flexibility in the following two reconstructions where we have compared:

- the interpolation of the pressure and velocity fields with pressure and velocity measurements: $\sigma_i^{(\mathbf{u},p)} := \sigma_i^{\mathbf{u}}(\mathbf{u}) + \sigma_i^p(p)$ (reconstruction 2).
- the interpolation of the pressure and velocity fields with pressure measurements only: $\sigma_i^{(\mathbf{u},p)} := \sigma_i^p(p)$. In other words, we are here studying if a velocity field can efficiently be reconstructed with the only knowledge of pressure measurements (reconstruction 3).

The sensor locations provided by the greedy algorithm are shown in figure and 5.9 and the results are summarized in figures 5.10 where an estimation of the interpolation error is plotted according to formula (5.35).

$$e_M^{(\mathbf{u},p)} = \max_{i\in\{1,\dots,196\}} \frac{\|(\mathbf{u},p)(\mu_i) - \mathcal{J}_M^{(\mathbf{u},p)}[(\mathbf{u},p)(\mu_i)]\|_{H^1(\Omega)^2\times L^2(\Omega)}}{\|(\mathbf{u},p)(\mu_i)\|_{H^1(\Omega)^2\times L^2(\Omega)}}. \tag{5.35}$$



(a) Reconstruction 2 **(b)** Reconstruction 3

Figure 5.9: Locations of the sensors for reconstructions 2 and 3.

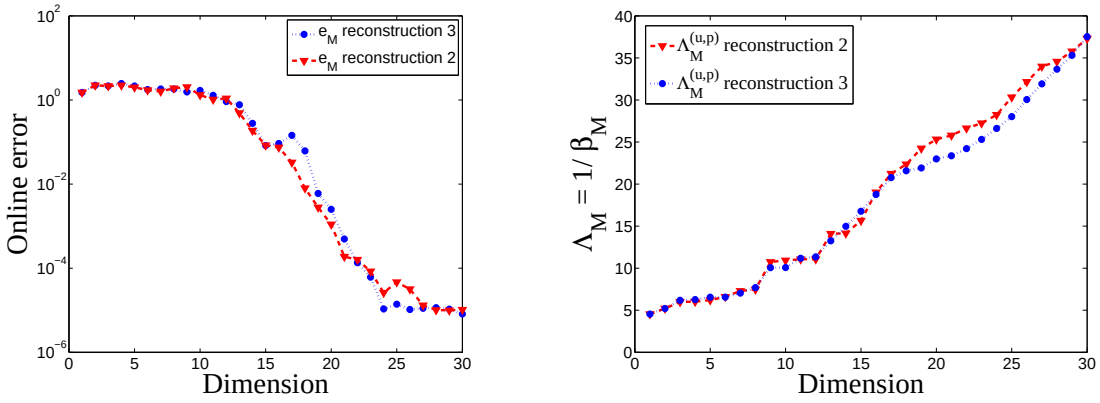


Figure 5.10: Reconstructions 2 and 3: A numerical estimation of the behavior of the interpolation error (left) and the Lebesgue constant (right) as a function of the dimension of the interpolating spaces X_M.

The interpolating error of the two types of reconstructions presents a very similar decay behavior in both cases and the convergence is also very similar to reconstruction 1. The most interesting consequence of this is that the velocity can efficiently be reconstructed with only pressure measurements. This result cannot probably be generalized to all types of situations but it proves that in some cases like the current one there is some redundancy in the datas and that, in this precise problem, there is no need in having velocity measurements in order to obtain a good accuracy in the approximation of the velocity field.

The Lebesgue constant

$$\Lambda_M^{(\boldsymbol{u},p)} = \sup_{(\boldsymbol{u},p)\in H^1(\Omega)^2\times L^2(\Omega)} \frac{\|\mathcal{J}_M^{(\mathbf{u},p)}(\mathbf{u},p)\|_{H^1(\Omega)^2\times L^2(\Omega)}}{\|(\mathbf{u},p)\|_{H^1(\Omega)^2\times L^2(\Omega)}} \tag{5.36}$$

has also been computed for reconstructions 2 and 3 as is shown in figure 5.10. Once again, the behavior is linear which is a moderate growth rate.

5.6 Conclusion and perspectives

After revisiting the foundations of GEIM for Banach spaces, the present work has focused on understanding the stability of the process and a relation between Λ_M and an inf-sup problem has been established in the particular case of Hilbert spaces. An interpretation of the generalized interpolant as an oblique projection has also been presented in that case. The derived formula for Λ_M has also allowed us to notice that the Greedy algorithm optimizes in some sense the Lebesgue constant.

A first analysis about the impact of the dictionary of linear functionals Σ on the Lebesgue constant has also been presented through a numerical test case. Furthermore, for a given dictionary Σ, the Lebesgue constant depends on the norm of the ambient space $\mathcal{X}$ (see formula (5.7)). A comparison of the behavior of (Λ_M) when $\mathcal{X} = L^2$ or H^1 has been provided in the case of a dictionary composed of simple local averages.

Beyond these results, there are still plenty of challenging theoretical open questions. Among the most important we mention:

- the obtention (if possible) of a general theory on the impact of Σ on the behavior of (Λ_M) and of a tighter upper bound than the one presented in (5.8).
- When the number of involved parameters is very large, how to deal with the offline phase in a reasonable time?
- How to include the bias between u_{true} and the manifold of solutions of our parameter dependent PDE? The works of [122] will probably be helpful to carry out this task.
- How to deal with noisy measurements? One can find some preliminary ideas in [76] and the works of [98].

Furthermore, the recent results of [27] lead us to think that it would be interesting to explore non-linear inputs of the form

$$\sigma\left(t(\varphi)\right),$$

where $\sigma \in \mathcal{L}(\mathcal{X})$, $t : \mathcal{X} \to \mathcal{X}$ is a non linear mapping and φ is an element of a compact set of small Kolmogorov n-width in $\mathcal{X}$. In an ongoing work, we are exploring this idea in the case of the Navier-Stokes equations.

On a second part of the paper, we have illustrated one of the most straightforward practical applications of GEIM that consists in monitoring in real-time a process. The idea is that GEIM could reconstruct in real-time physical quantities in the whole domain of an experiment by combining the real-time acquisition of measurements from sensors with mathematical models (parameter dependent PDE's).

This scheme has been applied to an example dealing with a parametrized lid-driven Stokes equation. The example shows a fast decrease in the interpolation error, which confirms that it is feasible to use GEIM to monitor experiments in real-time in cases where $d_n(F, \mathcal{X})$ is small enough (i.e. when the experiment is simple enough). The behavior of the Lebesgue constant seems to be linear and seems to be in accordance with previous works for the classical EIM (see [80]). The linear increase is far from the theoretical exponential upper bound of (5.8) and suggests that the bound might not be optimal in sets F of small Kolmogorov n-width. In the example, two types of sensors have been used (of pressure and velocity) and the idea of introducing different types of sensors could be extended to make more adequate distinctions among them.

By taking this method as a starting point, GEIM could be used to devise a more complete tool capable of supervising the safety of processes (see figure 5.11). The idea would be the following:

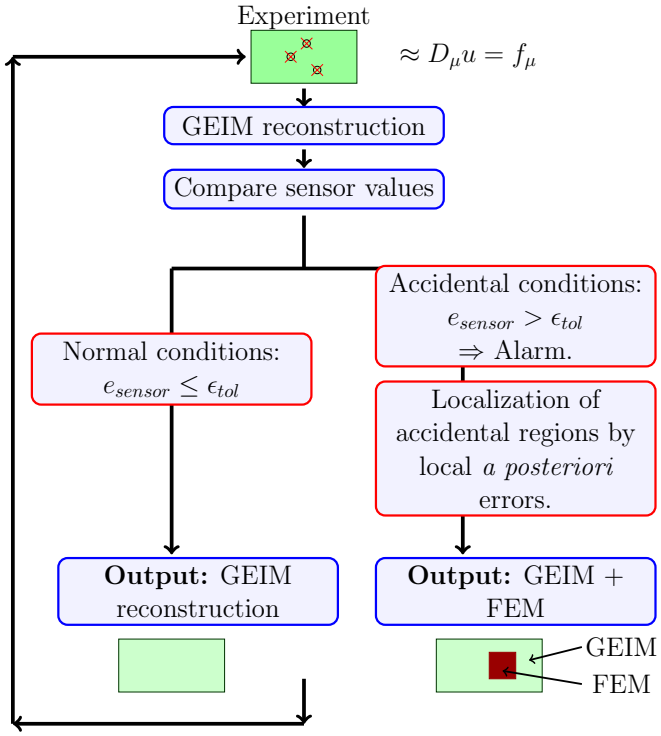


Figure 5.11: A tool to supervise in real-time the safety of an experiment.

given an experiment, we start by reconstructing it by GEIM. Let us assume that we have, e.g., $2M$ sensors at our disposal but that GEIM only needs the information of M of them to provide the reconstruction with the desired accuracy. We can then numerically compute the output of the rest of the sensors by using the generalized interpolant and compare this to the values coming from the experiment. If the values differ too much from each other, then we consider that an abnormal event has occurred in the experiment and an alarm can be launched to inform of the incident.

Further than this alarm information, we can seek to provide an accurate enough reconstruction of the solution during the incident by using the following strategy: through the computation of an posteriori error estimator in the regions where the sensor measurements are not in accordance, we could imagine to localize the spatial region(s) where the reconstruction is no longer accurate. The domain could then be split into:

- a subdomain with small Kolmogorov n-width where the reconstruction by GEIM is still accurate enough.
- a subdomain with big Kolmogorov n-width where the accident is located and GEIM is no longer accurate. The domain is computed by traditional discretization techniques such as finite elements complemented with Dirichlet boundary conditions from the GEIM reconstruction.

Under the hypothesis that the accidental subdomain is small, the reconstruction could still be done in a relatively quick time, preserving the real-time aspect of our device. The feasibility of decomposing the domain and coupling GEIM with other approximations has been explored in [76] in a simple Laplace problem.

Last but not least, it would also be interesting to explore the robustness of the method in cases where one or several sensors involved in the GEIM reconstruction fail.

Appendix A

Corollary 5.6.1. *Let $\mathcal{X}$ be a Hilbert space and E, F two subspaces of $\mathcal{X}$. Then, $\beta_{E,F} = \beta_{F^\perp,E^\perp}$, where:*

$$\beta_{E,F} \equiv \inf_{\substack{e\in E\\ \|e\|=1}} \sup_{\substack{f\in F\\ \|f\|=1}} , (e,f) \tag{5.37}$$

$$\beta_{F^\perp,E^\perp} \equiv \inf_{\substack{f\in F^\perp\\ \|f\|=1}} \sup_{\substack{e\in E^\perp\\ \|e\|=1}} (e,f). \tag{5.38}$$

Proof. Given $e \in \mathcal{X}$ of norm unity, we introduce f_e^* as

$$f_e^* = \arg \sup_{\substack{g\in F\\ \|g\|=1}} (e,g).$$

We can then show from optimality that $(e,h) = 0$ for all h in $\{q \in F \,|\, (q,f_e^*) = 0\}$ and hence

$$e = \lambda f_e^* + \varepsilon \tag{5.39}$$

for some $\lambda \in \mathcal{R}$ and $\varepsilon \in F^\perp$ such that $\lambda^2 + \|\varepsilon\|^2 = 1$ (from our normalization and orthogonality). We then deduce from (5.39), orthogonality, and Cauchy-Schwarz that

$$\sup_{\substack{p\in F\\ \|p\|=1}} (e,p) = \lambda$$

and

$$\sup_{\substack{p\in F^\perp\\ \|p\|=1}} (e,p) = \|\varepsilon\|.$$

Hence,

$$\Big(\sup_{\substack{p\in F\\ \|p\|=1}} (e,p) \Big)^2 + \Big(\sup_{\substack{p\in F^\perp\\ \|p\|=1}} (e,p) \Big)^2 = 1 \tag{5.40}$$

thanks to our normalization.

We may now note from (5.37) and (5.40) that

$$\begin{aligned}
\beta_{E,F} &= \inf_{\substack{e\in E\\ \|e\|=1}} \sqrt{1 - \Big(\sup_{\substack{p\in F^\perp\\ \|p\|=1}} (e,p) \Big)^2} \\
&= \sqrt{1 - \Big(\sup_{\substack{e\in E\\ \|e\|=1}} \sup_{\substack{p\in F^\perp\\ \|p\|=1}} (e,p) \Big)^2} \\
&= \sqrt{1 - \Big(\sup_{\substack{p\in F^\perp\\ \|p\|=1}} \sup_{\substack{e\in E\\ \|e\|=1}} (e,p) \Big)^2}
\end{aligned} \tag{5.41}$$

as we can exchange the two supremizer operations.

Finally, we define a second inf-sup constant,

$$\beta_{F^\perp,E^\perp} \equiv \inf_{\substack{f\in F^\perp\\ \|f\|=1}} \sup_{\substack{e\in E^\perp\\ \|e\|=1}} (e,f). \tag{5.42}$$

We can repeat the procedure above — E goes to $F^\perp$ and F goes to $E^\perp$ — to find

$$\beta_{F^\perp,E^\perp} = \sqrt{1 - (\sup_{\substack{p\in F^\perp \\ \|p\|=1}} \sup_{\substack{e\in (E^\perp)^\perp \\ \|e\|=1}} (e,p)\ )^2}, \tag{5.43}$$

and hence conclude from (5.41) and (5.43) that

$$\beta_{E,F} = \beta_{F^\perp,E^\perp}$$

since $(E^\perp)^\perp = E$. □

Appendix B

We propose here a practical method for the computation of

$$\sup_{\varphi\in\mathcal{X}} \frac{\|\mathcal{J}_M[\varphi]\|_{\mathcal{X}}}{\|\varphi\|_{\mathcal{X}}}. \tag{5.44}$$

The strategy consists in using a finite element Galerkin projection as an approximation of the elements of $\mathcal{X}$. We therefore propose to compute

$$\max_{\varphi\in V_h^k} \frac{\|\mathcal{J}_M[\varphi]\|_{V_h^k}}{\|\varphi\|_{V_h^k}}$$

as a surrogate of (5.44), where V_h^k is the classical continuous finite element approximating space of mesh size h that involves piece-wise $\mathbb{P}_k$ polynomials. Let $\mathcal{B} = \text{span}\{b_1,\dots,b_{\mathcal{N}}\}$ be a basis of V_h^k and let M be the $\mathcal{N}\times\mathcal{N}$ mass matrix of entries $M_{i,j} = (b_i,b_j)_{\mathcal{X}}$, $1\le i,j\le\mathcal{N}$. For any $\varphi\in V_h^k$, let

$$\boldsymbol{\varphi} = (\varphi_1,\dots,\varphi_{\mathcal{N}})^T \tag{5.45}$$

be the vector of coordinates of φ in the basis $\mathcal{B}$. In coherence with these notations, for any $1\le i\le M$, the vectors

$$\boldsymbol{q_i} = (q_{1,i},\dots,q_{\mathcal{N},i})^T \quad \text{and} \quad \boldsymbol{w_i} = (w_{1,i},\dots,w_{\mathcal{N},i})^T \tag{5.46}$$

will respectively denote the Galerkin projections onto V_h^k of the interpolating basis functions $q_i\in\mathcal{X}$ and of the Riesz representation of the i-th linear functional, σ_i. Furthermore, let Q^M be the $\mathcal{N}\times M$ matrix such that

$$Q^M = [\boldsymbol{q_1},\dots,\boldsymbol{q_M}],$$

and let C^M be the $M\times\mathcal{N}$ matrix such that:

$$C^M_{i,j} = \sigma_i(b_j) = (w_i,b_j)_{\mathcal{X}}, \quad \forall 1\le i\le M,\ 1\le j\le\mathcal{N}.$$

Finally, we recall that B^M is the $M\times M$ matrix defined in section 5.1 whose entries are

$$B^M_{i,j} = \sigma_i(q_j) = (w_i,q_j)_{\mathcal{X}}, \quad \forall 1\le i\le M,\ 1\le j\le M.$$

An approximation of the entries of B^M and C^M can easily be computed by using the finite element Galerkin projections of the involved functions:

$$\begin{cases} C^M_{i,j} \approx \boldsymbol{w_i}^T M \boldsymbol{b_j}, & \forall 1\le i\le M,\ 1\le j\le\mathcal{N} \\ B^M_{i,j} \approx \boldsymbol{w_i}^T M \boldsymbol{q_j}, & \forall 1\le i\le M,\ 1\le j\le M. \end{cases}$$

With these notations, we can easily prove

Lemma 5.6.2. *Let T be the $\mathcal{N} \times \mathcal{N}$ symmetric positive definite matrix:*

$$T := \left(Q^M (B^M)^{-1} C^M\right)^T M \left(Q^M (B^M)^{-1} C^M\right),$$

and let $\lambda_{\max}(T)$ be the largest eigenvalue of the generalized eigenvalue problem

$$\begin{cases} \text{Find } (\lambda, x) \in \mathbb{R} \times \mathbb{R}^{\mathcal{N}} \text{ such that:} \\ Tx = \lambda M x. \end{cases} \tag{5.47}$$

Then:

$$\max_{\varphi \in V_h^k} \frac{\|\mathcal{J}_M[\varphi]\|_{\mathcal{X}}}{\|\varphi\|_{\mathcal{X}}} = \sqrt{\lambda_{\max}}. \tag{5.48}$$

Proof. For any $\varphi \in V_h^k$ and any $1 \leq i \leq M$:

$$\sigma_i(\varphi) = \sum_{j=1}^{\mathcal{N}} \varphi_j \sigma_i(b_j) = \boldsymbol{e_i}^T C^M \boldsymbol{\varphi},$$

where $\boldsymbol{e_i}$ is the i-th canonical vector of dimension M. Furthermore, if

$$\mathcal{J}_M[\varphi] = \sum_{i=1}^{M} \alpha_i^M(\varphi) q_i \tag{5.49}$$

is the generalized interpolant of φ in dimension M, we have:

$$\sigma_i\left(\mathcal{J}_M[\varphi]\right) = \boldsymbol{e_i}^T B^M \boldsymbol{\alpha}, \quad \forall 1 \leq i \leq M,$$

where $\boldsymbol{\alpha} = \left(\alpha_1^M(\varphi), \ldots, \alpha_M^M(\varphi)\right)^T$. From the interpolation property stated in (5.3), it follows that

$$\boldsymbol{\alpha} = \left(B^M\right)^{-1} C^M \boldsymbol{\varphi}.$$

Then, the finite element Galerkin projection of the interpolant of (5.49) can be expressed as:

$$\mathcal{J}_M[\varphi] \approx Q^M \boldsymbol{\alpha} = Q^M \left(B^M\right)^{-1} C^M \boldsymbol{\varphi}.$$

Hence,

$$\begin{aligned} \max_{\varphi \in V_h^k} \frac{\|\mathcal{J}_M[\varphi]\|_{\mathcal{X}}}{\|\varphi\|_{\mathcal{X}}} &= \left(\max_{\boldsymbol{\varphi} \in \mathbb{R}^{\mathcal{N}}} \frac{\boldsymbol{\varphi}^T \left(Q^M (B^M)^{-1} C^M\right)^T M \left(Q^M (B^M)^{-1} C^M\right) \boldsymbol{\varphi}}{\boldsymbol{\varphi}^T M \boldsymbol{\varphi}} \right)^{1/2} \\ &= \sqrt{\lambda_{\max}(T)}. \end{aligned}$$

□

Remark 5.6.3. *The computation of $\Lambda_{\max}$ can easily be performed by, e.g., the power method scheme applied to the matrix T. However, note that the evaluation of Λ_M with formula (5.48) requires the construction of T, which is a large dense matrix of dimension $\mathcal{N} \times \mathcal{N}$. In cases where the storage of T is no longer possible, the Lebesgue constant can still be computed with formula (5.24), whose evaluation requires the construction of a much smaller matrix of dimension $M \times M$.*

Acknowledgments

This work was supported in part by the joint research program MANON between CEA-Saclay and University Pierre et Marie Curie-Paris 6. It has also been supported by Fondation Sciences Mathematiques de Paris, that hosts Anthony Patera on the Foundation's Senior Research Chair.

Chapter 6

Convergence analysis of the Generalized Empirical Interpolation Method

This is a submitted paper with Y. Maday and G. Turinici. Its reference in the manuscript is:

[79] Y. Maday, O. Mula, and G. Turinici. Convergence analysis of the Generalized Empirical Interpolation Method. Submitted, 2014.

6.1 Introduction

Let $\mathcal{X}$ be a Banach space of functions defined over a domain $\overline{\Omega} \in \mathbb{R}^d$ or $\mathbb{C}^d$, $X_n \in \mathcal{X}$ be a sequence of finite n-dimensional spaces and $S_n = \{x_1, \ldots, x_n\}$ be a set of n points in $\overline{\Omega}$. The problem of interpolating any function $f \in \mathcal{X}$ has traditionally been stated as:

$$\text{“Find } f_n \in X_n \text{ such that } f_n(x_i) = f(x_i),\ \forall i \in \{1, \ldots, n\}\text{”}, \tag{6.1}$$

where we note that it is implicitly assumed that $\mathcal{X}$ is a Banach space of continuous functions. Given X_n and S_n, among the most important issues raised by interpolation stand questions of existence and uniqueness of the interpolant of any $f \in \mathcal{X}$ and also about the stability of the process (via the study of the behavior of the Lebesgue constant – see [35] for this notion–). This, in turn, leads to an even more fundamental question related to the optimal choice of the interpolating space X_n together with the set of points S_n that provide the best interpolation properties. The difficulty of the task has usually led to restrict the study to lagrangian type approximations where the interpolating space X_n is spanned by algebraic polynomials, rational functions, Fourier series, etc. This approach is rather well documented and understood, especially in the case of polynomial interpolation where we know that, in one dimension, an almost optimal location for the interpolating points is given by the Gauss-Chebyshev nodes. More involved conditions are also known in higher dimensions in order for a polynomial interpolation to be well defined and we refer to [35] for more details on this topic.

Although the extension of the Lagrangian interpolation has already been explored in the literature (see, e.g. [114], [47] and also the activity concerning the kriging [65], [73] in the stochastic community), the question still remains on how to extend the concept of interpolation stated in (6.1) to general functions. One step in this direction is the Empirical Interpolation Method (EIM, [11], [55], [80]) that aims at interpolating continous functions belonging to a compact set $F \subset \mathcal{X}$ by interpolating spaces X_n spanned by functions that are not necessarily of polynomial

type. This is achieved by the construction of suitable sets of interpolating spaces and the selection of suitable interpolating points S_n thanks to a greedy selection procedure.

The empirical interpolation process is, by construction, problem dependent given the fact that the constructed X_n and S_n depend on F. Furthermore, it is clear that the successful approximation of any function in F by this method requires to suppose that the set F is approximable by linear combinations of small size. In particular, this is the case when the Kolmogorov n-width $d_n(F, \mathcal{X})$ of F in $\mathcal{X}$ is small. Indeed, $d_n(F, \mathcal{X})$ is defined by

$$d_n(F, \mathcal{X}) := \inf_{\substack{X_n \subset \mathcal{X} \\ dim(X_n)=n}} \sup_{x \in F} \inf_{y \in X_n} \|x - y\|_{\mathcal{X}}$$

and measures the extent to which F can be approximated by finite dimensional spaces $X_n \subset \mathcal{X}$ of dimension n (see [66]). Several reasons can account for the rapid decrease of the Kolmogorov n-width: if F is a set of functions defined over a domain, we can refer to regularity, or even to analyticity, of these functions with respect to the domain variable (as analyzed in the example in [66]). Another possibility is when $F = \{u(\mu, .),\ \mu \in D\}$, where D is a compact set of $\mathbb{R}^p$ and $u(\mu, .)$ is the solution of a PDE parametrized by μ. The approximation of any element $u(\mu, .) \in F$ by finite expansions is a classical problem addressed by reduced basis and the regularity of u in μ can also be a reason for having a small n-width as the results of [27] show.

In order to deal with functions that may not be continuous in space and also to account for experimental framework where data are acquired from sensors, an extension of this Lagrangian interpolation process has been proposed and is called GEIM as for Generalized Empirical Interpolation Method (see also [81], for another, though related approach to the problem of data assimilation). The method was first presented in [76] and consists in replacing the evaluation at interpolating points by application of a class of interpolating continuous linear forms chosen in a given dictionary $\Sigma \subset \mathcal{L}(\mathcal{X})$. In [77], it has been explained how GEIM can be extended to the frame of Banach spaces $\mathcal{X}$ and that EIM is a particular instance of it in the case where $\mathcal{X} = \mathcal{C}(\Omega)$ and the dictionary is composed of Dirac masses.

In this context, the present paper is a contribution to the understanding of the quality of this type of interpolation procedure through the analysis of the behavior of the interpolation error in GEIM in a framework of rapidly enough decreasing Kolmogorov n-width. To this purpose, the accuracy of the approximation in X_n of the elements of F will be compared to the best possible performance in an n-dimensional space which is measured by the Kolmogorov $n-$width $d_n(F, \mathcal{X})$. The present work is not the first contribution that studies the convergence rates of approximations of functions on spaces X_n constructed by greedy algorithms. Pioneer results in the case that $\mathcal{X}$ is a Hilbert can be found in [20] and [18]. An important extension of these works is [37] where the previous results were not only improved for the Hilbert framework but they were also extended to the case of Banach spaces. By employing the methodology proposed in [37], convergence rates for the generalized empirical interpolation were first presented in [78] when $\mathcal{X} = L^2(\Omega)$. As a sequel of [78] and still following the guidelines proposed in [37], we derive in this paper convergence rates for GEIM in the case of Banach spaces.

The document is organized as follows: in section 6.2 it will be shown that, under several hypothesis, the greedy algorithm of GEIM is of a weak greedy type (weak greedy algorithms are a category of greedy algorithms first identified in [18]). This observation is a preliminary step to analyze the convergence decay rates of the interpolation error. Section 6.3 provides these results in the case where $\mathcal{X}$ is a Banach space and in section 6.4 improved results will be derived in the particular case of Hilbert spaces.

6.2 The Generalized Empirical Interpolation Method

Let $\mathcal{X}$ be a Banach space of functions defined over a domain $\Omega \subset \mathbb{R}^d$, where $d = 1, 2, 3$. Its norm is denoted by $\|.\|_{\mathcal{X}}$. Let F be a compact set of $\mathcal{X}$ whose elements $f \in F$ are such that $\|f\|_{\mathcal{X}} \leq 1$. With $\mathcal{N}$ being some given large number, we assume that the dimension of the vectorial space spanned by F is larger than $\mathcal{N}$. Our goal is to build, for all $n < \mathcal{N}$, a sequence of n-dimensional subspaces of $\mathcal{X}$ that approximate well enough any element of F. Assume also that we have at our disposal a dictionary of linear forms $\Sigma \subset \mathcal{L}(\mathcal{X})$ with the following properties:

P1: $\forall \sigma \in \Sigma$, $\|\sigma\|_{\mathcal{L}(\mathcal{X})} = 1$.

P2: *Unisolvence property*: If $\varphi \in \text{span}\{F\}$ is such that $\sigma(\varphi) = 0,\ \forall \sigma \in \Sigma$, then $\varphi = 0$.

Given this setting, GEIM aims at building n-dimensional interpolating spaces X_n spanned by functions $\{\varphi_0, \varphi_1, \ldots, \varphi_{n-1}\}$ of F together with sets of n selected linear forms $\{\sigma_0, \sigma_1, \ldots, \sigma_{n-1}\}$ coming from Σ such that any $\varphi \in F$ is well approximated by its generalized interpolant $\mathcal{J}_n[\varphi] \in X_n$. $\mathcal{J}_n[\varphi]$ has the following interpolation property:

$$\mathcal{J}_n[\varphi] = \sum_{j=0}^{n-1} \beta_j \varphi_j, \text{ such that } \quad \sigma_i(\mathcal{J}_n[\varphi]) = \sigma_i(\varphi),\ \forall i = 0, \ldots, n-1. \tag{6.2}$$

The construction of the interpolation spaces X_n and the selection of the suitable associated elements of the dictionary is recursively carried out by a greedy algorithm. The search for the functions φ_i should ideally be done on F but this a too demanding task in practical applications. Hence, the search is in practice carried out over a discrete subset $\Xi_F \subset F$. For a fixed accuracy parameter $0 < \eta < 1$, there exists a discrete subset $\Xi_F^\eta \subset F$ such that the algorithm is of a weak greedy type as defined in section 1.3 of [18]. In the following, Ξ_F will denote this subset Ξ_F^η. Before proving its existence in lemma 6.2.1, let us momentarily assume this fact in order to explain how the search of the interpolating basis functions is carried out:

The first interpolating function φ_0 is chosen such that:

$$\|\varphi_0\|_{\mathcal{X}} = \max_{\varphi \in \Xi_F} \|\varphi\|_{\mathcal{X}} \geq \eta \sup_{\varphi \in F} \|\varphi\|_{\mathcal{X}},$$

the last inequality being a consequence of the definition of $\Xi_F \equiv \Xi_F^\eta$. The first interpolating linear form is

$$\sigma_0 = \arg\sup_{\sigma \in \Sigma} |\sigma(\varphi_0)|.$$

We then define the first basis function as $q_0 = \dfrac{\varphi_0}{\sigma_0(\varphi_0)}$ and the interpolation operator $\mathcal{J}_1 : \mathcal{X} \mapsto \text{span}\{q_0\}$ such that $\sigma_0(\varphi) = \sigma_0(\mathcal{J}_1[\varphi])$, for any $\varphi \in \mathcal{X}$. This yields the following expression:

$$\forall \varphi \in \mathcal{X}, \quad \mathcal{J}_1[\varphi] = \sigma_0(\varphi) q_0. \tag{6.3}$$

The second interpolating function φ_1 is chosen such that

$$\|\varphi_1 - \mathcal{J}_1[\varphi_1]\|_{\mathcal{X}} = \max_{\varphi \in \Xi_F} \|\varphi - \mathcal{J}_1[\varphi]\|_{\mathcal{X}} \geq \eta \sup_{\varphi \in F} \|\varphi - \mathcal{J}_1[\varphi]\|_{\mathcal{X}}.$$

The second interpolating linear form is

$$\sigma_1 = \arg\sup_{\sigma \in \Sigma} |\sigma(\varphi_1 - \mathcal{J}_1[\varphi_1])|,$$

and the second basis function is defined as

$$q_1 = \frac{\varphi_1 - \mathcal{J}_1[\varphi_1]}{\sigma_1(\varphi_1 - \mathcal{J}_1[\varphi_1])}.$$

We then proceed by induction. With $N_{\max} < \mathcal{N}$ being an upper bound fixed *a priori*, assume that, for a given $1 \leq n < N_{\max}$, we have built the set of interpolating functions $\{q_0, q_1, \ldots, q_{n-1}\}$ and the set of associated interpolating linear forms $\{\sigma_0, \sigma_1, \ldots, \sigma_{n-1}\}$ such that

$$\forall \varphi \in \mathcal{X}, \quad \mathcal{J}_n[\varphi] = \sum_{j=0}^{n-1} \alpha_j^n(\varphi) q_j$$

is well defined and the coefficients $\alpha_j^n(\varphi)$, $j = 0, \ldots, n-1$, are given by the interpolation problem

$$\begin{cases} \text{Find } \left(\alpha_j^n(\varphi)\right)_{j=0}^{n-1} \text{ such that:} \\ \sum_{j=0}^{n-1} \alpha_j^n(\varphi)\sigma_i(q_j) = \sigma_i(\varphi), \quad \forall i = 0, \ldots, n-1. \end{cases}$$

We now define

$$\forall \varphi \in \Xi_F, \quad \varepsilon_n(\varphi) = \|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}}.$$

We choose φ_n such that

$$\varepsilon_n(\varphi_n) = \max_{\varphi \in \Xi_F} \varepsilon_n(\varphi) \geq \eta \sup_{\varphi \in F} \varepsilon_n(\varphi)$$

and $\sigma_n = \arg\sup_{\sigma \in \Sigma} |\sigma(\varphi_n - \mathcal{J}_n[\varphi_n])|$. The next basis function is then

$$q_n = \frac{\varphi_n - \mathcal{J}_n[\varphi_n]}{\sigma_n(\varphi_n - \mathcal{J}_n[\varphi_n])}.$$

We finally set $X_{n+1} \equiv \text{span}\{q_j,\ j \in [0, n]\} = \text{span}\{\varphi_j,\ j \in [0, n]\}$. The interpolation operator $\mathcal{J}_{n+1} : \mathcal{X} \mapsto X_{n+1}$ is given by

$$\forall \varphi \in \mathcal{X}, \quad \mathcal{J}_{n+1}[\varphi] = \sum_{j=0}^{n} \alpha_j^{n+1}(\varphi) q_j \tag{6.4}$$

and the coefficients $\alpha_j^{n+1}(\varphi)$, $j = 0, \ldots, n$, are given by the interpolation problem

$$\begin{cases} \text{Find } \left(\alpha_j^{n+1}(\varphi)\right)_{j=0}^{n} \text{ such that:} \\ \sum_{j=0}^{n} \alpha_j^{n+1}(\varphi)\sigma_i(q_j) = \sigma_i(\varphi), \quad \forall i = 0, \ldots, n. \end{cases}$$

It has been proven in [80] (for EIM) and [77] (for GEIM) that for any $1 \leq n \leq N_{\max}$, the set $\{q_j,\ j \in [0, n-1]\}$ is linearly independent and that the generalized empirical interpolation procedure is well-posed in $\mathcal{X}$. It has also been proven that the interpolation error satisfies:

$$\forall \varphi \in F, \quad \|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq (1 + \Lambda_n) \inf_{\psi_n \in X_n} \|\varphi - \psi_n\|_{\mathcal{X}}, \tag{6.5}$$

where Λ_n is the Lebesgue constant in the $\mathcal{X}$ norm:

$$\Lambda_n := \sup_{\varphi \in \mathcal{X}} \frac{\|\mathcal{J}_n[\varphi]\|_{\mathcal{X}}}{\|\varphi\|_{\mathcal{X}}}. \tag{6.6}$$

Note that the parameter η quantifies the optimality of the greedy search: $\eta = 1$ will be the ideal case where $\Xi_F = F$ and the smaller the η, the worse Ξ_F will capture the interpolation behavior of the whole set F. Note also that Ξ_F^η cannot be easily determined in practice because its evaluation would require the computation of supremizers over the whole set F, which is not entirely possible in practice. The following lemma shows the existence of the discrete subset $\Xi_F = \Xi_F^\eta$, for any given η.

Lemma 6.2.1. *Let F be a compact subset of $\mathcal{X}$. Then, for any $0 < \eta < 1$, there exits a discrete subset Ξ_F^η such that*

$$\begin{cases} \max\limits_{\varphi\in\Xi_F^\eta} \|\varphi\|_{\mathcal{X}} \geq \eta \sup\limits_{\varphi\in F} \|\varphi\|_{\mathcal{X}}, \\ \max\limits_{\varphi\in\Xi_F^\eta} \|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \geq \eta \sup\limits_{\varphi\in F} \|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}}, \quad \forall\ n \in \{1,\dots,N_{\max}\}. \end{cases} \tag{6.7}$$

Proof. For a given $0 < \eta < 1$ and from the finite open cover property of the compact set F, there exists a discrete subset $\Xi_0^\eta \subset F$ and a function $\tilde{\varphi}_0 \in F$ such that:

$$\sup_{\varphi\in F} \inf_{\psi\in\Xi_0^\eta} \|\varphi - \psi\|_{\mathcal{X}} \leq (1-\eta)\|\tilde{\varphi}_0\|_{\mathcal{X}}.$$

Let $\varphi_0 = \arg\max\limits_{\psi\in\Xi_0^\eta} \|\psi\|_{\mathcal{X}}$ and $\varphi_0^{\sup} = \arg\sup\limits_{\varphi\in F} \|\varphi\|_{\mathcal{X}}$. Then, for any $\psi \in \Xi_0^\eta$:

$$\|\varphi_0\|_{\mathcal{X}} \geq \|\psi\|_{\mathcal{X}} \geq -\|\psi - \varphi_0^{\sup}\|_{\mathcal{X}} + \|\varphi_0^{\sup}\|_{\mathcal{X}} \geq -(1-\eta)\|\tilde{\varphi}_0\|_{\mathcal{X}} + \|\varphi_0^{\sup}\|_{\mathcal{X}} \geq \eta\|\varphi_0^{\sup}\|_{\mathcal{X}}.$$

This completes the proof of the first inequality of (6.7). The second inequality is derived following the same guidelines: for any $1 \leq n \leq N_{\max}$, the application

$$\begin{array}{rcl} r_n : \mathcal{X} & \mapsto & \mathcal{X} \\ \varphi & \mapsto & \varphi - \mathcal{J}_n[\varphi] \end{array}$$

is clearly continuous (with a norm that depends on Λ_n) and $r_n(F)$ is a compact subset of $\mathcal{X}$. From the finite open cover property of $r_n(F)$, there exists a discrete subset $\Xi_n^\eta \subset F$ and $\tilde{\varphi}_n \in F$ such that:

$$\sup_{\varphi\in F} \inf_{\psi\in\Xi_n^\eta} \|r_n[\varphi] - r_n[\psi]\|_{\mathcal{X}} \leq (1-\eta)\|r_n[\tilde{\varphi}_n]\|_{\mathcal{X}}.$$

Let $\varphi_n = \arg\max\limits_{\psi\in\Xi_n^\eta} \|r_n[\psi]\|_{\mathcal{X}}$ and $\varphi_n^{\sup} = \arg\sup\limits_{\varphi\in F} \|r_n[\varphi]\|_{\mathcal{X}}$. Then, for any $\psi \in \Xi_n^\eta$:

$$\begin{array}{rcl} \|\varphi_n - \mathcal{J}_n[\varphi_n]\|_{\mathcal{X}} & \geq & \|r_n[\psi]\|_{\mathcal{X}} \\ & \geq & -\|r_n[\psi] - r_n[\varphi_n^{\sup}]\|_{\mathcal{X}} + \|r_n[\varphi_n^{\sup}]\|_{\mathcal{X}} \\ & \geq & -(1-\eta)\|r_n[\tilde{\varphi}_n]\|_{\mathcal{X}} + \|r_n[\varphi_n^{\sup}]\|_{\mathcal{X}} \\ & \geq & \eta\ \|\varphi_n^{\sup} - \mathcal{J}_n[\varphi_n^{\sup}]\|_{\mathcal{X}}. \end{array}$$

The proof follows by taking

$$\Xi_F^\eta = \bigcup_{j=0}^{N_{\max}} \Xi_j^\eta.$$

□

Remark 6.2.2. *Note that the construction done in the proof is actually constructive in an adaptive and recursive way. Indeed, starting from the Ξ_0^η, that allows to define φ_0, the first interpolating function, the recursive update of the set Ξ^η can be done by adding a set Ξ_n^η defined similarly as $\Xi_0^{\eta_n}$, with $1 - \eta_n = \frac{(1-\eta)}{(1+\Lambda_n)}\|r_n[\tilde{\varphi}_n]\|_X$, the evaluation of Λ_n being explained in [77] in the Hilbertian context.*

Remark 6.2.3. *In a similar manner as in the case where F is an infinite set of functions, if the dictionary Σ is not a finite set of linear forms, the greedy search is in practice carried out over a discrete subset $\widetilde{\Sigma} \subset \Sigma$. The choice of the subset $\widetilde{\Sigma}$ will have an impact on the definition of the sequence of subsets $(\Xi_F^\eta)_{j=0}^{N_{\max}}$ described in the proof of lemma 6.2.1. The "coarser" the choice on $\tilde{\Sigma}$, the "finer" the subsets $(\Xi_F^\eta)_{j=0}^{N_{\max}}$ must be in order to satisfy relation (6.7).*

Remark 6.2.4. *The Lebesgue constant Λ_n defined in our interpolation procedure depends both on the set F and on the choice of the dictionary of continuous linear forms Σ. In the case of Hilbert spaces, a formula for Λ_n has been given in [77] where the impact of the selected linear forms is expressed more explicitly than in formula (6.6) and allows for an easier implementation. Although no theoretical analysis about the impact of F or Σ on the behavior of (Λ_n) has been possible so far, one can find in [77] an illustration of these interactions in a simple numerical example. In the same reference, it is also outlined how the generalized interpolant of a function can be efficiently computed in practice by a recursion formula.*

6.3 Convergence rates of GEIM in a Banach space

In order to have a consistent notation in what follows, we define $\varphi_n = 0$ and $X_n = X_{N_{\max}}$ for $n > N_{\max}$.

6.3.1 Preliminary notations and properties

We remind that $\mathcal{X}$ is a Banach space. To fix some notations, let K be a nonempty subset of $\mathcal{X}$. For every $\varphi \in \mathcal{X}$, the distance between φ and the set K is denoted by $\text{dist}(\varphi, K)$ and is defined by the following minimum equation:

$$\text{dist}(\varphi, K) = \inf_{y \in K} \|\varphi - y\|_{\mathcal{X}}.$$

For any $\varphi \in \mathcal{X}$, the metric projection of φ onto K is given by the set

$$P_K(\varphi) = \{z \in K : \|\varphi - z\|_{\mathcal{X}} = \text{dist}(\varphi, K)\}.$$

In general, this set can be empty or composed of one or more than one element. However, in the particular case where K is a finite dimensional space, $P_K(\varphi)$ is not empty. For any $n \geq 1$, the non empty set

$$P_n(\varphi) = \{z \in X_n : \|\varphi - z\|_{\mathcal{X}} = \text{dist}(\varphi, X_n)\} \tag{6.8}$$

will denote the metric projection of $\varphi \in \mathcal{X}$ onto X_n. Since, the uniqueness of the metric projection onto X_n is not necessarily ensured, in the following, $P_n(\varphi)$ will denote one of the elements of the set (6.8). We also define for any $1 \leq n \leq N_{\max}$:

$$\tau_n(F)_{\mathcal{X}} := \max_{f \in F} \|f - P_n(f)\|_{\mathcal{X}}, \quad n = 1, 2, \ldots \tag{6.9}$$

and

$$\gamma_n = \frac{\eta}{1 + \Lambda_n}, \quad \forall\ 1 \leq n \leq N_{\max}. \tag{6.10}$$

We will use the abbreviation τ_n and d_n for $\tau_n(F)_{\mathcal{X}}$ and $d_n(F, \mathcal{X})$. Likewise, (τ_n) and (d_n) will denote the sequences $(\tau_n(F)_{\mathcal{X}})_{n=1}^{\infty}$ and $(d_n(F, \mathcal{X}))_{n=1}^{\infty}$ respectively. We finish this section by proving the following lemma:

Lemma 6.3.1. *For any $n \geq 1$, $\|\varphi_n - P_n(\varphi_n)\|_{\mathcal{X}} \geq \gamma_n \tau_n$.*

Proof. From equation (6.5) applied to $\varphi = \varphi_n$ we have $\|\varphi_n - P_n(\varphi_n)\|_{\mathcal{X}} \geq \dfrac{1}{1 + \Lambda_n} \|\varphi_n - \mathcal{J}_n(\varphi_n)\|_{\mathcal{X}}$. But $\|\varphi_n - \mathcal{J}_n(\varphi_n)\|_{\mathcal{X}} \geq \eta \|\varphi - \mathcal{J}_n(\varphi)\|_{\mathcal{X}}$ for any $\varphi \in F$ according to the definition of φ_n. Thus $\|\varphi_n - P_n(\varphi_n)\|_{\mathcal{X}} \geq \gamma_n \|\varphi - \mathcal{J}_n(\varphi)\|_{\mathcal{X}} \geq \gamma_n \|\varphi - P_n(\varphi)\|_{\mathcal{X}}$. □

Thanks to lemma 6.3.1, we have proven that the weak greedy algorithm of GEIM has very similar properties as the abstract weak greedy algorithm analyzed in [37]. The difference is that, in our case, the parameter γ depends on the dimension n whereas in [37] γ was a constant. This observation will be the key to derive convergence decay rates in the sequence (τ_n) by extending the proofs of [37]. The main two lemmas that were derived in [37] (with γ independent of n) are recalled in lemmas 6.3.2 and 6.3.3 and section 6.3.2 presents their extension when γ depends on n. Then, by using equation (6.5), the results on the convergence of the interpolation error will easily follow (section 6.3.3).

Lemma 6.3.2 (Corollary 4.2 – (ii) of [37] – Polynomial decay rates for (τ_n) when $\gamma_n = \gamma$)**.** *If, for $\alpha > 0$, we have $d_n \leq C_0 n^{-\alpha}$, $n = 1, 2, \ldots$, then for any $0 < \beta < \min\{\alpha, 1/2\}$, we have $\tau_n \leq C_1 n^{-\alpha+1/2+\beta}$, $n = 1, 2, \ldots$, with*

$$C_1 := \max\left\{ C_0 4^{4\alpha+1}\gamma^{-4}\left(\frac{2\beta+1}{2\beta}\right)^{\alpha} ; \max_{n=1,\ldots,7} n^{\alpha-\beta-1/2} \right\}.$$

Lemma 6.3.3 (Corollary 4.2 – (iii) of [37] – Exponential decay rates for (τ_n) when $\gamma_n = \gamma$)**.** *If, for $\alpha > 0$, $d_n \leq C_0 e^{-c_1 n^{\alpha}}$, $n = 1, 2, \ldots$, then $\tau_n < \sqrt{2C_0}\gamma^{-1}\sqrt{n}e^{-c_2 n^{\alpha}}$, $n = 1, 2, \ldots$, where $c_2 = 2^{-1-2\alpha}c_1$. The factor $\sqrt{n}$ can be deleted by reducing the constant c_2.*

6.3.2 Convergence rates for (τ_n) in the case where (γ_n) is not constant

We look for an upper bound of the sequence (τ_n) that involves the sequence of Kolmogorov n-widths (d_n). The case $n = 1$ is addressed in

Lemma 6.3.4. *In the case where $n = 1$, we have the following upper bound for τ_1:*

$$\tau_1 \leq 2\left(1 + \frac{1}{\eta}\right) d_1.$$

Proof. Given the parameter η coming from the GEIM greedy algorithm, let $\beta > \dfrac{1}{\eta}$. We begin by recalling and defining some notations:

- φ_0 is the first interpolating function chosen by the greedy algorithm and $X_1 = \text{span}\{\varphi_0\}$.
- For any φ, $P_1(\varphi)$ is the metric projection of φ onto X_1.
- Let $\|\varphi_0^{\text{sup}}\|_{\mathcal{X}} = \sup_{\varphi \in F} \|\varphi\|_{\mathcal{X}}$. From the greedy selection procedure: $\|\varphi_0\|_{\mathcal{X}} \geq \eta \|\varphi_0^{\text{sup}}\|_{\mathcal{X}}$.
- Let X_μ be the one dimensional subspace associated to d_1. In other words,

$$X_\mu = \underset{\substack{X_1 \subset \mathcal{X} \\ dim(X_1)=1}}{\arg\inf} \sup_{x \in F} \inf_{y \in X_1} \|x - y\|_{\mathcal{X}}$$

  and

$$\forall\, \varphi \in \mathcal{X},\ \|\varphi - P_{X_\mu}(\varphi)\|_{\mathcal{X}} \leq d_1.$$

- Let $\varphi_\mu^{\text{sup}} = \underset{\varphi \in F}{\arg\max} \|P_{X_\mu}(\varphi)\|_{\mathcal{X}}$.

We now divide the proof by considering two complementary cases of values of $\|P_{X_\mu}(\varphi_\mu^{\text{sup}})\|_{\mathcal{X}}$. If $\|P_{X_\mu}(\varphi_\mu^{\text{sup}})\|_{\mathcal{X}} \leq \dfrac{1+\eta}{\eta - \frac{1}{\beta}}\, d_1$, we easily derive that

$$\begin{aligned} \forall \varphi \in F,\ \|\varphi - P_1(\varphi)\|_{\mathcal{X}} &\leq \|\varphi\|_{\mathcal{X}} \\ &\leq \|\varphi - P_{X_\mu}(\varphi)\|_{\mathcal{X}} + \|P_{X_\mu}(\varphi)\|_{\mathcal{X}} \\ &\leq d_1 + \|P_{X_\mu}(\varphi_\mu^{\text{sup}})\|_{\mathcal{X}} \\ &\leq \left(1 + \frac{1+\eta}{\eta - \frac{1}{\beta}}\right) d_1. \end{aligned}$$

If $\|P_{X_\mu}(\varphi_\mu^{\sup})\|_{\mathcal{X}} \geq \dfrac{1+\eta}{\eta - \frac{1}{\beta}}\, d_1$, we start by deriving the following inequality for $\|P_{X_\mu}(\varphi_0)\|_{\mathcal{X}}$:

$$\begin{aligned}
\|P_{X_\mu}(\varphi_0)\|_{\mathcal{X}} &\geq \|\varphi_0\|_{\mathcal{X}} - d_1 \\
&\geq \eta\|\varphi_0^{\sup}\|_{\mathcal{X}} - d_1 \\
&\geq \eta\|\varphi_\mu^{\sup}\|_{\mathcal{X}} - d_1 \\
&\geq \eta\left(\|P_{X_\mu}(\varphi_\mu^{\sup})\|_{\mathcal{X}} - d_1\right) - d_1 \\
&\geq \eta\|P_{X_\mu}(\varphi_\mu^{\sup})\|_{\mathcal{X}} - (1+\eta)d_1 \\
&\geq \frac{\|P_{X_\mu}(\varphi_\mu^{\sup})\|_{\mathcal{X}}}{\beta}.
\end{aligned} \tag{6.11}$$

From inequality (6.11), it follows that $\|P_{X_\mu}(\varphi_0)\|_{\mathcal{X}} > 0$ given that $\|P_{X_\mu}(\varphi_\mu^{\sup})\|_{\mathcal{X}}$ is strictly positive. Furthermore, for any $\varphi \in \mathcal{X}$, there exits $\lambda_\varphi \in \mathbb{R}_+$ such that:

$$P_{X_\mu}(\varphi) = \lambda_\varphi P_{X_\mu}(\varphi_0). \tag{6.12}$$

Hence the decomposition:

$$\begin{aligned}
\varphi &= P_{X_\mu}(\varphi) + \varphi - P_{X_\mu}(\varphi) \\
&= \lambda_\varphi P_{X_\mu}(\varphi_0) + \varphi - P_{X_\mu}(\varphi) \\
&= \lambda_\varphi(P_{X_\mu}(\varphi_0) - \varphi_0) + \lambda_\varphi\varphi_0 + \varphi - P_{X_\mu}(\varphi)
\end{aligned} \tag{6.13}$$

Equation (6.13) yields:

$$\begin{aligned}
\|\varphi - P_1(\varphi)\|_{\mathcal{X}} &\leq \|\varphi - \lambda_\varphi\varphi_0\|_{\mathcal{X}} \\
&\leq |\lambda_\varphi|\|P_{X_\mu}(\varphi_0) - \varphi_0\|_{\mathcal{X}} + \|\varphi - P_{X_\mu}(\varphi)\|_{\mathcal{X}} \\
&\leq (1 + |\lambda_\varphi|)d_1,
\end{aligned}$$

Furthermore, given that $\|P_{X_\mu}(\varphi_\mu^{\sup})\|_{\mathcal{X}} \geq \|P_{X_\mu}(\varphi)\|_{\mathcal{X}}$ for any $\varphi \in F$, we have

$$\|P_{X_\mu}(\varphi_\mu^{\sup})\|_{\mathcal{X}} \geq |\lambda_\varphi|\|P_{X_\mu}(\varphi_0)\|_{\mathcal{X}}, \tag{6.14}$$

where we have used equality (6.12). Inequalities (6.11) and (6.14) yield $|\lambda_\varphi| \leq \beta$ and therefore

$$\|\varphi - P_1(\varphi)\|_{\mathcal{X}} \leq (1+\beta)d_1.$$

Hence, we have proven that for any $\beta > 1/\eta$ and any $\varphi \in F$, we have

$$\|\varphi - P_1(\varphi)\|_{\mathcal{X}} \leq \max\left(1+\beta; 1 + \frac{1+\eta}{\eta - \frac{1}{\beta}}\right) d_1.$$

If we define

$$\forall \beta > 1/\eta, \quad g_\eta(\beta) := \max\left(1+\beta; 1 + \frac{1+\eta}{\eta - \frac{1}{\beta}}\right),$$

it follows that $\|\varphi - P_1(\varphi)\|_{\mathcal{X}} \leq \min\limits_{\beta > 1/2} g_\eta(\beta) d_1 = 2\left(1 + \dfrac{1}{\eta}\right) d_1$. □

For higher dimensions ($n > 1$), we first begin by proving

Theorem 6.3.5. *For any $N \geq 0$, consider a weak greedy algorithm with the property of lemma 6.3.1 and constant γ_N. We have the following inequalities between τ_N and d_N: for any $K \geq 1$, $1 \leq m < K$*

$$\prod_{i=1}^{K} \tau_{N+i}^2 \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^2} 2^K K^{K-m} \left(\sum_{i=1}^{K} \tau_{N+i}^2 \right)^m d_m^{2(K-m)} \tag{6.15}$$

Proof. This result is an extension of theorem 4.1 of [37] to the case where the parameter of the weak greedy algorithm (γ_N) depends on the dimension of the reduced space X_N. Its proof consists in a slight modification of the demonstration presented in [37]. The complete proof is given in the appendix section for the self-consistency of this paper. □

This theorem easily yields the following useful corollaries.

Corollary 6.3.6. *For any $n \geq 1$, we have:*

$$\tau_n \leq \frac{1}{\prod_{i=1}^{n} \gamma_i^{1/n}} \sqrt{2} \min_{1 \leq m < n} \left\{ n^{\frac{n-m}{2n}} \left(\sum_{i=1}^{n} \tau_i^2 \right)^{\frac{m}{2n}} d_m^{\frac{n-m}{n}} \right\} \tag{6.16}$$

In particular, for any $\ell \geq 1$:

$$\tau_{2\ell} \leq 2 \frac{1}{\prod_{i=1}^{2\ell} \gamma_i^{1/2\ell}} \sqrt{\ell d_\ell}. \tag{6.17}$$

Proof. We take $N = 0$, $K = n$ and any $1 \leq m < n$ in (6.15) and use the monotonicity of (τ_n) to obtain:

$$\tau_n^{2n} \leq \prod_{i=1}^{n} \tau_i^2 \leq \frac{1}{\prod_{i=1}^{n} \gamma_i^2} 2^n n^{n-m} \left(\sum_{i=1}^{n} \tau_i^2 \right)^m d_m^{2(n-m)}.$$

If we take the $2n$-th root on both sides, we arrive at (6.16). In particular, if $n = 2\ell$ and $m = \ell$, we have:

$$\tau_{2\ell} \leq \frac{1}{\prod_{i=1}^{2\ell} \gamma_i^{1/2\ell}} \sqrt{2} (2\ell)^{1/4} \left(\sum_{i=1}^{2\ell} \tau_i^2 \right)^{1/4} \sqrt{d_\ell} \leq \frac{1}{\prod_{i=1}^{2\ell} \gamma_i^{1/2\ell}} \sqrt{2} (2\ell)^{1/4} (2\ell)^{1/4} \sqrt{d_\ell} = 2 \frac{1}{\prod_{i=1}^{2\ell} \gamma_i^{1/2\ell}} \sqrt{\ell d_\ell},$$

where we have used the fact that all $\tau_i \leq 1$. □

Corollary 6.3.7. *For $N \geq 0$, $K \geq 1$ and $1 \leq m < K$:*

$$\tau_{N+K} \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^{1/K}} \sqrt{2K} \tau_{N+1}^{m/K} d_m^{1-m/K} \tag{6.18}$$

Proof. Given that (τ_n) is a monotonically decreasing sequence as is obtained by following the same lines as above, we derive from inequality (6.15) that:

$$\tau_{N+K}^{2K} \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^2} 2^K K^{K-m} \left(\sum_{i=1}^{K} \tau_{N+i}^2 \right)^m d_m^{2(K-m)}$$

Therefore,

$$\tau_{N+K} \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^{1/K}} \sqrt{2} K^{\frac{K-m}{2K}} \left(K\tau_{N+1}^2\right)^{m/2K} d_m^{1-m/K} \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^{1/K}} \sqrt{2K} \tau_{N+1}^{m/K} d_m^{1-m/K}.$$

□

We now derive convergence rates in (τ_n) for polynomial or exponential convergence of the Kolmogorov n-width. In the first two lemmas 6.3.8 and 6.3.9, the result is derived without making any assumption on the behavior of the sequence (γ_n) (that depends on the Lebesgue constant of GEIM).

Lemma 6.3.8 (Polynomial decay of (d_n))**.** *For any $n \geq 1$, let $n = 4\ell + k$ (where $\ell \in \{0, 1, \ldots\}$ and $k \in \{0, 1, 2, 3\}$). Assume that there exists a constant $C_0 > 0$ such that $\forall n \geq 1$, $d_n \leq C_0 n^{-\alpha}$, then*

$$\tau_n \leq C_0 \beta_n n^{-\alpha}, \tag{6.19}$$

where

$$\beta_n = \beta_{4\ell+k} := \begin{cases} 2\left(1 + \dfrac{1}{\eta}\right) & \text{if } n = 1 \\ \dfrac{1}{\prod_{i=1}^{\ell_2} \gamma_{\ell_1 - \lceil \frac{k}{4} \rceil + i}^{\frac{1}{\ell_2}}} \sqrt{2\ell_2 \beta_{\ell_1}} (2\sqrt{2})^{\alpha} & \text{if } n \geq 2 \end{cases}$$

and $\ell_1 = 2\ell + \lfloor \frac{2k}{3} \rfloor$, $\ell_2 = 2\left(\ell + \lceil \frac{k}{4} \rceil\right)$, where $\lfloor . \rfloor$ and $\lceil . \rceil$ are the floor and ceiling functions respectively.

Proof. The proof is done by recurrence over n and the case $n = 1$ directly follows from lemma 6.3.4. In the case $n \geq 2$:, we write $n = N + K$ with $N \geq 0$ and $K \geq 2$. Thanks to corollary 6.3.7, we have for any $1 \leq m < K$:

$$\tau_n = \tau_{N+K} \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^{1/K}} \sqrt{2K} \tau_{N+1}^{m/K} d_m^{1-m/K} \tag{6.20}$$

We now use that $d_m \leq C_0 m^{-\alpha}$ and the recurrence hypothesis $\tau_{N+1} \leq C_0 \beta_{N+1} (N+1)^{-\alpha}$ which yield:

$$\tau_{N+K} \leq C_0 \sqrt{2K} \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^{\frac{1}{K}}} \beta_{N+1}^{\frac{m}{K}} \xi(N, K, m)^{\alpha} (N+K)^{-\alpha}, \tag{6.21}$$

where $\xi(N, K, m) = \dfrac{N+K}{m} \left(\dfrac{m}{N+1}\right)^{\frac{m}{K}}$ for any $1 \leq m < K$ and any given index $n = N + K \geq 2$, where $N \geq 0$ and $K \geq 2$.
Furthermore, any $n \geq 2$ can be written as $n = 4\ell + k$ with $\ell \in \mathbb{N}$ and $k \in \{0, 1, 2, 3\}$. If $k = 1, 2$ or 3, it can easily be proven that the function ξ is bounded by $2\sqrt{2}$ by setting

$$\begin{cases} N = 2\ell - 1,\ K = 2\ell + 2,\ m = \ell + 1 \text{ and } \ell \geq 1 \text{ in the case } k = 1, \\ N = 2\ell,\ K = 2\ell + 2,\ m = \ell + 1 \text{ and } \ell \geq 0 \text{ in the case } k = 2, \\ N = 2\ell + 1,\ K = 2\ell + 2,\ m = \ell + 1 \text{ and } \ell \geq 0 \text{ in the case } k = 3. \end{cases}$$

These choices of N, K and m combined with the upper bound of ξ yield the result $\tau_n \leq C_0 \beta_n n^{-\alpha}$ in the case $k = 1, 2$ or 3.

To deal with the case $n = 4\ell$, we come back to estimate (6.20) and use the fact that $\tau_{N+1} \leq \tau_N$. It follows that:

$$\tau_n \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^{1/K}} \sqrt{2K} \tau_N^{m/K} d_m^{1-m/K}. \tag{6.22}$$

If we choose $N = K = 2\ell$ and $m = \ell$, the inequality (6.22) directly yields the desired result

$$\tau_{4\ell} \leq C_0 \sqrt{2} \sqrt{2\ell\beta_{2\ell}} \frac{1}{\prod_{i=1}^{2\ell} \gamma_{2\ell+i}^{\frac{1}{2\ell}}} (2\sqrt{2})^{\alpha} (4\ell)^{-\alpha}.$$

□

Lemma 6.3.9 (Exponential decay in (d_n))**.** *Assume that there exist constants $C_0 \geq 1$ and $\alpha > 0$ such that $\forall n \geq 1$, $d_n \leq C_0 e^{-c_1 n^{\alpha}}$, then*

$$\tau_n \leq C_0 \beta_n e^{-c_2 n^{\alpha}},$$

where $c_2 := c_1 2^{-2\alpha-1}$ and

$$\beta_n := \begin{cases} 2\left(1 + \dfrac{1}{\eta}\right), & \text{if } n = 1 \\ \sqrt{2} \dfrac{1}{\prod_{i=1}^{2\lfloor \frac{n}{2} \rfloor} \gamma_i^{\frac{1}{2\lfloor \frac{n}{2} \rfloor}}} \sqrt{n}, & \text{if } n \geq 2. \end{cases}$$

Proof. The case $n = 1$ easily follows from lemma 6.3.4. For $n = 2\ell$ ($\ell \geq 1$), inequality (6.17) directly yields:

$$\tau_{2\ell} \leq 2 \frac{1}{\prod_{i=1}^{2\ell} \gamma_i^{1/2\ell}} \sqrt{\ell d_\ell} \leq C_0 \sqrt{2} \frac{1}{\prod_{i=1}^{2\ell} \gamma_i^{1/2\ell}} \sqrt{2\ell} e^{-\frac{c_1}{2^{1+\alpha}} (2\ell)^{\alpha}}, \tag{6.23}$$

where we have used the fact that $d_\ell \leq C_0 e^{-c_1 (\ell)^{\alpha}}$ and that $C_0 \geq 1$. For $n = 2\ell + 1$, by using inequality (6.23) and the fact that $\tau_{2\ell+1} \leq \tau_{2\ell}$, we have:

$$\tau_{2\ell+1} \leq C_0 \sqrt{2} \frac{1}{\prod_{i=1}^{2\ell} \gamma_i^{1/2\ell}} \sqrt{2\ell} e^{-\frac{c_1}{2^{1+\alpha}} (2\ell)^{\alpha}} \leq C_0 \sqrt{2} \frac{1}{\prod_{i=1}^{2\ell} \gamma_i^{1/2\ell}} \sqrt{2\ell + 1} e^{-\frac{c_1}{2^{1+2\alpha}} (2\ell+1)^{\alpha}}. \tag{6.24}$$

□

The sequence (γ_n) is directly related to the Lebesgue constant sequence (Λ_n) and, in the particular case where (Λ_n) is monotonically increasing. It is therefore interesting to analyze the convergence rates that lemmas 6.3.8 and 6.3.9 provide in this particular case and the following corollary accounts for it.

Corollary 6.3.10. *In the case where (Λ_n) is monotically increasing (i.e. (γ_n) monotonically decreasing), the following bounds can be derived for τ_n:*

i) If $d_n \leq C_0 n^{-\alpha}$ for any $n \geq 1$, then $\tau_n \leq C_0 \tilde{\beta}_n n^{-\alpha}$, with

$$\tilde{\beta}_n := \begin{cases} 2\left(1 + \dfrac{1}{\eta}\right), & \text{if } n = 1 \\ 2^{3\alpha+1} \ell_2 \left(\dfrac{1}{\gamma_n}\right)^2, & \forall n \geq 2. \end{cases}$$

If we write n as $n = 4\ell + k$ (with $\ell \in \{0, 1, \ldots\}$ and $k \in \{0, 1, 2, 3\}$), then $\ell_2 = 2\left(\ell + \lceil \frac{k}{4} \rceil\right)$.

ii) If $d_n \leq C_0 e^{-c_1 n^\alpha}$ *for* $n \geq 1$ *and* $C_0 \geq 1$, *then* $\tau_n \leq C_0 \tilde{\beta}_n e^{-c_2 n^{-\alpha}}$, *with* $c_2 = c_1 2^{-2\alpha-1}$

$$\tilde{\beta}_n := \begin{cases} 2\left(1+\dfrac{1}{\eta}\right), & \text{if } n = 1 \\ \sqrt{2}\dfrac{1}{\gamma_n}\sqrt{n}, & \text{if } n \geq 2. \end{cases}$$

Proof.

i) The proof consists in showing by recursion that $\tilde{\beta}_n$ is larger than the coefficient β_n defined in lemma 6.3.8.
In the case $n = 1$, $\tilde{\beta}_1 = \beta_1$. Then, for $n > 1$, given that (γ_n) is monotonically decreasing, we have

$$\beta_n \leq \frac{1}{\gamma_n}\sqrt{2\ell_2\beta_{\ell_1}}\left(2\sqrt{2}\right)^\alpha \leq \frac{1}{\gamma_n}\sqrt{2\ell_2\tilde{\beta}_{\ell_1}}\left(2\sqrt{2}\right)^\alpha,$$

where we have used the recurrence hypothesis $\beta_{\ell_1} \leq \tilde{\beta}_{\ell_1}$ in the second inequality. Furthermore, since

$$\tilde{\beta}_{\ell_1} \leq 2^{3\alpha+1}\ell_2\gamma_n^{-2},$$

it follows that:

$$\beta_n \leq \gamma_n^{-1}\sqrt{2\ell_2 2^{3\alpha+1}\ell_2\gamma_n^{-2}}\left(2\sqrt{2}\right)^\alpha = 2^{3\alpha+1}\ell_2\gamma_n^{-2} = \tilde{\beta}_n.$$

ii) The result is straightforward and follows from the definition of β_n given in lemma 6.3.9.

□

In the case where (γ_n) is constant, corollary 6.3.10 shows that we obtain exactly the same result as the one derived in [37] for the exponential case (recalled in lemma 6.3.3 in this paper). In the case of polynomial decay, the result of corollary 6.3.10 provides a slightly degraded result with respect to the one presented in [37] (recalled in lemma 6.3.2). The most important difference relies in the fact that the authors get a convergence rate of order $\mathcal{O}(n^{-\alpha+1/2+\varepsilon})$ whereas the present results yields a convergence in $\mathcal{O}(n^{-\alpha+1})$.

It has so far not been possible to derive better convergence rates in the polynomial case for a general behavior of the sequence (Λ_n). Therefore, in an attempt to recover the convergence of order $\mathcal{O}(n^{-\alpha+1/2+\varepsilon})$ in the polynomial case, we propose to assume a particular behavior of the Lebesgue constant. In the case case where (Λ_n) presents a polynomial increasing behavior

$$\Lambda_n = \mathcal{O}(n^\zeta),$$

lemma 6.3.11 shows that the convergence is of order $\mathcal{O}(n^{-\alpha+\zeta+1/2+\varepsilon})$, which is, in some sense, similar to the result of [37].

Lemma 6.3.11 (Polynomial decay of (d_n) and polynomial increase in (Λ_n))**.**
If for $\alpha > 0$, *we have* $d_n \leq C_0 n^{-\alpha}$ *and* $\gamma_n^{-1} \leq C_\zeta n^\zeta$, $n \in \mathbb{N}^*$, *then for any* $\beta > 1/2$, *we have* $\tau_n \leq C_1 n^{-\alpha+\zeta+\beta}$, $n \in \mathbb{N}^*$, *where*

$$C_1 := \max\left\{C_0 2^{\frac{2\alpha^2}{\zeta}}\left(\frac{\zeta+\beta}{\beta-\frac{1}{2}}\right)^\alpha \max\left(1; C_\zeta^{\frac{\zeta+\beta}{\zeta}}\right); \max_{n=1,\ldots,2\lfloor 2(\zeta+\beta)\rfloor+1} n^{\alpha-\zeta-\beta}\right\}.$$

Note that in the above lemma, the constant β has no connection with β_n defined above.

Proof. It follows from the monotonicity of (τ_n) and inequality (6.15) for $N = K = n$ and any $1 \leq m < n$ that:

$$\tau_{2n} \leq \sqrt{2n} \frac{1}{\prod_{i=1}^{n} \gamma_{n+i}^{1/n}} \tau_n^{\delta} d_m^{1-\delta}, \quad \delta := \frac{m}{n} \tag{6.25}$$

Given $\beta > 1/2$, we define $m := \left\lfloor \frac{\beta - \frac{1}{2}}{\zeta + \beta} \right\rfloor + 1$ (so that $m < n$ for $n > 2(\zeta + \beta) > 2\zeta + 1$). It follows that

$$\delta = \frac{m}{n} \in \left(\frac{\beta - \frac{1}{2}}{\zeta + \beta}; \frac{\beta - \frac{1}{2}}{\zeta + \beta} + \frac{1}{n} \right) \tag{6.26}$$

We prove our claim by contradiction. Suppose it is not true and M is the first value where $\tau_M > C_1 M^{-\alpha+\zeta+\beta}$. Clearly, because of the definition of C_1 and the fact that $\tau_n \leq 1$, we must have $M > 2\lfloor 2(\zeta+\beta) \rfloor + 1$ (since $M \geq 2\lfloor 2(\zeta+\beta) \rfloor + 2$). We first consider the case $M = 2n$, and therefore $n \geq \lfloor 2(\zeta + \beta) \rfloor + 1$. From (6.25), we have:

$$\begin{aligned} C_1 (2n)^{-\alpha+\zeta+\beta} < \tau_{2n} &\leq \sqrt{2n} \frac{1}{\prod_{i=1}^{n} \gamma_{n+i}^{1/n}} \tau_n^{\delta} d_m^{1-\delta} \\ &\leq \sqrt{2n} C_\zeta (2n)^{\zeta} C_1^{\delta} n^{\delta(-\alpha+\zeta+\beta)} C_0^{1-\delta} (\delta n)^{-\alpha(1-\delta)}, \end{aligned} \tag{6.27}$$

where we have used the fact that $\tau_n \leq C_1 n^{-\alpha+\zeta+\beta}$ and $d_m \leq C_0 m^{-\alpha}$. It follows that

$$C_1^{1-\delta} < 2^{\alpha-\beta+\frac{1}{2}} C_\zeta C_0^{1-\delta} \delta^{-\alpha(1-\delta)} n^{\frac{1}{2}+\delta(\zeta+\beta)-\beta}$$

and therefore

$$C_1 < 2^{\frac{\alpha-\beta+\frac{1}{2}}{1-\delta}} C_\zeta^{\frac{1}{1-\delta}} C_0 \delta^{-\alpha} n^{\frac{\zeta+\beta}{1-\delta}\left(\delta - \frac{\beta-\frac{1}{2}}{\zeta+\beta}\right)}.$$

Since, for $n \geq \lfloor 2(\zeta + \beta) \rfloor + 1 > 2(\zeta + \beta)$, we have

$$\begin{aligned} \delta &< \frac{\beta - \frac{1}{2}}{\zeta + \beta} + \frac{1}{n} \\ &< \frac{\beta}{\zeta + \beta}, \end{aligned} \tag{6.28}$$

then,

$$\frac{1}{1-\delta} < \frac{\zeta + \beta}{\zeta}. \tag{6.29}$$

Hence,

$$\frac{\zeta+\beta}{1-\delta}\left(\delta - \frac{\beta-\frac{1}{2}}{\zeta+\beta}\right) < \left(\frac{\zeta+\beta}{1-\delta}\right)\frac{1}{n} < \frac{(\zeta+\beta)^2}{\zeta}\frac{1}{n}, \tag{6.30}$$

where we have used inequalities (6.28) and (6.29). By using (6.30), it follows that

$$n^{\frac{\zeta+\beta}{1-\delta}\left(\delta-\frac{\beta-\frac{1}{2}}{\zeta+\beta}\right)} < n^{\frac{(\zeta+\beta)^2}{\zeta}\frac{1}{n}} < 2^{\frac{(\zeta+\beta)^2}{\zeta}}. \tag{6.31}$$

This yields:

$$C_1 < 2^{\frac{\alpha-\beta+\frac{1}{2}}{1-\delta}} C_\zeta^{\frac{1}{1-\delta}} C_0 \delta^{-\alpha} 2^{\frac{(\zeta+\beta)^2}{\zeta}} < 2^{\left(\frac{\zeta+\beta}{\zeta}\right)(\alpha+\zeta+\frac{1}{2})} C_\zeta^{\frac{1}{1-\delta}} C_0 \delta^{-\alpha}. \tag{6.32}$$

Furthermore, for $-\alpha + \zeta + \beta < 0$ (which is the meaningful case), and using the fact that $\beta > \frac{1}{2}$, we have:

$$2^{\frac{\zeta+\beta}{\zeta}(\alpha+\zeta+\frac{1}{2})} < 2^{\frac{\alpha}{\zeta}(\alpha+\zeta+\beta)} < 2^{\frac{2\alpha^2}{\zeta}} \tag{6.33}$$

and

$$C_\zeta^{\frac{1}{1-\delta}} < \max\left(1; C_\zeta^{\frac{\zeta+\beta}{\zeta}}\right) \tag{6.34}$$

Also, from (6.26), we have

$$\delta^{-\alpha} < \left(\frac{\zeta+\beta}{\beta-\frac{1}{2}}\right)^{\alpha} \tag{6.35}$$

By inserting inequalities (6.33), (6.34) and (6.35) in relation (6.32), the desired contradiction follows:

$$C_1 < C_0 2^{\frac{2\alpha^2}{\zeta}} \left(\frac{\zeta+\beta}{\beta-\frac{1}{2}}\right)^{\alpha} \max\left(1; C_\zeta^{\frac{\zeta+\beta}{\zeta}}\right).$$

Likewise, if $M = 2n+1$, hence is odd, the actually $M \geq 2\lfloor 2(\zeta+\beta)\rfloor + 3$, implying that $n \geq \lfloor 2(\zeta+\beta)\rfloor + 1$:

$$C_1 2^{-\alpha+\zeta+\beta}(2n)^{-\alpha+\zeta+\beta} < C_1(2n+1)^{-\alpha+\zeta+\beta} < \tau_{2n+1} \leq \tau_{2n}. \tag{6.36}$$

But, since from equation (6.27) we have

$$\tau_{2n} \leq \sqrt{2n} C_\zeta (2n)^\zeta C_1^\delta n^{\delta(-\alpha+\zeta+\beta)} C_0^{1-\delta} (\delta n)^{-\alpha(1-\delta)}, \tag{6.37}$$

then, following the same argument as above, we get:

$$\begin{aligned} C_1 &< C_0 2^{\left(\frac{\zeta+\beta}{\zeta}\right)\left(\frac{1}{2}+2\alpha-\beta\right)} \left(\frac{\zeta+\beta}{\beta-\frac{1}{2}}\right)^{\alpha} \max\left(1; C_\zeta^{\frac{\zeta+\beta}{\zeta}}\right) && (6.38)\\ &< C_0 2^{\frac{2\alpha^2}{\zeta}} \left(\frac{\zeta+\beta}{\beta-\frac{1}{2}}\right)^{\alpha} \max\left(1; C_\zeta^{\frac{\zeta+\beta}{\zeta}}\right), && (6.39) \end{aligned}$$

where we have used the fact that $\beta > 1/2$ in the last inequality. □

6.3.3 Convergence rates of the interpolation error

Lemmas 6.3.8, 6.3.9 are the keys to derive the decay rates of the interpolation error of the GEIM greedy algorithm for any behavior of the sequence (γ_n). This is the purpose of the following theorem:

Theorem 6.3.12 (Convergence rates for GEIM in a Banach space)**.**

i) Assume that $d_n \leq C_0 n^{-\alpha}$ for any $n \geq 1$, then the interpolation error of the GEIM greedy selection process satisfies for any $\varphi \in F$ the inequality $\|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq (1+\Lambda_n) C_0 \beta_n n^{-\alpha}$, where the parameter β_n is defined as in lemma 6.3.8.

ii) Assume that $d_n \leq C_0 e^{-c_1 n^\alpha}$ for $n \geq 1$ and $C_0 \geq 1$, then the interpolation error of the GEIM greedy selection process satisfies for any $\varphi \in F$ the inequality $\|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq (1+\Lambda_n) C_0 \beta_n e^{-c_2 n^\alpha}$, where β_n and c_2 are defined as in lemma 6.3.9.

Proof. It can be inferred from equation (6.5) that, $\forall \varphi \in F$, $\|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq (1+\Lambda_n)\|\varphi - P_n(\varphi)\|_{\mathcal{X}} \leq (1+\Lambda_n)\tau_n$ according to the definition of τ_n. We conclude the proof by bounding τ_n thanks to lemmas 6.3.8, 6.3.9. □

From corollary 6.3.10 and lemma 6.3.11, we can also derive convergence rates in the case where (Λ_n) is a monotonically increasing sequence. This is summarized in

Corollary 6.3.13. *If (Λ_n) is a monotonically increasing sequence, then the sequence (γ_n) in the GEIM procedure is monotonically decreasing. The following decay rates in the generalized interpolation error can be inferred:*

i) If $d_n \leq C_0 n^{-\alpha}$ *for any* $n \geq 1$*, then the interpolation error of the GEIM greedy selection process can be bounded as*

$$\forall \varphi \in F, \quad \|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq \begin{cases} 2C_0 \left(1 + \frac{1}{\eta}\right)(1 + \Lambda_1), & \text{if } n = 1. \\ C_0 2^{3\alpha+1} \ell_2 \frac{(1+\Lambda_n)^3}{\eta^2} n^{-\alpha}, & \text{if } n \geq 2. \end{cases}$$

If we write n *as* $n = 4\ell + k$ *(with* $\ell \in \{0, 1, \ldots\}$ *and* $k \in \{0, 1, 2, 3\}$*), then* $\ell_2 = 2\left(\ell + \lceil \frac{k}{4} \rceil\right)$.

ii) If $d_n \leq C_0 e^{-c_1 n^{\alpha}}$ *for* $n \geq 1$ *and* $C_0 \geq 1$*, then the interpolation error of the GEIM greedy selection process can be bounded as*

$$\forall \varphi \in F, \quad \|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq \begin{cases} 2C_0 \left(1 + \frac{1}{\eta}\right)(1 + \Lambda_1), & \text{if } n = 1, \\ C_0 \sqrt{2} \frac{(1+\Lambda_n)^2}{\eta} \sqrt{n} e^{-c_2 n^{\alpha}}, & \text{if } n \geq 2, \end{cases}$$

where $c_2 = c_1 2^{-2\alpha-1}$.

iii) If $d_n \leq C_0 n^{-\alpha}$ *and that* $\gamma_n^{-1} \leq C_\zeta n^{\zeta}$ *for any* $n \geq 1$*, then the interpolation error of the GEIM greedy selection process satisfies for any* $\beta > 1/2$*,*

$$\forall \varphi \in F, \quad \|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq \eta C_\zeta C_1 n^{-\alpha+2\zeta+\beta},$$

where the parameter C_1 *is defined as in lemma 6.3.11.*

Proof. *i*) and *ii*) easily follow from corollary 6.3.10 and *iii*) is derived by using lemma 6.3.11. □

Remark 6.3.14. *The evolution of the Lebesgue constant* Λ_n *as a function of* n *is a subject of great interest. From the theoretical point of view, crude estimates exist and provide an exponential upper bound. This is however far from being what has been obtained in practical applications where, for a reasonable enough choice of the dictionary* Σ*, the sequence* (Λ_n) *presents, in the worst case scenario, a linearly increasing behavior (see [77] for a discussion about this issue and also [11], [55], [80] in the case of the traditional EIM). Assuming this type of behavior for* (Λ_n)*, from corollary 6.3.13-iii, it follows that a polynomial decrease of the Kolmogorov* n*-width of order* $\mathcal{O}\left(n^{-3}\right)$ *should be enough to ensure the convergence of the interpolation error of GEIM.*

6.4 Convergence rates of GEIM in a Hilbert space

6.4.1 Preliminary notations and properties

In this section, $\mathcal{X}$ is a Hilbert space equipped with its induced norm $\|f\|_{\mathcal{X}} = (f, f)_{\mathcal{X}}$, where $(.,.)_{\mathcal{X}}$ is the scalar product in $\mathcal{X}$.

In the same spirit as in the case of a Banach space, we define the sequence (τ_n) as in formula 6.9 but now, for any $f \in F$, $P_n(f)$ corresponds to the unique element of X_n that is the orthogonal projection of f onto X_n. Note that lemma 6.3.1 still holds in the Hilbert setting. We address the task of deriving convergence rates for the interpolation of GEIM by applying the same technique of section 6.3, i.e. by first deriving convergence rates on (τ_n) (see section 6.4.2). The obtained results will be compared to the ones presented in [37] in corollary 3.3 for the case $\gamma_n = \gamma$ and that are recalled here:

Lemma 6.4.1 (Corollary 3.3 – *(ii)* of [37] – Polynomial decay rates for (τ_n) when $\gamma_n = \gamma$)**.** *If* $d_n \leq C_0 n^{-\alpha}$ *for* $n = 1, 2, \ldots$*, then* $\tau_n \leq C_1 n^{-\alpha}$*,* $n = 1, 2, \ldots$*, with* $C_1 = 2^{5\alpha+1} \gamma^{-2} C_0$.

Lemma 6.4.2 (Corollary 3.3 – *(iii)* of [37] – Exponential decay rates for (τ_n) when $\gamma_n = \gamma$)**.** *If* $d_n \leq C_0 e^{-c_1 n^{\alpha}}$ *for* $n = 1, 2, \ldots$*, then* $\tau_n < \sqrt{2C_0} \gamma^{-1} e^{-c_2 n^{\alpha}}$*,* $n = 1, 2, \ldots$*, where* $c_2 = 2^{-1-2\alpha} c_1$.

6.4.2 Convergence rates for (τ_n)

In order to extend lemmas 6.4.1 and 6.4.2 to the more general case where γ depends on the dimension n, the following preliminary theorem is required:

Theorem 6.4.3. *For any $N \geq 0$, consider a weak Greedy algorithm with the property of lemma 6.3.1 and constant γ_N. We have the following inequalities between τ_N and d_N: for any $K \geq 1$, $1 \leq m < K$*

$$\prod_{i=1}^{K} \tau_{N+i}^2 \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^2} \left(\frac{K}{m}\right)^m \left(\frac{K}{K-m}\right)^{K-m} \tau_{N+1}^{2m} d_m^{2(K-m)}.$$

Proof. See appendix B. □

This theorem yields corollaries 6.4.4 and 6.4.5, that are the analogue for the Hilbert setting of corollaries 6.3.6 and 6.3.7.

Corollary 6.4.4. *For $N \geq 1$, we have*

$$\tau_n \leq \sqrt{2} \frac{1}{\prod_{i=1}^{n} \gamma_i^{1/n}} \min_{1 \leq m < n} d_m^{\frac{n-m}{n}}. \tag{6.40}$$

In particular,

$$\tau_{2n} \leq \sqrt{2} \frac{1}{\prod_{i=1}^{2n} \gamma_i^{\frac{1}{2n}}} \sqrt{d_n}. \tag{6.41}$$

Corollary 6.4.5. *For $N \geq 0$, $K \geq 1$ and $1 \leq m < K$:*

$$\tau_{N+K} \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^{1/K}} \sqrt{2} \tau_{N+1}^{m/K} d_m^{1-m/K}. \tag{6.42}$$

Proofs of corollaries 6.4.4 and 6.4.4. The proofs of these two results follow very similar guidelines as corolaries 6.3.6 and 6.3.7. The only difference is that here the staring point is theorem 6.4.3 instead of 6.3.5. □

The absence of the factor $\sqrt{n}$ in these corollaries will be the key to derive improved results in Hilbert spaces.

Using theorem 6.4.3, convergence rates in the sequence (τ_n) when (d_n) has a polynomial or an exponential decay can be inferred and lead to lemmas 6.4.6 and 6.4.7. In these results, no assumption on the behavior of (γ_n) has been made:

Lemma 6.4.6 (Polynomial decay of (d_n))**.** *For any $n \geq 1$, let $n = 4\ell + k$ (where $\ell \in \{0, 1, \ldots\}$ and $k \in \{0, 1, 2, 3\}$). Assume that there exists a constant $C_0 > 0$ such that $\forall n \geq 1$, $d_n \leq C_0 n^{-\alpha}$, then*

$$\tau_n \leq C_0 \beta_n n^{-\alpha}, \tag{6.43}$$

where

$$\beta_n = \beta_{4\ell+k} := \begin{cases} 2\left(1 + \dfrac{1}{\eta}\right) & \text{if } n = 1 \\ \sqrt{2\beta_{\ell_1}} \dfrac{1}{\prod_{i=1}^{\ell_2} \gamma_{\ell_1 - \lceil \frac{k}{4} \rceil + i}^{\frac{1}{\ell_2}}} (2\sqrt{2})^{\alpha} & \text{if } n \geq 2 \end{cases}$$

and $\ell_1 = 2\ell + \lfloor \frac{2k}{3} \rfloor$, $\ell_2 = 2\left(\ell + \lceil \frac{k}{4} \rceil\right)$.

Proof. The proof is similar to the one proposed in lemma 6.3.8: the case $n = 1$ directly follows from lemma 6.3.4 and in the case $n \geq 2$, we write $n = N + K$ with $N \geq 0$ and $K \geq 2$. Corollary 6.4.5 yields:

$$\tau_{N+K} \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^{1/K}} \sqrt{2} \tau_{N+1}^{m/K} d_m^{1-m/K}.$$

By using that $d_m \leq C_0 m^{-\alpha}$ and the recurrence hypothesis $\tau_{N+1} \leq \beta_{N+1}(N+1)^{-\alpha}$, we get:

$$\tau_{N+K} \leq C_0 \sqrt{2} \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^{\frac{1}{K}}} \beta_{N+1}^{\frac{m}{K}} \xi(N,K,m)^{\alpha} (N+K)^{-\alpha},$$

where $\xi(N,K,m) = \dfrac{N+K}{m} \left(\dfrac{m}{N+1} \right)^{\frac{m}{K}}$ for any $1 \leq m < K$ and any given index $n = N + K \geq 2$, where $N \geq 0$ and $K \geq 2$. It suffices now to decompose any $n \geq 2$ as $n = 4\ell + k$ with $\ell \in \{0, 1, \ldots\}$ and $k \in \{0, 1, 2, 3\}$ and use the same choices of N, K and m described in the proof of lemma 6.3.8 to derive the result. □

Lemma 6.4.7 (Exponential decay in (d_n))**.** *Assume that there exists a constant $C_0 \geq 1$ such that $\forall n \geq 1$, $d_n \leq C_0 e^{-c_1 n^{\alpha}}$, then*

$$\tau_n \leq C_0 \beta_n e^{-c_2 n^{\alpha}},$$

where $c_2 := c_1 2^{-2\alpha-1}$ and

$$\beta_n := \begin{cases} 2\left(1 + \dfrac{1}{\eta}\right), & \text{if } n = 1 \\ \sqrt{2} \dfrac{1}{\prod_{i=1}^{2\lfloor \frac{n}{2} \rfloor} \gamma_i^{\frac{1}{2\lfloor \frac{n}{2} \rfloor}}}, & \text{if } n \geq 2. \end{cases}$$

Proof. The proof is the same as lemma 6.3.9 but uses corollary 6.4.4 instead of corollary 6.3.6. □

As in the Banach cases, it is important to study the convergence rates in the particular case where (Λ_n) is monotonically increasing. The following corollary accounts for it.

Corollary 6.4.8. *In the case where (γ_n) is a monotonically decreasing sequence, the following bounds can be derived for τ_n:*

i) If $d_n \leq C_0 n^{-\alpha}$ for any $n \geq 1$, then $\tau_n \leq C_0 \tilde{\beta}_n n^{-\alpha}$, with

$$\tilde{\beta}_n := \begin{cases} 2\left(1 + \dfrac{1}{\eta}\right), & \text{if } n = 1 \\ 2^{3\alpha+1} \left(\dfrac{1}{\gamma_n}\right)^2, & \text{if } n \geq 2. \end{cases}$$

ii) If $d_n \leq C_0 e^{-c_1 n^{\alpha}}$ for $n \geq 1$ and $C_0 \geq 1$, then $\tau_n \leq C_0 \tilde{\beta}_n e^{-c_2 n^{-\alpha}}$, with

$$\tilde{\beta}_n := \begin{cases} 2\left(1 + \dfrac{1}{\eta}\right), & \text{if } n = 1 \\ \sqrt{2} \dfrac{1}{\gamma_n}, & \text{if } n \geq 2. \end{cases}$$

Proof. The proof is derived by following the same guidelines as the proof of corollary 6.3.10. □

Remark 6.4.9. *As a direct consequence of corollary 6.4.8, if γ_n is constant, we recover slighly better results as the ones in [37] for $n \geq 2$ (see lemmas 6.4.1 and 6.4.2 above).*

6.4.3 Convergence rates of the interpolation error

Lemmas 6.4.6 and 6.4.7 are the keys to derive the decay rates of the interpolation error of the GEIM Greedy algorithm. This is the purpose of the following theorem that is based on the inequality

$$\forall \varphi \in F,\ \|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq (1+\Lambda_n)\|\varphi - P_n(\varphi)\|_{\mathcal{X}} \leq (1+\Lambda_n)\tau_n,$$

together with lemmas 6.4.6 and 6.4.7 :

Theorem 6.4.10 (Convergence rates for GEIM)**.**

1. *Assume that $d_n \leq C_0 n^{-\alpha}$ for any $n \geq 1$, then the interpolation error of the GEIM Greedy selection process satisfies for any $\varphi \in F$ the inequality $\|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq (1+\Lambda_n)C_0\beta_n n^{-\alpha}$, where the parameter β_n is defined as in lemma 6.4.6.*
2. *Assume that $d_n \leq C_0 e^{-c_1 n^{\alpha}}$ for $n \geq 1$ and $C_0 \geq 1$, then the interpolation error of the GEIM Greedy selection process satisfies for any $\varphi \in F$ the inequality $\|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq (1+\Lambda_n)C_0\beta_n e^{-c_2 n^{\alpha}}$, where β_n and c_2 are defined as in lemma 6.4.7.*

Then, similarly as in the previous section, we derive

Corollary 6.4.11. *If (Λ_n) is a monotonically increasing sequence, using corollary 6.4.8, the following decay rates in the generalized interpolation error can be derived:*

- *For any $\varphi \in F$, if $d_n \leq C_0 n^{-\alpha}$ for any $n \geq 1$, then the interpolation error of the GEIM Greedy selection process can be bounded as*

$$\|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq \begin{cases} 2C_0\left(1+\dfrac{1}{\eta}\right)(1+\Lambda_1), & \text{if } n=1. \\ C_0 2^{3\alpha+1}\dfrac{(1+\Lambda_n)^3}{\eta^2} n^{-\alpha}, & \text{if } n \geq 2. \end{cases}$$

- *For any $\varphi \in F$, if $d_n \leq C_0 e^{-c_1 n^{\alpha}}$ for $n \geq 1$ and $C_0 \geq 1$, then the interpolation error of the GEIM Greedy selection process can be bounded as*

$$\|\varphi - \mathcal{J}_n[\varphi]\|_{\mathcal{X}} \leq \begin{cases} 2C_0\left(1+\dfrac{1}{\eta}\right)(1+\Lambda_1), & \text{if } n=1, \\ C_0\sqrt{2}\dfrac{(1+\Lambda_n)^2}{\eta} e^{-c_2 n^{\alpha}}, & \text{if } n \geq 2, \end{cases}$$

where $c_2 = 2^{-2\alpha-1}$.

6.5 Conclusion

Under the hypothesis of polynomial or exponential decay of the Kolmogorov n-width $d_n(F, \mathcal{X})$, it has been proven that the convergence rates of the interpolation error in GEIM are nearly-optimal and that the lack of optimality comes from the Lebesgue constant of the method that, depending on the case, impacts of the convergence by adding terms of order $\mathcal{O}(\Lambda_n^2)$ or $\mathcal{O}(\Lambda_n^3)$.

Given the fact that, for reasonable enough dictionaries Σ, it has been observed in practical applications that (Λ_n) is linear in the worst case scenario (see [11], [55], [80], [77]), our results prove that a decay of order $\mathcal{O}(n^{-3})$ in $d_n(F, \mathcal{X})$ should be enough to ensure the convergence of the interpolation errors of GEIM.

Acknowledgements

This work was supported in part by the joint research program MANON between CEA-Saclay and University Pierre et Marie Curie-Paris 6.

Appendix A: proof of Theorem 6.3.5

We begin by recalling a preliminary lemma for matrices that is proven in [37].

Lemma 6.5.1. *Let $G = (g_{i,j})$ be a $K \times K$ lower triangular matrix with rows $\boldsymbol{g_1}, \ldots, \boldsymbol{g_K}$, W be any m dimensional subspace of $\mathbb{R}^K$, and P be the orthogonal projection of $\mathbb{R}^K$ onto W. Then,*

$$\prod_{i=1}^{K} g_{i,i}^2 \leq \left\{ \frac{1}{m} \sum_{i=1}^{K} \|P\boldsymbol{g_i}\|_{\ell_2}^2 \right\}^m \left\{ \frac{1}{K-m} \sum_{i=1}^{K} \|\boldsymbol{g_i} - P\boldsymbol{g_i}\|_{\ell_2}^2 \right\}^{K-m} \tag{6.44}$$

where $\|.\|_{\ell_2}$ is the euclidean norm of a vector in $\mathbb{R}^K$.

For the proof of theorem 6.3.5, we consider a lower triangular matrix $A = (a_{i,j})_{i,j=1}^{\infty}$ defined in the following way. For each $j = 1, \ldots,$ we let $\lambda_j \in \mathcal{L}(\mathcal{X})$ be the linear functional of norm one that satisfies:

$$(i)\ \lambda_j(X_j) = 0, \qquad (ii)\ \lambda_j(\varphi_j) = \mathrm{dist}(\varphi_j, X_j), \tag{6.45}$$

where $X_j = \mathrm{span}\{\varphi_0, \ldots, \varphi_{j-1}\}$, $j = 1, 2, \ldots,$ is the interpolating space given by the greedy algorithm of GEIM. The existence of such a functional is a consequence of the Hahn-Banach theorem. We let A be the matrix with entries

$$a_{i,j} = \lambda_j(\varphi_i).$$

Lemma 6.5.2. *The matrix A has the following properties:*
P1: *The diagonal elements of A satisfy $\gamma_n \tau_n \leq a_{n,n} \leq \tau_n$*
P2: *For every $j < i$, one has: $|a_{i,j}| \leq \mathrm{dist}(\varphi_i, X_j) \leq \tau_j$.*
P3: *For every $j > i$, $a_{i,j} = 0$.*

Proof.

P1: We have

$$a_{j,j} = \lambda_j(\varphi_j) = \mathrm{dist}(\varphi_j, X_j) = \|\varphi_j - P_j(\varphi_j)\|_{\mathcal{X}} \leq \max_{\varphi \in F} \|\varphi - P_j(\varphi)\|_{\mathcal{X}} = \tau_j.$$

Lemma 6.3.1 directly yields the second part of the inequality: $a_{j,j} \geq \gamma_j \tau_j$.

P2: For any $j < i$ and any $g \in X_j$, we have

$$|a_{i,j}| = |\lambda_j(\varphi_i)| = |\lambda_j(\varphi_i - g)| \leq \|\lambda_j\|_{\mathcal{L}(\mathcal{X})} \|\varphi_j - g\|_{\mathcal{X}},$$

where we have used the fact that $\lambda_j(g) = 0$ because $g \in X_j$. Therefore, since $\|\lambda_j\|_{\mathcal{L}(\mathcal{X})} = 1$,

$$|a_{i,j}| \leq \|\varphi_j - g\|_{\mathcal{X}}, \quad \forall\ g \in X_j,$$

hence

$$|a_{i,j}| \leq \|\varphi_i - P_j(\varphi_i)\|_{\mathcal{X}} \leq \tau_j.$$

P3: Clearly, for $j > i$, $a_{i,j} = \lambda_j(\varphi_i) = 0$ because $\varphi_i \in X_j$ in this case.

□

We can now prove theorem 6.3.5, i.e.:

Theorem 6.5.3. *For any $N \geq 0$, consider a weak Greedy algorithm with the property of lemma 6.3.1 and constant γ_N. We have the following inequalities between τ_N and d_N: for any $K \geq 1$, $1 \leq m < K$*

$$\prod_{i=1}^{K} \tau_{N+i}^2 \leq \frac{1}{\prod_{i=1}^{K} \gamma_{N+i}^2} 2^K K^{K-m} \left(\sum_{i=1}^{K} \tau_{N+i}^2 \right)^m d_m^{2(K-m)} \tag{6.46}$$

Proof. We consider the $K \times K$ matrix G which is formed by the rows and columns of A with indices from $\{N+1, \ldots, N+K\}$. Let Y_m be the Kolmogorov subspace of $\mathcal{X}$ for which $\operatorname{dist}(F, Y_m) = d_m(F, \mathcal{X})$. For each i, there is an element $h_i \in Y_m$ such that

$$\|\varphi_i - h_i\|_{\mathcal{X}} = \operatorname{dist}(\varphi_i, Y_m) \leq d_m(F, \mathcal{X})$$

and therefore

$$|\lambda_j(\varphi_i) - \lambda_j(h_i)| = |\lambda_j(\varphi_i - h_i)| \leq \|\lambda_j\|_{\mathcal{L}(\mathcal{X})} \|\varphi_i - h_i\|_{\mathcal{X}} \leq d_m(F, \mathcal{X}). \tag{6.47}$$

We now consider the vectors $(\lambda_{N+1}(h), \ldots, \lambda_{N+K}(h))$, $h \in X_m$. They span a space $W \subset \mathbb{R}^K$ of dimension $\leq m$. We assume that $\dim(W) = m$ (a slight notational adjustment has to be made if $\dim(W) < m$). It follows from (6.47) that each row $\boldsymbol{g_i}$ of G can be approximated by a vector from W in the ℓ_∞ norm to accuracy d_m, and therefore in the ℓ_2 norm to accuracy $\sqrt{K} d_m$. Let P be the orthogonal projection of $\mathbb{R}^K$ onto W. Hence, we have

$$\|\boldsymbol{g_i} - P\boldsymbol{g_i}\|_{\ell_2} \leq \sqrt{K} d_m, \quad i = 1, \ldots, K. \tag{6.48}$$

It also follows from property P2 that

$$\|P\boldsymbol{g_i}\|_{\ell_2} \leq \|\boldsymbol{g_i}\|_{\ell_2} \leq \left\{ \sum_{j=1}^{i} \tau_{N+j}^2 \right\}^{1/2},$$

and therefore

$$\sum_{i=1}^{K} \|P\boldsymbol{g_i}\|_{\ell_2}^2 \leq \sum_{i=1}^{K} \sum_{j=1}^{i} \tau_{N+j}^2 \leq K \sum_{i=1}^{K} \tau_{N+i}^2. \tag{6.49}$$

Next, we apply lemma 6.5.1 for this G and W and use property P1 and estimates (6.48) and (6.49) to derive

$$\begin{aligned}
\prod_{i=1}^{K} \gamma_{N+i}^2 \tau_{N+i}^2 &\leq \left\{ \frac{K}{m} \sum_{i=1}^{K} \tau_{N+i}^2 \right\}^m \left\{ \frac{K^2}{K-m} d_m^2 \right\}^{K-m} \\
&= K^{K-m} \left(\frac{K}{m} \right)^m \left(\frac{K}{K-m} \right)^{K-m} \left\{ \sum_{i=1}^{K} \tau_{N+i}^2 \right\}^m d_m^{2(K-m)} \\
&\leq 2^K K^{K-m} \left\{ \sum_{i=1}^{K} \tau_{N+i}^2 \right\}^m d_m^{2(K-m)},
\end{aligned}$$

where we have used the fact that $x^{-x}(1 - x^{x-1}) \leq 2$ for $0 < x < 1$. This completes the proof. □

Appendix B: proof of Theorem 6.4.3

In this section, $\mathcal{X}$ is a Hilbert space. We will denote by $(\varphi_n^*)_{n\geq 0}$ the orthonormal system obtained from $(\varphi_n)_{n\geq 0}$ by Gram-Schmidt orthonormalisation. It follows that the orthogonal projector P_n from $\mathcal{X}$ onto X_n is given by

$$P_n\varphi = \sum_{i=0}^{n-1}(\varphi,\varphi_i^*)_{\mathcal{X}}\varphi_i^*, \quad n = 1,2,\dots$$

and, in particular,

$$\varphi_n = P_{n+1}\varphi_n = \sum_{j=0}^{n} a_{n,j}\varphi_j^*, \quad a_{n,j} = (\varphi_n,\varphi_j^*)_{\mathcal{X}},\ j\leq n.$$

There is no loss of generality in assuming that the infinite dimensional Hilbert space $\mathcal{X}$ is $\ell_2(\mathbb{N})$ and that $\varphi_j^* = e_j$, where e_j is the vector with a one in the coordinate indexed by j and is zero in all other coordinates, i.e. $(e_j)_i = \delta_{j,i}$.

In a similar manner as in the Banach space case, we associate with the greedy procedure of GEIM the lower triangular matrix:

$$A := (a_{i,j})_{i,j=0}^{\infty}, \quad a_{i,j} := 1,\ j > i.$$

This matrix incorporates all the information about the weak greedy algorithm on F. The following two properties characterize any lower triangular matrix A generated by such a greedy algorithm.

Lemma 6.5.4. *The matrix A has the following two properties:*

P1: *The diagonal elements of A satisfy $\gamma_n\tau_n \leq |a_{n,n}| \leq \tau_n$.*

P2: *For every $m \geq n$, one has $\sum_{j=n}^{m} a_{m,j}^2 \leq \tau_n^2$.*

Proof.

P1: For any $n \geq 1$, since $\varphi_n - P_n\varphi_n = a_{n,n}\varphi_n^*$, it follows that For any $n \geq 1$, $|a_{n,n}| = \|\varphi_n - P_n\varphi_n\| \leq \tau_n$. The fact that $|a_{n,n}| \geq \gamma_n\tau_n$ directly follows from lemma 6.3.1.

P2: For $m \geq n$, $\sum_{j=n}^{m} a_{m,j}^2 = \|\varphi_m - P_n\varphi_m\|^2 \leq \max_{\varphi\in F}\|\varphi - P_n\varphi\|^2 = \tau_n^2$.

□

We can now prove theorem 6.4.3, i.e.

Theorem 6.5.5. *For any $N \geq 0$, consider a weak Greedy algorithm with the property of lemma 6.3.1 and constant γ_N. We have the following inequalities between τ_N and d_N: for any $K \geq 1$, $1 \leq m < K$*

$$\prod_{i=1}^{K}\tau_{N+i}^2 \leq \frac{1}{\prod_{i=1}^{K}\gamma_{N+i}^2}\left(\frac{K}{m}\right)^m\left(\frac{K}{K-m}\right)^{K-m}\tau_{N+1}^{2m}d_m^{2(K-m)}.$$

Proof. We consider the $K\times K$ matrix $G = (g_{i,j})$ which is formed by the rows and columns of A with indices from $\{N+1,\dots,N+K\}$. Each row $\boldsymbol{g_i}$ is the restriction of φ_{N+i} to the coordinates $N+1,\dots,N+K$. Let Y_m be the Kolmogorov subspace of $\mathcal{X}$ for which $\operatorname{dist}(F,Y_m) = d_m(F,\mathcal{X})$. Then, $\operatorname{dist}(\varphi_{N+i},Y_m) \leq d_m,\ i = 1,\dots,K$. Let $\tilde{W}$ be the linear subspace which is the restriction of Y_m to the coordinates $N+1,\dots,N+K$. In general, $\dim(\tilde{W}) \leq m$. Let W be an m dimensional

space, $W \subset \text{span}\{e_{N+1}, \ldots, e_{N+K}\}$, such that $\tilde{W} \subset W$ and P and $\tilde{P}$ are the projections in $\mathbb{R}^K$ onto W and $\tilde{W}$, respectively. Clearly,

$$\|P\boldsymbol{g_i}\|_{\ell_2} \leq \|\boldsymbol{g_i}\|_{\ell_2} \leq \tau_{N+1}, \quad i = 1, \ldots, K, \tag{6.50}$$

where we have used property P2 in the last inequality. Note that

$$\|\boldsymbol{g_i} - P\boldsymbol{g_i}\|_{\ell_2} \leq \|\boldsymbol{g_i} - \tilde{P}\boldsymbol{g_i}\|_{\ell_2} = \text{dist}\left(\boldsymbol{g_i}, \tilde{W}\right) \leq \text{dist}\left(\varphi_{N+i}, Y_m\right) \leq d_m, \quad i = 1, \ldots, K. \tag{6.51}$$

It follows from property P1 that

$$\prod_{i=1}^{K} |a_{N+i,N+i}|^2 \geq \prod_{i=1}^{K} \gamma_{N+i}^2 \tau_{N+i}^2. \tag{6.52}$$

We now apply lemma 6.5.1 for this G and W, and use estimates (6.50), (6.51) and (6.52) to derive the result.

□

Chapter 7

Improvement of cheap approximations by a post-processing/reduced basis rectification method

This is an ongoing work with Y. Maday and B. Stamm.

Introduction

In chapters 4, 5 and 6, the properties of the Empirical Interpolation Methods (EIM-GEIM) have been explored. The replacement of point evaluations by the application of linear forms is the interesting feature to define interpolation processes not only for continuous functions like in the classical interpolation framework but also for functions in general Banach spaces. In addition to this, the use of linear forms models in a more faithful manner sensor measurements and makes of GEIM an interesting tool for the reconstruction of (real) experiments. The accuracy of the GEIM approximations has been analyzed and compared with the best approximation provided by the Kolmogorov n-width, in particular, the loss in accuracy has been quantified. But, what if the sensor locations lead to an unsatisfactory approximation with GEIM despite the selection of the Greedy algorithm? Assuming that we have at our disposal another more accurate — and therefore more computationally expensive— approximation operator, we can think of improving the accuracy of our approximation by using the rectification method used in [21] and [59]. The idea consists in a reduced basis post-processing strategy aiming at recovering the accuracy of the good approximation without sacrificing on the computational complexity.

In this framework, the purpose of the present work is to revisit the original idea of rectification of [21] and [59] in order to:

- enlarge the range of application of the method
- try to explain its efficiency

For this, we will place ourselves in the following general framework: let $\mathcal{X}$ be a Hilbert space equipped with the norm $\|.\|_{\mathcal{X}}$ and the scalar product $(.,.)_{\mathcal{X}}$.

The goal is to accurately approximate any $f \in F$ where F is a given set in $\mathcal{X}$ by elements of some given finite dimensional subspace $X_M \subset \mathcal{X}$ of small dimension M. Suppose that we have at our disposal two approximation operators:

- $\pi_M : \mathcal{X} \mapsto X_M$ that provides a computationally expensive and accurate approximation of the elements of F, i.e. such that
$$\sup_{f \in F} \|f - \pi_M[f]\|_{\mathcal{X}}$$
is small enough for the application under consideration,

- $\mathcal{J}_M : \mathcal{X} \mapsto X_M$ that provides a cheap and inaccurate approximation of the elements of F, i.e. such that

$$\sup_{f \in F} \|f - \mathcal{J}_M[f]\|_{\mathcal{X}}$$

  is not small enough for our standards.

We wish to discuss about the hypothesis under which one can build a rectification operator $\tilde{\pi}_M : \mathcal{X} \mapsto X_M$ from evaluations of $\mathcal{J}_M$ that preserves a comparable accuracy as π_M in the sense that

$$\sup_{f \in F} \|f - \tilde{\pi}_M[f]\|_{\mathcal{X}} \approx \sup_{f \in F} \|f - \pi_M[f]\|_{\mathcal{X}},$$

but that circumvents the computational cost of π_M. This general problem in $\mathcal{X}$ seems to be hopeless if we do not set an additional hypothesis on the set F where we want to have the previous property. This one can be for instance to assume that F is a compact subset of $\mathcal{X}$ of small Kolmogorov n-width (F can be, e.g., the set of solutions of a parameter dependent PDE as was the case in [21] and [59]). Let the $\mathcal{S}_P = \text{span}\{h_j\}_{j=1}^P$ denote a P-dimensional space with $1 \leq P$ and whose deviation from F

$$E(F, \mathcal{S}_P) = \sup_{f \in F} \inf_{h \in \mathcal{S}_P} \|f - h\|_{\mathcal{X}}$$

is not far from achieving the Kolmogorov width $d_P(F, \mathcal{X})$. For this reason, let us call it "representative space of F".

The only hypothesis on the accuracy of the approximations π_M and $\mathcal{J}_M$ is that we assume that both sets $\{\pi_M[h_1], \ldots, \pi_M[h_M]\}$ and $\{\mathcal{J}_M[h_1], \ldots, \mathcal{J}_M[h_M]\}$ are linearly independent.

Remark 7.0.6. *In the context of reduced basis methods, $\mathcal{S}_P$ would be called the reduced basis space of the manifold F. Note also that in this paper, there is no need to choose X_M equal to S_P.*

The idea is to use the existence of these "special functions" within the representative space of F to help in improving the approximation mapping $\mathcal{J}_M$.

As an example, in the works of [21] and [59], the operators π_M and $\mathcal{J}_M$ were often Galerkin projections:

- in [21], π_M was a finite element Galerkin projection on a fine spatial mesh whose accuracy was recovered by post-processing a coarse mesh finite element Galerkin projection $\mathcal{J}_M$. In that case $X_M = S_P$ and was spanned by accurate approximation based on the fine finite element mesh.
- in [59], π_M was an orthogonal projection but $\mathcal{J}_M$ was a reduced basis Galerkin projection. In that also case $X_M = S_P$ but was spanned by externally given reduced basis — in particular not provided by a Galerkin process.

Since, in this work, we place ourselves in a general case, $\pi_M/\mathcal{J}_M$ can represent any kind of linear or non linear expensive and accurate/cheap and inaccurate approximation.

We will therefore be able to cover other interesting cases, like, e.g.

- π_M is an orthogonal projection and $\mathcal{J}_M$ is the generalized interpolant, or
- π_M is the interpolation with Chebyshev points and $\mathcal{J}_M$ is the interpolation with equidistant points.

The chapter is organized as follows: in section 7.1, we will revisit the definition of the rectification operator $\tilde{\pi}_M$ and show in what sense our formulation is an extension of the previous works. We will also discuss about the hypothesis under which $\tilde{\pi}_M$ can recover the accuracy of the accurate operator π_M without significantly increasing the computational cost of the cheap approximation $\mathcal{J}_M$. Section 7.2 will be devoted to the derivation of a simple formula to do rectification in practice. Finally, in section 7.3, we will present a first numerical example in which π_M is the interpolation with Chebyshev points and $\mathcal{J}_M$ is the interpolation with equidistant points.

7.1 Definition of the rectification operator

7.1.1 Definition of the rectification operator in the linear case

In the case where π_M and $\mathcal{J}_M$ are both linear, and as long as $P = M$, the rectification is constructed following the works presented in [21] and [59].

The rectification process is a linear operator (a matrix) that allows to map each $\mathcal{J}_M[h_j]$ over $\pi_M[h_j]$ for $j = 1, \dots, M$. Let $q_1, \dots, q_M$ be a basis set of X_M, then the rectification matrix R_M simply maps each vector $[\beta_{\cdot,j}]_i$ on the vector $[\alpha_{\cdot,j}]_i$ where $\mathcal{J}_M[h_j] = \sum_i \beta_{i,j} q_i$ and $\pi_M[h_j] = \sum_i \alpha_{i,j} q_i$ (let us recall that we have made the assumption that both $\{\pi_M[h_1], \dots, \pi_M[h_M]\}$ and $\{\mathcal{J}_M[h_1], \dots, \mathcal{J}_M[h_M]\}$ are linearly independent).

The rectification process then maps, for any $f \in F$ the approximation $\mathcal{J}_M[f] = \sum_i \beta_i q_i$ on $\tilde{\pi}_M[f]$ defined by $\tilde{\pi}_M[f] = \sum_i \alpha_i q_i$ where $[\alpha]_i = R_P[\beta]_i$. The construction is explained in section 7.2.

The following theorem explains to which extent the rectification process allows to get a better accuracy than $\mathcal{J}_M[f]$:

Theorem 7.1.1. *If $\mathcal{J}_M$ and π_M are linear operators, then*

$$\forall f \in F, \quad \|f - \tilde{\pi}_M[f]\|_{\mathcal{X}} \leq \|f - \pi_M[f]\|_{\mathcal{X}} + \|\pi_M - \tilde{\pi}_M\|_{\mathcal{L}(\mathcal{X})} \inf_{h \in \mathcal{S}_{\mathcal{P}}} \|f - h\|_{\mathcal{X}}.$$

Proof. Due to relation (7.4) and the fact that $\mathcal{J}_M$ and π_M are linear, we have that

$$\tilde{\pi}_M[h] = \pi_M[h], \quad \forall h \in \mathcal{S}_M. \tag{7.1}$$

For any $f \in F$,

$$\|f - \tilde{\pi}_M[f]\|_{\mathcal{X}} \leq \|f - \pi_M[f]\|_{\mathcal{X}} + \|\pi_M[f] - \tilde{\pi}_M[f]\|_{\mathcal{X}}.$$

But, from the linearity of π_M and $\tilde{\pi}_M$ and relation (7.1),

$$\|\pi_M[f] - \tilde{\pi}_M[f]\|_{\mathcal{X}} = \|\pi_M[f - h] - \tilde{\pi}_M[f - h]\|_{\mathcal{X}}, \quad \forall h \in \mathcal{S}_M.$$

Therefore, for any $f \in F$ and any $h \in \mathcal{S}_M$,

$$\begin{aligned} \|f - \tilde{\pi}_M[f]\|_{\mathcal{X}} &\leq \|f - \pi_M[f]\|_{\mathcal{X}} + \|\pi_M[f - h] - \tilde{\pi}_M[f - h]\|_{\mathcal{X}} \\ &= \|f - \pi_M[f]\|_{\mathcal{X}} + \|(\pi_M - \tilde{\pi}_M)[f - h]\|_{\mathcal{X}} \\ &\leq \|f - \pi_M[f]\|_{\mathcal{X}} + \|\pi_M - \tilde{\pi}_M\|_{\mathcal{L}(\mathcal{X})} \|[f - h]\|_{\mathcal{X}}, \end{aligned}$$

and the result easily follows. □

Remark 7.1.2. *Theorem 7.1.1 provides interesting informations to explain the quality of the approximation of $\tilde{\pi}_M$. Indeed, the best possible performance for our rectification procedure is to provide the same error as π_M. Hence, the smaller the product*

$$\|\pi_M - \tilde{\pi}_M\|_{\mathcal{L}(\mathcal{X})} \inf_{h \in \mathcal{S}_{\mathcal{P}}} \|f - h\|_{\mathcal{X}}, \tag{7.2}$$

the more efficient the rectification procedure. Unfortunately, the operator norm $\|\pi_M - \tilde{\pi}_M\|_{\mathcal{L}(\mathcal{X})}$ involves somehow the operator norm $\|\mathcal{J}_M\|_{\mathcal{L}(\mathcal{X})}$ and can be ill conditioned. However, $\inf_{h \in \mathcal{S}_M} \|f - h\|_{\mathcal{X}}$ is small since

$$\inf_{h \in \mathcal{S}_M} \|f - h\|_{\mathcal{X}} \leq \sup_{f \in F} \inf_{h \in \mathcal{S}_M} \|f - h\|_{\mathcal{X}} = E(F, \mathcal{S}_M)$$

and $E(F, \mathcal{S}_M)$ is close to $d_M(F, \mathcal{X})$. The trade-off between these two terms will therefore determine the success of our approximation.

Remark 7.1.3. *Note that the results of [21] and [59] satisfy the hypothesis of theorem 7.1.1.*

7.1.2 Definition of the rectification operator in the general case

We now propose here a construction of a rectification operator $\tilde{\pi}_M : \mathcal{X} \mapsto X_M$ that is extends the previous construction to the case where, possibly, $\tilde{\pi}_M$ can be, in full generality, a non-linear operator and $P \leq M$.

Our starting point is the same as before and we define:

$$\tilde{\pi}_M[f] := R_M \circ \mathcal{J}_M[f], \quad \text{if } f \in \mathcal{X} \setminus X_M, \tag{7.3}$$

where $R_M : X_M \to X_M$ will be called the rectification map. The difference with respect to [21] and [59] is the way R_M is now going to be defined.

In order to define R_M, we require that

$$\tilde{\pi}_M[h_j] = \pi_M[h_j] \tag{7.4}$$

for the elements h_j of the basis of $\mathcal{S}_P$. Relation (7.4) can equivalently be written as

$$\pi_M[h_j] = R_M\left[\mathcal{J}_M[h_j]\right], \ j \in \{1, \dots, P\}. \tag{7.5}$$

That is, we impose that the rectification is exact for all elements of the basis h_j. We now define the spaces:

$$X^{(\pi)} = \text{span}\{\pi_M[h_1], \dots, \pi_M[h_P]\} \tag{7.6}$$

and

$$X^{(\mathcal{J})} = \text{span}\{\mathcal{J}_M[h_1], \dots, \mathcal{J}_M[h_P]\}, \tag{7.7}$$

whose dimension is lower or equal to P. For any, $\varphi \in X_M$, we define the rectification map as:

$$R_M[\varphi] = r(P[\varphi]), \tag{7.8}$$

where $P[\varphi]$ is the orthogonal projection of φ onto $X^{(\mathcal{J})}$ in the $\mathcal{X}$ norm and $r : X^{(\mathcal{J})} \to X^{(\pi)}$ is a linear mapping. Its construction can be found thanks to relation (7.5) but this point will be explained in detail in section 7.2.

Note that, in the present construction, the rectification map R_M given in (7.8) is linear. However, since $\mathcal{J}_M$ could be non linear, the rectification operator $\tilde{\pi}_M$ will be, in general, non linear. The definition of $\tilde{\pi}_M$ in the above described form is an extension of the framework under which [21] and [59] have worked. Indeed, in the particular case where $P = M$ and $\{\mathcal{J}_M[h_1], \dots, \mathcal{J}_M[h_M]\}$ are linearly independent, then the spaces $X^{(\mathcal{J})}$ and X_M are equal. Therefore $P[\varphi] = \varphi$, $\forall \varphi \in X_M$, and R_M is only a linear mapping. This is exactly the rectification scheme used in [21] and [59], where, in addition, $\mathcal{J}_M$ and π_M were linear.

Note that the process allows to build a rectification operator when the representative space $\mathcal{S}_P$ of F has a smaller dimension than the approximation space X_M.

We now present the following theorem, which is an attempt to explain the hypothesis under which rectification can recover the accuracy of π_M.

Theorem 7.1.4. *Assume that $\tilde{\pi}_M$ is Lipschitz continuous with continuity constant k and that there exists $\varepsilon_1,\ \varepsilon_2 > 0$ such that:*

$$\begin{cases} E(F, \mathcal{S}_P) \leq \varepsilon_1 \sup\limits_{f \in F} \|f - \pi_M[f]\|_{\mathcal{X}} \\ \|\pi_M[h] - \tilde{\pi}_M[h]\|_{\mathcal{X}} \leq \varepsilon_2, \quad \forall h \in \pi_{\mathcal{S}_P}(F), \end{cases} \tag{7.9}$$

then,

$$\forall f \in F, \quad \|f - \tilde{\pi}_M[f]\|_{\mathcal{X}} \leq (1 + \varepsilon_1(1 + k)) \sup_{f \in F} \|f - \pi_M[f]\|_{\mathcal{X}} + \varepsilon_2. \tag{7.10}$$

Proof. For any $f \in F$, let $f_{\mathcal{S}_P}$ be the projection of f onto $\mathcal{S}_P$, i.e,

$$f_{\mathcal{S}_P} = \arg\inf_{h \in \mathcal{S}_P} \|f - h\|_{\mathcal{X}}.$$

It therefore follows that:

$$\begin{aligned}
\|f - \tilde{\pi}_M[f]\|_{\mathcal{X}} &\leq \|f - f_{\mathcal{S}_P}\|_{\mathcal{X}} + \|f_{\mathcal{S}_P} - \tilde{\pi}_M[f]\|_{\mathcal{X}} \\
&\leq \|f - f_{\mathcal{S}_P}\|_{\mathcal{X}} + \|f_{\mathcal{S}_P} - \pi_M[f_{\mathcal{S}_P}]\|_{\mathcal{X}} \\
&\qquad + \|\pi_M[f_{\mathcal{S}_P}] - \tilde{\pi}_M[f_{\mathcal{S}_P}]\|_{\mathcal{X}} + \|\tilde{\pi}_M[f_{\mathcal{S}_P}] - \tilde{\pi}_M[f]\|_{\mathcal{X}} \\
&\leq (1+k)\|f - f_{\mathcal{S}_P}\|_{\mathcal{X}} + \|f_{\mathcal{S}_P} - \pi_M[f_{\mathcal{S}_P}]\|_{\mathcal{X}} + \|\pi_M[f_{\mathcal{S}_P}] - \tilde{\pi}_M[f_{\mathcal{S}_P}]\|_{\mathcal{X}},
\end{aligned}$$

where we have used the Lipschitz continuity in the last inequality. By noticing that

$$\|f - f_{\mathcal{S}_P}\|_{\mathcal{X}} \leq E(F, \mathcal{S}_P)$$

and using the hypothesis of (7.9), we derive the desired result:

$$\|f - \tilde{\pi}_M[f]\|_{\mathcal{X}} \leq (1 + \varepsilon_1(1+k)) \sup_{f \in F} \|f - \pi_M[f]\|_{\mathcal{X}} + \varepsilon_2.$$

□

Remark 7.1.5. *Note that the previous lemma provides a strategy to improve the accuracy of the rectification approximation: if, for a given dimension M, one wishes to improve the accuracy, it is possible to achieve this (up to some extend) by increasing the dimension $P \leq M$ of $\mathcal{S}_P$. Indeed, this will decrease the term $\inf_{h \in \mathcal{S}_P} \|f - h\|_{\mathcal{X}}$ and help minimize the product (7.2). We refer to section 7.3 for an illustration of this idea in a numerical example.*

In the following section we will present in detail how to derive R_M in practice.

7.2 A formula to derive the rectification map R_M in practice

Let us remind that $\{q_1, \ldots, q_M\}$ is a basis of X_M and let un introduce the symmetric positive definite matrix Q_M given by

$$Q_M = (Q_{i,j})_{1 \leq i,j \leq M}, \quad Q_{i,j} = (q_i, q_j)_{\mathcal{X}}, \ 1 \leq i, j \leq M. \tag{7.11}$$

For any $j = 1, \ldots, P$, we will write

$$\mathcal{J}_M[h_j] = \sum_{i=1}^{M} \beta_{i,j}^{(M)} q_i$$

and

$$\pi_M[h_j] = \sum_{i=1}^{M} \alpha_{i,j}^{(M)} q_i,$$

for the basis functions h_j of the reduced basis $\mathcal{S}_P$. With these notations, we have:

Lemma 7.2.1. *The rectification map $R_M : X_M \to X_M$ expressed in the basis $\{q_1, \ldots, q_M\}$ is the $M \times M$ matrix:*

$$R_M = P_M \tilde{I}_M^+, \tag{7.12}$$

where $P_M = \left(\alpha_{i,j}^{(M)}\right)_{\substack{1 \leq i \leq M \\ 1 \leq j \leq P}}$, $I_M = \left(\beta_{i,j}^{(M)}\right)_{\substack{1 \leq i \leq M \\ 1 \leq j \leq P}}$ and

$$\tilde{I}_M^+ := \left(I_M^T Q_M I_M\right)^{-1} I_M^T Q_M.$$

Proof. Any $\varphi = \sum_{i=1}^{M} \varphi_i q_i$ of X_M can be expressed as a vector of $\mathbb{R}^M$, $\boldsymbol{\varphi} = \left(\varphi_1, \ldots, \varphi_M\right)^T$. The orthogonal projection $P(\varphi)$ of φ onto $X^{(\mathcal{J})}$ satifies the minimization problem

$$\begin{aligned}
\|\varphi - P(\varphi)\|_{\mathcal{X}} &= \min_{y \in X^{(\mathcal{J})}} \|\varphi - y\|_{\mathcal{X}} \\
&= \min_{\boldsymbol{y} \in \mathbb{R}^P} \| \sum_{i=1}^{M} \varphi_i q_i - \sum_{j=1}^{P} y_j \mathcal{J}_M[h_j] \|_{\mathcal{X}} \\
&= \min_{\boldsymbol{y} \in \mathbb{R}^P} \| \sum_{i=1}^{M} \varphi_i q_i - \sum_{i=1}^{M} \sum_{j=1}^{P} \beta_{i,j} y_j q_i \|_{\mathcal{X}} \\
&= \min_{\boldsymbol{y} \in \mathbb{R}^P} \langle \boldsymbol{\varphi} - I_M \boldsymbol{y}, Q_M \left(\boldsymbol{\varphi} - I_M \boldsymbol{y}\right) \rangle_M ,
\end{aligned}$$

where $\langle ., . \rangle_M$ denotes the euclidean scalar product in $\mathbb{R}^M$. Since Q_M is symmetric positive definite, we can use its Choleski factorization and write

$$Q_M = L_M L_M^T, \tag{7.13}$$

where L_M is a lower triangular matrix. Hence,

$$\|\varphi - P(\varphi)\|_{\mathcal{X}} = \min_{\boldsymbol{y} \in \mathbb{R}^P} \| L_M^T \left(\boldsymbol{\varphi} - I_M \boldsymbol{y}\right) \|_M \tag{7.14}$$

where $\|.\|_M$ denotes the euclidean norm in $\mathbb{R}^M$. Equation (7.14) is a classical least squares problem. The solution $\tilde{\boldsymbol{y}} \in \mathbb{R}^P$ is not necessarily unique and satisfies the normal equations

$$\begin{aligned}
& \left(L_M^T I_M\right)^T \left(L_M^T I_M\right) \tilde{\boldsymbol{y}} = \left(L_M^T I_M\right)^T L_M^T \boldsymbol{\varphi} \\
\Leftrightarrow \quad & I_M^T Q_M I_M \tilde{\boldsymbol{y}} = I_M^T Q_M \boldsymbol{\varphi}.
\end{aligned} \tag{7.15}$$

Among these, we define $\boldsymbol{P(\varphi)} \in \mathbb{R}^P$ as the unique solution with minimum ℓ_2-norm, which is given by

$$\boldsymbol{P(\varphi)} = \tilde{I}_M^+ \boldsymbol{\varphi}, \tag{7.16}$$

with $\tilde{I}_M^+ := \left(I_M^T Q_M I_M\right)^{-1} I_M^T Q_M$. We now look for the expression of the linear map $r : X_M \to X_M$ in the basis $\{q_1, \ldots, q_M\}$. The map r can be represented by a matrix $(r_{i,j})_{1 \leq i,j \leq M}$ whose coefficients can be found thanks to relation (7.4). For any $j \in \{1, \ldots, P\}$, if we represent $\pi[h_j]$ as $\boldsymbol{\pi[h_j]} = (\alpha_{1,j}, \ldots, \alpha_{M,j})^T$ in the canonical basis of $\mathbb{R}^M$, we can derive thanks to (7.4) that:

$$\boldsymbol{\pi[h_j]} = \begin{pmatrix} \alpha_{1,j} \\ \vdots \\ \alpha_{M,j} \end{pmatrix} = r\left(P(\mathcal{J}_M[h_j])\right) = r\left(\tilde{I}_M^+ \begin{pmatrix} \beta_{1,j} \\ \vdots \\ \beta_{M,j} \end{pmatrix}\right) = r e_j = \begin{pmatrix} r_{1,j} \\ \vdots \\ r_{M,j} \end{pmatrix}, \tag{7.17}$$

where we have used that $\tilde{I}_M^+ \begin{pmatrix} \beta_{1,j} \\ \vdots \\ \beta_{M,j} \end{pmatrix} = e_j$, where e_j is the j-th canonical vector of $\mathbb{R}^M$. It follows that $r = P_M$, hence the final formula $R_M = P_M \tilde{I}_M^+$. □

The particular case treated in [21] and [59] is explained in

Corollary 7.2.2. *If $P = M$ and $\{\mathcal{J}_M[h_1], \dots, \mathcal{J}_M[h_M]\}$ are linearly independent, then*

$$R_M = P_M I_M^{-1},$$

which is the formula used in [21] and [59].

Proof. Since, in this case, the spaces $X^{(\mathcal{J})}$ and X_M are equal, we will have that the projection $P(\varphi)$ of any $\varphi \in X_M$ onto $X^{(\mathcal{J})}$ is φ itself, the rectification map defined in (7.8) reads in this case:

$$\forall \varphi \in X_M,\ R_M[\varphi] = r(\varphi).$$

Furthermore, from the linear independence property of the set $\{\mathcal{J}_M[h_1], \dots, \mathcal{J}_M[h_M]\}$, the matrix I_M is invertible. If we represent $\pi[h_j]$ as $\boldsymbol{\pi[h_j]} = (\alpha_{1,j}, \dots, \alpha_{M,j})^T$ and $\mathcal{J}_M[h_j]$ as $\boldsymbol{\mathcal{J}_M[h_j]} = (\beta_{1,j}, \dots, \beta_{M,j})^T$ and r as $\boldsymbol{r} = (r_{i,j})_{1 \le i,j \le M}$ in the canonical basis of $\mathbb{R}^M$, relation (7.4) yields:

$$\boldsymbol{\pi[h_j]} = \begin{pmatrix} \alpha_{1,j} \\ \vdots \\ \alpha_{M,j} \end{pmatrix} = \boldsymbol{r}\boldsymbol{\mathcal{J}_M[h_j]} = \boldsymbol{r} \begin{pmatrix} \beta_{1,j} \\ \vdots \\ \beta_{M,j} \end{pmatrix}, \quad \forall j = 1, \dots, M. \tag{7.18}$$

Hence

$$P_M = \boldsymbol{r} I_M, \tag{7.19}$$

and the result follows from the invertibility of I_M.

□

Once R_M has been obtained, any $f \in F$ can be quickly approximated by computing $\mathcal{J}_M[f]$ in the basis $\{q_1, \dots, q_M\}$ and then applying R_M (whose use costs $\mathcal{O}(M^2)$ additional operations). If the hypothesis of theorem 7.1.4 are fulfilled, the accuracy of the approximation with $\tilde{\pi}$ will be the one given by π. We however point out that the proposed strategy will be efficient if it is divided in two phases:

- An offline (and costly) stage in which R_M is derived. This phase is costly because one needs to find the reduced space $\mathcal{S}_P$ and the computation of R_M requires the evaluation of the elements $\pi[h_j]$, $j \in \{1, \dots, M\}$) as well as the evaluation of $\tilde{I}_M^+$.
- An online stage in which, given any $f \in F$, $\tilde{\pi}[f]$ is quickly performed.

7.3 A numerical result

We present here an example in 2D over the region $\Omega = \{(x, y) \in [-1, 1] \times [-1, 1]\}$ in which we consider the rectification process in the L^2-norm of the following compact set of continuous functions:

$$F = \left\{ f(., .; \mu_x, \mu_y, c_x, c_y) \mid (\mu_x, \mu_y, c_x, c_y) \in [0, 5]^4 \right\}, \tag{7.20}$$

where, for any $(x, y) \in \Omega$,

$$f(x, y; \mu_x, \mu_y, c_x, c_y) = \left(\frac{1}{1 + (25 + \mu_x \cos(c_x \pi x))x^2} \right) \left(\frac{1}{1 + (25 + \mu_y \cos(c_y \pi y))y^2} \right).$$

We are going to approximate the elements of F in the finite dimensional polynomial space:

$$X_M = \text{span}\{x^n y^p,\ 0 \le n, p \le m\},$$

whose dimension is $\dim(X_M) = M = (m + 1)^2$. In our particular example, we define:

- π_M as the L^2 orthogonal projection onto X_M.

- $\mathcal{J}_M$ as the interpolation on equidistant points.

While π_M is the best polynomial approximation that one can build in the L^2 norm, $\mathcal{J}_M$ is far from optimal in our case. Indeed, for any $f \in \mathcal{C}(\Omega)$:

$$\|f - \mathcal{J}_M[f]\|_{L^2(\Omega)} \leq |\Omega| \|f - \mathcal{J}_M[f]\|_{L^\infty(\Omega)} \leq |\Omega|(1 + \Lambda_M) \inf_{y \in X_M} \|f - y\|_{L^\infty(\Omega)},$$

where Λ_M is the Lebesgue constant on equidistant points. The exponential growth of Λ_M with M leads to an unstable interpolation operator. Given that π_M is far more computationally expensive than $\mathcal{J}_M$, we wish to recover the accuracy that π_M can provide by computing $\mathcal{J}_M$ and applying the rectification process.

For this, we extract in an offline phase a P-dimensional reduced basis $\mathcal{S}_P$ of F by proper orthogonal decomposition and build the rectification matrix R_M following (7.12). Once R_M has been derived, we have access to $\tilde{\pi}_M = R_M \circ \mathcal{J}_M$ that can be used in the online stage as a cheap approximation operator. The performances of $\tilde{\pi}_M$ have been tested over 1000 snapshots of F and the worst errors in L^2 are shown in figure 7.1. Several values have been tested for the parameter P.

We can observe first of all that the performances of the rectification process improve as the number P of POD functions involved to construct R_M is increased. As pointed out in remark 7.1.2, this is to be expected due to the fact that the deviation $E(F, \mathcal{S}_P)$ between F and $\mathcal{S}_P$ decreases as the dimension P increases. The quantity ε_{POD} is given in the legend as an estimation of this deviation. The case $P = 1$ is of particular interest to comment because it shows that, in this example, the rectification process with only one POD function already improves dramatically the quality of the approximation with respect to the original interpolation. Also, note that, in the case $P = M$, the rectification process allows not only to improve the error regarding interpolation but we recover **exactly** the accuracy provided by the orthogonal projection. This reveals the fact that the product (7.2) becomes negligible and we are currently verifying this fact.

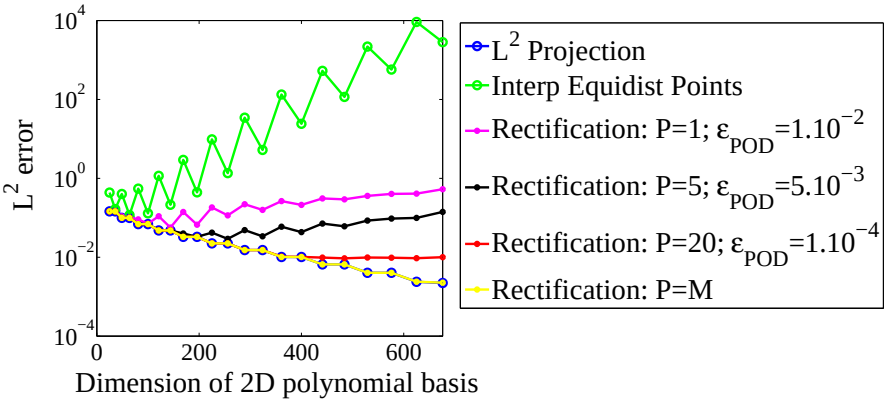


Figure 7.1: Performances of the three different polynomial approximation techniques to approximate the set F: interpolation on equidistant points, projection and rectification.

Acknowledgments

This work was supported in part by the joint research program MANON between CEA-Saclay and University Pierre et Marie Curie-Paris 6.

Conclusions and perspectives

Damit das Mögliche entsteht, muss immer wieder das Unmögliche versucht werden. (Hermann Hesse)

In the present work, we have first of all included in MINARET a time-dependent module to solve the multigroup neutron transport S_N equations with a discontinuous Galerkin finite element discretization for the space. A very special stress has been put on the inclusion of acceleration techniques in order to deal with the computational complexity in reasonable computing times. Some of the methods take advantage of modern computer architectures like the parallelization of the angular and time variables. The latter has been implemented with the parareal in time algorithm.

This development will be useful for several applications. First of all, the tool is important for safety calculations in the nuclear industry for the analysis of fast transients and in cases of strong anisotropy of the flux. We note on this direction the ongoing PhD of A. Targa [116] in which the aim is to study fast accidents with coupled neutronics, thermal-hydraulics and fuel-mechanics (MINARET will be used for the neutronics part).

Furthermore, since the resolution of the time-dependent transport equation is done through the resolution of a source problem, the implemented module will also be used for other studies. A very interesting example are vessel fluence studies in which a project on this topic is currently ongoing (see [97]). The main goal is to see whether vessel fluence calculations can be performed by deterministic calculations instead of the classical Monte Carlo ones. This would dramatically reduce the time computation and open the door to uncertainty quantification studies. It is expected that this kind of analysis could shed some light in the uncertainty of the flux measured by a sensor when there exists uncertainty in the knowledge of the cross-sections. In this study, it has been necessary at some point to compare MINARET's calculations with reference Monte Carlo ones (obtained with the TRIPOLI ® code, [19]) and therefore the work is also being useful for MINARET's validation. The results on this topic are very satisfactory so far.

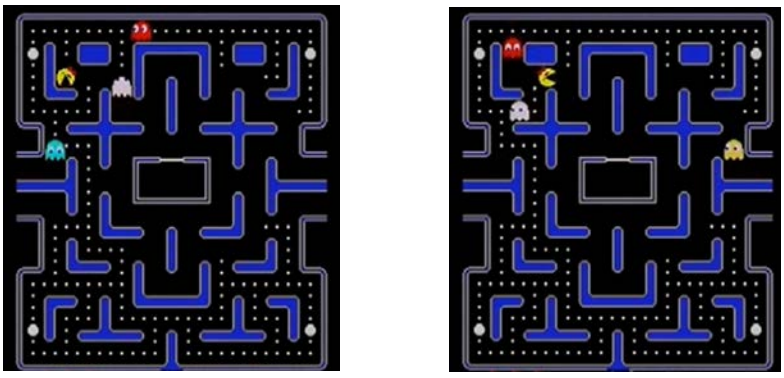
The second main contribution of this work has been to build an extension of the Empirical Interpolation Method (GEIM). The idea consists in working with interpolating continuous linear functionals instead of interpolating points. This presents the major advantage of relaxing the classical continuity requirement in the target functions and defines a concept of interpolation that is applicable in Banach spaces. The main theoretical properties such as the well-posedness and convergence decay rates (under the hypothesis of small Kolmogorov n-width) have been explored. Special attention has been given to the understanding of the concept of Lebesgue constant Λ_M in a Banach space. In the particular case of Hilbert spaces, an explicit formula for Λ_M has been derived thanks to the fact that the generalized interpolant can be seen as an oblique projection in this case. It has also been explained that the Greedy algorithm optimizes in some sense this constant. Despite this advances, no theoretical knowledge of the behavior of Λ_M in EIM nor GEIM is known and a very interesting (and challenging) task would be to explore this topic in future works.

The numerical results shown in chapters 4 and 5 illustrate the potential applications of this development: used in a reduced basis framework, GEIM can be used to build a tool for the real-

time monitoring of a physical or industrial process. This can be done by combining measurements collected from the process itself with a mathematical model (a parameter dependent PDE) that represents the physical understanding of the process under consideration. This idea has been illustrated through a parameter dependent Stokes problem and also a Laplace problem in which the geometry was considered as a parameter. Taking advantage of this idea, it has been explained at the end of chapter 5 how this scheme could be refined to build an adaptive tool for the supervision of experiments that could distinguish between normal and accidental conditions. We believe that this tool could help in taking real-time decisions regarding the security of processes. A potential application of special importance to this work is the monitoring of the neutron population during the operation of a reactor core by combining MINARET's calculations with measurements coming from the core itself. Note also that the uncertainty quantification problem for vessel fluence issues has many ingredients that GEIM requires (a parameter dependent problem with sensors) and it might be possible to use GEIM for this kind of study.

There are however several issues that still need to be addressed in GEIM to accomplish such a task (see figure 7.2). Among the most important stand:

- The treatment of transport problems with reduced basis on a regular basis. This is a challenging task and an analysis of this issue has been explored in [30] where particular Petrov-Galerkin variational formulations combined with certain stabilization techniques seem to be the key to obtain good convergence results.
- When the number of involved parameters is very large, how to deal with the offline phase in a reasonable time?
- How to include the bias between the true experiment and the manifold of solutions of our parameter dependent PDE? The works of [122] will probably be inspiring to carry out this task.
- How to deal with noisy measurements? One can find some preliminary ideas in [76] and the works of [98].



(a) GEIM over? **(b)** ... No!

Figure 7.2: A state of the art of GEIM through PacMan.

Last but not least, two ongoing works have been presented in chapters 3 and 7. The first one aims at solving the loss in the speed-up performances in the parareal algorithm that can arise when it is coupled with other iterative techniques. This is being done by the study of a numerical scheme in which the internal iterations are truncated and the convergence is obtained across the parareal iterations. The second ongoing work aims at providing a better theoretical understanding to the so-called "rectification method" that has been proposed in the field of reduced basis.

Bibliography

[1] M. L. Adams and E. W. Larsen. Fast iterative methods for discrete-ordinates particle transport calculations. *Progress in Nuclear Energy*, 40(1):3 – 159, 2002.

[2] C. Ahrens and G. Beylkin. Rotationally invariant quadratures for the sphere. *Proceedings of the Royal Society A: Mathematical, Physical and Engineering Science*, 465:3103–3125, 2009.

[3] E.J. Allen. A finite element approach for treating the energy variable in the numerical solution of the neutron transport equation. *Transp. Theory Stat. Phys.*, 15(4):449–478, 1986.

[4] P. F. Antonietti and B. Ayuso. Schwarz domain decomposition preconditioners for discontinuous galerkin approximations of elliptic problems: non-overlapping case. *ESAIM, Math. Model. Numer. Anal.*, 41:21–54, 2007.

[5] M. Asadzadeh. L^2 error estimates for the discrete ordinates method for three-dimnesional neutron transport. *Transp. Theory Stat. Phys.*, 17(1):1–24, 1988.

[6] E. Aubanel. Scheduling of tasks in the parareal algorithm. *Parallel Computing*, 37:172–182, 2011.

[7] L. Baffico, S. Bernard, Y. Maday, G. Turinici, and G. Zérah. Parallel-in-time molecular-dynamics simulations. *Phys. Rev. E*, 66, Nov 2002.

[8] G. Bal. *Couplage d'équations et homogénéisation en transport neutronique*. PhD thesis, Paris VI, 1997.

[9] G. Bal. On the convergence and the stability of the parareal algorithm to solve partial differential equations. In *Domain Decomposition Methods in Science and Engineering*, volume 40 of *Lect. Notes Comp. Sci.*, pages 425–432. Springer Berlin Heidelberg, 2005.

[10] G. Bal and Y. Maday. A "parareal" time discretization for non-linear PDE's with application to the pricing of an American put. *Recent developments in domain decomposition methods*, 23:189–202, 2002.

[11] M. Barrault, Y. Maday, N.C. Nguyen, and A.T. Patera. An empirical interpolation method: Application to efficient reduced-basis discretization of partial differential equations. *C. R. Acad. Sci. Paris, Série I.*, 339:667–672, 2004.

[12] A.-M. Baudron and J.-J. Lautard. MINOS: a simplified P_N solver for core calculation. *Nucl. Sci. Eng.*, 155(2):250–263, 2007.

[13] A.-M. Baudron, J.-J. Lautard, Y. Maday, and O. Mula. The parareal in time algorithm applied to the kinetic neutron diffusion equation. In *21st International Conference on Domain Decomposition Methods*, 2012.

[14] A.-M. Baudron, J.-J. Lautard, K. Riahi, Y. Maday, and J. Salomon. Parareal in time 3D numerical solver for the LWR Benchmark neutron diffusion transient model. *Submitted*, 2014.

[15] R. Becker, M. F. Modest, R. Koch, and H.-J. Bauer. A finite element treatment of the angular dependency of the even-parity equation of radiative transfer. *Journal of Heat Transfer*, 132(023404), 2009.

[16] A.F. Bennett. *Array design by inverse methods*, volume 15. 1985.

[17] L.A. Berry, W. Elwasif, J.M. Reynolds-Barredo, D. Samaddar, R. Sanchez, and D.E. Newman. Event-based parareal: A data-flow based implementation of parareal. *J. Comput. Phys.*, 231(17):5945 – 5954, 2012.

[18] P. Binev, A. Cohen, W. Dahmen, R.A. DeVore, G. Petrova, and P. Wojtaszczyk. Convergence rates for greedy algorithms in reduced basis methods. *SIAM J. Math. Anal.*, 43(3):1457–1472, 2011.

[19] E. Brun, E. Dumonteil, F.X. Hugot, N. Huot, C. Jouanne, Y.K. Lee, F. Malvagi, A. Mazzolo, O. Petit, J.C. Trama, and A. Zoia. Overview of TRIPOLI-4 ® version 7 Continuous-energy Monte Carlo Transport code. In *Proceedings of ICAPP*, 2011.

[20] A. Buffa, Y. Maday, A.T. Patera, C. Prud'Homme, and G. Turinici. A priori convergence of the greedy algorithm for the parametrized reduced basis method. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46 (3):595–603, 2012.

[21] R. Chakir and Y. Maday. A two-grid finite-element/ reduced basis scheme for the approximation of the solution of parametric dependent PDE's. *C. R. Acad. Sci. Paris*, Série I, 347:435–440, 2009.

[22] P. Chartier and B. Philippe. A parallel shooting technique for solving dissipative ODE's. *Computing*, 51(3-4):209–236, 1993.

[23] S. Chauvet. *Méthode multi-échelle pour la résolution des équations de la cinétique neutronique.* PhD thesis, Université de Nantes, 2008.

[24] S. Chauvet, A. Nachaoui, A.-M. Baudron, and J.-J. Lautard. A multi-scale approach for the neutronic kinetics equation using the mixed dual solver minos. In *Joint International Conference on Mathematics and Computation and Supercomputing in Nuclear Applications*, 2007.

[25] A. Chkifa, A. Cohen, and C. Schwab. High-dimensional adaptive sparse polynomial interpolation and applications to parametric PDE's. *Foundations of Computational Mathematics*, pages 1–33, 2013.

[26] B. Cockburn and J. Guzmán. Error Estimates for the Runge-Kutta Discontinuous Galerkin Method for the Transport Equation with Discontinuous Initial Data. *SIAM J. Num. Anal.*, 46(3):1364–1398, 2008.

[27] A. Cohen and R. DeVore. Kolmogorov widths under holomorphic mappings. *Submitted*, 2014.

[28] M. Coste-Delclaux. *Modélisation du phénomène d'autoprotection dans le code de transport multigroupe APOLLO2.* PhD thesis, Conservatoire National des Arts et Métiers, 2006.

[29] D.E. Cullen, C.J. Clouse, R. Procassini, and R.C. Little. Static and dynamic criticality: are they different? Technical Report UCRL-TR-201506, Lawrence Livermore National Laboratory, 2003.

[30] W. Dahmen, C. Plesken, and G. Welper. Double greedy algorithms: Reduced basis methods for transport dominated problems. *ESAIM, Math. Model. Numer. Anal.*, 48:623–663, 5 2014.

[31] X. Dai and Y. Maday. Stable parareal in time method for first- and second-order hyperbolic systems. *SIAM J. Sci. Comput.*, 35(1):A52–A78, 2013.

[32] I. Daubechies. *Ten Lectures on Wavelets.* Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, 1992.

[33] R. Dautray and J.-L. Lions. *Analyse mathématique et calcul numérique.* Masson, 1984.

[34] G.G. Davidson, T.M. Evans, J.J. Joshua, C.G. Basker, and R.N. Slaybaugh. Massively parallel, three-dimensional transport solutions for the k-eigenvalue problem. *Nucl. Sci. Eng.*, 2013. To appear.

[35] P. J. Davis. *Interpolation and Approximation*. Blaisdell Publishing Company, 1963.

[36] V. Deniz. The theory of neutron leakage in reactor lattices. In Y. Ronen, editor, *Handbook of nuclear reactor calculations*, volume II, pages 409–508. CRC, 1986.

[37] R. A. DeVore, G. Petrova, and P. Wojtaszczyk. Greedy algorithms for reduced bases in Banach spaces. *Constructive Approximation*, pages 1–12, 2012.

[38] M. Drohman, B. Haasdonk, and M. Ohlberger. Reduced Basis Approximation for Nonlinear Parametrized Evolution Equations based on Empirical Operator Interpolation. *SIAM J. Sci. Comput.*, 34:A937–A969, 2012.

[39] S. Dulla, E.H. Mund, and P. Ravetto. Accuracy of the predictor-corrector quasi-static method for space-time reactor dynamics. In *PHYSOR, Advances in Nuclear Analysis and Simulation*, 2006.

[40] S. Dulla, E.H. Mund, and P. Ravetto. The quasi-static method revisited. *Progress in Nuclear Energy*, 50:908–920, 2008.

[41] J.L. Eftang, A.T. Patera, and E.M. Rønquist. An" hp" certified reduced basis method for parametrized elliptic partial differential equations. *SIAM J. Sci. Comput.*, 32(6):3170–3200, 2010.

[42] J.L. Eftang and B. Stamm. Parameter multi-domain "hp" empirical interpolation. *Int. J. Numer. Methods Eng.*, 90(4):412–428, 2012.

[43] M. Emmet and M. Minion. Toward an efficient parallel in time method for partial differential equations. *Comm. App. Math. and Comp. Sci.*, 1(1), 2012.

[44] J. Erhel and S. Raoult. Algorithme parallèle pour le calcul d'orbites - Parallélisation à travers le temps. *Techniques et Sciences informatiques*, 19(5), 2000.

[45] A. Ern and J. L. Guermond. *Theory and practice of finite elements*. Springer Verlag, 2004.

[46] P.F. Fischer, F. Hecht, and Y. Maday. A parareal in time semi-implicit approximation of the Navier-Stokes equations. In *Proceedings of Fifteen International Conference on Domain Decomposition Methods*, pages 433–440. Springer Verlag, 2004.

[47] M.S. Floater and K. Hormann. Barycentric rational interpolation with no poles and high rates of approximation. *Numer. Math.*, 107(2):315–331, August 2007.

[48] D. Fournier and R. Le Tellier. An adaptive energy discretization of the neutron transport equation based on a wavelet Galerkin method. *Discrete Wavelet Transforms - Algorithms and Applications, Prof. Hannu Olkkonen (Ed.)*, 2011. Chapter 16.

[49] M.J. Gander and S. Vandewalle. Analysis of the parareal time-parallel time-integration method. *SIAM J. Sci. Comput.*, 29(2):556–578, March 2007.

[50] L. Gastaldi. On a domain decomposition for the transport equation: theory and finite element approximation. *IMA J. Numer. Anal.*, 14(1):111–135, 1994.

[51] E.M. Gelbard. Simplified spherical harmonics equations and their use in shielding problems. Technical Report WAPD-T-1182, Westinghouse report, 1961.

[52] E. Girardi, P. Guérin, S. Dulla, and P. Ravetto. Comparison of direct and quasi-static methods for neutron kinetic calculations with the EDF R&D COCAGNE code. In *PHYSOR, Advances in Reactor Physics*, 2012.

[53] H. Golfier, R. Lenain, C. Calvin, J.-J. Lautard, A.-M. Baudron, P. Fougeras, P. Magat, E. Martinolli, and Y. Dutheillet. APOLLO3: a common project of CEA, AREVA and EDF for the development of a new deterministic multi-purpose code for core physics analysis. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2009.

[54] F. Golse, S. Jin, and C. D. Levermore. A domain decomposition analysis for a two-scale linear transport problem. *ESAIM, Math. Model. Numer. Anal.*, 37:869–892, 2003.

[55] M.A. Grepl, Y. Maday, N.C. Nguyen, and A.T. Patera. Efficient reduced-basis treatment of nonaffine and nonlinear partial differential equations. *ESAIM, Math. Model. Numer. Anal.*, 41(3):575–605, 2007.

[56] R. Guetat. *Méthode de parallélisation en temps: Application aux méthodes de décomposition de domaine*. PhD thesis, Paris VI, 2012.

[57] T.P. Hamilton and P. Pulay. Direct inversion in the iterative subspace (DIIS) optimization of openshell, excited state, and small multiconfiguration SCF wave functions. *Journal of Chemical Physics*, 84:5728, 1986.

[58] F. Hecht. New developments in Freefem++. *J. Numer. Math.*, 20(3-4):251–265, 2012.

[59] H. Herrero, Y. Maday, and F. Pla. RB (Reduced basis) for RB (Rayleigh-Bénard). *Comput. Methods Appl. Mech. Eng.*, 261-262:132–141, 2013.

[60] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer Publishing Company, Incorporated, 2007.

[61] E. Jamelot, A.-M. Baudron, and J.-J. Lautard. Domain Decomposition for the SP_N Solver MINOS. *Transp. Theory Stat. Phys.*, 41(7):495–512, 2012.

[62] E. Jamelot, J. Dubois, J.-J. Lautard, C. Calvin, and A.-M. Baudron. High performance 3D neutron transport on petascale and hybrid architectures within APOLLO3 code. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2011.

[63] C. Johnson and J. Pitkäranta. An analysis of the discontinuous Galerkin method for a scalar hyperbolic equation. *Math. Comp.*, 46:1–26, 1986.

[64] C. T. Kelley. *Iterative Methods for Linear and Nonlinear Equations*. Number 16 in Frontiers in Applied Mathematics. SIAM, 1995.

[65] J. Kleijnen and W. van Beers. Robustness of kriging when interpolating in random simulation with heterogeneous variances: Some experiments. *European Journal of Operational Research*, 165(3):826 – 834, 2005.

[66] A. Kolmogoroff. Über die beste Annäherung von Funktionen einer gegebenen Funktionenklasse. *Annals of Mathematics*, 37:107–110, 1936.

[67] S. Langenbuch, W. Maurer, and W. Werner. Coarse-mesh flux expansion method for the analysis of space-time effects in large light water reactor cores. *Nucl. Sci. Eng.*, 63:437–456, 1977.

[68] J.-J. Lautard, Y. Maday, and O. Mula. MINARET: Towards a parallel 3D time-dependent neutron transport solver. *Submitted*, –:–, 2014.

[69] P. Le Saint and P.A. Raviart. On a finite element method for solving the neutron transport equation. *Mathematical aspects of finite elements in partial differential equations, Academic press, New York*, pages 89–145, 1974.

[70] R. Le Tellier, D. Fournier, and J.M. Ruggieri. A wavelet-based finite element method for the self-shielding issue in neutron transport. *Nucl. Sci. Eng.*, 163(1):34–55, 2009.

[71] Q. Lin and A.H. Zhou. Convergence of the discontinuous Galerkin methods for a scalar hyperbolic equation. *Acta Math. Sci.*, 13:207–210, 1993.

[72] J.L. Lions, Y. Maday, and G. Turinici. Résolution d'EDP par un schéma en temps pararéel. *C. R. Acad. Sci. Paris*, 2001. t. 332, Série I, p. 661-668.

[73] H. Liu and S. Maghsoodloo. Simulation optimization based on taylor kriging and evolutionary algorithm. *Applied Soft Computing*, 11(4):3451 – 3462, 2011.

[74] L. Lunéville. Méthode multibande aux ordonnées discretes. Formalisme et résultats. Technical Report 2832, CEA, 1998.

[75] Y. Maday. The 'Parareal in Time' Algorithm. In F. Magoulès, editor, *Substructuring Techniques and Domain Decomposition Methods*, chapter 2, pages 19–44. Saxe-Coburg Publications, 2010.

[76] Y. Maday and O. Mula. A generalized empirical interpolation method: Application of reduced basis techniques to data assimilation. In Franco Brezzi, Piero Colli Franzone, Ugo Gianazza, and Gianni Gilardi, editors, *Analysis and Numerics of Partial Differential Equations*, volume 4 of *Springer INdAM Series*, pages 221–235. Springer Milan, 2013.

[77] Y. Maday, O. Mula, A.T. Patera, and M. Yano. The generalized Empirical Interpolation Method: stability theory on Hilbert spaces with an application to the Stokes equation. *Submitted*, –:–, 2014.

[78] Y. Maday, O. Mula, and G. Turinici. A priori convergence of the Generalized Empirical Interpolation Method. In *10th international conference on Sampling Theory and Applications (SampTA 2013)*, pages 168–171, 2013.

[79] Y. Maday, O. Mula, and G. Turinici. Convergence analysis of the Generalized Empirical Interpolation Method. *Submitted*, –:–, 2014.

[80] Y. Maday, N.C. Nguyen, A.T. Patera, and G.S.H. Pau. A general multipurpose interpolation procedure: the magic points. *Comm. Pure Appl. Anal.*, 8(1):383–404, 2009.

[81] Y. Maday, A.T. Patera, J.D. Penn, and M. Yano. A Parametrized-Background Data-Weak approach to variational data assimilation: formulation, analysis and application to accoustics. *Int. J. Num. Meth. Eng.*, 2014. Submitted.

[82] Y. Maday, A.T. Patera, and G. Turinici. A priori convergence theory for reduced-basis approximations of single-parameter elliptic partial differential equations. *Journal of Scientific Computing*, 17(1-4):437–446, 2002.

[83] Y. Maday, J. Salomon, and G. Turinici. Monotonic time-discretized schemes in quantum control. *Numerische Mathematik*, 103(2):323–338, 2006.

[84] Y. Maday and B. Stamm. Locally adaptive greedy approximations for anisotropic parameter reduced basis spaces. *SIAM J. Sci. Comput.*, 35(6), 2013.

[85] Y. Maday and G. Turinici. A parareal in time procedure for the control of partial differential equations. *C. R. Math. Acad. Sci. Paris*, Série I, 335(4):387 – 392, 2002.

[86] Y. Maday and G. Turinici. The Parareal in Time Iterative Solver: a Further Direction to Parallel Implementation. In *Domain Decomposition Methods in Science and Engineering*, pages 441–448. Springer Berlin Heidelberg, 2005.

[87] M. Minion. A hybrid parareal spectral deferred corrections method. *Comm. App. Math. and Comp. Sci.*, 5(2), 2010.

[88] M. Mokhtar-Kharroubi. *Mathematical topics in neutron transport theory: new aspects*, volume 46 of *Series on advances in mathematics for applied sciences*. Singapore River Edge, N.J. World Scientific, 1997.

[89] J.-Y. Moller. *Éléments finis courbes et accélération pour le transport de neutrons*. PhD thesis, Université Henri Poincaré , 2012.

[90] J.-Y. Moller and J.-J. Lautard. Minaret, a deterministic neutron transport solver for nuclear core calculations. In *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering*, 2011.

[91] P. Mosca. *Concéption et développement d'un mailleur énergétique adaptatif pour la génération des bibliothèques multigroupes des codes de transport.* PhD thesis, Université Paris XI, 2009.

[92] O. Mula. *Some contributions towards the parallel simulation of time dependent neutron transport and the integration of observed data in real time.* PhD thesis, Paris VI, 2014.

[93] A. Napov and Y. Notay. An algebraic multigrid method with guaranteed convergence rate. *SIAM J. Sci. Comput.*, 34(2):A1079–A1109, 2012.

[94] L. Naymeh. *Analyse et développement d'un schéma de discrétisation numérique de l'équation de transport des neutrons en géométrie tridimensionnelle.* PhD thesis, Paris XI, 2013.

[95] N. C. Nguyen, K. Veroy, and A.T. Patera. Certified real-time solution of parametrized partial differential equations. In *Handbook of Materials Modeling*, pages 1523–1558. Springer, 2005.

[96] J. Nievergelt. Parallel methods for integrating ordinary differential equations. *Communications of the ACM*, 7(12), 1964.

[97] S. Pastoris. Validation of the MINARET solver through vessel fluence source computations. Master's thesis, Politecnico di Torino, 2014.

[98] A.T. Patera and E.M. Rønquist. Regression on parametric manifolds: Estimation of spatial fields, functional outputs, and parameters from noisy data. *C.R. Acad. Sci. Paris*, Series I 350(9-10):543–547, 2012.

[99] A.T. Patera and G. Rozza. *Reduced Basis Approximation and A Posteriori Error Estimation for Parametrized Partial Differential Equations.* Version 1.0, Copyright MIT 2006, to appear in (tentative rubric) MIT Pappalardo Graduate Monographs in Mechanical Engineering., 2006.

[100] A. Pautz and A. Birkhofer. DORT-TD: A transient neutron transport code with fully implicit time integration. *Nucl. Sci. Eng.*, 145:299–319, 2003.

[101] S.D. Pautz. An algorithm for parallel S_N sweeps on unstructured meshes. *Nucl. Sci. Eng.*, 34:111–136, 2002.

[102] T.E. Peterson. A note on the convergence of the discontinuous Galerkin method for a scalar hyperbolic equation. *SIAM J. Numer. Anal.*, 28:133–140, 1991.

[103] J. Pitkäranta and R. Scott. Error estimates for the combined spatial and angular approximations of the transport equation for slab geometry. *SIAM J. Numer. Anal.*, 20(5):922–950, 1983.

[104] C. Prud'homme, D. Rovas, K. Veroy, Y. Maday, A.T. Patera, and G. Turinici. Reliable real-time solution of parametrized partial differential equations: Reduced-basis output bound methods. *J. Fluids Eng.*, 124(1):70–80, 2002.

[105] J.C. Ragusa, J.-L. Guermond, and G. Kanschat. A robust S_N-DG-approximation for radiation transport in optically thick and diffusive regimes. *J. Comput. Physics*, 231(4):1947–1962, 2012.

[106] W.H. Reed and T.R. Hill. Triangular mesh methods for the neutron transport equation. *Los Alamos Scientific Laboratory report*, pages LA–UR–73–479, 1973.

[107] P. Reuss. *Précis de neutronique.* EDP Sciences, Collection Génie Atomique, 2003.

[108] P. Ribon and J.-M. Maillard. Les tables de probabilité – Application au traitement des sections efficaces pour la neutronique. Technical Report 2485, CEA, 1986.

[109] G. Rozza, M. Andrea, and F. Negri. Reduction strategies for PDE-constrained optimization problems in haemodynamics. Technical Report 26.2012, MATHICSE, 2012.

[110] D. Samaddar, D. E. Newman, and R. Sánchez. Parallelization in time of numerical simulations of fully-developed plasma turbulence using the parareal algorithm. *J. Comput. Phys.*, 229(18):6558–6573, 2010.

[111] R. Sanchez and L. Bourhrara. Existence result for the kinetic neutron transport problem with a general albedo boundary condition. *Transp. Theory Stat. Phys.*, 40(2):69–84, 2011.

[112] A. Saubert, A. Sureda, J. Bader, J. Lapins, M. Buck, and E. Laurien. The 3-D time-dependent transport code TORT-TD and its coupling with the 3D thermal-hydraulic code ATTICA3D for HTGR applications. *Nuclear Engineering and Design*, 251:173–180, 2012.

[113] R.N. Slaybaugh, T.M. Evans, G.G. Davidson, and P.P. Wilson. Rayleigh quotient iteration in 3d, deterministic neutron transport. In *PHYSOR, Advances in Reactor Physics*, 2012.

[114] I. H. Sloan. Nonpolynomial interpolation. *J. Approx. Theory*, 39(2):97 – 117, 1983.

[115] G.A. Staff and E.M. Rønquist. Stability of the parareal algorithm. *Lect. Notes Comp. Sci.*, 40:449–456, 2005.

[116] A. Targa. *Développement de modélisations multi-physiques best-estimate pour une analyse fine des réacteurs à eau pressurisée en conditions de fonctionnement normal et accidentel.* PhD thesis, Ecole Polytechnique, 2016 (expected).

[117] R. S. Varga. *Matrix iterative analysis.* Springer series in computational mathematics. Springer Verlag, Berlin, Heidelberg, Paris, 2000.

[118] K. Veroy, C. Prud'homme, D.V. Rovas, and A.T. Patera. A posteriori error bounds for reduced-basis approximation of parametrized noncoercive and nonlinear elliptic partial differential equations. In *Proceedings of the 16th AIAA Computational Fluid Dynamics Conference*, 2003. AIAA Paper 2003-3847.

[119] H. D. Victory Jr. Convergence properties of discrete ordinates solutions for neutron transport in three-dimensional media. *SIAM J. Numer. Anal.*, 17:71–83, 1980.

[120] E.L. Wachspress. *Iterative solution of elliptic systems and applications to the neutron diffusion equations of reactor physics.* Prentice-Hall, Inc., 1966.

[121] W. Yang, H. Wu, Y. Zheng, and L. Cao. Application of wavelets scaling function expansion method in resonance self-shielding calculation. *Annals of Nuclear Energy*, 37(5):653–663, 2010.

[122] M. Yano, J.D. Penn, and A.T. Patera. A model-data weak formulation for simultaneous estimation of state and model bias. *Comptes Rendus Mathematique*, 351(23–24):937 – 941, 2013.