Résolution numérique de l'équation du transport de neutrons par la méthode des harmoniques sphériques et une méthode de Galerkin discontinue

Kenneth Assogba

INSTITUT POLYTECHNIQUE DE PARIS

NNT : 2023IPPAX154

Thèse de doctorat

ÉCOLE POLYTECHNIQUE

IP PARIS

On a numerical scheme to solve the neutron transport equation using spherical harmonics and a discontinuous Galerkin method

Thèse de doctorat de l'Institut Polytechnique de Paris
préparée au Service d'Études des Réacteurs et de Mathématiques Appliquées (SERMA), CEA Paris-Saclay

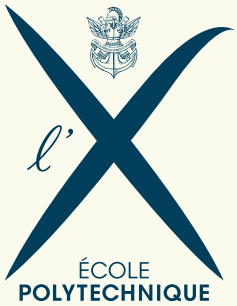
École doctorale n°574 École doctorale de mathématiques Hadamard (EDMH)
Spécialité de doctorat : Mathématiques appliquées

Thèse présentée et soutenue à Gif-sur-Yvette, le 15/12/2023, par

KENNETH ASSOGBA

Composition du Jury :

Jean Ragusa Professeur, Texas A&M University	Rapporteur, Président
Alexandre Ern Professeur, École des Ponts	Rapporteur
Olga Mula Associate Professor, Eindhoven University of Technology	Examinatrice
Teddy Pichard Professeur assistant, École polytechnique	Examinateur
Richard Sanchez Directeur de recherche, CEA	Examinateur
Grégoire Allaire Professeur, École polytechnique	Directeur de thèse
Lahbib Bourhrara Ingénieur de recherche, CEA	Co-directeur de thèse



On a numerical scheme to solve the neutron transport equation using spherical harmonics and a discontinuous Galerkin method

---

A manuscript submitted by

Kenneth Assogba

École polytechnique
Institut Polytechnique de Paris
Palaiseau, France

Commissariat à l'Énergie
Atomique et aux énergies
alternatives
Gif-sur-Yvette, France

Grégoire Allaire	Professor at École polytechnique	Thesis Director
Lahbib Bourhrara	Research Engineer at CEA	Thesis Co-Director

On a numerical scheme to solve the neutron transport equation using spherical harmonics and a discontinuous Galerkin method

Abstract

In this thesis, we carry out the mathematical analysis and performance study of a numerical scheme for solving the Boltzmann model of neutral particle transport. Our study applies in the field of nuclear reactor physics to the numerical simulation of reactor cores and fuel assemblies. The studied numerical scheme is based on combined spherical harmonics and discontinuous finite element methods, respectively for the angular and space variables. It handles a large class of unstructured prismatic meshes, allowing all the geometries describing fuel cells to be processed without any simplification or homogenization. Taking advantage of the similarity of some cells, our matrix-vector product algorithm is highly optimized and parallelized.

The resulting transport numerical solver has a wide range of applications, it can be used for a core calculation as well as for a precise 281 energy groups lattice calculation accounting for anisotropic scattering. To assess the accuracy and performance of this numerical scheme, it was applied to one, two and three-dimensional reactor core and fuel assembly calculations. In practice, it produces accurate solutions even in the case of complex core and assemblies geometries.

On the other hand, we prove the convergence and provide error estimates of this numerical scheme. Using $\mathbb{P}^k$ discontinuous Galerkin finite elements (in space) on a mesh of size h and a spherical harmonics approximation of order N (in the angular variable), the convergence rate is of order $\mathcal{O}\big(N^{-t} + h^k\big)$ for a smooth solution which admits partial derivatives of order $k+1$ and t with respect to the spatial and angular variables respectively. For $k = 0$ (piecewise constant finite elements) we also obtain a convergence result of order $\mathcal{O}\big(N^{-t} + h^{1/2}\big)$. Numerical experiments in 1, 2 and 3 dimensions are provided, showing a better convergence behavior for the L^2-norm, typically of one more order in space, $\mathcal{O}\big(N^{-t} + h^{k+1}\big)$.

Finally, we leverage the benefits of the discretization to wrap the proposed solver, in a domain decomposition framework. We conduct robustness, strong, and weak scalability experiments on a petaflop cluster. We reach and maintain a strong scaling efficiency of 100 % up to 4096 CPU cores and 80 % up to 8192 CPU cores. Moreover, the parallel solver is robust, with variations in the obtained solutions compared to the sequential solver being close to the machine precision.

Keywords: boltzmann transport equation, spherical harmonics, discontinuous finite element, unstructured meshes

Institut Polytechnique de Paris
91120 Palaiseau, France

Résumé

Dans cette thèse, nous étudions un schéma numérique pour la résolution de l'équation du transport de neutrons. Notre étude s'applique dans le domaine de la physique des réacteurs nucléaires pour la simulation numérique de cœurs de réacteurs et d'assemblages combustible. Le schéma numérique étudié est basé sur la méthode des harmoniques sphériques pour la variable angulaire et celle des éléments finis discontinus pour la variable spatiale. Ce schéma numérique permet de traiter les maillages non-structurés, non-conformes avec des faces 2D courbes, typiquement des cercles et des arcs de cercle. La prise en compte de maillages courbes permet de représenter de façon exacte la géométrie des crayons combustibles de réacteurs à eau pressurisée. Par ailleurs, tirant profit de la périodicité des motifs dans un cœur, les produits matrice-vecteur sont optimisés et parallélisés.

Le solveur de flux qui en résulte a un large éventail d'applications. Afin d'évaluer sa précision et ses performances, il a été appliqué à des calculs de cœurs de réacteurs et d'assemblages de combustible en une, deux et trois dimensions. Les solutions obtenues sont conformes aux standards de l'industrie nucléaire et les temps de calcul compétitifs, même dans le cas de géométries complexes de cœurs et d'assemblages.

D'autre part, nous prouvons la convergence et fournissons des estimations d'erreur de ce schéma numérique. En utilisant des éléments finis discontinus $\mathbb{P}^k$ sur un maillage de taille h et une approximation angulaire d'ordre N, le taux de convergence obtenu est $\mathcal{O}\big(N^{-t} + h^k\big)$ pour une solution régulière qui admet des dérivées partielles d'ordre $k+1$ et t par rapport aux variables spatiales et angulaires respectivement. Pour $k = 0$ (éléments finis constants par morceaux), nous obtenons également un résultat de convergence d'ordre $\mathcal{O}\big(N^{-t} + h^{1/2}\big)$. Des expériences numériques en une, deux et trois dimensions sont présentées, montrant un meilleur taux de convergence en norme L^2, typiquement d'un ordre supplémentaire en espace, $\mathcal{O}\big(N^{-t} + h^{k+1}\big)$.

Enfin, nous développons une méthode de décomposition de domaine afin d'effectuer des calculs parallèles en mémoire distribuée. Les tests numériques menés montrent que la méthode permet d'obtenir un passage à l'échelle quasi-linéaire. Nous maintenons une efficacité en scalabilité forte de 100 % jusqu'à 4096 cœurs de calcul et de 80 % jusqu'à 8192 cœurs. Le solveur est en outre robuste, les variations sur les solutions obtenues par rapport au solveur séquentiel étant proches de la précision machine.

Mots clés : équation de transport de boltzmann, harmoniques sphériques, éléments finis discontinus, maillages non structurés

Institut Polytechnique de Paris
91120 Palaiseau, France

Remerciements

J'ai adoré cette aventure. Elle m'a transformé personnellement et, je l'espère, a contribué un peu au projet de l'équipe qui m'a accueilli. Par-dessus tout, j'ai adoré travailler et partager de bons moments avec eux. C'est donc l'occasion de les remercier.

J'ai eu beaucoup de chance d'avoir Grégoire et Lahbib en tant que directeurs de thèse, et je tiens avant tout à les remercier chaleureusement. Grégoire, je te remercie infiniment pour le temps précieux que tu as consacré à me guider. Ta disponibilité à creuser chaque question en détail a été inestimable, et j'ai énormément appris à tes côtés. Lahbib, je tiens à te remercier du fond du cœur pour ta pédagogie et ton soutien constant. Ta disponibilité pour m'aider et m'encourager a été d'une importance capitale, et je te suis reconnaissant pour tout ce que j'ai appris grâce à toi. Enfin, je souhaite remercier Igor. Passionné et souriant, il a toujours trouvé les bons mots pour m'encourager et m'aider à avancer. Je lui en suis vraiment reconnaissant.

Je suis sincèrement honoré de vous avoir rencontrés et d'avoir pu travailler avec vous.

Je voudrais remercier l'ensemble des membres du jury pour s'être intéressés à mon travail et pour avoir participé à ma soutenance. Merci beaucoup à Alexandre Ern et Jean Ragusa pour votre relecture de mon manuscrit. Je suis très reconnaissant et fier que vous ayez accepté cette tâche. Merci également à Olga Mula, Teddy Pichard et Richard Sanchez d'avoir accepté de faire partie de mon jury, j'en suis tout autant honoré.

Je souhaiterais remercier le CEA pour avoir rendu cette thèse possible ainsi que l'ensemble du SERMA pour leur accueil. Je tiens à exprimer ma gratitude envers la hiérarchie du SERMA, notamment le chef de service Loïc de Carlan et le chef du LLPR Fadhel Malouch, pour leur soutien tout au long de cette thèse. J'aimerais remercier en particulier les doctorants que j'ai croisés pendant ces trois années et qui sont devenus des amis proches aujourd'hui. Je remercie mes collègues de Siemens qui m'ont accordé du temps pour préparer sereinement ma soutenance et sont venus me soutenir.

Je remercie Nadège et ses enfants pour leur accueil chaleureux à mon arrivée en France. Enfin, et pour conclure ces remerciements, je souhaite exprimer toute ma gratitude à mes chers parents et à ma famille pour leur soutien tout au long de mon parcours académique.

Contents

List of Figures

List of Tables

List of algorithms

Introduction (en français)

Sommaire du présent chapitre

Contexte de la thèse

Le Commissariat à l'Énergie Atomique et aux Énergies Alternatives (CEA) développe des plateformes de simulation[1] et des infrastructures expérimentales pour des applications dans les domaines de l'énergie, de la santé et de l'industrie. Ces outils sont utilisés notamment pour la production de radio-isotopes médicaux ; l'analyse du risque de criticité dans les installations où de la matière fissile est utilisée ; l'étude de la thermomécanique du combustible ; la vérification de la tenue sismique des installations nucléaires ; la recherche de nouveaux matériaux et l'étude de bien d'autres phénomènes.

Dans le domaine de la physique des réacteurs, l'un des principaux enjeux est de simuler numériquement, à haut niveau de précision les cœurs de réacteurs nucléaires, voir Figure 1. La simulation permet de réduire le temps et les coûts de développement lors des études de conception. En exploitation, elle permet de piloter efficacement le cœur en tenant compte de la demande sur le réseau électrique et des autres moyens de production (hydraulique, éolien et solaire notamment). Par ailleurs, lors des campagnes de renouvellement du combustible, il est nécessaire de simuler des milliers de configurations afin d'optimiser les plans de chargements du cœur. En outre, les outils de calcul scientifique développés pour les cœurs de réacteurs sont également utilisés en sûreté nucléaire, en radioprotection, en dosimétrie et en analyse de criticité.

[1] Logiciels de calcul scientifique et machines massivement parallèles.

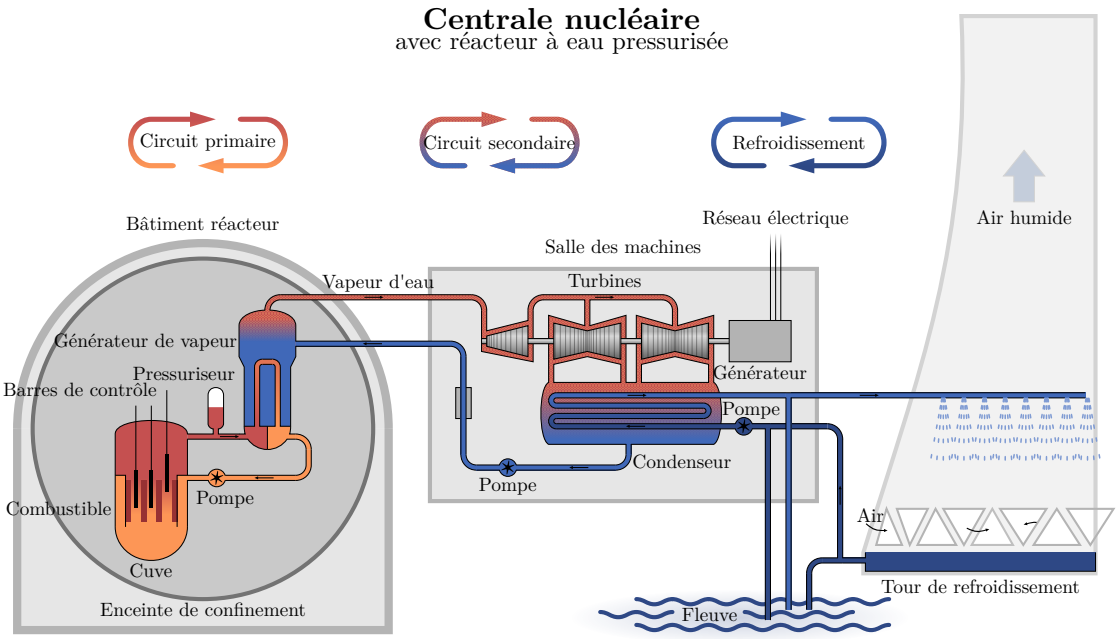


FIGURE 1 : Schéma d'une centrale nucléaire avec un réacteur à eau pressurisée.

Une étape cruciale pour *calculer* un réacteur est la résolution numérique de l'équation de transport de neutrons. Cette équation fournit la distribution de puissance dans le cœur en tenant compte des collisions et absorptions de neutrons et des fissions induites par ces neutrons. À chaque instant, la distribution de puissance varie en fonction de la position, la direction et de l'énergie des neutrons. La résolution numérique de ce problème nécessite donc quatre approximations : en espace, en angle, en énergie et en temps. Dans ce travail, nous nous intéressons aux approximations en angle et en espace, et le cadre de nos développements numériques est la plateforme de physique des réacteurs APOLLO3®.

La plateforme APOLLO3®

Développée au CEA et partagée avec les industriels EDF et Framatome, APOLLO3® [122, 98] est un outil de calcul scientifique multifilière, multi-échelle et multidimensionnel pour la physique des réacteurs. À cet effet, APOLLO3® intègre

- des outils de modélisation : description de géométrie, générateurs de maillage et bibliothèques de données nucléaires,
- des solveurs de flux pour la résolution numérique des équations de transport et de diffusion,
- et des outils de post-traitement et de couplage avec d'autres physiques : thermique, mécanique, thermohydraulique.

Concernant les solveurs de flux, le schéma de calcul standard en neutronique se fait en deux étapes. Une première étape – dite de réseau – à l'échelle d'un assemblage où on produit des bibliothèques de sections efficaces. Ces sections efficaces sont homogénéisées en espace et condensées en énergie en vue d'un calcul à l'échelle du cœur.

	Échelle		Dim			Discrétisation		Maillage		
	Réseau	Cœur	1	2	3			Polygonal	Courbe	MPI
IDT	✓	✗	✓	✓	✓	S_N	sMOC	✓	✓	✗
TDT	✓	✗	✗	✓	✓	S_N	lMOC	✓	✓	✗
Minaret	✗	✓	✗	✓	✓	S_N	DG	✗	✗	✓
Minos	✗	✓	✓	✓	✓	SP_N	FEM	✗	✗	✓
Pastis	✗	✓	✓	✓	✓	P_N	Nodale	✗	✗	✗
NYMO	✓	✓	✓	✓	✓	P_N	DG	✓	✓	✓

Tableau 1 : Revue des solveurs de APOLLO3®.

À chacune de ces échelles, APOLLO3® fournit aux utilisateurs de nombreux solveurs de flux leur permettant ainsi de modéliser tout type de géométrie et tout type de filière. Ces solveurs se différencient notamment par : le type de mailles (triangle, quadrangle, polygone) et de maillages (cartésien, hexagonal, non-structuré, non-conforme, courbe) traités, la prise en compte de géométries tridimensionnelles, l'utilisation de méthodes d'ordre élevé (quadratique, cubique, d'ordre arbitraire), l'optimisation de l'empreinte mémoire et le calcul parallèle en mémoire distribuée.

Une comparaison de ces solveurs est fournie au Tableau 1. À l'échelle réseau la méthode des caractéristiques est utilisée dans sa forme longue (lMOC) dans TDT [121, 124, 56] et courte (sMOC) dans IDT[2] [96, 95, 15]. Les solveurs de cœur sont Minaret [83, 82, 16] basé sur les méthodes S_N et DG, Minos [17] basé sur l'approximation P_N simplifiée et les éléments finis de Raviart-Thomas et Pastis [87, 28] basé sur la méthode P_N.

Le solveur NYMO

Développé au sein de la plateforme APOLLO3®, le solveur NYMO [22, 21, 23, 20] résout numériquement l'équation du transport de neutrons en utilisant la méthode des harmoniques sphériques pour la variable angulaire et celle des éléments finis discontinus pour la variable spatiale.

La méthode des harmoniques sphériques (P_N) [93, 94, 44] consiste à approcher le flux angulaire de neutrons par un développement en série sur la base des harmoniques sphériques réelles. L'approximation spatiale utilise une méthode de Galerkin discontinue (DG[3]). Le flux angulaire discret ainsi obtenu est injecté dans une formulation faible du problème de transport et les fonctions tests choisies sont les fonctions de bases de l'espace d'approximation. La méthode de Galerkin discontinue offre la possibilité de traiter des maillages non-structurés, non-conformes et possédant des faces courbes. Par ailleurs, la résolution peut être faite localement, le flux dans chaque maille ne dépendant que du flux dans les cellules adjacentes. Cela permet d'envisager des simulations parallèles distribuées

[2] L'implémentation d'une methode de décomposition de domaine dans IDT avec MPI est opérationnelle en dehors d'APOLLO3® [84].

[3] De l'anglais *discontinuous Galerkin*.

performantes avec peu de communications entre les unités de calcul. Enfin, la méthode permet d'implémenter facilement des éléments finis d'ordre arbitrairement élevés.

Le solveur qui résulte de ce schéma numérique peut traiter des maillages 2D non-structurés, non-conformes avec des faces courbes, typiquement des cercles et des arcs de cercle. La prise en compte de maillages courbes permet de représenter de façon exacte la fine couronne de zirconium qui enrobe les crayons combustibles de réacteurs à eau pressurisée.

Démarche

Cette thèse étend les travaux précédemment décrits sur 3 axes :

- Nous démontrons la convergence et fournissons des estimations d'erreur *a priori* du schéma numérique de `NYMO` [8].
- Nous implémentons dans le solveur `NYMO` la prise en compte des géométries 3D extrudées (maillages prismatiques) et utilisons ce solveur sur des cas de test de type benchmark et sur des problèmes plus représentatifs des études neutroniques [11]. Ces tests sont accompagnés de comparaisons avec les autres solveurs de `APOLLO3`® : TDT et IDT pour les calculs de réseau ; Minaret et Minos pour les calculs de cœur.
- Nous implémentons une méthode de décomposition de domaine pour la parallélisation en mémoire distribuée [9].

Contributions et plan de la thèse

Chapitres 1 et 2 Ces chapitres proposent une brève introduction à la physique des réacteurs nucléaires. Nous y introduisons les éléments de modélisation mathématique et les méthodes numériques utiles pour ce travail.

Chapitre 3 Dans ce chapitre, nous présentons le schéma numérique proposé par [20] pour résoudre l'équation de transport de Boltzmann. Ce schéma numérique combine la méthode des harmoniques sphériques en angle à celle des éléments finis discontinus en espace. Cette méthode est implémentée dans le solveur de flux `NYMO`. Le solveur `NYMO` peut traiter toutes les géométries décrivant un assemblage combustible sans simplification ou homogénéisation.

Chapitre 4 Dans le chapitre 4, nous étendons le solveur `NYMO` aux géométries 3D extrudées. Pour cela, nous calculons les coefficients des matrices élémentaires issues de la discrétisation en 3D à partir de celles de la géométrie 2D. Ensuite, nous avons validé l'implémentation de ce schéma sur des cas de test de type benchmark. Enfin, nous utilisons le solveur développé pour effectuer des études neutroniques aux échelles réseau et cœur.

Une partie de ce chapitre a été publiée sous les références,

[10] K. Assogba, L. Bourhrara, I. Zmijarevic et G. Allaire. "Precise 3D Reactor Core Calculation Using Spherical Harmonics and Discontinuous Galerkin Finite Element Methods". *Proceedings of International Conference on Physics of Reactors 2022 (PHYSOR 2022)*. Pittsburgh, PA, United States : American Nuclear Society, mai 2022, p. 1224-1233,

[11] K. Assogba, L. Bourhrara, I. Zmijarevic, G. Allaire et A. Galia. "Spherical Harmonics and Discontinuous Galerkin Finite Element Methods for the Three-Dimensional Neutron Transport Equation : Application to Core and Lattice Calculation". *Nuclear Science and Engineering* 197.8 (août 2023), p. 1584-1599.

Chapitre 5 L'objectif du chapitre 5 est de fournir une preuve mathématique de la convergence du schéma numérique objet de la présente thèse. En utilisant des éléments finis discontinus $\mathbb{P}^k$ sur un maillage de taille h et une approximation angulaire d'ordre N, le taux de convergence obtenu est $\mathcal{O}\left(N^{-t} + h^k\right)$ pour une solution régulière qui admet des dérivées partielles d'ordre $k+1$ et t par rapport aux variables spatiales et angulaires respectivement. Pour $k = 0$ (éléments finis constants par morceaux), nous obtenons également un résultat de convergence d'ordre $\mathcal{O}\left(N^{-t} + h^{1/2}\right)$. Des expériences numériques en une, deux et trois dimensions sont présentées, montrant un meilleur taux de convergence en norme L^2, typiquement d'un ordre supplémentaire en espace, $\mathcal{O}\left(N^{-t} + h^{k+1}\right)$.

La version de prépublication de ce travail est référencée comme [8] K. Assogba, G. Allaire et L. Bourhrara. *Analysis of a Combined Spherical Harmonics and Discontinuous Galerkin Discretization for the Boltzmann Transport Equation*. (Hal-04196435). Août 2023.

Chapitre 6 Le chapitre 6 rend compte du développement d'une méthode de décomposition de domaine pour le solveur de transport `NYMO`. Un élément important de cette approche est la structure par blocs des matrices issues de la discrétisation. Les blocs hors diagonale décrivent alors les interactions entre les mailles adjacentes et induisent un couplage des sous-domaines voisins par la transmission du flux entrant. Ce couplage s'effectue au niveau du produit matrice vecteur. Les communications MPI sont effectuées de manière non bloquante, permettant ainsi le recouvrement des communications par les calculs. L'implémentation dans le code est peu intrusive et l'exécution en parallèle ne nécessite aucun effort supplémentaire des utilisateurs finaux.

Les tests numériques menés montrent que la méthode permet d'obtenir un passage à l'échelle quasi-linéaire. Le solveur est en outre robuste, les variations sur les solutions obtenues par rapport au solveur séquentiel étant proches de la précision machine.

Une partie de ce chapitre a été publiée avec la référence [9] K. Assogba et L. Bourhrara. "The PN Form of the Neutron Transport Problem Achieves Linear Scalability Through Domain Decomposition". *Proceedings of International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2023)*. Niagara Falls, Ontario, Canada, août 2023.

Résumé étendu des contributions

Dans cette section, nous présentons plus en détail le modèle de transport de neutrons, la discrétisation utilisée puis nous décrivons les contributions de cette thèse.

On souhaite déterminer le flux de neutrons multigroupe $u^g(x,\omega)$ en tout point (x,ω) de l'espace des phases $X = D \times \mathbf{S}^2$. La variable spatiale est notée x et D est le domaine spatial. La direction angulaire est notée ω et $\mathbf{S}^2$ est le domaine angulaire. L'indice g représente le groupe d'énergie et, sauf nécessité, est omis dans ce qui suit. Dans l'espace des phases X avec un flux entrant f via le bord entrant Γ_-, le problème s'écrit,

$$\omega \cdot \nabla u + \sigma u = q \quad \text{dans} \quad X \tag{1a}$$

$$u = f \quad \text{sur} \quad \Gamma_- \tag{1b}$$

où σ est la section efficace macroscopique totale. La fonction q représente les sources de neutrons et prend en compte la diffusion et éventuellement la fission et les sources externes de neutrons. De plus amples détails sont fournis dans les Chapitres 1, 2 et 3.

Le bord sortant est noté Γ_+ et on défini l'espace variationnel W comme

$$W = \left\{ v \in L^2(X);\ \omega \cdot \nabla v \in L^2(X);\ v|_{\Gamma_+} \in L^2_+ \right\}, \tag{2}$$

$v|_{\Gamma_+}$ désignant la restriction de v à Γ_+ et $L^2_+ = L^2(\Gamma_+, |\omega \cdot n|\,\mathrm{d}s\,\mathrm{d}\omega)$.

Soit $v \in W$ une fonction test, nous multiplions l'équation (1a) par $(v + \frac{1}{\sigma}\omega \cdot \nabla v)$ puis intégrons le tout sur l'espace des phases. Après utilisation de la formule de Green (3.6) et des conditions aux limites (1b), nous obtenons le problème variationnel :

$$\text{trouver } u \in W \ \text{ tel que } \ a(u,v) = L(v), \ \ \forall v \in W, \tag{3}$$

avec

$$a(u,v) = \int_X \left(\frac{1}{\sigma}(\omega \cdot \nabla u)(\omega \cdot \nabla v) + \sigma u v \right) + \int_{\Gamma_+} uv(\omega \cdot n), \tag{4}$$

$$L(v) = \int_X q \left(v + \frac{1}{\sigma}(\omega \cdot \nabla v) \right) - \int_{\Gamma_-} fv(\omega \cdot n). \tag{5}$$

Sous des hypothèses sur les données f, q et σ, ce problème est bien posé. Il admet une solution unique et cette solution dépend continument des données du problème [22].

La méthode des harmoniques sphériques ou méthode $\mathrm{P_N}$ consiste à faire un développement limité à l'ordre N du flux angulaire sur la base des harmoniques sphériques réelles notées $y_n^m(\omega)$,

$$u^N(x,\omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} u_n^m(x) y_n^m(\omega). \tag{6}$$

Les inconnues u_n^m résultant de ce développement sont appelées *moments du flux*. Les moments du flux sont approchés par des fonctions polynomiales par morceaux de degré au plus k. L'espace des polynômes par morceaux de degré au plus k est noté dans la

suite $\mathbb{P}^k = \text{span}\{\varphi_1, \ldots, \varphi_J\}$. En fin de compte,

$$u_h^N(x,\omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} \sum_{j=1}^{J} u_{n,j}^m \varphi_j(x) y_n^m(\omega). \tag{7}$$

L'espace d'approximation ainsi construit est noté $\widetilde{W}_{N,k}$. Le problème discret est le suivant :

$$\text{trouver } u \in \widetilde{W}_{N,k} \text{ tel que } a(u,v) = L(v), \quad \forall v \in \widetilde{W}_{N,k}. \tag{8}$$

Extension du solveur numérique aux géométries 3D extrudées

Du fait du développement de nouveaux concepts de réacteurs nucléaires, les industriels sont demandeurs de méthodes de résolution pour l'équation de transport de neutrons en trois dimensions. Par ailleurs, certains effets multi-physiques ne peuvent être modélisés qu'en trois dimensions. Dans les réacteurs rapides par exemple, la dilatation thermique induit un déplacement radial non-uniforme qui affecte la réactivité [53].

L'objectif du Chapitre 4 est de présenter une extension des capacités de `NYMO` aux maillages 3D extrudés, voir Figure 2. Ce solveur est ensuite utilisé pour effectuer aussi bien des calculs de cœur de réacteur que d'assemblage combustible.

Maillages prismatiques extrudés

En deux dimensions, le domaine spatial D est maillé en éléments disjoints (également appelés régions) D_r. Le maillage peut être non structuré et non-conforme. Des éléments courbes (cercle ou arc de cercle) peuvent être utilisés, permettant ainsi de représenter la gaine entourant les cellules combustibles. En trois dimensions, les maillages extrudés sont suffisants pour modéliser la géométrie de la plupart des types de réacteurs nucléaires. Chaque région 3D, notée D_r^{3D} est donnée comme une face de base D_r^{2D} extrudée le long de l'axe z, on écrit

$$D_r^{3D} = D_r^{2D} \times [z_0, z_1]. \tag{9}$$

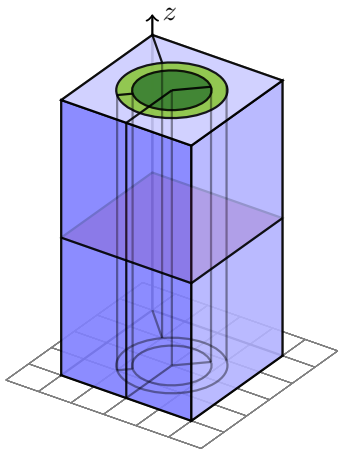


FIGURE 2 : Exemple de maillage 3D d'une cellule de combustible traité par `NYMO`.

Chaque face F^{3D} de ∂D_r^{3D} est soit une face horizontale, soit une face verticale. Les faces verticales sont obtenues en extrudant un segment de ligne, un cercle ou un arc de cercle dans le plan horizontal le long de l'axe z. Le traitement des faces horizontales est identique à celui des régions 2D.

$$F^{3D} = \begin{cases} F^{2D} \times [z_0, z_1] \text{ si } F^{3D} \text{ est verticale,} \\ D_r^{2D} \times \{z_0\} \text{ sinon.} \end{cases}$$

Calcul des matrices élémentaires en 3D

Les matrices élémentaires résultant de la discrétisation 3D se calculent en se ramenant à des calculs de matrices élémentaires 2D. Trois types d'intégrales peuvent être obtenues après la discrétisation :

- les intégrales de volume $\int_{D_r} \bullet \, \mathrm{d}x$,
- les intégrales sur la variable angulaire $\int_{\mathbf{S}^2} \bullet \, \mathrm{d}\omega$,
- et les intégrales sur la frontière $\Gamma_\pm(D_r)$ d'une région.

Du fait de l'équation (9), les intégrales de volumes s'écrivent comme produit de deux intégrales que l'on sait calculer. Les intégrales sur la variable angulaire se calculent de la même façon en 2 et 3 dimensions et nous nous référons au Chapitre 3 pour plus de détails sur le calcul 2D. Les intégrales sur la frontière d'une région couplent les deux premiers types d'intégrales. En notant n_F la normale sortante unitaire d'une face F de D_r, le bord de la région est donné par

$$\Gamma_\pm(D_r) = \bigcup_{F \in \partial D_r} \left\{ (x, \omega) \in F \times \mathbf{S}^2 \ / \ \pm \omega \cdot n_F > 0 \right\}.$$

Les intégrales de frontière s'écrivent alors

$$\sum_{F \in \partial D_r} \int_F \int_{\pm(\omega \cdot n_F) > 0} \bullet (\omega \cdot n_F) \, \mathrm{d}\omega \, \mathrm{d}s \,. \tag{10}$$

La transformation des coordonnées cartésiennes en coordonnées sphériques d'angles (θ_n, φ_n) permet d'écrire le vecteur normal sous la forme (n_x, n_y, n_z) avec $n_x = \sin\theta_n \cos\varphi_n$, $n_y = \sin\theta_n \sin\varphi_n$ et $n_z = \cos\theta_n$. L'intégrale (10) se calcule alors en utilisant

$$\int_{\pm(\omega \cdot n_F) > 0} y(\omega)(\omega \cdot n_F) \, \mathrm{d}\omega = \int_{\theta_0}^{\theta_1} f(\theta) \, \mathrm{d}\theta \int_{\varphi_0}^{\varphi_1} g(\varphi) \, \mathrm{d}\varphi \,.$$

Il suffit enfin de déterminer θ_0, θ_1, φ_0 et φ_1, selon que la face est horizontale ou verticale.

Faces horizontales Si la face F est la base inférieure de D_r, alors n_F est de coordonnées $(0, 0, -1)$ et $\omega \cdot n_F = -\cos\theta$. Sur la demi-sphère entrante $\mathbf{S}^2_-$, $\omega \cdot n_F < 0$ implique $\cos(\theta) > 0$. Par conséquent, $\theta \in \left[-\frac{\pi}{2}, \frac{\pi}{2}\right] \cap [0, \pi]$ et finalement $\theta_0 = 0$ et $\theta_1 = \frac{\pi}{2}$. Symétriquement pour la demi-sphère sortante $\mathbf{S}^2_+$, $\theta_0 = \frac{\pi}{2}$ et $\theta_1 = \pi$. Dans les deux cas, $\varphi_0 = 0$ et $\varphi_1 = 2\pi$. Les mêmes développements s'appliquent aux faces horizontales supérieures.

Faces verticales Si la face F est une face latérale plane de D_r, l'utilisation de $\theta_n = \frac{\pi}{2}$ implique $n_F = (\cos\varphi_n, \sin\varphi_n, 0)$ puis $\omega \cdot n_F = \sin\theta\cos(\varphi - \varphi_n)$. Puisque $\theta \in [0, \pi]$, le signe de $\omega \cdot n_F$ ne dépend que du signe de $\cos(\varphi - \varphi_n)$. Sur la demi-sphère entrante, $\omega \cdot n_F < 0$ implique $\varphi_0 = \varphi_n + \frac{\pi}{2}$ et $\varphi_1 = \varphi_n + 3\frac{\pi}{2}$. Des arguments symétriques s'appliquent à la demi-sphère sortante. Dans les deux cas $\theta_0 = 0$ et $\theta_1 = \pi$.

Dans le cas des faces latérales courbes, la normale à la face est paramétrée par un point M, et on note $n = n(M)$ avec $n_x = \cos(\varphi_n(M))$, $n_y = \sin(\varphi_n(M))$ et $n_z = 0$.

Applications numériques

Le solveur développé est ensuite utilisé pour effectuer du calcul de cœur et de réseau en deux et trois dimensions. Le premier calcul est le cœur Stepanek [126], le deuxième est la suite de problèmes Takeda [127], le troisième est le cœur C5G7 3D [89] et le dernier est un assemblage typique de réacteur à eau pressurisée [137]. Ces problèmes fournissent différentes configurations, qui offrent des géométries cartésiennes, hexagonales, non structurées et courbes avec différentes propriétés optiques, représentatives des réacteurs à neutrons rapides et des réacteurs à eau légère. Les simulations effectuées sont des calculs critiques et de distribution de puissance.

Calcul de cœur Sur la base des données fournies dans [104], nous comparons `NYMO` avec les solveurs IDT, Minaret, Minos et Pastis. Les solutions de référence utilisées sont fournies par `TRIPOLI-4`® et nous calculons les erreurs de `NYMO` en fonction de celles-ci.

Le Tableau 2 compare différents solveurs `APOLLO3`® pour le calcul de cœur. À l'exception du cas 1 du modèle 1 de Takeda, les erreurs relatives sont inférieures à 50 pcm[4], avec en particulier quatre pcm obtenus pour le cas 2 du modèle 2. Pour le cas 1 du modèle 1, il est possible de raffiner le maillage pour réduire l'erreur, comme cela a été fait pour Minaret par exemple. Les calculs des valeurs propres des modèles 1, 2 et 3 sont effectués en moins d'une minute. Ces temps de calcul sont inférieurs à ceux donnés dans [104] pour les autres solveurs.

Tableau 2 : Erreurs en pcm sur le k_{eff} relativement à la solution `TRIPOLI-4`®.

	Stepanek	Takeda 1		Takeda 2		Takeda 3			Takeda 4		
Cas		C1	C2	C1	C2	C1	C2	C3	C1	C2	C3
IDT	-3	30	25	-4	-11	-71	-20	-22	-	-	-
Minaret	-2	-16	-10	-11	-20	-85	-11	-13	34	20	-36
Minos	7	233	164	-29	-47	-130	-104	-4	-394	-269	7
Pastis	-3	-189	-35	-2	-2	-12	-10	-3	-1	-23	1
`NYMO`	-3	142	25	7	4	-48	12	-21	34	29	31

Calcul de cœur hétérogène C5G7 [89] est un problème conçu pour évaluer la capacité des codes de transport déterministes à effectuer des calculs de cœur de réacteurs sans

[4] pour cent mille

homogénéisation spatiale. Nous présentons ici le calcul de la nappe de puissance du cœur.

Le Tableau 3 présente l'erreur maximale (MAX), moyenne (AVG), quadratique moyenne (RMS) et relative moyenne (MRE) sur la distribution de puissance. Les calculs sont effectués en P_5 $\mathbb{P}^2$ sans raffinement axial pour les deux premiers cas et P_6 $\mathbb{P}^1$ pour la configuration Rodded B. Pour les deux premiers cas, les calculs présentent des erreurs AVG, RMS et MRE inférieures à 0,4 %. Dans le dernier cas, les erreurs AVG, RMS et MRE sont inférieures à 0,8 %. On peut supposer que l'erreur est plus élevée ici en raison des fortes discontinuités introduites par l'insertion des barres de contrôle. Dans l'ensemble, ces résultats sont conformes aux erreurs admises en neutronique.

Tableau 3 : Erreurs sur la distribution de la puissance pour le cœur C5G7.

Configurations		Unrodded	Rodded A	Rodded B
Erreur par cellule (%)	AVG	0.225	0.301	0.743
	RMS	0.280	0.398	0.835
	MRE	0.171	0.249	0.764
	MAX	0.818	1.100	1.708
Erreur par assemblage (%)	UOx Interne	0.014	-0.178	-0.980
	MOx	-0.135	0.041	0.476
	UOx Externe	0.365	0.662	0.866

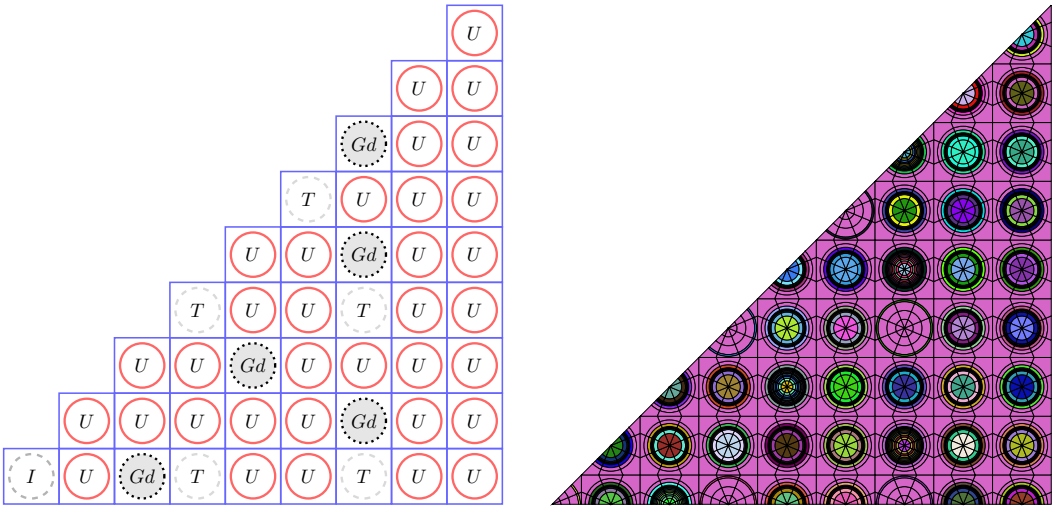
Calcul d'assemblage

Cette partie est consacrée au calcul d'un assemblage typique de réacteur à eau pressurisée (REP), Figure 3. Les données nucléaires utilisées sont issues de [137, §III. 1]. Il s'agit d'un assemblage typique 17 par 17 contenant du combustible UO_2, du gadolinium et des tubes guides. Cette étude vise à présenter un problème réaliste avec 281 groupes d'énergie en tenant compte de la diffusion anisotrope, ici d'ordre 3. Les résultats obtenus sont comparés avec des solveurs déterministes, ici TDT (méthode des caractéristiques) [121, 124, 56] et IDT (méthode des caractéristiques courtes) [96, 95, 15].

Le Tableau 4 présente les valeurs propres, le temps de calcul et l'erreur sur la distribution de la puissance en comparant `NYMO` à TDT et IDT. Nous obtenons un k_{eff} de 0,999568, soit 121 pcm de plus que la solution TDT. L'écart de 400 pcm par rapport à IDT peut s'expliquer par l'absence de secteurs annulaires dans le maillage de IDT. Les nappes de puissance sont normalisées de manière à ce que la puissance totale soit égale au nombre de cellules combustible. Les erreurs sont de moins de 1 % par rapport à TDT et sont inférieures à 1,47 % en ce qui concerne IDT. La nature des approximations étant très différente, ce résultat est très satisfaisant.

Convergence et analyse d'erreur du schéma numérique

Dans cette partie, nous cherchons à évaluer l'erreur *a priori* $u - u_h^N$ de la solution approchée u_h^N par rapport à la solution exacte u du problème continu (1). Nous introduisons pour cela une nouvelle forme bilinéaire $\tilde{a}$ (5.22), un espace variationnel $\widetilde{W}$ (5.10) et une



(a) Géométrie (b) Maillage

FIGURE 3 : Assemblage combustible de REP, géométrie 1/8ème et maillage.

TABLEAU 4 : Valeur propre et nappe de puissance de NYMO comparée à TDT et IDT pour un assemblage de REP.

	NYMO	TDT	IDT
k_{eff}	0.999568	0.998360	0.995493
Δk_{eff} (pcm)	-	121	407
Max (%)	-	0.794	1.468
AVG (%)	-	0.314	0.461
RMS (%)	-	0.384	0.607
MRE (%)	-	0.295	0.381
temps (s)	142	176	-

formulation variationnelle dite *brisée* (5.21). Nous invitons le lecteur à se reporter au Chapitre 5 pour une définition rigoureuse de la formulation variationnelle et du problème discret. Nous nous contentons dans ce résumé de présenter les résultats dans l'ordre classique : coercitivité, continuité, lemme de Céa et enfin estimation d'erreur.

Pour toute fonction discontinue v, $[\![v]\!]$ et $\{\!\{v\}\!\}$ désignent respectivement le saut et la moyenne des traces de cette fonction de part et d'autre d'une face du maillage. Posons $L^2_+ = L^2(\Gamma_+, (\omega \cdot n)\,\mathrm{d}s\,\mathrm{d}\omega)$ et définissons une norme $\|v\|_{\widetilde{W}^\star}$ sur $\widetilde{W}$ par

$$\|v\|^2_{\widetilde{W}^\star} = \sum_r \|v\|^2_{L^2(X_r)} + \sum_r \|\omega \cdot \nabla v\|^2_{L^2(X_r)} + \|v\|^2_{L^2_+} + \sum_{F \in \mathcal{F}_h^i} \int_{F \times \mathbf{S}^2} [\![v]\!]^2 |\omega \cdot n_F|. \tag{11}$$

Proposition 1. *La forme bilinéaire* $\tilde{a}$ (5.22) *est coercive sur* $\widetilde{W}$ (5.10) *pour la norme* (11)*. Ainsi, il existe une constante* α *indépendante du maillage, tel que*

$$\tilde{a}(v, v) \geqslant \alpha \|v\|^2_{\widetilde{W}^\star} \quad \forall v \in \widetilde{W}. \tag{12}$$

Démonstration. Voir Proposition 5.3.7 □

Posons $L^2_F \coloneqq L^2(F \times \mathbf{S}^2, |\omega \cdot n_F| \,\mathrm{d}s \,\mathrm{d}\omega)$ et $\|\{\!\{v\}\!\}\|^2_{L^2_F} = \int_{F\times\mathbf{S}^2} \{\!\{v\}\!\}^2 |\omega \cdot n_F|$. Nous introduisons une seconde norme

$$\|u\|^2_{\widetilde{W}^+} = \|u\|^2_{\widetilde{W}^\star} + \sum_{F\in\mathcal{F}_h^i} \|\{\!\{u\}\!\}\|^2_{L^2_F}. \tag{13}$$

Par définition $\|v\|_{\widetilde{W}^\star} \leqslant \|v\|_{\widetilde{W}^+}$, $\forall v \in \widetilde{W}$.

Proposition 2. *La forme bilinéaire $\tilde{a}$, est continue dans le sens où il existe $M > 0$ indépendant du choix du maillage tel que*

$$|\tilde{a}(u,v)| \leqslant M\|u\|_{\widetilde{W}^+}\|v\|_{\widetilde{W}^\star} \quad \textit{pour tout } (u,v) \in \widetilde{W}\times\widetilde{W}. \tag{14}$$

Démonstration. Voir Proposition 5.3.10 □

Résultat principal

Lemme 1. *(de Céa) Soit $u \in W$ l'unique solution du problème de transport et supposons que $u \in \widetilde{W}$. Soit $u_h^N \in \widetilde{W}_{N,k}$ la solution discrète. Alors, il existe une constante C indépendante de k, h et N tel que*

$$\left\|u - u_h^N\right\|_{\widetilde{W}^\star} \leqslant C \inf_{v_h^N\in\widetilde{W}_{N,k}} \left\|u - v_h^N\right\|_{\widetilde{W}^+}. \tag{15}$$

Démonstration. Voir Lemme 5.5.1. □

Ce résultat démontre que u_h^N réalise la meilleure approximation de u dans $\widetilde{W}_{N,k}$ à une constante multiplicative C près.

On considère désormais une suite de maillages $(D_h)_{h>0}$ tel que le diamètre de maille h tende vers zéro. La solution u de l'équation de transport (1) est supposée suffisamment régulière. Plus précisément, nous introduisons des espaces de Sobolev anisotropes. Soit $s \in \mathbb{N}$ et $t \in \mathbb{N}$, deux entiers positifs, $\alpha \in \mathbb{N}^d$ et $\beta \in \mathbb{N}^{d-1}$ deux multi-indices, avec $|\alpha| = \alpha_1 + \cdots + \alpha_d$. Notons ∂_x^α la α-dérivée faible par rapport à x et ∂_ω^β la β-dérivée faible par rapport à ω. Nous définissons ensuite

$$H^{s,t}(D\times\mathbf{S}^2) = \left\{u \in L^2(D\times\mathbf{S}^2);\ \partial_x^\alpha\partial_\omega^\beta u \in L^2(D\times\mathbf{S}^2),\ |\alpha| \leqslant s,\ |\beta| \leqslant t\right\}.$$

L'espace $H^{s,t}(D\times\mathbf{S}^2)$ est muni de la norme

$$\|u\|^2_{H^{s,t}(D\times\mathbf{S}^2)} = \sum_{\substack{|\alpha|\leqslant s\\ |\beta|\leqslant t}} \left\|\partial_x^\alpha\partial_\omega^\beta u\right\|^2_{L^2(D\times\mathbf{S}^2)}.$$

Théorème 1. *Supposons que l'unique solution u du problème continu (1) appartienne à $H^{k+1,t}(D\times\mathbf{S}^2)$. Pour $N\geqslant 1$ et $k\geqslant 1$, notons u_h^N la solution discrète de (8). Alors nous avons l'estimation d'erreur suivante,*

$$\left\|u-u_h^N\right\|_{\widetilde{W}^\star}\leqslant c\|u\|_{H^{k+1,t}(D\times\mathbf{S}^2)}\left(\frac{1}{N^t}+h^k\right).$$

Démonstration. Voir section 5.6. □

Remarque 1. *Ce résultat est sous-optimal pour la norme $L^2(D)$ puisque nous attendons un ordre de convergence d'ordre $\mathcal{O}\left(h^{k+1}\right)$ en espace. Les expériences numériques fournissent quant à elles l'ordre de convergence attendu.*

Remarque 2. *Il convient de noter que la formulation variationnelle classique obtenue en multipliant l'équation (1a) par v (au lieu de $(v+\frac{1}{\sigma}\omega\cdot\nabla v)$) : trouver $u\in W$*

$$\text{tel que } \int_X(-u(\omega\cdot\nabla v)+\sigma uv)+\int_{\Gamma_+}uv(\omega\cdot n)=\int_X qv-\int_{\Gamma_-}fv(\omega\cdot n)$$
$$\forall v\in W,$$

bénéficie du terme $u(\omega\cdot\nabla v)$ pour obtenir l'ordre de convergence classique $\mathcal{O}\left(h^{k+\frac{1}{2}}\right)$ des méthodes DG [75]. En effet, en utilisant l'inégalité de Cauchy-Schwarz, on obtient $\int_{X_r}u(\omega\cdot\nabla v)\leqslant\left(h_r^{-1}\int_{X_r}u^2\right)^{\frac{1}{2}}\left(h_r\int_{X_r}(\omega\cdot\nabla v)^2\right)^{\frac{1}{2}}$ ce qui implique

$$\begin{aligned}\int_{X_r}u(\omega\cdot\nabla v)&\leqslant h^{-1}\|u\|^2_{L^2(X_r)}+h\|\omega\cdot\nabla v\|^2_{L^2(X_r)},\\&\leqslant\mathcal{O}\left(h^{2(k+\frac{1}{2})}\right).\end{aligned}$$

Il suffit donc de pondérer les termes de la norme $\|\cdot\|_{\widetilde{W}^+}$ (13) par des poids en h [46, §2.3]. Dans la formulation variationnelle utilisée dans ce travail (3), le terme $u(\omega\cdot\nabla v)$ est remplacé par $(\omega\cdot\nabla u)(\omega\cdot\nabla v)$, ce qui empêche l'utilisation de cette technique.

Dans le cas d'une approximation constante par morceaux $k=0$ (qui correspond à la méthode des volumes finis), le Théorème 1 peut être amélioré. En effet le terme de dérivée advective $\int_{X_r}(\omega\cdot\nabla u)(\omega\cdot\nabla v)$ disparaît de la forme bilinéaire $\tilde{a}$ lorsque l'un de ses deux arguments, u ou v, est constant par morceaux. Introduisons une norme plus faible $\|\cdot\|_{\widetilde{W}_0^\star}$ que $\|\cdot\|_{\widetilde{W}^\star}$, mais sans le terme dérivée advective, à savoir

$$\|u\|^2_{\widetilde{W}_0^\star}=\sum_r\|u\|^2_{L^2(X_r)}+\|u\|^2_{L^2_+}+\sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![u]\!]^2|\omega\cdot n_F|. \tag{16}$$

Théorème 2. *Pour $k=0$, supposons que la solution unique u de l'équation de transport (1) appartient à $H^{1,t}(D\times\mathbf{S}^2)$. Pour $N\geqslant 1$, considérons $u_h^N\in\widetilde{W}_{N,0}$ comme la solution discrète de (8). Alors nous avons l'estimation d'erreur*

$$\left\|u-u_h^N\right\|_{\widetilde{W}_0^\star}\leqslant c\|u\|_{H^{1,t}(D\times\mathbf{S}^2)}\left(\frac{1}{N^t}+h^{\frac{1}{2}}\right).$$

Démonstration. Voir section 5.6. □

Application numérique

L'objectif ici est de comparer les erreurs observées numériquement avec les estimations d'erreurs théoriques. Nous nous intéressons au problème à source, c'est-à-dire que le flux entrant f est considéré nul dans (1). Une solution manufacturée u est prescrite, et nous en déduisons la source correspondante q de sorte que l'équation (1) soit satisfaite. Les solutions manufacturées sont construites avec un nombre fini de moments angulaires. Les ordres d'approximation angulaire sont alors choisis pour être suffisamment élevés, de sorte qu'il n'y ait pas d'erreur en angle, et seule l'erreur en espace est étudiée. La solution est choisie de sorte qu'elle soit nulle sur le bord du domaine. Les moments de flux choisis sont polynomiaux, et lors du calcul de la source q, leurs dérivées sont calculées exactement. Considérons un domaine cubique $D = [0,1]^3$. Le maillage est uniformément raffiné dans chaque direction comme le montre la Figure 4. La solution manufacturée choisie est

$$u(x,\omega) = \sum_{n=0}^{N=1} \sum_{m=-n}^{m=n} u_n^m(x) y_n^m(\omega),$$

avec

$$u_n^m(x) = \frac{m+2}{n+1} x_1^{n+1}(1-x_1) x_2^{m+2}(1-x_2) x_3(1-x_3), \ \forall n \in \{0,1\}, \ m \in \{-n,\dots,n\}.$$

Le domaine est hétérogène et les sections efficaces totales sont constantes par morceaux, comme indiqué sur la Figure 4. Le Tableau 5 présente les ordres de convergence du flux angulaire et de sa dérivée. L'ordre de convergence obtenu sur la dérivée advective est conforme à l'ordre théorique fourni par le théorème 1, tandis qu'il est supérieur d'un ordre pour le flux angulaire.

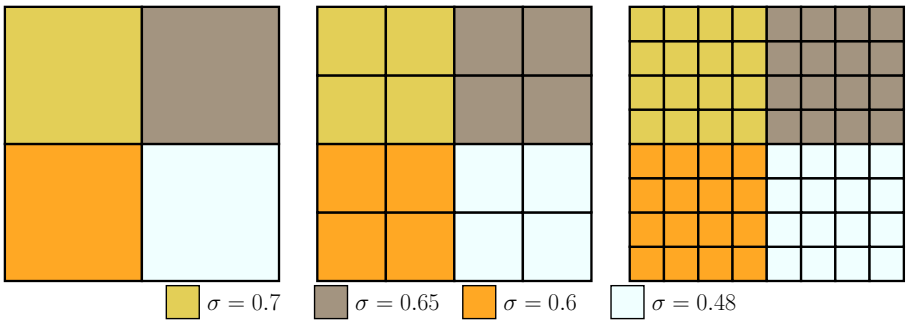


FIGURE 4 : Coupe radiale et raffinement uniforme du maillage.

Calcul parallèle distribué par décomposition de domaine

Le Chapitre 6 est consacré à la résolution parallèle et distribuée du problème de transport. Les machines cible sont des supercalculateurs constitués de nœuds de calcul indépendants. Étant donné que chaque nœud de calcul ne contiendra qu'une petite partie des données

Tableau 5 : Ordres de convergence du flux angulaire et de sa dérivée.

Nbr. d' éléments	$\|u - u_h\|_{L^2(X)}$	Ordre (flux angulaire)	$\|\omega \cdot \nabla(u - u_h)\|_{L^2(X)}$	Ordre (dérivée)
$k = 3$ approximation spatiale, $N = 2$ approximation angulaire				
8	1.619e-04	-	3.869e-03	-
64	1.041e-05	3.96	6.521e-04	2.57
512	6.496e-07	4.00	8.772e-05	2.89
4096	4.073e-08	4.00	1.116e-05	2.97
$k = 4$ approximation spatiale, $N = 2$ approximation angulaire				
8	4.028e-05	-	1.295e-03	-
64	1.485e-06	4.76	1.043e-04	3.63
512	4.880e-08	4.93	6.924e-06	3.91
4096	1.551e-09	4.97	4.391e-07	3.98
$k = 5$ approximation spatiale, $N = 2$ approximation angulaire				
8	9.808e-06	-	3.309e-04	-
64	1.853e-07	5.73	1.227e-05	4.75
512	3.066e-09	5.92	3.992e-07	4.94
4096	4.911e-11	5.96	1.260e-08	4.99

d'origine, de très gros problèmes avec des milliards d'inconnues par exemple pourront être résolus. Toutefois, pour atteindre l'efficacité souhaitée, il est crucial de prévoir comment seront communiquées les données entre les nœuds. L'objectif final étant de fournir un solveur de transport adapté aux supercalculateurs exaflopiques. APOLLO3® étant une plateforme industrielle, nous attachons une grande importance à l'expérience utilisateur. Du point de vue de l'utilisateur, l'utilisation du programme distribué doit être identique à celle du programme séquentiel, et aucune modification des jeux de données de l'utilisateur ne doit être nécessaire. Pour toutes ces raisons, notre choix s'est porté sur une méthode de décomposition de domaine.

Décomposition de domaine

La décomposition de domaine a été proposée à l'origine dans [123] pour prouver l'existence et l'unicité de solutions à l'équation de Laplace $\Delta u = 0$ avec des conditions aux limites de Dirichlet sur un domaine arbitraire. L'exemple original est un domaine constitué de l'union d'un disque et d'un carré, Figure 5. Une solution explicite de l'équation de Laplace est connue sur chacun de ces sous-domaines en utilisant les séries de Fourier. Schwarz montre ensuite que la méthode consistant à résoudre alternativement ce problème sur chacun de ces sous-domaines converge.

Avec l'avènement des ordinateurs parallèles, l'idée d'utiliser cette méthode pour effectuer des simulations numériques parallèles s'est développée. Une revue de ces méthodes peut être trouvée dans [47].

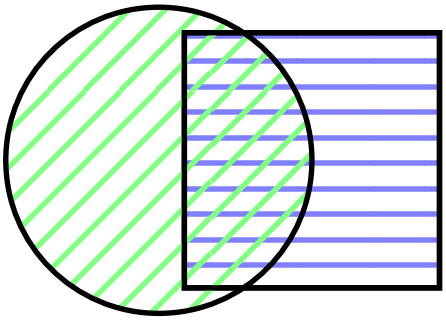
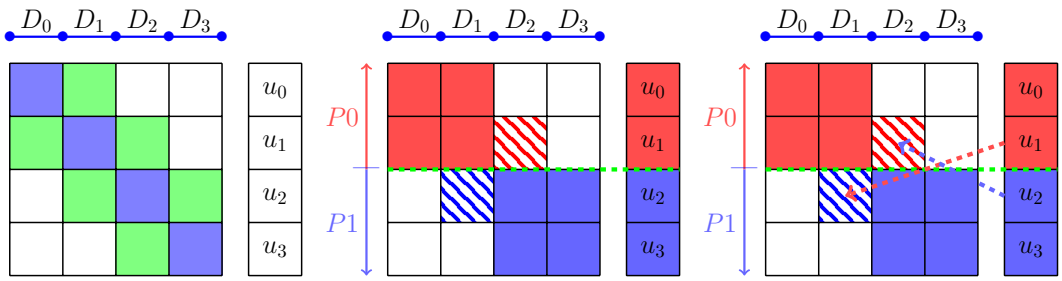


FIGURE 5 : Un domaine formé par l'union d'un disque et d'un carré, utilisé pour décrire l'algorithme alternatif de Schwarz.

Dans ce travail, nous commençons par partitionner le domaine en deux ou plusieurs sous-domaines. D'un point de vue algébrique, le partitionnement du maillage induit un partitionnement par ligne de la matrice et des vecteurs du système linéaire global, Figure 6. Chaque sous-système peut ainsi être résolu indépendamment des autres. Cependant, la partie hors diagonale du système requiert de communiquer des portions de flux provenant des sous-domaines voisins. Ce couplage est réalisé à l'intérieur du produit matrice - vecteur en utilisant MPI. Les communications MPI sont effectuées de manière non bloquante, permettant ainsi le recouvrement des communications par des calculs.

En se positionnant à l'échelle algébrique, les maillages non-conformes et non structurés avec des faces incurvées peuvent être traités sans effort supplémentaire. Nous résolvons le problème du partitionnement de ces types de maillages en implémentant une stratégie de partitionnement basée sur les courbes remplissantes de Hilbert [72], ainsi qu'un partitionnement naïf, utilisant la numérotation des mailles.

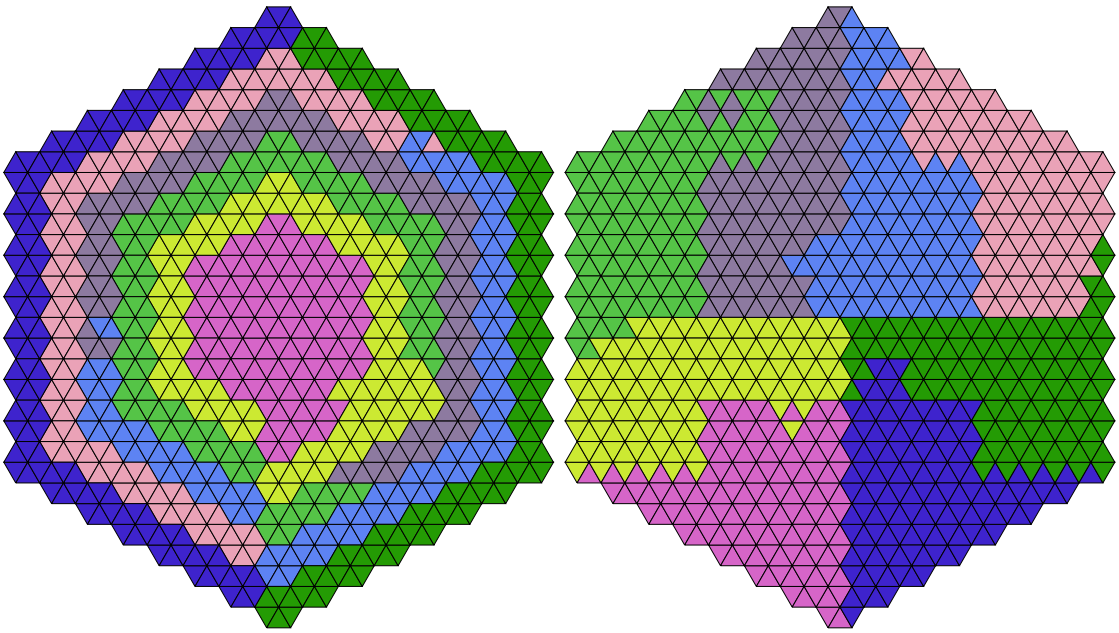


(a) Système linéaire issu de la discrétisation. (b) Décomposition du problème en deux sous-domaines. (c) Communications requises pour la résolution.

FIGURE 6 : Exemple de calcul distribué sur un maillage 1D. (a) La matrice du problème assemblé sans décomposition de domaine. Les blocs hors diagonale décrivent les interactions entre cellules voisines. (b) Le problème est partitionné entre 2 unités de calcul indépendantes. (c) La résolution implique des communications de flux entre les sous-domaines.

Partitionnement du maillage

Notons par n_d le nombre de processus MPI dans le communicateur global. La résolution parallèle commence par la partition du maillage initial D_h en n_d sous-domaines. Il convient de garantir un équilibrage de la charge entre les processus tout en minimisant les communications entre ceux-ci. Ce critère est satisfait en imposant que le nombre de régions à l'intérieur de chaque sous-domaine soit le même (plus ou moins un). Deux partitionneurs de maillage souvent utilisés sont Metis [79] et Scotch [33]. Ils ne permettent cependant pas de traiter les maillages non-conformes et les éléments courbes. Nous avons donc implémenté deux stratégies de partitionnement. La première est basée sur la numérotation du maillage et est appelée partitionnement *simple* dans la suite. La seconde est basée sur les courbes remplissantes de Hilbert [72] et est appelée partitionnement *géométrique*.



(a) Partitionnement *simple* (b) Partitionnement *géométrique*

FIGURE 7 : Maillage hexagonal où chaque hexagone est découpé en triangle. Dans ce cas, le partitionnement simple (a) conduit à des partitions emboîtées, ce qui n'est pas idéal pour les communications.

Résolution du système linéaire

Solveur linéaire Bien que toutes les expériences numériques des chapitres 3, 4 et 5 soient réalisées avec GMRes [118], nous utilisons BiCGStab [132] dans cette partie. En effet, dans l'algorithme GMRes, un ensemble de transformations de Householder doit être construit et appliqué. La construction des transformations de Householder demande la connaissance de tous les blocs de la matrice A. Ainsi dans le cas où les blocs de la matrice sont distribués, il est nécessaire d'adapter l'algorithme de construction et d'application des transformations de Householder en ajoutant des communications globales, typiquement

des Broadcast, et des All_reduce. Ce qui pourrait introduire des goulots d'étranglement. D'où le choix de BiCGStab qui est entièrement local, à l'exception des produits matrice - vecteur et scalaires.

Produit matrice - vecteur Lors de la résolution du système linéaire, tous les processus opèrent simultanément. Les sous-domaines voisins sont couplés et s'échangent le flux amont sur leurs interfaces communes. Ces échanges se font à chaque produit matrice - vecteur. L'utilisation de communications non bloquantes permet de recouvrir le temps de communication par le temps de calcul.

Le produit matrice - vecteur se déroule en trois étapes principales et est décrit dans l'Algorithme 1. Nous commençons par envoyer une requête `MPI_Irecv` pour commencer à recevoir le flux entrant $x^{\uparrow}$ des sous-domaines voisins. Ensuite, pour chaque sous-domaine voisin, la partie sortante du flux est copiée dans un conteneur temporaire x_{out} et une requête `MPI_Isend` est envoyée. L'étape de copie est nécessaire car le flux dans les cellules à l'interface entre deux sous-domaines n'est pas nécessairement contigu à l'intérieur du vecteur de flux. Cette opération est accélérée à l'aide de OpenMP et le surcoût est négligeable. Le calcul commence et la partie du produit impliquant uniquement les degrés de liberté locaux est calculée. Ensuite, on attend que les communications entrantes et sortantes se terminent. Lorsque le recouvrement fonctionne bien, le temps d'attente est négligeable. Enfin, on utilise le flux reçu pour calculer la partie restante du produit.

Algorithm 1 : Produit vectoriel matriciel du point de vue d'un sous-domaine.

Input : A_{loc}, x_{loc}, y_{loc}

1 **foreach** $i_d \in domaines_{voisins}$ **do**
2 $\quad reqs_{in} \leftarrow$ MPI_Irecv($x^{\uparrow}$) // async
3 $\quad$ Copie dans x_{out} de la portion de x_{loc} à destination de i_d
4 $\quad reqs_{out} \leftarrow$ MPI_Isend(x_{out}) // async
5 $y \leftarrow y + A_{|\text{diag}} x$ // $A_{|\text{diag}} = A^0 + A^1 + A^+ + A^-_{\text{loc}}$
6 MPI_Waitall($reqs_{in}, reqs_{out}$)
7 $y \leftarrow y + A_{|\text{offd}} x^{\uparrow}$ // $A_{|\text{offd}} = A^-_{\text{comm}}$

Pour conclure cette partie, il est utile de noter que les produits scalaires impliquent une opération de réduction collective entre tous les sous-domaines.

Applications numériques

Le cœur C5G7 [89] est utilisé ici pour évaluer la performance de l'implémentation distribuée. Trois expériences sont menées : fiabilité, scalabilité forte et scalabilité faible. Tous les calculs sont effectués sur la partition Milan du cluster pétaflopique Topaze du CEA. Milan compte 864 nœuds, chacun abritant deux sockets AMD EPYC 7763 à 2,45 GHz, équipés de 64 cœurs chacun.

Fiabilité Nous évaluons la variation du k_{eff}, de la puissance et du nombre d'itérations externes lorsque l'on augmente le nombre de sous-domaines. L'objectif est de vérifier que

l'erreur sur les valeurs propres et sur le flux en parallèle est négligeable par rapport à la solution séquentielle.

Le Tableau 6 montre les résultats de l'expérience de fiabilité. Nous observons qu'il n'y a pas de variation sur le k_{eff} et sur le nombre d'itérations. L'écart maximal sur le taux de fission est de 4 pcm. Comme le flux est calculé et stocké en simple précision, ces écarts sont négligeables. Ils peuvent être justifiés par le fait qu'en parallèle, les mêmes opérations sont effectuées qu'en séquentiel, mais dans un ordre différent.

Tableau 6 : Expérience de fiabilité. Les valeurs propres et les taux de fission obtenus avec la décomposition de domaine sont comparés à un calcul de référence sans décomposition.

n_d	k_{eff}	erreur (pcm)	avg (%)	rms (%)	mre (%)	max (%)	#externes
1	1.07864	-	-	-	-	-	31
2	1.07864	0	0.0006	0.0008	0.0005	0.0038	31
4	1.07864	0	0.0008	0.0009	0.0007	0.0025	31
8	1.07864	0	0.0010	0.0013	0.0009	0.0040	31

Scalabilité forte L'objectif de la scalabilité forte est d'évaluer le gain de temps potentiel lorsque davantage de ressources de calcul sont disponibles. Ainsi, le nombre d'unités de calcul augmente alors que la taille du problème reste constante. Nous mesurons l'accélération (*speedup*) $s(d) = {}^{t(0)}/_{t(d)}$ et l'efficacité (*efficiency*) obtenue par la parallélisation $e(d) = {}^{s(d)}/_{n_d}$, $t(d)$ étant le temps écoulé pour n_d sous-domaines. Nous utilisons ce test pour comparer le partitionnement *simple* et *géométrique*. Chaque tranche axiale est divisée en 4, ce qui porte le nombre de cellules du maillage à 665 856 et le nombre total de degrés de liberté à 913 554 432.

La Figure 8 montre les résultats de cette expérience. Nous montons jusqu'à 128 sous-domaines, ce qui correspond à 8192 cœurs de calcul. On remarque tout d'abord que quel que soit le partitionnement choisi, le temps mesuré diminue linéairement. En outre, il n'y a pas de réelle différence de performance entre le partitionnement simple et le partitionnement géométrique. L'efficacité est presque parfaite jusqu'à 64 sous-domaines, puis diminue légèrement à 128 sous-domaines. À 128 sous-domaines, chaque domaine possède 5 216 mailles, on peut raisonnablement supposer que la charge de travail est faible dans ce cas. En outre, le temps de communication représente environ 50% du temps total écoulé. Le temps de communication est dominé par les opérations de réduction globale.

Scalabilité faible Ici, le nombre de sous-domaines augmente de 16 à 32 puis 64, tout en maintenant constante la taille du problème par processus. Pour ce faire, le maillage est raffiné axialement. À partir du maillage non raffiné, chaque plan est divisé en 2 puis en 4. La mesure de performance utilisée est l'efficacité de scalabilité faible, qui est définie comme le rapport entre le temps d'exécution sur 16 processus et le temps d'exécution sur n_d processus. L'objectif est d'atteindre une efficacité proche de 1, ce qui indique que l'algorithme s'adapte efficacement à l'ajout de processus de calcul.

Le Tableau 7 présente les résultats de cette expérience. L'efficacité obtenue étant

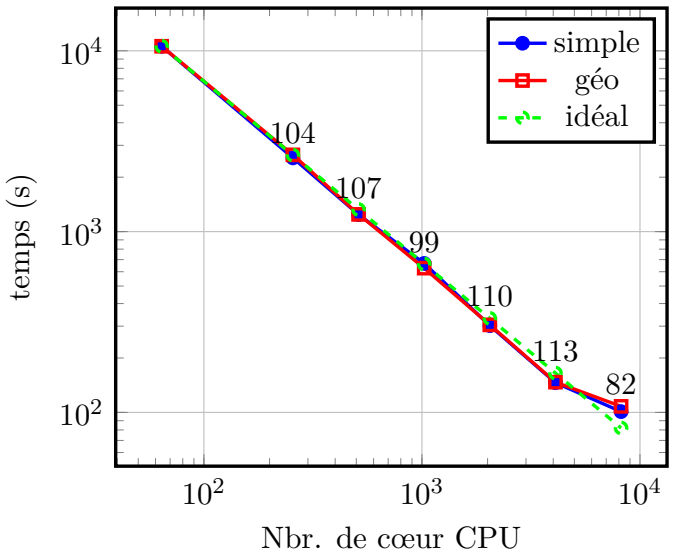


FIGURE 8 : Expérience de scalabilité forte. Le partitionnement simple et le partitionnement géométrique sont comparés à un passage à l'échelle idéal. L'efficacité obtenue pour le partitionnement simple est affichée le long de la courbe.

proche de 100%, ajoutés à ceux de la scalabilité forte, ils démontrent que notre implémentation parallèle est très efficace.

TABLEAU 7 : Expérience de scalabilité faible.

			Partitionnement simple		Partitionnement géométrique	
n_d	#cœurs	#ddl	temps (s)	efficacité (%)	temps (s)	efficacité (%)
16	1024	228 388 608	145	-	142	-
32	2048	456 777 216	154	94	157	90
64	4096	913 554 432	146	99	147	96

Introduction (in english)

Outline of the current chapter

Thesis background

The CEA[5] develops simulation platforms[6] and experimental facilities for research and industrial applications in the fields of low-carbon energy, healthcare and digital technology. These tools are used in particular for the production of medical radioisotopes; criticality safety analysis in the handling of fissile material; the study of nuclear fuel thermo-mechanics; verification of the seismic resistance of nuclear facilities; research on new materials and the study of many other phenomena.

In the field of reactor physics, one of the main challenges is the high-fidelity numerical simulation of nuclear reactor cores, see Figure 9. The simulation reduces development time and costs during conception studies. In operation, it enables the core to be controlled efficiently, taking into account demand on the power grid and other means of production (hydroelectricity, wind and solar in particular). In addition, during fuel renewal campaigns, thousands of configurations need to be simulated in order to optimize core load plans. In addition, the scientific computing tools developed for reactor cores are also used in nuclear safety, radiation protection, dosimetry and criticality analysis.

A key step in reactor analysis is the numerical resolution of the neutron transport equation. This equation provides the power distribution in the core, taking into account neutron collisions, neutron absorptions and neutron-induced fissions. At each instant, the power distribution varies as a function of neutron position, direction and energy. The numerical solution to this problem therefore requires four approximations: in

[5]The French Atomic Energy and Alternative Energies Commission, in french *Commissariat à l'Énergie Atomique et aux Énergies Alternatives*.

[6]Scientific computing software and supercomputers.

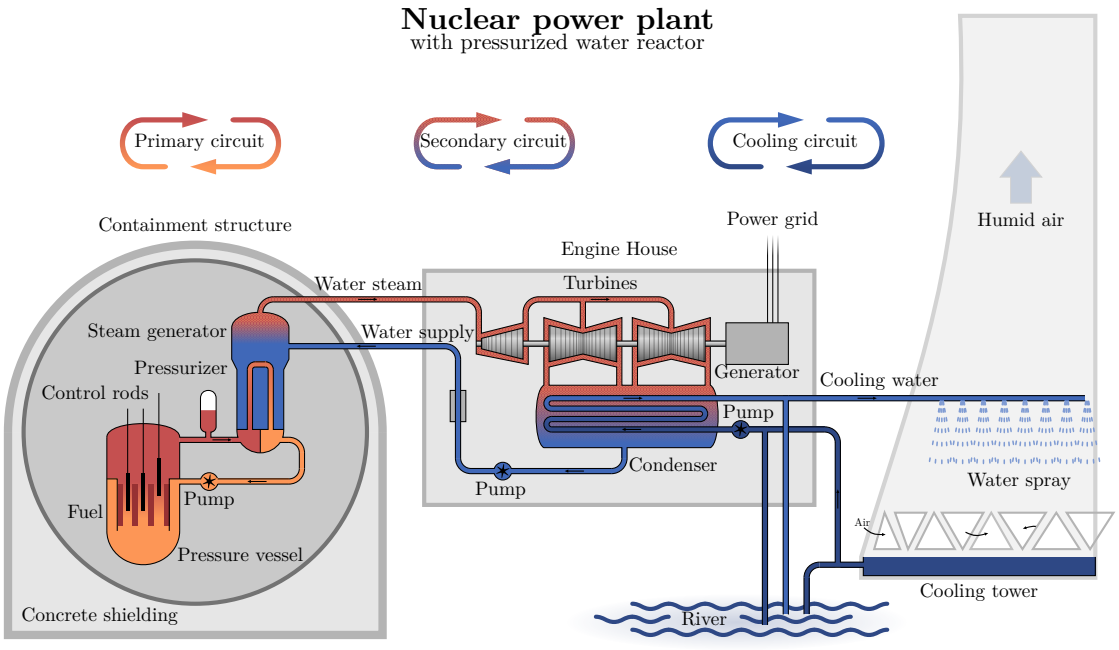


Figure 9: Nuclear power plant with a pressurized water reactor.

space, in angle, in energy and in time. In this work, we focus on the angle and space approximations, and the framework for our numerical developments is the reactor physics platform APOLLO3®.

The APOLLO3® platform

Developed at CEA and shared with industrial partners EDF and Framatome, APOLLO3® [122, 98] is a multipurpose, multi-scale, multi-dimensional scientific computing software for reactor physics. For this purpose, APOLLO3® integrates

- modeling tools: nuclear data libraries, geometry and mesh generators,
- flux solvers for the numerical resolution of transport and diffusion equations,
- and post-processing tools for coupling with other physics: thermodynamics, mechanics, thermohydraulics.

Regarding flux solvers, the standard calculation scheme in neutronics consists of two stages. In the first stage, known as the lattice step, cross-section libraries are produced at an assembly scale. These cross-sections are then homogenized in space and condensed in energy for calculation at the core scale.

At each of these scales, APOLLO3® provides users with numerous flux solvers, enabling them to model any type of geometry and any type of reactor. These solvers can be distinguished by: the type of cells (triangle, quadrangle, polygon) and meshes (Cartesian, hexagonal, unstructured, non-conformal, curved) processed, the ability to handle three-dimensional geometries, the use of high-order methods (quadratic, cubic, arbitrary order), memory footprint optimization and parallel distributed computing.

	Scale		Dim					Mesh		
	Lattice	Core	1	2	3	Discretization		Polygonal	Curved	MPI
IDT	✓	✗	✓	✓	✓	S_N	sMOC	✓	✓	✗
TDT	✓	✗	✗	✓	✓	S_N	lMOC	✓	✓	✗
Minaret	✗	✓	✗	✓	✓	S_N	DG	✗	✗	✓
Minos	✗	✓	✓	✓	✓	SP_N	FEM	✗	✗	✓
Pastis	✗	✓	✓	✓	✓	P_N	Nodal	✗	✗	✗
NYMO	✓	✓	✓	✓	✓	P_N	DG	✓	✓	✓

Table 8: Review of APOLLO3® solvers.

A comparison of these solvers is provided in Table 8. At lattice scale the method of characteristics is used in its long form (lMOC) in TDT [121, 124, 56] and short (sMOC) in IDT[7] [96, 95, 15]. The core solvers are Minaret [83, 82, 16] based on the S_N and DG methods, Minos [17] based on the simplified P_N approximation and Raviart-Thomas finite elements and Pastis [87, 28] based on the P_N method.

The NYMO solver

Developed within the APOLLO3® platform, the NYMO solver [22, 21, 23, 20] solves the neutron transport equation, using the spherical harmonics method for the angular variable and the discontinuous finite elements method for the spatial variable.

The spherical harmonics method (P_N) [93, 94, 44] consists in approximating the angular flux of neutrons by a truncated series on the real spherical harmonics. The spatial approximation uses a discontinuous Galerkin method (DG). The discrete angular flux thus obtained is injected into a weak formulation of the transport problem, and the test functions chosen are the basis functions of the approximation space. The discontinuous Galerkin method enables to handle unstructured, non-conforming meshes with curved faces. Furthermore, the resolution can be performed locally, with the flux in each mesh cell depending only on the flux in adjacent cells. This enables high-performance distributed parallel simulations to be carried out with little communication between the computing units. Finally, the method makes it easy to implement finite elements of arbitrarily high order.

The solver resulting from this numerical scheme can handle unstructured and non-conforming 2D meshes with curved faces, typically circles and arcs. By taking curved meshes into account, it is possible to accurately represent the thin zirconium ring that coats the fuel rods of pressurized water reactors.

[7] A domain decomposition method implemented in IDT with MPI is available as a standalone application outside APOLLO3® [84].

Scope of the thesis

This thesis extends the work described earlier in 3 directions:

- We demonstrate convergence and provide *a priori* error estimates of the numerical scheme of NYMO [8].
- We implement in the NYMO solver the support of three-dimensional extruded geometries (prismatic meshes). We use this solver on benchmark test cases and on problems more representative of neutron studies [11]. These tests are accompanied by comparisons with other APOLLO3® solvers: TDT and IDT for lattice calculations; Minaret and Minos for core calculations.
- We implement a domain decomposition method for distributed memory parallelization [9].

Thesis outline and contributions

Chapters 1 et 2 These chapters provide a brief introduction to nuclear reactor physics. We introduce the elements of mathematical modeling and numerical methods useful for this work.

Chapter 3 In this chapter, we present the numerical scheme proposed by [20] to solve the Boltzmann transport equation. This numerical scheme combines the spherical harmonics method in angle with the discontinuous finite elements method in space. This method is implemented in the flux solver NYMO. The NYMO solver can handle all geometries describing a fuel assembly without simplification or homogenization.

Chapter 4 In chapter 4, we extend the NYMO solver to prismatic meshes. To this end, we calculate the coefficients of the elementary matrices resulting from the 3D discretization from those of the 2D geometry. We then validated the implementation of this scheme on benchmark test cases. Finally, we use the developed solver to carry out lattice and core calculations.

A part of this chapter has been published with the references,

[10] K. Assogba, L. Bourhrara, I. Zmijarevic, and G. Allaire. “Precise 3D Reactor Core Calculation Using Spherical Harmonics and Discontinuous Galerkin Finite Element Methods”. *Proceedings of International Conference on Physics of Reactors 2022 (PHYSOR 2022)*. Pittsburgh, PA, United States: American Nuclear Society, May 2022, pp. 1224–1233,

[11] K. Assogba, L. Bourhrara, I. Zmijarevic, G. Allaire, and A. Galia. “Spherical Harmonics and Discontinuous Galerkin Finite Element Methods for the Three-Dimensional Neutron Transport Equation: Application to Core and Lattice Calculation”. *Nuclear Science and Engineering* 197.8 (Aug. 2023), pp. 1584–1599.

Chapter 5 The aim of chapter 5 is to provide a mathematical proof of the convergence of the numerical scheme described in this thesis. Using $\mathbb{P}^k$ discontinuous Galerkin finite elements (in space) on a mesh of size h and a spherical harmonics approximation of order N (in the angular variable), the convergence rate is of order $\mathcal{O}\left(N^{-t} + h^k\right)$ for a smooth solution which admits partial derivatives of order $k+1$ and t with respect to the spatial and angular variables respectively. For $k = 0$ (piecewise constant finite elements) we also obtain a convergence result of order $\mathcal{O}\left(N^{-t} + h^{1/2}\right)$. Numerical experiments in one, two and three dimensions are provided, showing a better convergence behavior for the L^2-norm, typically of one more order in space, $\mathcal{O}\left(N^{-t} + h^{k+1}\right)$.

The preprint version of this work is referenced as [8] K. Assogba, G. Allaire, and L. Bourhrara. *Analysis of a Combined Spherical Harmonics and Discontinuous Galerkin Discretization for the Boltzmann Transport Equation.* (Hal-04196435). Aug. 2023.

Chapter 6 Chapter 6 reports on the development of a domain decomposition method for the transport solver `NYMO`. An important component of this approach is the block structure of the matrices resulting from discretization. The off-diagonal blocks describe the interactions between adjacent cells and enable neighboring sub-domains to be coupled through the transmission of an incoming flux. This coupling takes place at the level of the matrix-vector product. The MPI communications are carried out in a non-blocking manner, allowing the communications to be overlapped by the calculations. The implementation in the code is minimally intrusive, and parallel execution requires no additional effort on the part of the end-users.

The numerical tests carried out show that the method achieves quasi-linear scaling. Moreover, the solver is robust, with variations in the obtained solutions compared to the sequential solver being close to the machine precision.

A subset of this chapter has been published with the reference [9] K. Assogba and L. Bourhrara. "The PN Form of the Neutron Transport Problem Achieves Linear Scalability Through Domain Decomposition". *Proceedings of International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2023).* Niagara Falls, Ontario, Canada, Aug. 2023.

Chapter 1

Modeling of neutron transport in nuclear reactor physics

The purpose of this chapter is to introduce the context and the problems addressed in this study. We introduce the concepts of nuclear reactor physics required for the applications covered. We start by describing how a nuclear power plant with a pressurized water reactor works. Then we describe the interactions between neutrons and matter, and the associated physical quantities, known as cross-sections. We introduce the Boltzmann equation used to model neutron transport. Finally, we provide existence and uniqueness properties of solutions to this problem.

We refer to [18, 115, 48, 114, 38, 29, 69, 92] for a more comprehensive overview of reactor physics, and to [42, 3] for the mathematical analysis of the problems encountered in this discipline, in particular the results of existence and uniqueness of solutions to the transport problem.

Outline of the current chapter

1.1 Operating principle of a Pressurized Water Reactor

1.1.1 Geometry and materials

Most of the nuclear reactor units installed in the world today are Pressurized Water Reactors (PWR), see Figure 9. The core consists of a square array of fuel assemblies immersed in light water. Water plays both the role of a moderator and a coolant. A moderator is a material that slows down neutrons by successive collisions with light nuclei, in our case hydrogen. A coolant is a fluid that cools the fuel assemblies and transfers the thermal energy to a secondary circuit through a heat exchanger. The assemblies are made either of uranium oxide (UOX) or of a mixture of uranium oxide and gadolinium, a poison that limits the reactivity of the assembly. Some power plants use MOX, a mixture of uranium and plutonium oxides produced by recycling used fuel from nuclear power plants. Recycling spent fuel in the form of MOX reduces the volume of high-level radioactive waste. Figure 1.1 shows the different spatial scales of a PWR core.

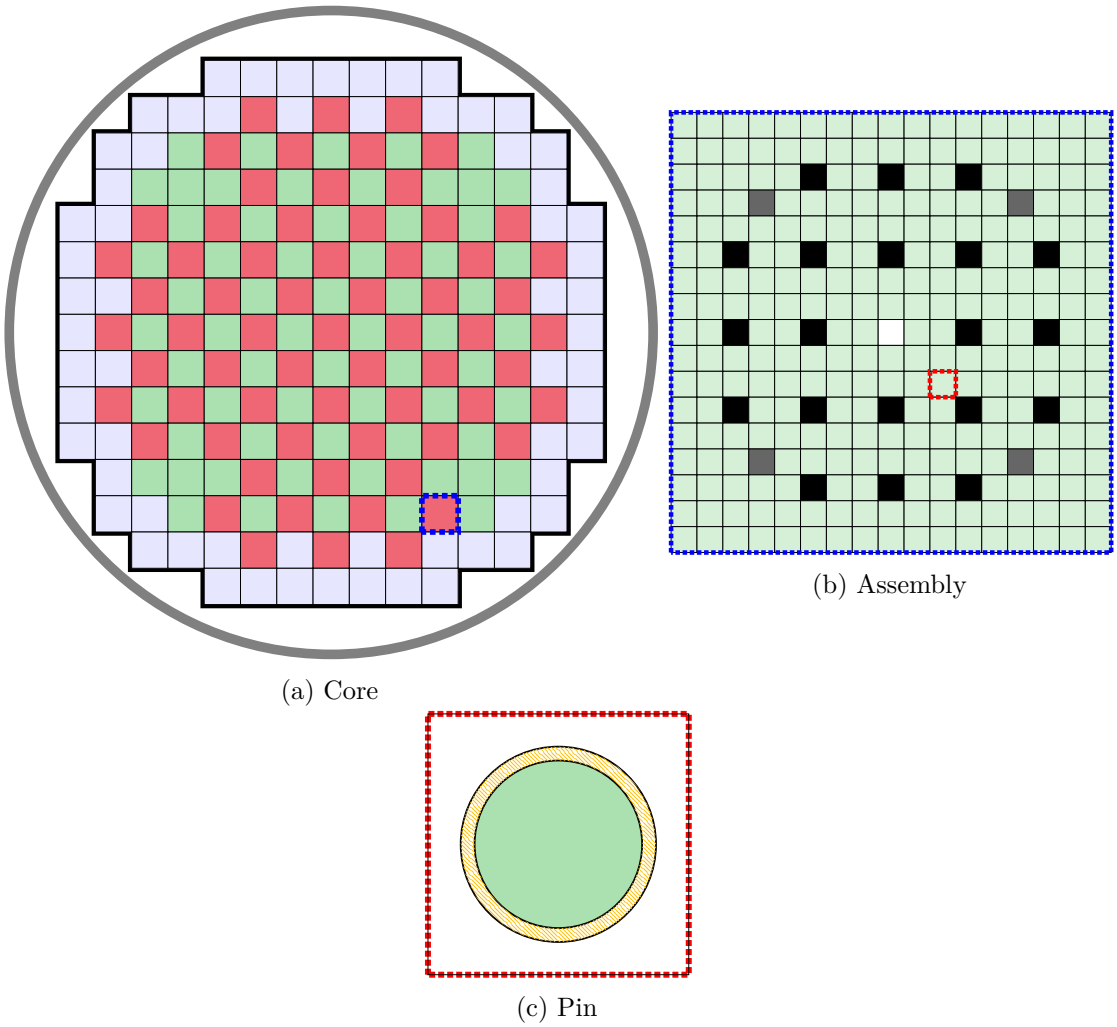
Each assembly contains at least 264 fuel pins in a 17-by-17 cell lattice arrangement. The fuel rods are held together by metal grids. Guide tubes allow the insertion of instrumentation tools or control rods, which can be used to control reactivity. Each rod measures 4 m and consists of a stack of 272 uranium pellets. Each pellet is 8.2 mm in diameter. The fuel is embedded in a thin zirconium alloy cladding about 1 mm thick.

1.1.2 Operating principle

The energy released by the fission of uranium nuclei heats the water in a first circuit, called the primary circuit. This water, heated from 300 to 320 °C is kept under high pressure, at 155 bar, to remain in liquid phase. This circuit is closed, and the thermal energy from the primary circuit is transmitted to a secondary circuit by means of heat exchangers. Steam generators transform the water in the secondary circuit into steam. This pressurized steam rotates a turbine, producing electricity through an alternator. The electrical yield of a PWR is between 30 and 35% [38]. A third loop, connected to a river or the sea, cools the steam that passes through the turbine.

The reactor can be operated in such a way as to adjust its energy production in line with demand from the power grid. The reactor operates in one-and-a-half-year cycles, then is shut down for 5 to 6 weeks for refueling and maintenance. On this occasion, one-quarter to one-third of the fuel is replaced. In addition to this, in France, a major ten-yearly inspection is carried out by the nuclear safety authority[1]. Nuclear power plants in France are designed to operate for at least 40 years. Studies are currently underway to extend their life to 60 years.

[1]ASN - Autorité de sûreté nucléaire



(a) Core

(b) Assembly

(c) Pin

Figure 1.1: The three scales of a Pressurized Water Reactor core. The core is made up of 286 assemblies. Each assembly is a grid of 17 by 17 cells. Each fuel cell is a uranium pellet surrounded by a zircaloy cladding.

1.2 The other nuclear reactor technologies

All reactors used to generate electricity in France are Pressurized Water Reactors, although there are other types. In recent years, the revival of the nuclear industry has been accompanied by innovative concepts.

1.2.1 Boiling water reactor

In a boiling water reactor or BWR, as with the PWR, the water flowing through the core acts both as a coolant and as a moderator. This water is under pressure, but at a lower pressure, 70 to 80 bars as opposed to 155 bars. The cooling water is partially vaporized in the core. The fuel used is enriched uranium. Both PWRs and BWRs belong to the family of light-water reactors (LWR).

1.2.2 Heavy-water reactor

A heavy water reactor uses heavy water (deuterium oxide D_2O) both as a coolant and as a neutron moderator. This allows natural uranium to be used as a fuel, which means it can operate without expensive uranium enrichment facilities. The use of natural uranium is made possible by the fact that deuterium captures fewer neutrons than hydrogen.

Most of the reactors in operation in Canada are heavy water reactors and are known as CANDU[2].

1.2.3 Fast-neutron reactor

In a fast-neutron reactor or fast-spectrum reactor, the fission chain reaction is sustained by fast neutrons[3], which removes the need for a moderator. This is in contrast with the slow or thermal neutrons used in PWRs. The fuel used is enriched uranium or plutonium. They can generate fissile material, in which case they are referred to as breeder reactors. The coolant is a liquid metal (such as sodium) or a gas (such as carbon dioxide or helium).

The French reactors Phénix and Superphénix were based on this technology.

1.3 Nuclear chain reaction

After being emitted in the reactor core, the neutrons move with fixed flight directions and interact with the other elements in this medium. As they do so, they lose energy and are said to *thermalize*. After absorption by a heavy nucleus, a uranium 235 or plutonium 239 atom for example, a slow neutron (also referred to as *thermal neutron*) can induce the fission of the said nucleus.

$$^{235}\mathrm{U} + \mathrm{n} \longrightarrow \text{fission products} + 2.4\ \mathrm{n} + 192.9\ \text{MeV}.$$
$$^{239}\mathrm{Pu} + \mathrm{n} \longrightarrow \text{fission products} + 2.9\ \mathrm{n} + 198.5\ \text{MeV}.$$

The fission of the uranium or plutonium produces:

- a large amount of energy, of the order of ≈ 200 MeV;
- two[4] new, lighter atoms, called fission products;
- and ν new neutrons.

After a fission, ν new neutrons are emitted. These new neutrons are emitted at high energies and are known as *fast neutrons*. The energy distribution of these neutrons is called the *fission spectrum* and is denoted χ. In addition, some unstable fission products are likely to emit a neutron. The neutrons from this second wave are called *delayed neutrons*, as opposed to the *prompt neutrons* from the first wave. After thermalization, these new neutrons in turn break other atoms, and so on. The new neutrons emitted following fission open up the possibility of a chain reaction.

[2]Canada Deuterium Uranium
[3]With kinetic energies above 1 MeV (eV stands for electronvolt).
[4]More rarely three.

The ratio of the number of neutrons between two successive generations of fission *'production'* is called the *multiplication factor*. Some *'absorptions'* do not induce fission and are referred to as sterile. Neutrons that are not absorbed may leave the core, and this is referred to as *'leakage'*. Taking leakage into account, the *effective multiplication factor* is defined as

$$k_{\text{eff}} = \frac{\text{production}}{\text{absorptions} + \text{leakage}}.$$

The effective multiplication factor k_{eff} measures the asymptotic evolution of the neutron population.

Critical. If $k_{\text{eff}} = 1$, the system is said to be critical. On average, each fission generates another fission. This results in a stable reaction and constant core power.

Subcritical. If $k_{\text{eff}} < 1$, the system is said to be subcritical. The system cannot sustain a chain reaction, and any chain reaction that starts will die out over time.

Supercritical. If $k_{\text{eff}} > 1$, the system is said to be supercritical. The number of fissions grows exponentially.

1.4 The cross-sections

The neutrons in the core not only induce fission but also many other types of nuclear reactions. The probability density function of a nuclear reaction is called *microscopic cross-section*. There are as many microscopic cross-sections as there are possible nuclear reactions between an incident neutron and a targeted nucleus. We are interested in macroscopic cross-sections[5], referred to here as σ which are defined for each nucleus as the product of the corresponding microscopic cross-section $\mu(E)$ given as a function of the incident neutron energy E and the concentration $N(x)$ of target nucleus around the position x.

The macroscopic cross-sections are therefore the probability for a neutron to interact with a target per unit length. The unit of microscopic cross-sections is the barn, and 1 barn $= 10^{-24}$ cm^2. The concentrations are in barn^{-1}cm^{-1}, so macroscopic cross-sections are in cm^{-1}. Between two collisions, the neutrons fly in a straight line. The distance between two collisions is called the *mean free path*. The neutron mean free path is between 10^{-3} and 10^{-2} m. In a pressurized water reactor core, the neutrons remain very close to the fuel assembly where they are emitted.

The neutrons interact with the nuclei in the medium mainly by *absorption* and *scattering*, see Figure 1.2.

1.4.1 Scattering

The neutron is simply reflected in a different direction from the one in which it arrived. Neutrons slow down after a collision, they are also said to thermalize. Figure 1.2 shows a diagram of neutron thermalization. The scattering reactions leave the target nucleus

[5]It should be noted that in the reactor physics literature, macroscopic cross-sections are generally referred to as Σ and microscopic ones as σ.

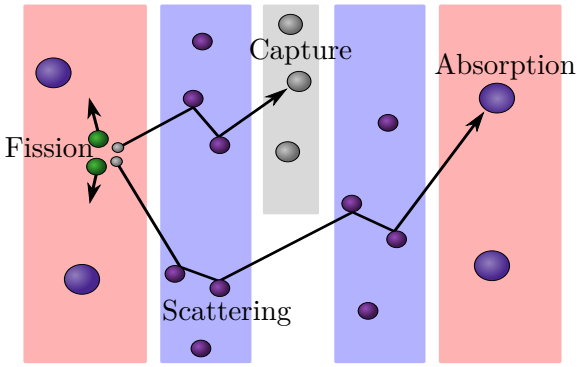


Figure 1.2: Neutron path and post-fission thermalization. After being emitted in the core, the neutrons fly in fixed directions and undergo nuclear reactions with other elements in the environment, Fuel, Moderator, Control rod.

unchanged. The associated macroscopic cross-section is denoted by σ_s. The scattering cross-section depends on the position x, the direction ω' before and ω after the collision, the energy before E' and after E the collision of the incident neutron, we will note $\sigma_s(x, \omega, \omega', E, E')$. The isotropic nature of the medium towards neutrons means that the scattering cross-section only depends on (ω, ω') through the angle between these two directions. That is $\sigma_s(x, \omega, \omega', E, E') = \sigma_s(x, \omega \cdot \omega', E, E')$.

Remark 1.4.1. *There is a difference between isotropic medium and isotropic scattering. The scattering is said to be isotropic if σ_s is independent of $\omega \cdot \omega'$.*

Remark 1.4.2. *A priori, a neutron should lose energy after collision, and we would have $E' \leqslant E$. However, a portion of the nucleus' thermal energy can be transferred in the form of kinetic energy to the incident neutron. In this case, we can see small energy rises, $E' > E$. This is known as up-scattering.*

1.4.2 Absorption

A neutron can be absorbed when approaching a nucleus, thus modifying its composition. The absorption can be *sterile* or *fertile*. Since the medium is assumed to be isotropic, the absorption cross-sections do not depend on incidence and reflection directions.

- After capturing a neutron, a nucleus may decay by emitting a photon. This reaction is referred to as sterile capture. The sterile capture macroscopic cross-section is denoted $\sigma_c(x, E)$.
- Some high-energy nuclei undergo fission, releasing energy, new neutrons and fission fragments. The fission macroscopic cross-section is denoted $\sigma_f(x, E)$.

If we neglect the other types of capture reaction, for example the (n, xn)[6] or the (n, α)[7], the absorption macroscopic cross-section is given by $\sigma_a = \sigma_c + \sigma_f$.

[6] Capture of a neutron followed by the emission of x new neutrons.
[7] Capture of a neutron producing an alpha particle.

1.4.3 Total cross-section

The sum of all the cross-sections is referred to as *total cross-section* $\sigma_t(x, E)$,

$$\sigma_t(x, E) = \sigma_a(x, E) + \int_{\mathbb{R}^+} \int_{\mathbf{S}^2} \int_{\mathbf{S}^2} \sigma_s(x, \omega \cdot \omega', E, E') \, \mathrm{d}\omega \, \mathrm{d}\omega' \, \mathrm{d}E' \,. \tag{1.1}$$

1.4.4 Resonant nuclei

Figure 1.3 shows the tabulated microscopic scattering, fission and capture cross-section of an incident neutron on uranium 235 and uranium 238 at 293 K using ENDF/B cross-section data [25]. At intermediate energies, the probability of interaction of an incident neutron with a heavy nucleus can become very high, sharp and rapid variations of cross-section are observed. At these energies, the nucleus is said to be resonant. One of the challenges of computational reactor physics is to approximate properly the cross-sections in the resonance domain. This problem is treated by the self-shielding theory, see Section 2.2.1.

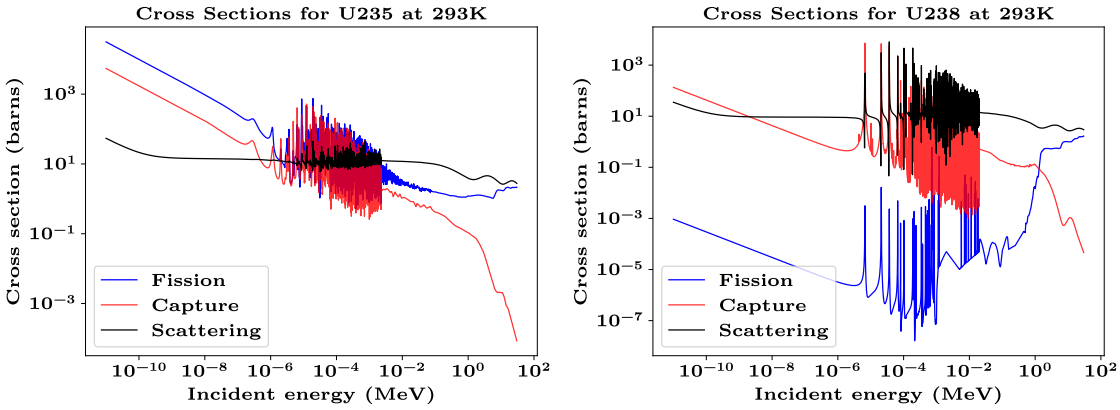


Figure 1.3: Tabulated microscopic scattering, fission and capture cross-section of uranium 235 and uranium 238 at 293 K. Note the presence of resonances between 10^{-6} and 10^{-1} MeV.

1.5 Physical quantities of interest

The quantities of interest in reactor physics are the neutron population and the isotopic composition of the medium. These quantities must be known at all times and at all points in the reactor. The neutron phase space is defined by the position coordinates x, the propagation direction unit vector $\omega = v/|v|$ with v the velocity vector, and the kinetic energy $E = 1/2 \; m|v|^2$, where m is the neutron's mass.

The *neutron density*, denoted $n(x, \omega, E, t)$ is the number of neutrons per unit of volume at point x, per unit of solid angle in the direction ω, per unit of energy in E, at a given time t. It is used to determine the *angular flux* u,

$$u(x, \omega, E, t) = |v| n(x, \omega, E, t). \tag{1.2}$$

Table 1.1: Variables and quantities of interest in reactor physics.

Variables	
Position	x
Direction	ω
Energy	E
Time	t
Nuclear data	
Macroscopic cross-section of the interaction i	σ_i
Average number of neutrons emitted by fission of the isotope α	ν_α
Fission spectrum	χ_α
Physical quantities	
Multiplication factor	k_{eff}
Angular flux	u
Scalar flux	u_0
Current vector	u_1
Reaction rate	τ_i

Let us denote $\mathbf{S}^2$ the unit sphere of $\mathbb{R}^3$, by definition $\omega \in \mathbf{S}^2$. The *scalar flux* is defined as the integral of the angular flux on all angular directions

$$u_0(x, E, t) = \int_{\mathbf{S}^2} u(x, \omega, E, t)\, \mathrm{d}\omega\,. \tag{1.3}$$

The *current vector* is defined by

$$u_1(x, E, t) = \int_{\mathbf{S}^2} \omega u(x, \omega, E, t)\, \mathrm{d}\omega\,. \tag{1.4}$$

Recalling that macroscopic cross-section of the nuclear interaction (fission, capture, scattering, absorption) i is denoted σ_i, the *reaction rate* for i is given by

$$\tau_i(x, E, t) = \sigma_i(x, E) u_0(x, E, t). \tag{1.5}$$

In steady state, these rates are generally integrated over a subset of the physical domain and an energy range to obtain a scalar value.

Table 1.1 summarizes the fundamental variables and physical quantities in reactor physics. The neutron angular flux and the k_{eff} are linked by the Boltzmann equation which we introduce in Section 1.6.

1.6 The Boltzmann transport equation

Let D denote the physical domain, such that x belongs to D. The domain D is a connected open subset of $\mathbb{R}^3$, with boundary ∂D and outward normal n.

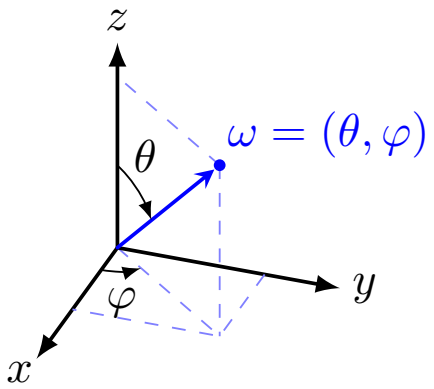


Figure 1.4: Spherical coordinates.

The unit sphere of $\mathbb{R}^3$ is denoted $\mathbf{S}^2$ and the direction $\omega \in \mathbf{S}^2$. The angular domain $\mathbf{S}^2$ is parametrized by two angles $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi]$ such that $\omega(\theta, \varphi) \in \mathbf{S}^2$, with components $\omega_x = \sin\theta\cos\varphi$, $\omega_y = \sin\theta\sin\varphi$ and $\omega_z = \cos\theta$, see Figure 1.4. In addition, $\mathrm{d}\omega = {}^1\!/_{4\pi} \sin\theta \,\mathrm{d}\theta \,\mathrm{d}\varphi$. The integral over $\mathbf{S}^2$ is defined by

$$\int_{\mathbf{S}^2} f(\omega)\,\mathrm{d}\omega = \frac{1}{4\pi}\int_0^{2\pi}\left(\int_0^{\pi} f(\theta,\varphi)\sin\theta\,\mathrm{d}\theta\right)\mathrm{d}\varphi\,.$$

Let us denote $E_{\min}$ and $E_{\max}$ the minimum and maximum energy values[8], thus $E \in [E_{\min}, E_{\max}]$. The phase space is denoted $X = D \times \mathbf{S}^2 \times [E_{\min}, E_{\max}]$. In addition $t \in \mathbb{R}^+$.

The Boltzmann equation is a balance of the neutron population taking into account the nuclear reactions that can create or destroy neutrons in an elementary volume $\mathrm{d}x\,\mathrm{d}\omega\,\mathrm{d}E$ around the point $(x, \omega, E) \in X$, in the elementary time interval $[t, t + \mathrm{d}t]$. Since the neutron density is very small compared to that of atoms in the propagation medium, the neutron-neutron interactions are neglected. Five terms are involved in this balance.

Losses by crossing the elementary space volume. The balance of neutron arrivals and leaks across the boundary of the element $\mathrm{d}x$,

$$-\omega \cdot \nabla u(x, \omega, E, t)\,\mathrm{d}x\,\mathrm{d}\omega\,\mathrm{d}E\,\mathrm{d}t\,. \tag{1.6}$$

Losses through absorption and scattering. The absorption causes neutrons to vanish while scattering causes them to change direction and energy. We use the macroscopic total cross-section, which is the sum of the absorption and scattering cross-sections (1.1),

$$-\sigma_t(x, E)u(x, \omega, E, t)\,\mathrm{d}x\,\mathrm{d}\omega\,\mathrm{d}E\,\mathrm{d}t\,. \tag{1.7}$$

[8]Typical values in neutronics are $E_{\min} = 10^{-5}$ eV and $E_{\max} = 20$ MeV.

Arrival by scattering. The neutrons which before the collision were in another direction ω' and energy E' and which pass to direction ω and energy E after collision,

$$\int_{E_{\min}}^{E_{\max}} \int_{\mathbf{S}^2} \sigma_s(x, \omega \cdot \omega', E, E') u(x, \omega', E', t) \, \mathrm{d}\omega' \, \mathrm{d}E' \;\; \mathrm{d}x \, \mathrm{d}\omega \, \mathrm{d}E \, \mathrm{d}t \, . \tag{1.8}$$

Arrival by fission. The neutrons emitted in the system after a fission reaction,

$$\sum_{\alpha} \chi_\alpha(x, E) \int_{E_{\min}}^{E_{\max}} \nu_\alpha \sigma_{f,\alpha}(x, E') \int_{\mathbf{S}^2} u(x, \omega', E', t) \, \mathrm{d}\omega' \, \mathrm{d}E' \;\; \mathrm{d}x \, \mathrm{d}\omega \, \mathrm{d}E \, \mathrm{d}t \, . \tag{1.9}$$

The sum $\sum_\alpha$ is a sum over all fissile isotopes, $\chi_\alpha(x, E)$ is the fission energy spectrum, and ν is the average number of neutrons emitted by fission.

External neutron sources. The neutrons entering the system from an external autonomous source

$$q(x, \omega, E, t) \, \mathrm{d}x \, \mathrm{d}\omega \, \mathrm{d}E \, \mathrm{d}t \, . \tag{1.10}$$

The variation of the angular flux over time $\frac{1}{|v|} \frac{\partial u}{\partial t} \, \mathrm{d}x \, \mathrm{d}\omega \, \mathrm{d}E \, \mathrm{d}t$ is equal to the sum of the quantities (1.6) - (1.10). After simplifying the elementary volumes $\mathrm{d}x \, \mathrm{d}\omega \, \mathrm{d}E \, \mathrm{d}t$, the linear Boltzmann transport equation is written,

$$\begin{aligned} \frac{1}{|v|} \frac{\partial u}{\partial t} + \omega \cdot \nabla u + \sigma_t(x, E) u = & \int_{E_{\min}}^{E_{\max}} \int_{\mathbf{S}^2} \sigma_s(x, \omega \cdot \omega', E, E') u(x, \omega', E', t) \, \mathrm{d}\omega' \, \mathrm{d}E' \\ & + \sum_{\alpha} \chi_\alpha(x, E) \int_{E_{\min}}^{E_{\max}} \nu_\alpha \sigma_{f,\alpha}(x, E') \int_{\mathbf{S}^2} u(x, \omega', E', t) \, \mathrm{d}\omega' \, \mathrm{d}E' \\ & + q(x, \omega, E, t). \end{aligned} \tag{1.11}$$

Remark 1.6.1. *The phase space X is of dimension 6. The time variable is added, so the problem is of dimension 7. For computational reasons, we often have to restrict ourselves to one- or two-dimensional configurations. In this case, $D \subset \mathbb{R}^d$, with $d = 1,\ 2$ or 3.*

To the equation (1.11) must be added the boundary conditions, they are described in Section 1.7.1. The standard use of the transport model in reactor physics is the steady-state form of (1.11). The remaining development is therefore related to the time-independent transport equation.

1.7 The steady-state Boltzmann equation

At steady state, the term $\frac{1}{|v|}\frac{\partial u}{\partial t}$ disappears, the flux and the external sources are no longer time-dependent, the Boltzmann equation is written

$$\begin{aligned}\omega\cdot\nabla u+\sigma_t(x,E)u=&\int_{E_{\min}}^{E_{\max}}\int_{\mathbf{S}^2}\sigma_s(x,\omega\cdot\omega',E,E')u(x,\omega',E')\,\mathrm{d}\omega'\,\mathrm{d}E'\\&+\sum_\alpha\chi_\alpha(x,E)\int_{E_{\min}}^{E_{\max}}\nu\sigma_{f,\alpha}(x,E')\int_{\mathbf{S}^2}u(x,\omega',E')\,\mathrm{d}\omega'\,\mathrm{d}E'\\&+q(x,\omega,E).\end{aligned}\tag{1.12}$$

The stationary state can be reached only if the system is critical. Criticality is reached under the data assumptions (1.19). From here, let us introduce the scattering operator,

$$(Hu)(x,\omega,E)=\int_{E_{\min}}^{E_{\max}}\int_{\mathbf{S}^2}\sigma_s(x,\omega\cdot\omega',E,E')u(x,\omega',E')\,\mathrm{d}\omega'\,\mathrm{d}E'\,,\tag{1.13}$$

and the fission operator,

$$(Fu)(x,\omega,E)=\sum_\alpha\chi_\alpha(x,E)\int_{E_{\min}}^{E_{\max}}\nu\sigma_{f,\alpha}(x,E')\int_{\mathbf{S}^2}u(x,\omega',E')\,\mathrm{d}\omega'\,\mathrm{d}E'\,.\tag{1.14}$$

To the equation (1.12) must be added the boundary conditions.

1.7.1 Boundary conditions

In this section, we describe the boundary conditions for the transport equation. Let us recall that the physical domain D is an open bounded set in $\mathbb{R}^d$ ($d=1,\ 2$ or 3), with boundary ∂D and outward normal n. The boundary $\Gamma=\partial D\times\mathbf{S}^2$ of $X=D\times\mathbf{S}^2$ is divided into an incoming boundary Γ_- and outgoing boundary Γ_+ defined by

$$\begin{aligned}\Gamma_-&=\Big\{(x,\omega)\in\partial D\times\mathbf{S}^2,\ \omega\cdot n(x)<0\Big\},\\\Gamma_+&=\Big\{(x,\omega)\in\partial D\times\mathbf{S}^2,\ \omega\cdot n(x)>0\Big\}.\end{aligned}$$

The remainder $\Gamma\setminus(\Gamma_-\cup\Gamma_+)$ of the boundary, where the normal is not defined or $\omega\cdot n=0$, is assumed to be a set of measure zero in $\mathbb{R}^d\times\mathbf{S}^2$ and is ignored.

Vacuum boundary condition

Here we assume that the space D is a convex domain and that the exterior is occupied by vacuum. The *vacuum boundary condition* means that no neutron from the outside can enter the physical domain and no neutron leaving the medium can re-enter it (hence the convexity hypothesis). It is written

$$u(x,\omega,E)=0\quad\text{on}\quad\Gamma_-.$$

Non-homogeneous boundary condition

A more general case of vacuum conditions is the case where a neutron flux $f(x, \omega, E)$ is imposed on the incoming boundary. The following problem, known as the *source problem*, is then defined as follows.

Problem 1.7.1 (Source problem)**.** *Find* $u(x, \omega, E)$ *which satisfies*

$$\omega \cdot \nabla u + \sigma_t(x, E)u = Hu + Fu + q(x, \omega, E) \qquad in \quad X, \tag{1.15a}$$

$$u = f(x, \omega, E) \qquad on \quad \Gamma_-. \tag{1.15b}$$

Albedo boundary condition

Let us consider a direction ω' defined as the specular reflection of an incident direction ω on the boundary ∂D, see Figure 1.5. The vector n is the normal at the point of incidence and $\times$ the cross product of two vectors of $\mathbb{R}^3$. The reflected direction ω' is uniquely defined by

$$n \cdot \omega = -n \cdot \omega' \quad \text{and} \quad (\omega \times \omega') \cdot n = 0,$$

or equivalently,

$$\omega' = \omega - 2(\omega \cdot n)n.$$

The *albedo boundary condition* consists in imposing that the incoming flux at the point x following ω is proportional to the outgoing flux following the reflected direction ω',

$$u(x, \omega, E) = \beta(E)u(x, \omega', E) \quad \text{on} \quad \Gamma_-.$$

The *specular reflection* boundary condition refers to the particular case $\beta(E) = 1$. It is applied on the boundary corresponding to a symmetry plane of D.

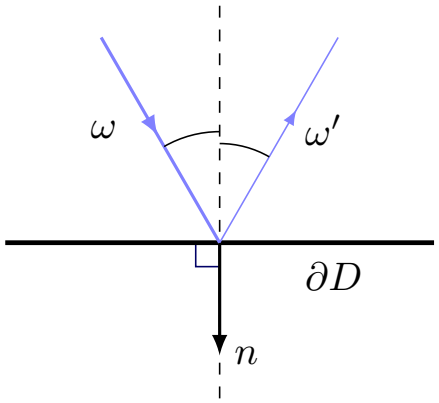


Figure 1.5: Specular reflection.

There are other boundary conditions, such as translation, periodic boundary conditions, and white boundary conditions. They are described in the references provided at the beginning of the chapter.

1.7.2 The eigenvalue problem

When there is no external source ($q = 0$) and the incoming flux is zero, if u is a solution to (1.12), then so is every αu function, with $\alpha \in \mathbb{R}$. An additional parameter is therefore introduced and the following problem, known as the *critical* or *eigenvalue problem*, is addressed.

Problem 1.7.2 (Eigenvalue problem)**.** *Find the largest* $\lambda \in \mathbb{R}$ *such that the pair* (u, λ) *satisfies*

$$\omega \cdot \nabla u + \sigma_t(x, E)u = Hu + \frac{1}{\lambda}Fu \qquad \text{in} \quad X, \tag{1.16a}$$

$$u = 0 \qquad \text{on} \quad \Gamma_-. \tag{1.16b}$$

The problem (1.16) is a generalized eigenvalue problem. The largest λ is denoted k_{eff} and is the *effective multiplication factor* described in Section 1.3.

1.7.3 Existence and uniqueness of solutions

We rely on [42, Chapter XXI. §2], where the pair of variables (ω, E) is replaced by the velocity $v \in V$. Let us recall that $E = 1/2\ m|v|^2$ and $\omega = v/|v|$. The velocity space $V \subset \mathbb{R}^3$ is the support of a positive Radon measure μ on $\mathbb{R}^3$, with $\mu(\{0\}) = 0$. In addition, $\mathrm{d}\omega\,\mathrm{d}E = m/|v|\ \mathrm{d}v$. The phase space is denoted $X = D \times V$.

Let us denote by L^p, with p a real number such that $1 \leqslant p < \infty$, the space of measurable functions which are p-integrable,

$$L^p(X,\ \mathrm{d}x\,\mathrm{d}\mu) = \left\{ v \text{ Lebesgue measurable};\ \int_X |v|^p\,\mathrm{d}x\,\mathrm{d}\mu < \infty \right\}.$$

In the case $p = \infty$, $L^\infty(X,\ \mathrm{d}x\,\mathrm{d}\mu)$ refers to the space of measurable functions that are bounded almost everywhere on X. As [22], our setting will be that of L^2-spaces, suitable for physics applications. The general L^p case is addressed in [42]. The surface element of ∂D is denoted $\mathrm{d}s$, let us define

$$\begin{aligned} L^2 &= L^2(X,\ \mathrm{d}x\,\mathrm{d}\mu), \\ L^2_- &= L^2(\Gamma_-, |\omega \cdot n|\,\mathrm{d}s\,\mathrm{d}\mu), \\ L^2_+ &= L^2(\Gamma_+, |\omega \cdot n|\,\mathrm{d}s\,\mathrm{d}\mu). \end{aligned}$$

The weighted spaces L^2_- and L^2_+ are defined as spaces of measurable functions with respective finite integrals

$$\|v\|_{L^2_-} = \int_{\Gamma_-} v^2|\omega \cdot n|\,\mathrm{d}s\,\mathrm{d}\mu < \infty \quad \text{and} \quad \|v\|_{L^2_+} = \int_{\Gamma_+} v^2|\omega \cdot n|\,\mathrm{d}s\,\mathrm{d}\mu < \infty.$$

With this new set of variables, the problem (1.15) is rewritten as follows (the details of the calculations can be found in [13, 100]).

Problem 1.7.3 (Source problem)**.** *Given $q \in L^2$, $f \in L^2_-$, and the non-negative cross-sections σ_t and σ_s, find $u(x,v)$ which satisfies*

$$\omega \cdot \nabla u + \sigma_t(x,v)u = Hu + q(x,v) \qquad \text{in} \quad X, \tag{1.17a}$$

$$u = f(x,v) \qquad \text{on} \quad \Gamma_-, \tag{1.17b}$$

with H the scattering operator

$$(Hu)(x,v) = \int_V \sigma_s(x,v,v')u(x,v')\,\mathrm{d}\mu(v')\,. \tag{1.18}$$

We introduce the space W defined by

$$W = \Big\{ v \in L^2 \,;\, \omega \cdot \nabla v \in L^2,\ v|_{\Gamma_+} \in L^2_+ \Big\}.$$

The space W is a Hilbert space when equipped with the scalar product

$$(u,v)_W = \int_X uv + (\omega \cdot \nabla u)(\omega \cdot \nabla v)\,\mathrm{d}x\,\mathrm{d}\mu + \int_{\Gamma_+} uv|\omega \cdot n|\,\mathrm{d}s\,\mathrm{d}\mu\,.$$

Let us denote by $\|\cdot\|_W$ the associated norm, $\|v\|_W = (v,v)_W^{1/2}$. The main result of this section is stated as follows.

Theorem 1.7.4. *Let us assume that $\sigma_t \in L^\infty(X)$ with $\sigma_t(x,v) \geqslant 0$ and $\alpha > 0$ a constant. Under the so-called subcriticality conditions*

$$\sigma_t(x,v) - \int_V \sigma_s(x,v',v)\,\mathrm{d}\mu(v') \geqslant \alpha > 0, \tag{1.19a}$$

$$\sigma_t(x,v) - \int_V \sigma_s(x,v,v')\,\mathrm{d}\mu(v') \geqslant \alpha > 0, \tag{1.19b}$$

the problem (1.17) has a unique solution $u \in W$, which depends continuously on the problem data f and q,

$$\|u\|_W \leqslant c\Big(\|q\|_{L^2} + \|f\|_{L^2_-}\Big),$$

where c is a positive constant.

Proof. For vacuum boundary conditions the proof is presented in [42, Chapter XXI. §2, section 4, Theorem 4]. For heterogeneous boundary conditions, the proof for $u \in L^\infty(X)$ is given in [42, Chapter XXI. §4, Proposition 7], as well as the idea of the proof for the L^p case. □

In the same way, the eigenvalue problem (1.16), is rewritten with the velocity vector v.

Problem 1.7.5 (Eigenvalue problem)**.** *Given non-negative cross-sections σ_t, σ_s and σ_f, find $u(x,v)$ and $\lambda \in \mathbb{R}$ which satisfies*

$$\omega \cdot \nabla u + \sigma_t(x,v)u = Hu + \frac{1}{\lambda}Fu \qquad \text{in} \quad X, \tag{1.20a}$$

$$u = 0 \qquad \text{on} \quad \Gamma_-, \tag{1.20b}$$

where H is the scattering operator defined at (1.18) *and F is the fission operator*

$$(Fu)(x,v) = \int_V \nu\sigma_f(x,v')u(x,v')\,\mathrm{d}\mu(v')\,. \tag{1.21}$$

The following result is proved in [3].

Theorem 1.7.6. *Let us assume that D is a bounded open domain with a regular boundary. Under the condition of theorem 1.7.4, there exists a smallest real and simple eigenvalue denoted $1/k_{eff}$ solution of the problem* (1.20) *such that the associated eigenvector u is the only one that is strictly positive in $D \times V$.*

Proof. [3, Proposition 6.5.7]. □

Chapter 2

Numerical methods for neutron transport

In this chapter the main numerical methods used to solve the neutron transport equation are presented.

Outline of the current chapter

2.1 Introduction

In this chapter, we present the main numerical methods for the solution of the neutron transport equation. Successive discretizations in energy, angle and space lead to a fully discretized problem. To keep the presentation simple, we progressively simplify the problems treated, moving from the multigroup transport (2.6) to linear advection equation (2.22).

For a closer look at the elements presented here, we refer to [119, 88, 108, 1, 3] for computational reactor physics and to [76, 36, 12, 2, 50, 46] for numerical analysis.

2.2 The multigroup discretization

In the targeted applications, the energy variable E is treated by dividing the interval $[E_{\min}, E_{\max}]$ into sub-intervals called *energy groups*. A common convention is that the numbering of the energy groups is ordered by decreasing energy,

$$E_{\min} = E_G < \cdots < E_1 < E_0 = E_{\max}.$$

Such a discretization of the energy domain is called the *multigroup approximation*. The number of energy groups varies between two (fast and slow neutrons) and thousands of groups. A typical discretization for a pressurized water reactor lattice calculation is 281 groups [71].

Using the notation

$$\int_g \cdot \,\mathrm{d}E = \int_{E_g}^{E_{g-1}} \cdot \,\mathrm{d}E\,,$$

let us introduce the angular flux in the g group defined by

$$u^g(x,\omega) = \int_g u(x,\omega,E)\,\mathrm{d}E\,. \tag{2.1}$$

The multigroup angular flux is then denoted $u = (u^g(x,\omega))_{g=1,\cdots,G}$.

On each energy interval $[E_g, E_{g-1}]$ the angular flux is approximated by

$$u(x,\omega,E) \approx w(E)u^g(x,\omega), \quad E_g < E \leqslant E_{g-1}. \tag{2.2}$$

The function w is called *weighting flux* and is normalized inside each group,

$$\int_g w(E)\,\mathrm{d}E = 1.$$

The multigroup cross-sections are defined in order to preserve the reaction rates from the continuous energy problem to the multigroup problem,

$$\sigma_i^g(x,\omega)u^g(x,\omega) = \int_g \sigma_i(x,E)u(x,\omega,E)\,\mathrm{d}E\,. \tag{2.3}$$

It should be noted that the multigroup cross-sections obtained from equation (2.3) depend on the angular variable. In practical applications, the scalar flux is used instead of the angular flux, giving angle-independent cross-sections. The cross-sections and sources are redefined as follows,

$$\begin{aligned}
\sigma_t^g(x) &= \int_g w(E)\sigma_t(x,E)\,\mathrm{d}E\,,\\
\sigma_s^{g,g'}(x,\omega\cdot\omega') &= \int_g\int_{g'} w(E')\sigma_s(x,\omega\cdot\omega',E,E')\,\mathrm{d}E'\,\mathrm{d}E\,,\\
\chi^g(x) &= \int_g \chi(x,E)\,\mathrm{d}E\,,
\end{aligned}$$

and

$$\nu\sigma_f^{g'}(x) = \int_g w(E)\nu\sigma_f(x,E)\,\mathrm{d}E\,.$$

The approximation (2.2) is inserted into the problem (1.15), then the equation is integrated on $[E_g, E_{g-1}]$. The scattering operator is now written as

$$(H^g u)(x,\omega) = \sum_{g'=1}^{G} \int_{\mathbf{S}^2} \sigma_s^{g,g'}(x,\omega\cdot\omega')u^{g'}(x,\omega')\,\mathrm{d}\omega'\,, \tag{2.4}$$

the fission operator as

$$(F^g u)(x,\omega) = \sum_{\alpha} \chi_\alpha^g(x) \sum_{g'=1}^{G} \nu\sigma_{f,\alpha}^{g'}(x) \int_{\mathbf{S}^2} u^{g'}(x,\omega')\,\mathrm{d}\omega'\,, \tag{2.5}$$

the external sources and the incoming flux as

$$q^g(x,\omega) = \int_g q(x,\omega,E)\,\mathrm{d}E\,, \quad f^g(x,\omega) = \int_g f(x,\omega,E)\,\mathrm{d}E\,.$$

The phase space is denoted $X = D\times\mathbf{S}^2$. The multigroup transport problem is stated.

Problem 2.2.1 (Multigroup transport problem)**.** *Find* $u = (u^g(x,\omega))_{g=1,\cdots,G}$ *solution of: for all* $g = 1,\cdots,G$

$$\omega\cdot\nabla u^g + \sigma_t^g(x)u^g = H^g u + F^g u + q^g(x,\omega) \quad \text{in} \quad X, \tag{2.6a}$$
$$u^g = f^g(x,\omega) \quad \text{on} \quad \Gamma_-. \tag{2.6b}$$

All the energy groups are coupled through the scattering and fission operators.

2.2.1 The self-shielding

To determine the multigroup cross-sections, it is therefore necessary to know the weighting flux w. However, as this weighting flux itself depends on the angular flux to be calculated, it is not computable. From the nuclear data and a weighting flux representative of the physics of a generic reactor core, it is possible to obtain first cross-sections called *infinite dilution cross-sections*. The process of correcting the cross-sections previously obtained, taking into account the geometry of the problem under study, the composition of the medium and the resonance of the cross-sections, is called *self-shielding* and the cross-sections obtained are called *self-shielded cross-sections*. One refers to [115, 39, 69, 117] for a complete review on the subject.

The evaluation of the error resulting from this process is out of the scope of this thesis, and in the rest of this work we consider the self-shielded cross-sections to be exact.

2.3 Angular discretization

Here we describe the treatment of the angular dependence of the problem (2.6). We focus on the spherical harmonics $\mathrm{P_N}$ and discrete ordinate $\mathrm{S_N}$ methods. Alternative methods include Simplified $\mathrm{P_N}$ [57, 59, 58, 109], finite elements in angle [66, 101] and wavelet [27, 136].

Since the energy dependency and the fission source are not used in the following developments, let us replace the transport problem (2.6) by the one-group transport problem.

Problem 2.3.1 (One-group transport problem)**.** *Find $u(x,\omega)$ solution of*

$$\omega\cdot\nabla u+\sigma_t(x)u=\int_{\mathbf{S}^2}\sigma_s(x,\omega\cdot\omega')u(x,\omega')\,\mathrm{d}\omega'+q(x,\omega)\qquad in\quad X, \tag{2.7a}$$

$$u=f(x,\omega)\qquad on\quad \Gamma_-. \tag{2.7b}$$

2.3.1 The spherical harmonics

Let us introduce the spherical harmonics as defined by [20, Appendix A]. Other useful references are [12, p. 315] and [88, Appendix A]. Let us recall that $\omega(\theta,\varphi)\in\mathbf{S}^2$ has components $\omega_x=\sin\theta\cos\varphi$, $\omega_y=\sin\theta\sin\varphi$ and $\omega_z=\cos\theta$. In addition $\mathrm{d}\omega=1/4\pi\sin\theta\,\mathrm{d}\theta\,\mathrm{d}\varphi$.

Definition 2.3.1. *Let $\omega(\theta,\varphi)\in\mathbf{S}^2$, n and m two integers. The real spherical harmonics $y_n^m(\omega)$ of degree n and order m are defined by*

$$y_n^0(\omega)=(2n+1)^{\frac12}P_n(\cos\theta),\qquad m=0, \tag{2.8}$$

$$y_n^m(\omega)=(2C_n^m)^{\frac12}P_n^m(\cos\theta)\cos(m\varphi),\qquad m\in[\![1,n]\!], \tag{2.9}$$

$$y_n^{-m}(\omega)=(2C_n^m)^{\frac12}P_n^m(\cos\theta)\sin(m\varphi),\qquad m\in[\![1,n]\!]. \tag{2.10}$$

The functions P_n are the Legendre polynomials and the associated Legendre functions are denoted by P_n^m. The constant C_n^m is defined by

$$C_n^m=(2n+1)\frac{(n-m)!}{(n+m)!}. \tag{2.11}$$

The Legendre polynomial of degree n, P_n are defined by,

$$P_0(\mu)=1, \tag{2.12}$$

$$P_n(\mu)=\frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}\mu^n}(\mu^2-1),\quad n\geqslant 1. \tag{2.13}$$

The associated Legendre functions are denoted P_n^m,

$$P_n^0(\mu)=P_n(\mu),\qquad n\geqslant 0, \tag{2.14}$$

$$P_n^m(\mu)=(-1)^m(1-\mu)^{\frac{m}{2}}\frac{\mathrm{d}^m}{\mathrm{d}\mu^m}P_n(\mu),\qquad n\geqslant 0;\ m\in[\![1,n]\!]. \tag{2.15}$$

Let us denote $(\cdot,\cdot)_{\mathbf{S}^2}$ the inner product of $L^2(\mathbf{S}^2)$, given by,

$$(u,v)_{\mathbf{S}^2} = \int_{\mathbf{S}^2} uv \,\mathrm{d}\omega\,.$$

For all integers $n,\ l$ and $m \in [\![-n,n]\!]$, $k \in [\![-l,l]\!]$, the spherical harmonics satisfy

$$\left(y_n^m, y_l^k\right)_{\mathbf{S}^2} = \delta_{n,l}\delta_{m,k}, \tag{2.16}$$

with $\delta_{n,l}$ the Kronecker symbol.

Proposition 2.3.2. *The family* $\{y_n^m;\ n \in \mathbb{N},\ m \in [\![-n,n]\!]\}$ *is a complete orthonormal basis of the space* $L^2(\mathbf{S}^2)$*. Moreover*

$$P_n(\omega\cdot\omega') = \frac{1}{2n+1}\sum_{m=-n}^{n} y_n^m(\omega)y_n^m(\omega'), \tag{2.17}$$

$$y_n^m(-\omega) = (-1)^n y_n^m(\omega). \tag{2.18}$$

The equation (2.17) *is known as the spherical harmonic addition theorem and* (2.18) *means that the parity of* y_n^m *is the same as* n*.*

Proof. See [12, p. 315] and [90, p. 128]. □

2.3.2 Approximation of scattering cross-sections

The traditional way of approximating the scattering cross-sections σ_s is a truncated Legendre series expansion

$$\sigma_s(x,\omega\cdot\omega') \approx \sum_{n=0}^{N} \sigma_s^n(x)P_n(\omega\cdot\omega').$$

In general, the quadrature is stopped at a, the order of anisotropy in the medium around x. Hence, when $n > a$, $\sigma_s^n = 0$. Using the spherical harmonics addition theorem (2.17), it follows

$$\sigma_s(x,\omega\cdot\omega') \approx \sum_{n=0}^{N} \sigma_s^n(x) \sum_{m=-n}^{n} y_n^m(\omega)y_n^m(\omega').$$

The same discretization applies to multigroup scattering cross-sections $\sigma_s^{g,g'}$. We denote by $u_n^m(x) = (u, y_n^m)_{\mathbf{S}^2}$ the angular flux moments. The problem (2.7) is rewritten

$$\begin{aligned} \omega\cdot\nabla u + \sigma_t(x)u = \sum_{n=0}^{N}\sigma_s^n(x)\sum_{m=-n}^{n} u_n^m(x)y_n^m(\omega) + q(x,\omega) \qquad & \text{in} \quad X, \\ u = f(x,\omega) \qquad & \text{on} \quad \Gamma_-. \end{aligned}$$

This being said, in the remainder of the presentation, we assume isotropic scattering, i.e. $\sigma_s(x,\omega\cdot\omega') = \sigma_s(x)$ or equivalently $a = 0$. The problem (2.7) becomes

Problem 2.3.3 (One-group isotropic scattering transport)**.** *Find* $u(x,\omega)$ *solution of*

$$\omega \cdot \nabla u + \sigma_t(x)u = \sigma_s(x) \int_{\mathbf{S}^2} u(x,\omega')\,\mathrm{d}\omega' + q(x,\omega) \qquad \textit{in} \quad X, \tag{2.19a}$$

$$u = f(x,\omega) \qquad \textit{on} \quad \Gamma_-. \tag{2.19b}$$

Here we recognize the scalar flux $u_0(x) = \int_{\mathbf{S}^2} u(x,\omega')\,\mathrm{d}\omega'$*, which is the flux moment of order zero.*

2.3.3 The discrete ordinate method

The discrete ordinate or $\mathrm{S_N}$ method [32] is a quadrature on the angular variable. The direction ω is replaced by discrete directions ω_i and the integral on the unit sphere $\mathbf{S}^2$ is approximated by the quadrature

$$\int_{\mathbf{S}^2} u(x,\omega)\,\mathrm{d}\omega \approx \sum_{i=1}^{N} \mathrm{w}_i u(x,\omega_i),$$

where w_i are the quadrature weights, and N is the number of finite directions used.

Let us denote $u(x,\omega_i) = u_i(x)$, $q(x,\omega_i) = q_i(x)$ and $f(x,\omega_i) = f_i(x)$. The angular flux $u(x,\omega)$ is replaced by a discrete angular flux $(u_i(x))_{1\leqslant i\leqslant N}$ in problem (2.19). The result is a system of N coupled partial differential equations on D.

Problem 2.3.4 (Discrete ordinate transport problem)**.** *For all* $1 \leqslant i \leqslant N$*, find* $u_i(x)$ *solution of*

$$\omega_i \cdot \nabla u_i + \sigma_t(x)u_i = \sigma_s(x) \sum_{i=1}^{N} \mathrm{w}_i u_i(x) + q_i(x) \qquad \textit{in} \quad D, \tag{2.20a}$$

$$u_i = f_i(x) \qquad \textit{on} \quad \partial D_i^-, \tag{2.20b}$$

with $\partial D_i^- = \{x \in \partial D \ ;\ \omega_i \cdot n(x) < 0\}$.

After a spatial approximation, one obtains a fully discrete scheme for solving (2.6).

The discrete ordinate method is known to suffer from nonphysical oscillations, also referred to as ray effects [81], in a medium with little or no scattering, a source is propagated preferentially along the ordinate directions, leading to artificially non-monotoneous flux behavior.

To the question of the convergence of the discrete ordinate approximation for the neutron transport problem, an affirmative answer is given by [107] for slab geometry, [78] for two-dimensional geometry, and [7] on three-dimensional geometry.

2.3.4 The spherical harmonics method

The spherical harmonics or $\mathrm{P_N}$ method consists in approximating the angular neutron flux $u(x,\omega)$ by the truncated series on $y_n^m(\omega)$ the real spherical harmonics, defined in

Section 2.3.1,

$$u(x,\omega) \approx u^N(x,\omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} u_n^m(x) y_n^m(\omega). \tag{2.21}$$

The functions u_n^m are called *flux moments* and depend only on the spatial variable. The sources q and f are approximated in the same way. Since the angular flux is approximated for all directions, the spherical harmonics method is immune to ray effects.

The classic P_N method [93, 94, 44] consists of injecting these fluxes and sources into the equation (2.6a). This equation is then multiplied by y_l^k, knowing that the spherical harmonics are orthonormal, see equation (2.16). Finally, we integrate over ω to obtain a system of coupled equations, which are independent of the angular variable. It raises the question of the boundary conditions to be used to replace (2.6b) [120]. Mark [93] and Marshak [94] propose heuristics for writing these conditions.

In this work, we adopt a different approach to the P_N method by using the Galerkin method. Starting from (2.21), the flux moments are discretized using a finite element method. The result is injected into a weak formulation of the problem (2.6), and the test functions chosen are the basis functions of the approximation space. Furthermore, the question of boundary conditions does not arise [20].

This approach is presented in more depth in Chapter 3 and a detailed analysis is performed in Chapter 5.

2.4 Spatial discretization

Starting from the S_N transport equation (2.20) and considering a single fixed direction $\beta = \omega_i$, one obtains a one-group one-direction transport problem, here referred to as advection-reaction problem. We will describe the main features of the spatial discretization using this equation as a model problem.

Problem 2.4.1 (Advection-reaction problem)**.** *Find $u(x)$ solution of*

$$\beta \cdot \nabla u + \sigma u = q(x) \qquad \text{in} \quad D, \tag{2.22a}$$

$$u = f(x) \qquad \text{on} \quad \partial D^-, \tag{2.22b}$$

with $\partial D^- = \{x \in \partial D \ ; \ \beta \cdot n(x) < 0\}$.

It is a typical linear hyperbolic equation. The existence and uniqueness of solutions to this problem are discussed in [86, Theorem 3] and [50, Proposition 5.9]. The classic methods for solving it are the finite element method, the discontinuous Galerkin finite element method and the method of characteristics.

2.4.1 Finite element method

The idea of the finite element method [76, 1, 36, 12, 2, 50] is to find a solution to the equation (2.22) as a linear combination of finite-dimensional basis functions. It transforms the continuous problem (2.22) into a linear system $AU = b$, whose discrete solution

approaches the solution of the continuous problem more accurately as the size of the system increases. The process begins by partitioning the domain D into triangles D_r to obtain a conformal triangular mesh D_h. In order to obtain an accurate approximation, the triangles are not allowed to be arbitrarily flat. We then introduce the finite-dimensional space V_h of functions that are continuous on $\overline{D}$ (the closure of D) and polynomial of degree at most k on each cell D_r,

$$V_h = \left\{ v \in C^0(\overline{D});\ \forall D_r \in D_h,\ v|_{D_r} \in \mathbb{P}^k(D_r) \right\}.$$

Lemma 2.4.2. *Since V_h consists of continuous piecewise polynomials, the first derivative of these functions exists and belongs to $L^2(D)$,*

$$V_h \subset H^1(D).$$

Proof. See [76, Chapter 3] and [50, Chapter 1]. □

We consider the discrete variational formulation with weakly imposed boundary conditions,

Problem 2.4.3. ([76, §9.5]) *Find $u_h \in V_h$ such that*

$$a(u_h, v_h) = l(v_h) \quad \forall v_h \in V_h, \tag{2.23}$$

with

$$\begin{aligned} a(u_h, v_h) &= \int_D (\beta \cdot \nabla u_h + \sigma u_h) v_h - \int_{\partial D^-} u_h v_h (\beta \cdot n), \\ l(v_h) &= \int_D q v_h - \int_{\partial D^-} f v_h (\beta \cdot n). \end{aligned}$$

Remark 2.4.4. *The exact solution u of (2.22) is also solution of the weak formulation (2.23) in general,*

$$a(u, v_h) = l(v_h) \quad \forall v_h \in V_h.$$

Of course, V_h is finite-dimensional and u does not belong to V_h.

The integer $J = \dim V_h$ is the number of unknowns in the discrete problem, known as degrees of freedom. Let $\{\varphi_1, \cdots, \varphi_J\}$ be a linearly independent family of functions that span V_h. Any function $u_h \in V_h$ is uniquely defined by its degrees of freedom $u_j \in \mathbb{R}$,

$$u_h(x) = \sum_{j=1}^{J} u_j \varphi_j(x).$$

We denote the vector of degrees of freedom by $U = (u_j)_{j=1}^J$. By choosing the functions $(\varphi_i)_{1 \leqslant i \leqslant J}$ as test functions, the weak formulation can then be rewritten,

$$\sum_{j=1}^{J} a(\varphi_j, \varphi_i) u_j = l(\varphi_i), \quad \forall\ 1 \leqslant i \leqslant J.$$

In other words,

$$AU = b,$$

with

$$A = (a(\varphi_j, \varphi_i))_{1 \leqslant i,j \leqslant J} \quad \text{and} \quad b = (l(\varphi_i))_{1 \leqslant i \leqslant J}.$$

Calculating the coefficients of the matrix A and vector b involves calculating integrals of basis functions over triangles. This calculation is generally reduced to a calculation on a reference triangle.

2.4.2 Discontinuous Galerkin method

If the exact solution to the advection problem is not smooth, the numerical solutions using the continuous finite element method may exhibit spurious oscillations around the discontinuity. With the discontinuous Galerkin method (DG) [112, 86, 76], the continuity constraint of the V_h space is removed. The new space defined in this way, still denoted V_h, is

$$V_h = \left\{ v \in L^2(D);\ \forall D_r \in D_h,\ v|_{D_r} \in \mathbb{P}^k(D_r) \right\}.$$

The functions of V_h can be discontinuous at the interfaces between two cells. It is therefore necessary to describe what happens at the interfaces, and this is referred to as numerical flux. The classic numerical flux for advection problems is the upwind flux [112, 24]: according to the sign of the normal velocity, the flux entering a cell through a face is the flux outgoing from its neighbouring cell on the other side of the face. Since the direction β is fixed, for a given cell D_r we know its incoming faces $F \subset \partial D_r^-$ also known as incident faces and its outgoing faces $F \subset \partial D_r^+$, where $\partial D_r^\pm = \{x \in \partial D_r\ ;\ \pm\beta \cdot n_r(x) > 0\}$, see Figure 2.1.

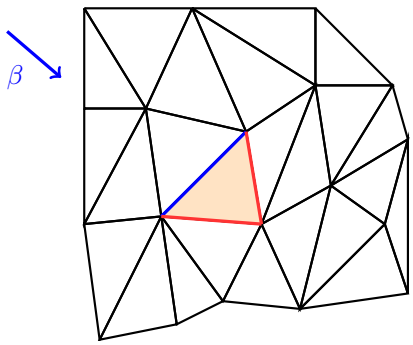


Figure 2.1: Incoming (▬) and outgoing (▬) faces of a mesh cell.

On any incident face F, the upwind flux is defined by

$$u^\uparrow = \begin{cases} f & \text{if } F \subset \partial D \cap \partial D_r^-, \\ u_r^F & \text{else}, \end{cases}$$

where u_r^F is the trace on F of the flux in the adjacent region to D_r by the face F. In each element we can define a local problem to be solved.

Problem 2.4.5. ([46, §2.3.4]) *For each $D_r \in D_h$, find $u_h \in \mathbb{P}^k(D_r)$ such that*

$$a_r(u_h, v_h) = l_r(v_h) \quad \forall v_h \in \mathbb{P}^k(D_r), \tag{2.24}$$

with

$$\begin{aligned} a_r(u_h, v_h) &= \int_{D_r} -u_h(\beta \cdot \nabla v_h) + \sigma u_h v_h + \int_{\partial D_r^+} u_h v_h(\beta \cdot n), \\ l_r(v_h) &= \int_{D_r} q v_h - \int_{\partial D_r^-} u_h^{\uparrow} v_h(\beta \cdot n). \end{aligned}$$

In each cell, the only data required for resolution is the flux $u_h^{\uparrow}$ entering the cell. The discontinuous Galerkin method therefore enables to solve the problem by successive elimination following a propagation front starting from the incoming boundary ∂D^- of the domain. This process is called *sweeping*.

Further details of the discontinuous Galerkin method are given in Chapters 3 and 5. For a complete survey and a historical overview of the subject, we refer to [5, 24, 70, 134, 46, 37]. The recent developments in the discontinuous Galerkin method for radiation transport include for example the development of diffusion-limit preserving numerical fluxes [111, 67], adaptive mesh refinement [135] and *hp*-refinement [54, 101, 68].

Chapter 3

Numerical solution of the Boltzmann transport equation by spherical harmonics and discontinuous Galerkin methods

This chapter describes the numerical scheme proposed by [20] to solve the Boltzmann transport equation.

Outline of the current chapter

3.1 Introduction

Based on the variational formulation proposed in [22], a new approximation method for the Boltzmann transport equation is introduced in [20]. The resulting solver called `NYMO` has been implemented in the frame of CEA reactor physics platform `APOLLO3`® [122, 98]. The proposed numerical method can treat 2D unstructured and non-conforming meshes with line segments, circles and circular arcs [128].

This chapter is organized as follows. In Section 3.2 the neutron transport problem and the weak formulation used are introduced. Then the angle-space approximation framework is described in sections 3.3, 3.4 and 3.5. Finally in Section 3.6, the method for solving the linear system resulting from the discretization without matrix-assembly is presented.

3.2 The continuous problem

In this section we introduce the notations used, the transport problem is recalled and then we derive the variational formulation.

3.2.1 Multigroup neutron transport

Let us consider $x \in D$ with D an open bounded set in $\mathbb{R}^d$ ($d = 1$, 2 or 3), with boundary ∂D and outward normal n. Let $\omega \in \mathbf{S}^2$ with $\mathbf{S}^2$ the unit sphere of $\mathbb{R}^3$. The angular domain $\mathbf{S}^2$ is parametrized by two angles $\theta \in [0, \pi]$ and $\varphi \in [0, 2\pi]$ such that $\omega(\theta, \varphi) \in \mathbf{S}^2$, with components $\omega_x = \sin\theta\cos\varphi$, $\omega_y = \sin\theta\sin\varphi$ and $\omega_z = \cos\theta$. The integral over $\mathbf{S}^2$ is defined by

$$\int_{\mathbf{S}^2} f(\omega)\,\mathrm{d}\omega = \frac{1}{4\pi}\int_0^{2\pi}\left(\int_0^{\pi} f(\theta,\varphi)\sin\theta\,\mathrm{d}\theta\right)\mathrm{d}\varphi\,.$$

Let us denote by g the energy group index, and G the total number of energy groups, with $1 \leqslant g \leqslant G$. The multigroup angular flux is denoted by $u = (u^g(x,\omega))_{g=1,\cdots,G}$. The multigroup neutron transport problem is defined on the phase space $X = D \times \mathbf{S}^2$. The boundary $\Gamma = \partial D \times \mathbf{S}^2$ consists of an incoming boundary Γ_- and outgoing boundary Γ_+ defined by

$$\Gamma_\pm = \left\{(x,\omega) \in \partial D \times \mathbf{S}^2,\ \pm\omega\cdot n(x) > 0\right\}.$$

Next, let us recall the operators involved in the transport equation in their multigroup form.

Scattering operator The scattering operator H^g models the transfer of neutrons from an energy group g' to another group g after a collision. It is defined by

$$(H^g u)(x,\omega) = \sum_{g'=1}^{G} \int_{\mathbf{S}^2} \sigma_s^{g,g'}(x, \omega\cdot\omega') u^{g'}(x,\omega')\,\mathrm{d}\omega' , \tag{3.1}$$

with $\sigma_s^{g,g'}$, the transfer or scattering cross-section, the probability for a neutron of energy g' to arrive in the energy group g after collision with a nuclide. The scattering cross-sections are generally anisotropic.

Fission operator The fission operator F^g quantifies the emission of new neutrons resulting from a fission reaction. It is defined as

$$(F^g u)(x,\omega) = \sum_{\alpha} \chi_\alpha^g(x) \sum_{g'=1}^{G} \nu\sigma_{f,\alpha}^{g'}(x) \int_{\mathbf{S}^2} u^{g'}(x,\omega')\,\mathrm{d}\omega' . \tag{3.2}$$

The quantities $\nu\sigma_{f,\alpha}^g$ and χ_α^g are respectively the fission production and the fission emission spectrum for isotope α, and are assumed isotropic. The sum over α represents the sum over fissile isotopes.

The total cross-section is denoted σ^g and the absorption section σ_a^g. The problem to be solved is then stated.

Problem 3.2.1 (Multigroup transport source problem)**.** *Given an external source $q = (q^g)_{g=1,\cdots,G}$, an incoming flux $f = (f^g)_{g=1,\cdots,G}$ through the incoming boundary Γ_-, and the nuclear data, find the multigroup neutron flux $u = (u^g(x,\omega))_{g=1,\cdots,G}$ solution of: for all $g = 1,\cdots,G$,*

$$\omega\cdot\nabla u^g + \sigma^g u^g = H^g u + F^g u + q^g \qquad \text{in} \quad X, \tag{3.3a}$$

$$u^g = f^g \qquad \text{on} \quad \Gamma_-. \tag{3.3b}$$

In the absence of external source and incoming flux, that is q and f are zero, equation (3.3) is studied as an eigenvalue problem, the objective being to obtain the effective multiplication factor $k_{\text{eff}} = \lambda$.

Problem 3.2.2 (Multigroup transport critical problem)**.** *Find the multigroup neutron flux $u = (u^g(x,\omega))_{g=1,\cdots,G}$ and the associated eigenvalue λ solution of: for all $g = 1,\cdots,G$,*

$$\omega\cdot\nabla u^g + \sigma^g u^g = H^g u + \frac{1}{\lambda} F^g u \qquad \text{in} \quad X, \tag{3.4a}$$

$$u^g = 0 \qquad \text{on} \quad \Gamma_-. \tag{3.4b}$$

In the following, we focus on solving the source problem (3.3). The critical problem is solved by successive iterations of the source problem, as described in Section 3.6.

3.2.2 Variational formulation

The variational space used is the space W where lives the unique solution of the transport problem [42],

$$W = \left\{ v \in L^2(X) \;/\; \omega \cdot \nabla v \in L^2(X),\ v|_{\Gamma_+} \in L^2(\Gamma_+, |\omega \cdot n| \,\mathrm{d}s \,\mathrm{d}\omega) \right\}. \tag{3.5}$$

From here on, the total cross section σ^g is assumed to be a positive non-vanishing function. In addition, we use the Green's formula [42, p. 225].

Lemma 3.2.3 (Green's formula)**.** *For any* $(u, v) \in W \times W$*,*

$$\int_X ((\omega \cdot \nabla u)v + u(\omega \cdot \nabla v)) \,\mathrm{d}x \,\mathrm{d}\omega = \int_\Gamma uv(\omega \cdot n) \,\mathrm{d}s \,\mathrm{d}\omega \,. \tag{3.6}$$

The variational formulation proposed in [22] consists in multiplying (3.3a) by $\left(v + \frac{1}{\sigma^g}(\omega \cdot \nabla v)\right)$, with $v \in W$ a test function,

$$\begin{aligned} &\frac{1}{\sigma^g}(\omega \cdot \nabla u^g)(\omega \cdot \nabla v) + \sigma^g u^g v + u^g(\omega \cdot \nabla v) + (\omega \cdot \nabla u^g)v \\ &= \left(H^g u + F^g u + q^g\right)\left(v + \frac{1}{\sigma^g}(\omega \cdot \nabla v)\right). \end{aligned}$$

After integrating over the phase space X and applying Green's formula (3.6), it follows

$$\begin{aligned} &\int_X \left(\frac{1}{\sigma^g}(\omega \cdot \nabla u^g)(\omega \cdot \nabla v) + \sigma^g u^g v\right) \mathrm{d}\omega \,\mathrm{d}x + \int_\Gamma u^g v(\omega \cdot n) \,\mathrm{d}\omega \,\mathrm{d}s \\ &= \int_X \left(H^g u + F^g u + q^g\right)\left(v + \frac{1}{\sigma^g}(\omega \cdot \nabla v)\right) \mathrm{d}\omega \,\mathrm{d}x \,. \end{aligned}$$

By writing the integral on Γ as the sum of two integrals on Γ_- and Γ_+ and using the boundary conditions (3.3b), the variational formulation obtained reads,

Problem 3.2.4 (Variational formulation)**.** *For all groups* $1 \leqslant g \leqslant G$*,*

$$\textit{find } u^g \in W \;\; \textit{such that } \; a^g(u^g, v) = h^g(u, v) + p^g(u, v) + L^g(v), \quad \forall v \in W \tag{3.7}$$

with

$$a^g(u^g, v) = \int_X \left(\frac{1}{\sigma^g}(\omega \cdot \nabla u^g)(\omega \cdot \nabla v) + \sigma^g u^g v\right) \mathrm{d}\omega \,\mathrm{d}x + \int_{\Gamma_+} u^g v(\omega \cdot n) \,\mathrm{d}\omega \,\mathrm{d}s \,, \tag{3.8}$$

$$h^g(u, v) = \int_X \left((H^g u)v + \frac{1}{\sigma^g}(H^g u)(\omega \cdot \nabla v)\right) \mathrm{d}\omega \,\mathrm{d}x \,, \tag{3.9}$$

$$p^g(u, v) = \int_X \left((F^g u)v + \frac{1}{\sigma^g}(F^g u)(\omega \cdot \nabla v)\right) \mathrm{d}\omega \,\mathrm{d}x \,, \tag{3.10}$$

$$L^g(v) = \int_X \left(q^g v + \frac{1}{\sigma^g} q^g(\omega \cdot \nabla v)\right) \mathrm{d}\omega \,\mathrm{d}x - \int_{\Gamma_-} f^g v(\omega \cdot n) \,\mathrm{d}\omega \,\mathrm{d}s \,. \tag{3.11}$$

Under the assumptions $q^q \in L^2$, $f^g \in L^2_-$ and the subcriticality conditions (1.19), it has been proved in [22] that the variational formulation (3.7) is equivalent to the original

transport problem (3.3). Equivalence of this formulation with transport self-adjoint angular flux equation (SAAF) [97], least-squares approach and the even (or odd) parity formulation [88, §6] is discussed in [20, §3].

Since σ^g is assumed to be non-zero, fully voided regions cannot be modeled. However, numerical tests show that almost voided regions, with σ^g of the order of 10^{-6} are supported.

3.3 The discrete setting

The numerical scheme used to solve problem (3.7) combines the spherical harmonics method for the treatment of angular dependence and the discontinuous Galerkin finite element method for spatial discretization. The link between mesh regions is done using upwind flux at cell interfaces.

3.3.1 Spherical harmonics method for the angular discretization

The neutron angular flux $u^g(x,\omega)$ can be expanded in terms of real-valued spherical harmonics functions $y_n^m(\omega)$ defined at Section 2.3.1. Introducing the angular flux moments $u_n^{m,g}(x) = (u^g, y_n^m)_{\mathbf{S}^2}$ we write

$$u^g(x,\omega) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} u_n^{m,g}(x) y_n^m(\omega). \tag{3.12}$$

The first $u_0^{0,g}(x)$ and second $(u_1^{m,g}(x))_{m\in\{-1,0,1\}}$ angular moments of u^g are respectively the scalar flux and current vector. The $\mathrm{P_N}$ method consists in truncating the expansion (3.12) of the angular flux to the order N, that is:

$$u^g(x,\omega) \approx u^{N,g}(x,\omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} u_n^{m,g}(x) y_n^m(\omega). \tag{3.13}$$

The error resulting from this truncation is studied in Chapter 5.

Remark 3.3.1. *The number of spherical harmonics of degree lower or equal to $N \in \mathbb{N}$ is $d_S = (N+1)^2$. Due to symmetries in 2D, one take into account only the m that have the same parity as n and $d_S = 1/2(N+1)(N+2)$, see [1, p. 192]. In 1D, one only takes into account $m = 0$ and $d_S = N+1$. In the general case $d_S = 1/2(N+1)((d-1)N+2)$.*

In the following, we tackle the discretization of the flux moments $u_n^{m,g}(x)$.

3.3.2 Polygonal meshes with curved faces

The spatial domain D is meshed into disjoint elements D_r also referred to as regions. The mesh is denoted D_h and

$$\overline{D} = \bigcup_{D_r \in D_h} \overline{D}_r. \tag{3.14}$$

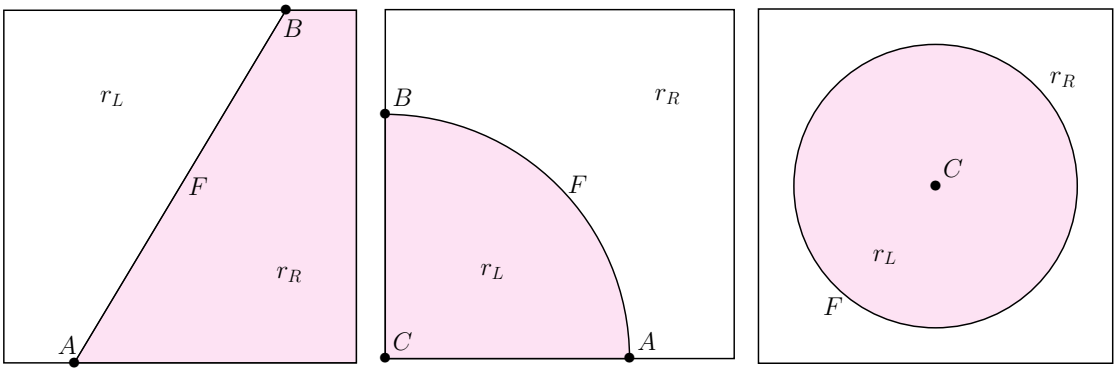


Figure 3.1: Types of faces supported by `NYMO` and their description.

In dimension 2, the mesh D_h can be unstructured and non-conforming, that is with hanging nodes. Figure 3.1 shows the types of faces supported by `NYMO`. Curved faces, like circles and circular arcs, are considered to be able to represent cladding surrounding the fuel rods.

The basic element is the point, defined by its coordinates $(x_1, \cdots, x_d)$. The mesh is then described as a cloud of points connected by lines, arcs or circles. Each point in the mesh has a unique identifier. The mesh faces are described according to their shape:

- A line segment is described by its endpoints A and B, the region at the left r_L and the region at the right r_R along the direction A to B.
- A circle arc is described by its radius, endpoints A and B and center C. To this, we add left r_L and right r_R regions in the direction A to B.
- A circle is defined by its radius, its center C, the inner region r_L and the outer region r_R.

We proceed to the spatial discretization using discontinuous Galerkin finite elements.

3.3.3 Discontinuous Galerkin method for the spatial discretization

To obtain a full discretization, the angular flux moments $u_n^{m,g}(x)$ are approximated by piecewise polynomials over the mesh D_h of the domain D. With any region $D_r \in D_h$, we associate a finite-dimensional space $\mathbb{P}^k(D_r)$ of d-variate polynomials of total degree at most k on D_r. Then we collect all those spaces to form the space of piecewise polynomials over D_h,

$$\mathbb{P}_h^k = \left\{ v \in L^2(D);\ \forall D_r \in D_h,\ v|_{D_r} \in \mathbb{P}^k(D_r) \right\}. \tag{3.15}$$

Due to the discontinuous nature of the spatial approximation, a numerical flux at mesh cell interfaces has to be set. The natural choice when dealing with particle transport is the upwind numerical flux [112, 86]. From [24] we know that DG discretization of the linear advection equation based on upwind flux is stable. For an interior face F shared by two cells D_{r1} and D_{r2} with respective outward normal n_1, n_2, the numerical flux across F is given by

$$u_F^{\uparrow} = \begin{cases} u|_{D_{r2}}, & \text{if } \omega \cdot n_1 < 0, \\ u|_{D_{r1}}, & \text{if } \omega \cdot n_1 > 0. \end{cases} \tag{3.16}$$

Else if $F \in \partial D$ is a boundary face with outward normal n, the numerical flux is given by

$$u_F^{\uparrow} = \begin{cases} f, & \text{if } \omega \cdot n < 0, \\ u, & \text{if } \omega \cdot n > 0. \end{cases} \tag{3.17}$$

In particular, inside a region D_{r1} the incoming flux on a face $F \in \partial D_{r1}$ is given either by the boundary condition f if $F \in \partial D_{r1} \cap \partial D$ or by the flux in the adjacent region D_{r2} if F is an interface.

3.3.4 The full approximation, a global point of view

Given a basis, let say $(\varphi_j)_{j=1,\cdots,J}$ of $\mathbb{P}_h^k$, the fully discrete solution $u_h^{N,g}$ can be written on the basis $\varphi_j y_n^m$,

$$u_h^{N,g} = \sum_{n=0}^{N} \sum_{m=-n}^{n} \sum_{j=1}^{J} u_{n,j}^{m,g} \varphi_j(x) y_n^m(\omega). \tag{3.18}$$

Here $J = \dim \mathbb{P}_h^k = \mathrm{card}(D_h)\binom{k+d}{d}$ is the total number of spatial degree of freedom. The coefficients $u_{n,j}^{m,g}$ fully determine the approximated flux. The incoming flux f and the source term q are approximated in the same way. The approximation space is

$$W_h = \mathrm{span}(\varphi_j y_n^m), \tag{3.19}$$

and it is useful to note that

$$\dim W_h = \dim(\mathbb{P}_h^k) d_S = \mathrm{card}(D_h) \binom{k+d}{d} \left(\frac{1}{2}\right) (N+1)((d-1)N+2).$$

Taking into account the energy discretization, the total number of degrees of freedom is

$$N_{\mathrm{dof}} = G \dim W_h.$$

The convergence of the full approximation is studied in Chapter 5. Next, we look at a local formulation.

3.3.5 A local point of view

From here on, and for the rest of the chapter, we adopt a local point of view. For the sake of simplicity, the notations a, h, p and L (3.8) - (3.11), will henceforth be used to designate their restriction to the local phase space $X_r = D_r \times \mathbf{S}^2$. In the same way, Γ_-^r and Γ_+^r are used to denote the incoming boundary and the outgoing boundary of X_r. Hence we can write,

$$\Gamma_\pm^r = \bigcup_{F \in \partial D_r} \Gamma_\pm^r(F) \quad \text{with} \quad \Gamma_\pm^r(F) = \left\{ (x,\omega) \in F \times \mathbf{S}^2 \ / \ \pm \omega \cdot n_F > 0 \right\}. \tag{3.20}$$

The local approximation space W_h^r is obtained using the space $\mathbb{P}^k(D_r)$ instead of the whole polynomial space $\mathbb{P}_h^k$. The local discretized problem is then stated.

Problem 3.3.2. *For all cells $D_r \in D_h$, find $u^g \in W_h^r$ such that*

$$a^g(u^g, v) = h^g(u, v) + p^g(u, v) + L^g(v), \quad \forall v \in W_h^r \tag{3.21}$$

with

$$L^g(v) = \int_{X_r} \left(q^g v + \frac{1}{\sigma^g} q^g (\omega \cdot \nabla v) \right) \mathrm{d}\omega \, \mathrm{d}x - \sum_{F \in \partial D_r} \int_{\Gamma_-^r(F)} u_F^{\uparrow g} v (\omega \cdot n) \, \mathrm{d}\omega \, \mathrm{d}s \,. \tag{3.22}$$

Following definitions (3.16) and (3.17) it is important to notice that in (3.22), $u_F^{\uparrow g}$ is independent of the flux in the region D_r, and is either the boundary flux f or the flux coming from the neighboring regions of D_r, since the integral is on the incoming boundary.

In practice, the local formulation is the one used in the implementation. We write (3.18), with $J = \dim \mathbb{P}^k(D_r)$, the local number of spatial degrees of freedom. As described in the section 3.6, the matrices resulting from the discretization are not assembled. It is however possible to obtain the global formulation by summing (3.21) over all mesh elements. Table 3.1 provides the number of degrees of freedom per element induced by a $\mathrm{P_N}$-DG discretization in 3D.

Table 3.1: Number of degrees of freedom per element and per energy group involved in a 3D $\mathrm{P_N}$-DG discretization.

N_{dof}/D_r	P_0	P_1	P_2	P_3	P_4	P_5	P_6
$\mathbb{P}_h^0$	1	4	9	16	25	36	49
$\mathbb{P}_h^1$	4	16	36	64	100	144	196
$\mathbb{P}_h^2$	10	40	90	160	250	360	490

The fully discretized problem is obtained by replacing in (3.21), $v = \varphi_i(x) y_n^m(\omega)$ and discrete functions by their energy-spatial-angular approximations,

$$u_h^{N,g} = \sum_{l=0}^{N} \sum_{k=-l}^{l} \sum_{j=1}^{J} u_{l,j}^{k,g} \varphi_j(x) y_l^k(\omega), \tag{3.23}$$

$$q_h^{N,g} = \sum_{l=0}^{N} \sum_{k=-l}^{l} \sum_{j=1}^{J} q_{l,j}^{k,g} \varphi_j(x) y_l^k(\omega), \tag{3.24}$$

$$f_h^{N,g} = \sum_{l=0}^{N} \sum_{k=-l}^{l} \sum_{j=1}^{J} f_{l,j}^{k,g} \varphi_j(x) y_l^k(\omega). \tag{3.25}$$

One obtains

$$a^g(u_h^{N,g}, \varphi_i y_n^m) = h^g(u_h^N, \varphi_i y_n^m) + p^g(u_h^N, \varphi_i y_n^m) + L^g(\varphi_i y_n^m). \tag{3.26}$$

It is a linear system of unknowns $u_{l,j}^{k,g}$ with $l \in [\![0, N]\!]$, $k \in [\![-l, l]\!]$, $j \in [\![1, J]\!]$ and $g \in [\![1, G]\!]$. We describe the matrices of this system and the solving method in the following.

3.3.6 Spatial basis functions

Here we describe the 3D case, the other cases being constructed similarly. Let us define a 3-index, $\alpha \in \mathbb{N}^3$, with the convention $|\alpha| = \sum_{i=1}^{3} \alpha_i$. Let us note by (x_0, y_0, z_0) the coordinates of a local origin of the region D_r. On each cell D_r, we choose to use *translated monomials*, defined as follows,

$$\phi_\alpha = (x - x_0)^{\alpha_1}(y - y_0)^{\alpha_2}(z - z_0)^{\alpha_3}. \tag{3.27}$$

Proposition 3.3.3. *The set of translated monomials* $\{\phi_\alpha;\ |\alpha| \leqslant k\}$ *is a basis of* $\mathbb{P}^k(D_r)$.

Proof. It is straightforward to show that $\{\phi_\alpha;\ |\alpha| \leqslant k\}$ is a linearly independent family that span $\mathbb{P}^k(D_r)$. □

The benefits of using translated monomials is described in Section 3.6. In the following, we index this basis by $j \in [\![1, J]\!]$, with $J = \dim(\mathbb{P}^k(D_r))$.

The family $(\phi_j)_{j\in[\![1,J]\!]}$ is then orthogonalized on the region D_r with respect to the $L^2(D_r)$-product using the Gram-Schmidt process, Algorithm 2. The $L^2(D_r)$-product is here denoted $(u, v)_r = \int_{D_r} uv \,\mathrm{d}x$ and $\|v\|_r = (v, v)_r^{1/2}$. The only thing to do for the first basis function is a normalization, $\varphi_1 = \|\phi_1\|_r$. Next, one compute the projection p_{21} of ϕ_2 onto the function φ_1, that is $p_{21} = (\phi_2, \varphi_1)_r \varphi_1$. Then, one subtract the projection from ϕ_2, that is $\tilde{\varphi}_2 = \phi_2 - p_{12}$. The new function $\tilde{\varphi}_2$ is orthogonal to φ_1. This step is completed by normalizing $\varphi_2 = \|\tilde{\varphi}_2\|_r$. Next, we move on to φ_3, and so on.

Algorithm 2 : Gram-Schmidt algorithm.

input : $(\phi_j)_{j\in[\![1,J]\!]}$ basis of $\mathbb{P}^k(D_r)$
output : $(\varphi_j)_{j\in[\![1,J]\!]}$ orthonormal basis of $\mathbb{P}^k(D_r)$

```
for j = 1 to J do
    φ̃_j = φ_j
    for i = 1 to j − 1 do
        φ̃_j = φ̃_j − (φ_j, φ_i)_r φ_i
    φ_j = φ̃_j / ‖φ̃_j‖_r
```

3.4 The elementary matrices

Here, one develops each term of (3.26). The triple sum $\sum_{l=0}^{N} \sum_{k=-l}^{l} \sum_{j=1}^{J}$ is hereafter noted $\sum_{l,k,j}$. In the following, the indexes N, h are dropped from $u_h^{N,g}$ for greater clarity. Let us consider $x = (x_1, \cdots, x_d) \in D_r$, the symbol ∂_p, with $p \in [\![1, d]\!]$ denotes the partial derivative with respect to x_p. In the same way, $(\omega_1, \omega_2, \omega_3)$ are the components of $\omega \in \mathbf{S}^2$ in the Cartesian basis of $\mathbb{R}^3$. The notation $\delta_{n,l}$ is the Kronecker symbol.

3.4.1 The bilinear form $a^g(u^g, \varphi_i y_n^m)$

Since $a^g(\cdot,\cdot)$ is bilinear,

$$a^g(u^g, \varphi_i y_n^m) = \sum_{l,k,j} u_{l,j}^{k,g} a^g(\varphi_j y_l^k, \varphi_i y_n^m). \tag{3.28}$$

The vector $u^g = \left(u_{l,j}^{k,g}\right)_{l\in[\![0,N]\!];\, j\in[\![1,J]\!]}^{k\in[\![-l,l]\!]}$ is the discrete solution in group g and region D_r and $a^g(\varphi_j y_l^k, \varphi_i y_n^m)$ are the coefficients of a certain matrix, which we define below.

Let us note

$$a_0^g(u^g, v) = \int_{X_r} \frac{1}{\sigma^g}(\omega\cdot\nabla u^g)(\omega\cdot\nabla v)\,\mathrm{d}\omega\,\mathrm{d}x\,,$$

$$a_1^g(u^g, v) = \int_{X_r} \sigma^g u^g v\,\mathrm{d}\omega\,\mathrm{d}x\,,$$

$$a^+(u^g, v) = \int_{\Gamma_+^r} u^g v(\omega\cdot n)\,\mathrm{d}\omega\,\mathrm{d}s\,,$$

such that from (3.8),

$$a^g(u^g, v) = a_0^g(u^g, v) + a_1^g(u^g, v) + a^+(u^g, v).$$

Using the discrete solution (3.23) and the bilinearity of a_0^g, a_1^g and a^+,

$$a_0^g(u^g, \varphi_i y_n^m) = \sum_{l,k,j} A_r^0(i,n,m;j,l,k)\frac{1}{\sigma^g} u_{l,j}^{k,g}, \tag{3.29}$$

where

$$A_r^0(i,n,m;j,l,k) = \sum_{p=1}^{3}\sum_{q=1}^{3}\left(\int_{D_r}\partial_p\varphi_i\partial_q\varphi_j\,\mathrm{d}x\right)\left(\int_{\mathbf{S}^2}\omega_p\omega_q y_n^m y_l^k\,\mathrm{d}\omega\right). \tag{3.30}$$

When n and l are of opposite parity, the term inside the integral $\int_{\mathbf{S}^2}$ is odd in ω, hence $A_r^0(i,n,m;j,l,k) = 0$. Next

$$a_1^g(u^g, \varphi_i y_n^m) = \sum_{l,k,j} A_r^1(i,n,m;j,l,k)\sigma^g u_{l,j}^{k,g}, \tag{3.31}$$

where

$$A_r^1(i,n,m;j,l,k) = \delta_{n,l}\delta_{m,k}\int_{D_r}\varphi_i\varphi_j\,\mathrm{d}x\,.$$

Since the basis $(\varphi_j)_{j=1,\cdots,J}$ is orthonormalized

$$A_r^1(i,n,m;j,l,k) = \delta_{n,l}\delta_{m,k}\delta_{i,j}. \tag{3.32}$$

Finally

$$a^+(u^g, \varphi_i y_n^m) = \sum_{l,k,j} A_r^+(i,n,m;j,l,k) u_{l,j}^{k,g}, \tag{3.33}$$

with

$$A_r^+(i,n,m;j,l,k) = \sum_{F\in\partial D_r} \int_F \varphi_i\varphi_j \int_{(\omega\cdot n)>0} y_n^m y_l^k (\omega\cdot n)\,\mathrm{d}\omega\,\mathrm{d}s \tag{3.34}$$

The A_r^0 A_r^1 and A_r^+ matrices are symmetric, and we calculate their coefficients exactly in Section 3.5.

3.4.2 The bilinear scattering source $h^g(u, \varphi_i y_n^m)$

One rewrites the scattering operator using the spherical harmonic addition theorem. Let a denote the order of anisotropy in the region D_r. Using the spherical harmonics addition theorem (2.17),

$$\sigma_s^{g,g'}(x, \omega\cdot\omega') = \sum_{n=0}^{a} \sigma_s^{n;g,g'} \sum_{m=-n}^{n} y_n^m(\omega) y_n^m(\omega').$$

In the following, when $n > a$, $\sigma_s^{n;g,g'} = 0$. Thus using the discrete solution (3.23) inside (3.1),

$$(H^g u)(x,\omega) = \sum_{l,k,j} \varphi_j(x) y_j^k(\omega) \sum_{g'=1}^{G} \int_{\mathbf{S}^2} \sigma_s^{l;g,g'} u_{l,j}^{k,g'}.$$

Then, let us note

$$h_1^g(u,v) = \int_{X_r} (H^g u) v\,\mathrm{d}\omega\,\mathrm{d}x\,,$$

$$h_2^g(u,v) = \int_{X_r} \frac{1}{\sigma^g}(H^g u)(\omega\cdot\nabla v)\,\mathrm{d}\omega\,\mathrm{d}x\,,$$

such that from (3.9),

$$h^g(u,v) = h_1^g(u,v) + h_2^g(u,v).$$

Using the discrete solution (3.23) and the bilinearity of h_1^g and h_2^g,

$$h_1^g(u, \varphi_i y_n^m) = \sum_{l,k,j} A_r^1(i,n,m;j,l,k) \sum_{g'=1}^{G} \sigma_s^{l;g,g'} u_{l,j}^{k,g'}, \tag{3.35}$$

where A_r^1 has already been defined at (3.32).

$$h_2^g(u, \varphi_i y_n^m) = \sum_{l,k,j} A_r^2(i,n,m;j,l,k) \sum_{g'=1}^{G} \frac{\sigma_s^{l;g,g'}}{\sigma^g} u_{l,j}^{k,g'}, \tag{3.36}$$

where

$$A_r^2(i,n,m;j,l,k) = \sum_{p=1}^{3} \left(\int_{D_r} (\partial_p \varphi_i) \varphi_j \, \mathrm{d}x \right) \left(\int_{\mathbf{S}^2} \omega_p y_n^m y_l^k \, \mathrm{d}\omega \right). \tag{3.37}$$

When n and l have the same parity, the term inside the integral $\int_{\mathbf{S}^2}$ is odd in ω, hence $A_r^2(i,n,m;j,l,k) = 0$.

3.4.3 Rewriting the scattering operator

As we will discuss in Section 3.6, solving the discrete problem involves three nested loops where we successively calculate fission and collision sources. The innermost loop - called *inner iterations* - is an iteration on neutron collisions of energy g, which remain at the energy level g after collision, this is self-scattering. To skip inner iterations, we send self-scattering to the left-hand side of equality (3.26).

Let us define

$$h_1^{g,g}(u^g, \varphi_i y_n^m) = \sum_{l,k,j} A_r^1(i,n,m;j,l,k) \sigma_s^{l;g,g} u_{l,j}^{k,g},$$

$$h_2^{g,g}(u^g, \varphi_i y_n^m) = \sum_{l,k,j} A_r^2(i,n,m;j,l,k) \frac{\sigma_s^{l;g,g}}{\sigma^g} u_{l,j}^{k,g},$$

and

$$h^{\star,g} = h_1^{\star,g} + h_2^{\star,g}.$$

The terms $h_1^{\star,g}$ and $h_2^{\star,g}$ are written as h_1^g (3.35) and h_2^g (3.35) by substituting the sum $\sum_{g'=1}^{G}$ by the same sum without self-scattering $\sum_{g'=1;g'\neq g}^{G}$. Thus, equality (3.26) can be now written

$$(a^g - h_1^{g,g} - h_2^{g,g})(u^g, \varphi_i y_n^m) = h^{\star,g}(u, \varphi_i y_n^m) + p^g(u, \varphi_i y_n^m) + L^g(\varphi_i y_n^m). \tag{3.38}$$

We can notice that

$$(a_1^g - h_1^{g,g})(u^g, \varphi_i y_n^m) = \sum_{l,k,j} A_r^1(i,n,m;j,l,k) \sigma_a^{l;g,g} u_{l,j}^{k,g} \tag{3.39}$$

with $\sigma_a^{l,g} = \sigma^g - \sigma_s^{l;g,g}$ the absoption cross-section.

3.4.4 The bilinear fission source $p^g(u, \varphi_i y_n^m)$

Let us note

$$p_1^g(u,v) = \int_{X_r} (F^g u) v \, \mathrm{d}\omega \, \mathrm{d}x \,,$$

$$p_2^g(u,v) = \int_{X_r} \frac{1}{\sigma^g} (F^g u)(\omega \cdot \nabla v) \, \mathrm{d}\omega \, \mathrm{d}x \,,$$

such that from (3.10),

$$p^g(u,v) = p_1^g(u,v) + p_2^g(u,v).$$

Using the discrete solution (3.23) and the bilinearity of p_1^g and p_2^g,

$$p_1^g(u, \varphi_i y_n^m) = \sum_{l,k,j} F_r^1(i,n,m;j,l,k) \sum_\alpha \chi_\alpha^g \sum_{g'=1}^G \nu \sigma_{f,\alpha}^{g'} u_{0,j}^{0,g'} \,, \tag{3.40}$$

and

$$p_2^g(u, \varphi_i y_n^m) = \sum_{l,k,j} F_r^2(i,n,m;j,l,k) \sum_\alpha \frac{\chi_\alpha^g}{\sigma^g} \sum_{g'=1}^G \nu \sigma_{f,\alpha}^{g'} u_{0,j}^{0,g'} \,, \tag{3.41}$$

where

$$F_r^1(i,n,m;j,l,k) = \delta_{n,0}\delta_{m,0}\delta_{l,0}\delta_{k,0}\delta_{i,j},$$

and

$$F_r^2(i,n,m;j,l,k) = \delta_{n,1}\delta_{l,0}\delta_{k,0} \sum_{p=1}^3 \int_{D_r} (\partial_p \varphi_i)\varphi_j \, \mathrm{d}x \int_{\mathbf{S}^2} \omega_p y_n^m \, \mathrm{d}\omega \,. \tag{3.42}$$

3.4.5 The linear source $L^g(\varphi_i y_n^m)$

To simplify the notations, let us refer to the upwind flux defined at (3.16)-(3.17) as f. Since here we are concerned by an integral on the incoming boundary of the element, f designates either the boundary condition if one considers a face $F \in \partial D_r \cap \partial D$ at the boundary of D, or the flux in the adjacent region to D_r if F is an interface. Let us note

$$L_1^g(v) = \int_{X_r} q^g v \, \mathrm{d}\omega \, \mathrm{d}x \,,$$

$$L_1^g(v) = \int_{X_r} \frac{1}{\sigma^g} q^g (\omega \cdot \nabla v) \, \mathrm{d}\omega \, \mathrm{d}x \,,$$

and

$$L_-^g(v) = \int_{\Gamma_-^r} f^g v (\omega \cdot n) \, \mathrm{d}\omega \, \mathrm{d}s \,,$$

such that from (3.22),

$$L^g(v) = L_1^g(v) + L_2^g(v) - L_-^g(v).$$

Since the forms L_1^g, L_2^g and L_-^g are linear, using the approximations (3.24) and (3.25),

$$L_1^g(\varphi_i y_n^m) = \sum_{l,k,j} A_r^1(i,n,m;j,l,k) q_{l,j}^{k,g}, \tag{3.43}$$

$$L_2^g(\varphi_i y_n^m) = \sum_{l,k,j} A_r^2(i,n,m;j,l,k) \frac{1}{\sigma^g} q_{l,j}^{k,g}, \tag{3.44}$$

$$L_-^g(\varphi_i y_n^m) = \sum_{F\in\partial D_r} \sum_{l,k,j} A_F^-(i,n,m;j,l,k) f_{l,j}^{k,g}. \tag{3.45}$$

The matrices A_r^1 and A_r^2 are defined at (3.32) and (3.37). In particular, since the polynomial basis $(\varphi_j)_{j=1,\cdots,J}$ is orthonormalized, A_r^1 is the identity matrix and $L_1^g(\varphi_i y_n^m) = q_{n,i}^{m,g}$. The matrix A_F^- is defined by,

$$A_F^-(i,n,m;j,l,k) = \int_F \varphi_i\varphi_j \int_{(\omega\cdot n)<0} y_n^m y_l^k (\omega\cdot n)\,\mathrm{d}\omega\,\mathrm{d}s\,. \tag{3.46}$$

3.5 Exact calculation of matrix coefficients

In Section 3.4, we presented the matrices obtained after discretization. Classically in the finite element literature [76, 36, 12, 2], the coefficients of these matrices are approximated by a quadrature $\int_{D_r} f(x)\,\mathrm{d}x \approx \sum_{i=1}^n w_i f(x_i)$, with w_i and x_i well-chosen weights and nodes.

In this work, the coefficients of the elementary matrices are calculated exactly. In this section, we describe the procedure used. The reader interested in more details can refer to [20].

Three kinds of integrals are encountered after discretization: integrals over the angular variable, integrals over a region and integrals over the (incoming or outgoing) boundary of a region. Let us describe the calculation of the first two, as the last type can be obtained from them with a few adjustments.

3.5.1 Integrals over the angular variable

The coefficients of matrices involving integrals on the angular variable are

$$\int_{\mathbf{S}^2} \omega_p\omega_q y_n^m y_l^k\,\mathrm{d}\omega\,, \quad \int_{\mathbf{S}^2} \omega_p y_n^m y_l^k\,\mathrm{d}\omega \quad \text{and} \quad \int_{\pm(\omega\cdot n)>0} y_n^m y_l^k (\omega\cdot n)\,\mathrm{d}\omega\,.$$

Let us recall that, the spherical harmonics y_n^m are defined in Section 2.3.1, ω_p are the components of $\omega\in\mathbf{S}^2$ in spherical coordinates, $\omega_1 = \sin\theta\cos\varphi$, $\omega_2 = \sin\theta\sin\varphi$ and $\omega_3 = \cos\theta$; and $\mathrm{d}\omega = 1/4\pi \sin\theta\,\mathrm{d}\theta\,\mathrm{d}\varphi$.

It can be shown that these integrals can be rewritten

$$\int_{\theta_0}^{\theta_1}\int_{\varphi_0}^{\varphi_1} f(\theta)g(\varphi)\,\mathrm{d}\theta\,\mathrm{d}\varphi = \int_{\theta_0}^{\theta_1} f(\theta)\,\mathrm{d}\theta \int_{\varphi_0}^{\varphi_1} g(\varphi)\,\mathrm{d}\varphi\,,$$

with f and g functions written as linear combinations of sin and cos functions,

$$f(\theta) = a_0 + \sum_i a_c^i \cos(i\theta) + \sum_j a_s^j \sin(j\theta).$$

It remains to determine θ_0, θ_1, φ_0 and φ_1.

For integrals on the entire sphere $\mathbf{S}^2$, $\theta_0 = 0$, $\theta_1 = \pi$, $\varphi_0 = 0$ and $\varphi_1 = 2\pi$. The analytical expression of these integrals and the `C++` code to compute them are presented in [19].

In 2D, the integrals on the incoming half-sphere ($\omega \cdot n < 0$), $\theta_0 = 0$, $\theta_1 = \pi$, $\varphi_0 = \varphi_n + \pi/2$ and $\varphi_1 = \varphi_n + 3\pi/2$, with φ_n the measure of the angle between the x-axis and the normal n. For the outgoing half-sphere, $\varphi_0 = \varphi_n - \pi/2$ and $\varphi_1 = \varphi_n + \pi/2$.

The 3D case is described in Chapter 4.

3.5.2 Integrals of a monomial on a region

Here we describe what happens in dimension 2, the 3D case being presented in Chapter 4. Let us recall that $(\varphi_j)_{j=1,\cdots,J}$ is the polynomial basis of $\mathbb{P}^k$. The external unit normal of an element D_r is denoted $n = (n^x, n^y)$ and that of a face $n_F = (n_F^x, n_F^y)$. The integrals to be calculated here are given by

$$\int_{D_r} \varphi_i\varphi_j\,\mathrm{d}x\,, \quad \int_{D_r} (\partial_p\varphi_i)\varphi_j\,\mathrm{d}x\,, \quad \text{and} \quad \int_{D_r} (\partial_p\varphi_i)(\partial_q\varphi_j)\,\mathrm{d}x\,.$$

In dimension 2, these integrals reduce to a sum of integrals over monomials $x^i y^j$.

Let us choose a vector field $\mathbf{F}$ such that $\operatorname{div}\mathbf{F} = x^i y^j$. Using the divergence theorem, we write

$$\int_{D_r} x^i y^j\,\mathrm{d}x\,\mathrm{d}y = \int_{D_r} \operatorname{div}\mathbf{F}\,\mathrm{d}x\,\mathrm{d}y = \int_{\partial D_r} \mathbf{F}\cdot n\,\mathrm{d}s = \sum_{F\in\partial D_r}\int_F \mathbf{F}\cdot n_F\,\mathrm{d}s\,.$$

Here we choose

$$\mathbf{F} = \left(\frac{1}{i+1}x^{i+1}y^j, 0\right),$$

and we have $\operatorname{div}\mathbf{F} = x^i y^j$. Hence,

$$\int_{D_r} x^i y^j\,\mathrm{d}x\,\mathrm{d}y = \frac{1}{i+1}\sum_{F\in\partial D_r}\int_F x^{i+1}y^j n_F^x\,\mathrm{d}s\,.$$

We have therefore moved from an integral on the element D_r to an integral on each of its faces F. Now we just need to proceed according to whether F is a line segment, a circular arc or a circle. In each of these cases, we know how to express explicitly the

limits of integration, the component n_F^x of the normal to F and ds [20].

3.6 Assembly-free linear system resolution

After the discretization described in Sections 3.3 and 3.4, we obtain local linear system by group and by region

$$A_r^g u^g = Q_r^g u, \tag{3.47}$$

with $u^g = \left(u_{l,j}^{k,g}\right)$ the discrete solution in group g and region D_r and $u = (u^g)_g$ the discrete multi-group solution since all energy groups are coupled by the scattering and fission operators.

Then, let us define the diagonal cross-section matrices

$$d_r^{0,g}(i,n,m;j,l,k) = \delta_{n,l}\delta_{m,k}\delta_{i,j}\left(\frac{1}{\sigma^g}\right),$$

$$d_r^{1,g}(i,n,m;j,l,k) = \delta_{n,l}\delta_{m,k}\delta_{i,j}\left(\sigma_a^{l,g}\right),$$

$$d_r^{2,g}(i,n,m;j,l,k) = \delta_{n,l}\delta_{m,k}\delta_{i,j}\left(\frac{\sigma_s^{l;g,g}}{\sigma^g}\right),$$

such that using the elementary matrices A_r^0 (3.30), A_r^1 (3.32), A_r^2 (3.37), A_r^+ (3.34), A_F^- (3.46) and the upwind flux defined at (3.16)-(3.17),

$$A_r^g u^g = \left(A_r^0 d_r^{0,g} + A_r^1 d_r^{1,g} - A_r^2 d_r^{2,g} + A_r^+\right)u^g + \sum_{F\in\partial D_r} A_F^- u_F^{\uparrow g} \tag{3.48}$$

We recall that on the incoming part of a region D_r the upwind flux u_F is either the boundary condition if one considers a face F at the boundary of D, or the flux in the adjacent region to D_r if F is an interface.

The right-hand side Q_r^g includes the scattering source, the fission source and possibly external sources. Using the matrices A_r^1 (3.32), A_r^2 (3.37), F_r^1 (3.4.4) and F_r^2 (3.42), Q_r^g is defined by

$$Q_r^g u = A_r^1(\widehat{s}^g) + A_r^2\left(\frac{1}{\sigma^g}\widehat{s}^g\right) + \left(F_r^1(\widehat{p}^g) + F_r^2\left(\frac{1}{\sigma^g}\widehat{p}^g\right)\right) + A_r^1(q^g) + A_r^2\left(\frac{1}{\sigma^g}q^g\right), \tag{3.49}$$

with $\widehat{s}^g$ and $\widehat{p}^g$ written using degrees of freedom $u_{l,j}^{k,g}$

$$\widehat{s}_{l,j}^{k,g} = \sum_{g'=1,g'\neq g}^{G} \sigma_s^{l;g,g'} u_{l,j}^{k,g'},$$

$$\widehat{p}_{l,j}^{k,g} = \delta_{l,0}\delta_{k,0}\sum_{\alpha}\chi_\alpha^g \sum_{g'=1}^{G} \nu\sigma_{f,\alpha}^{g'} u_{0,j}^{0,g'}.$$

It is possible to assemble the linear system (3.47) to obtain a global problem $A^g u^g =$

$Q^g u$. In the next section, taking advantage of the fact that the matrices A_r^0, A_r^1, A_r^2, A_r^+ and A_F^- do not depend on the cross-sections but only on the shape of the D_r element, we describe how to solve this problem without having to assemble the system.

3.6.1 The resolution algorithm

Calculating the right-hand side of (3.47) requires knowledge of the flux in all energy groups. This problem is solved by the power iteration algorithm 3. At each step of the algorithm, the right-hand side is computed using the flux u of the previous step. The algorithm is considered converged when the eigenvalue λ and the fission source $F^g u$ for all groups are converged. It consists of three nested loops in which we successively calculate the fission source, then the scattering source decomposed into down-, self- and up-scattering. As described in Section 3.4.3, the problem has been rewritten by sending the self-scattering in the left-hand member, to avoid inner iterations. The other two loops are described below.

Outer iterations At the end of each step $(n+1)$, the fission source $F^g u$ in all groups and the eigenvalue λ are computed using the flux u at the previous step (n) of the loop. The eigenvalue λ is updated using

$$\lambda^{(n+1)} = \lambda^{(n)} \frac{\left(w, Fu^{(n+1)}\right)}{\left(w, Fu^{(n)}\right)}, \tag{3.50}$$

with $(\cdot,\cdot)$ the standard dot product and the weight vector w usually chosen is $w = Fu^{(n)}$ or the vector of all ones $w = \vec{1}$. The loop stops when the convergence criteria are reached.

Thermal iterations The cross-sections $\sigma_s^{g,g'}$ of neutrons of energy g' arriving in group g after collision can be put into matrix form $\left(\sigma_s^{g,g'}\right)_{1\leqslant g\leqslant G; 1\leqslant g'\leqslant G}$. By convention, we put the incident groups g' by column and the destination group g by row. Since neutrons lose energy after a collision (they are said to thermalize), this matrix should be lower triangular. However, low-energy neutrons – known as thermal neutrons – can gain energy by colliding with a target nucleus. This phenomenon is called up-scattering.

Consequently, the matrix corresponding to the collision operator H^g can be decomposed into the sum of three matrices: (i) a self-scattering operator H_{ss}^g with the corresponding matrix being block-diagonal, (ii) an up-scattering operator H_{up}^g with the corresponding matrix being block upper-triangular, and (iii) a down-scattering operator H_{dw}^g with the corresponding matrix being block lower-triangular.

Neutrons are grouped into two packets according to their energy level: fast neutrons for $g \in [\![1, G_f]\!]$ and thermal neutrons for $g \in [\![G_f+1, G]\!]$, with G_f the number of fast neutron groups.

For fast neutrons, $H_{up}^g = 0$, they are therefore treated by simple descent, by substitution of the precedent group fluxes in the scattering source. This descent can be seen as a Gauss-Siedel method for the energy.

Thermal neutrons require additional processing, as they are coupled together by the H^g_{up} operator. Thermal iterations involve iterating on the up-scattering source to process the thermal neutrons.

Algorithm 3 : Power iteration algorithm.

Initialization of λ and (u^g) $\forall g \in [\![1, G]\!]$
while $F^g u$ *and* λ *not converged* **do** // Outers indexed by (n+1)
  Compute fission source $\mathcal{F}^{g,(n)} = \frac{1}{\lambda^{(n)}} F^g u^{(n)}$, $\forall g$
  for $g = 1$ **to** G_f **do** // Fast neutrons
    Compute source $\mathcal{Q}^g = H^g_{dw} u^{(n+1)} + \mathcal{F}^{g,(n)}$
    Solve $A^g u^g = \mathcal{Q}^g$
  while $H_{up} u$ *not converged* **do** // Thermals indexed by (t+1)
    Compute up-scattering source $\mathcal{H}^{g,(t)}_{up} = H^g_{up} u^{(n+1,t)}$
    for $g = G_f$ **to** G **do** // Slow neutrons
      Compute down-scattering source $\mathcal{H}^{g,(t+1)}_{dw} = H^g_{dw} u^{(n+1,t+1)}$
      Compute source $\mathcal{Q}^g = \mathcal{H}^{g,(t+1)}_{dw} + \mathcal{H}^{g,(t)}_{up} + \mathcal{F}^{g,(n)}$
      Solve $A^g u^g = \mathcal{Q}^g$
  Update $F^g u$ and λ

3.6.2 Mesh equivalence classes

Geometrically, a reactor core consists of a set of assemblies, similarly each assembly is made up of a set of fuel rods. One can therefore consider using these repetitive patterns to significantly reduce the memory footprint and the simulation time. Indeed, due to the choice of the basis functions $(\varphi_j)_j$, presented in Section 3.3, from two mesh elements identical by translation, the same elementary matrices are produced. More precisely, we refer to geometric elementary matrices, as they do not depend on the cross-sections in the region. These elements can be grouped by clusters of elements identical up to a translation, thus defining an equivalence class, Figure 3.2.

We define the equivalence class $[D_r]$ of an element $D_r \in D_h$ as the set of cells $D_s \in D_h$ such that D_s is the image of D_r by a translation, let us note $D_s \sim D_r$,

$$D_s \in [D_r] \Leftrightarrow D_s \sim D_r.$$

Two elements of the mesh D_h are equivalent to each other if and only if they belong to the same equivalence class. Any element of $[D_r]$ characterizes the class and may be used to represent it. In practice, the chosen representative is the cell with the smallest r number.

First, we need to determine the set of equivalence classes of the mesh, for the equivalence relation $\sim$, called the quotient set. The underlying idea is to work in the quotient set as on the mesh, but without distinguishing between equivalent elements. It is therefore sufficient to calculate the geometric elementary matrices of each class representative. Likewise, the matrix-vector product operator is written using the canonical

application $D_r \mapsto D_r^\star$, which associates to each element of D_h its class representative. The reuse of data already present inside the cache memory reduces the need to repeatedly fetch data from the main memory.

In the end, the solution of the linear system is performed without the need to assemble the matrix A resulting from the bilinear form $a(\cdot, \cdot)$ (3.8). It is sufficient to pass the matrix-vector product operator to the Krylov solver. The solvers implemented in `NYMO` based on this principle are BICGSTAB [132] and GMRES [118]. The preconditioners used are Jacobi and block-Jacobi, the blocks being the matrix per region.

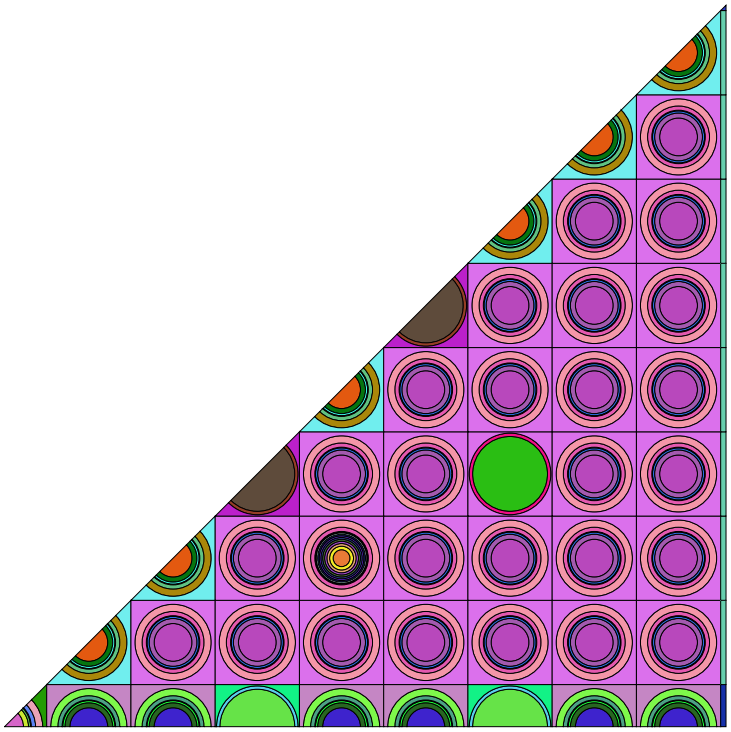


Figure 3.2: 312-cell mesh of an eighth PWR assembly. The elements of the same colors are equal up to a translation and thus belong to the same equivalence class. There are 49 classes, dividing the number of geometric matrices to be computed and stored by 6.

Chapter 4

The three dimensional spherical harmonics – discontinuous Galerkin method and application to core and lattice calculation

In Chapter 3, we presented a spherical harmonics – discontinuous Galerkin numerical scheme for the neutron transport problem and described its implementation in the `NYMO` solver. In this chapter, we extend this solver to extruded 3D geometries. To assess the accuracy of this numerical scheme, it has been used for reactor core and lattice calculations. These calculations point out that the proposed method is able to produce precise solutions while the developed solver can handle complex 3D core and assembly geometries.

A part of this chapter has been published with the reference [11] K. Assogba, L. Bourhrara, I. Zmijarevic, G. Allaire, and A. Galia. "Spherical Harmonics and Discontinuous Galerkin Finite Element Methods for the Three-Dimensional Neutron Transport Equation: Application to Core and Lattice Calculation". *Nuclear Science and Engineering* 197.8 (Aug. 2023), pp. 1584–1599.

Outline of the current chapter

4.1 Motivation

Due to the development of new nuclear reactor concepts, deterministic solutions of the Boltzmann transport equation on unstructured, non-conforming, curved and 3D meshes are in high demand in the industry. Moreover, some multi-physics effects can only be modeled in three dimensions. In fast reactors, for example, thermal dilatation induces an axially non-uniform radial displacement that affects the reactivity [53].

In Chapter 3, we described a combined spherical harmonics and discontinuous Galerkin numerical scheme for the multigroup neutron transport problem. The discontinuous Galerkin method provides the flexibility to handle non-conforming and unstructured meshes with curved faces. The ability to handle curved meshes allows for an accurate representation of annular sectors, which is typical of a fuel pellet, see Figure 4.1.

The resulting solver called `NYMO` has been implemented in the frame of CEA reactor physics platform `APOLLO3`® [122, 98]. The proposed numerical method can treat 2D unstructured and non-conforming meshes with line segments, circles and circular arcs [128]. Hence heterogeneous pin-cell geometries with concentric annular regions can be modeled without any approximation, see Figure 4.1. To leverage the common concern about memory usage, a matrix assembly-free method is used, thus high-order approximation can be considered. The code is written in `C++` and compute-intensive tasks like matrix-vector product operators are parallelized using OpenMP.

The purpose of this chapter is to present an extension of the `NYMO` capabilities to 3D prismatic meshes, Figure 4.2. The novelty here is the versatile use of this solver to efficiently perform 3D reactor core and fuel assembly calculations.

This chapter is organized as follows. In Section 4.2, we describe the calculations of the coefficients of the elementary matrices in 3D. We set up and describe the numerical experiments proposed in Section 4.3. The Sections 4.4, 4.5 and 4.6 and are dedicated to core and lattice calculation with various geometries and optical properties.

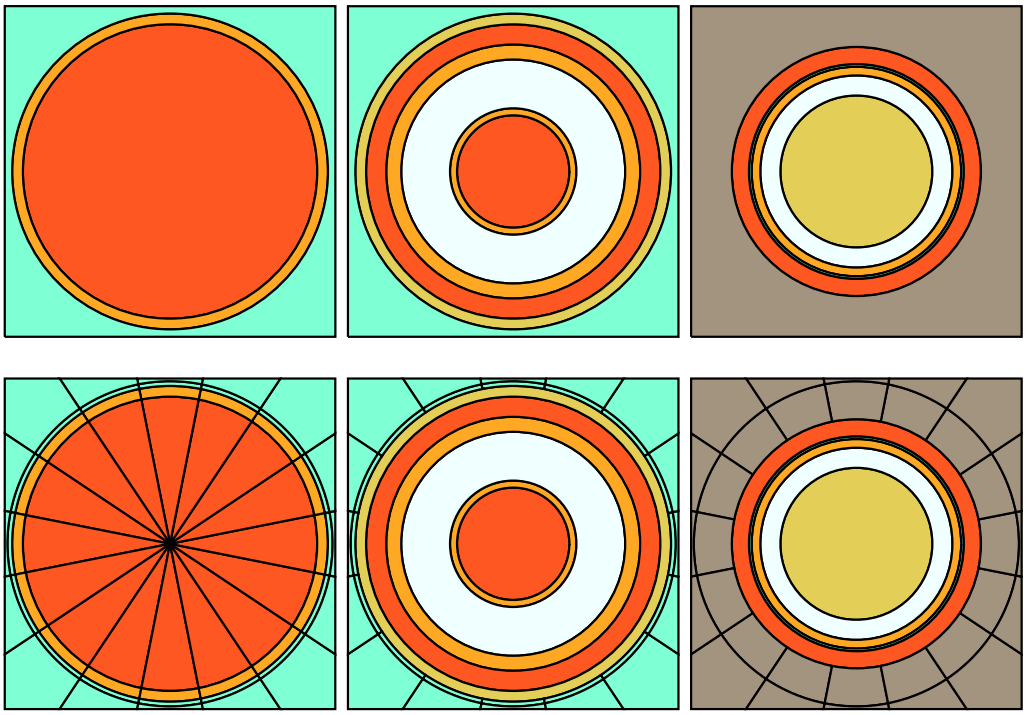


Figure 4.1: Typical Pressurized Water Reactor fuel-cells and meshes used by reactor physics engineers. All these meshes are supported by `NYMO`. In 2D `NYMO` supports non-conforming and unstructured meshes with curved faces.

4.2 Extension to three-dimensional extruded meshes

4.2.1 Extruded meshes

The spatial domain D is meshed into disjoint elements (also called regions) D_r. The mesh can be unstructured and non-conforming, that is with hanging nodes. Curved elements (circle or circular arc) are considered to be able to represent cladding surrounding the fuel rods, see Figure 4.1. Extruded meshes are sufficient to model the geometry of most types of nuclear reactors. Each 3D region D_r^{3D} is given as a base face D_r^{2D} extruded along the z direction,

$$D_r^{3D} = D_r^{2D} \times [z_0, z_1]. \tag{4.1}$$

Each face F^{3D} of ∂D_r^{3D} is either a horizontal face or a vertical face. The vertical faces are obtained by extruding a line segment, circle or a circular arc in the horizontal plane along the z axis. Horizontal faces treatment is identical to that of 2D regions.

$$F^{3D} = \begin{cases} F^{2D} \times [z_0, z_1] \text{ if } F^{3D} \text{ is vertical,} \\ D_r^{2D} \times \{z_0\} \text{ else.} \end{cases}$$

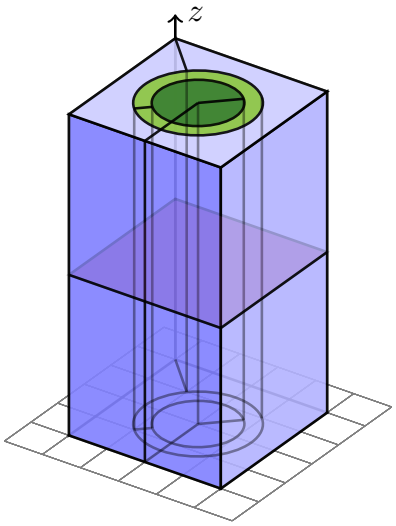


Figure 4.2: An example of a 3D fuel-cell mesh supported by `NYMO`. In 3D, `NYMO` supports extruded meshes also called prismatic meshes.

4.2.2 3D Elementary Matrices

Following the 3D mesh cells and faces classification presented in Subsection 4.2.1, one can notice that the 3D matrix resulting from the discretization can be recast into 2D matrix calculations. We refer to Chapter 3 for more details about 2D matrix calculation.

Three kinds of integrals are obtained after discretization:

- volume integrals $\int_{D_r} \bullet \, \mathrm{d}x$,
- integrals over the angular variable $\int_{\mathbf{S}^2} \bullet \, \mathrm{d}\omega$,
- and integrals over the boundary $\Gamma_{\pm}(D_r)$ of a region.

Given the equation (4.1), volume integrals are written as the product of two integrals that we know how to calculate. Integrals on the angular variable are calculated in the same way in 2 and 3 dimensions, and we refer to Chapter 3 for more details on 2D calculations. Integrals on the boundary of a region couple the first two types of integrals, $\int_{\partial D_r} \int_{\pm(\omega\cdot n)>0} \bullet (\omega \cdot n) \, \mathrm{d}\omega \, \mathrm{d}s$. Let us detail how to calculate this type of integral depending on the type of face, horizontal or vertical.

A region boundary can be rewritten

$$\Gamma_{\pm}(D_r) = \bigcup_{F \in \partial D_r} \Big\{ (x, \omega) \in F \times \mathbf{S}^2 \ / \ \pm \omega \cdot n_F > 0 \Big\} . \tag{4.2}$$

Let us denote n the unit outgoing normal. The map from Cartesian to spherical coordinates is $(n_x, n_y, n_z) \mapsto (\theta_n, \varphi_n)$ with $n_x = \sin\theta_n \cos\varphi_n$, $n_y = \sin\theta_n \sin\varphi_n$ and $n_z = \cos\theta_n$. The third integral type can then be rewritten using

$$\int_{\pm(\omega\cdot n)>0} y(\omega)(\omega \cdot n) \, \mathrm{d}\omega = \int_{\theta_0}^{\theta_1} f(\theta) \, \mathrm{d}\theta \int_{\varphi_0}^{\varphi_1} g(\varphi) \, \mathrm{d}\varphi \, .$$

It remains to determine θ_0, θ_1, φ_0 and φ_1.

Horizontal faces

If the face F is a bottom base of D_r, then $n(0,0,-1)$ and $\omega \cdot n = -\cos\theta$. On the incoming half-sphere $\mathbf{S}^2_-$, $\omega \cdot n < 0$ implies $-\cos(\theta) > 0$. Therefore $\theta \in [-\frac{\pi}{2}, \frac{\pi}{2}] \cap [0, \pi]$ and finally $\theta_0 = 0$ and $\theta_1 = \frac{\pi}{2}$. Symmetrically for the outgoing half-sphere $\mathbf{S}^2_+$, $\theta_0 = \frac{\pi}{2}$ and $\theta_1 = \pi$. In the two cases $\varphi_0 = 0$ and $\varphi_1 = 2\pi$. The same developments apply to upper horizontal faces.

Vertical faces

If the face F is a flat lateral face of D_r, using $\theta_n = \frac{\pi}{2}$ implies $n(\cos\varphi_n, \sin\varphi_n, 0)$ then $\omega \cdot n = \sin\theta\cos(\varphi - \varphi_n)$. Since $\theta \in [0, \pi]$, the sign of $\omega \cdot n$ only depends on the sign of $\cos(\varphi - \varphi_n)$. On the incoming half-sphere, $\omega \cdot n < 0$ implies $\varphi_0 = \varphi_n + \frac{\pi}{2}$ and $\varphi_1 = \varphi_n + 3\frac{\pi}{2}$. Symmetrical arguments apply to the outgoing half-sphere. In the two cases $\theta_0 = 0$ and $\theta_1 = \pi$.

In the case of curved lateral faces, the normal to the face is parametrized by a point M, and we denote $n = n(M)$ with $n_x = \cos(\varphi_n(M))$, $n_y = \sin(\varphi_n(M))$ and $n_z = 0$.

4.3 Numerical experiments

The numerical method described in Chapter 3 has been employed to solve two- and three-dimensional core and lattice problems. The first one is the Stepanek core [126] the second is the Takeda benchmark [127], the third is the 3D C5G7 core [89] and the last one is a typical pressurized water reactor assembly [137]. These problems provide different configurations, which offer Cartesian, hexagonal, unstructured and curved geometries with different optical properties, representative of fast breeder reactors and light water reactors.

The simulations performed are eigenvalue and power distribution calculations. Two error criteria determine the convergence, one on the eigenvalue, and the other on the fission source in L^2 norm. The calculation stops when for two successive iterations, the variations on these two values are both below 10^{-5}. The linear solver used is GMRES [118], using the assembly-free framework described in Section 3.6.

The eigenvalue errors are relative errors and in *pcm*[1] regarding the reference solution, in most cases the Monte-Carlo solution provided by the benchmark authors

$$e_r = \frac{\lambda_{\text{approx}} - \lambda_{\text{ref}}}{\lambda_{\text{ref}}}.$$

Pin power distributions presented in Sections 4.5 and 4.6 are normalized such that the total power is equal to the number of fuel pins. To compare the pin power solution

[1] Per cent mille, 1 pcm $= 10^{-5}$.

obtained to the reference distribution, four errors are calculated: the average (AVG), root mean square (RMS), mean relative (MRE) and maximum (MAX) pin power error in percent (%). Let us denote by e_n the relative error on the n-th fuel pin, and the number of fuel pins is denoted by N. The reference pin power and the reference average power are respectively denoted by p_n and p_{avg}.

$$\text{AVG} = \frac{1}{N}\sum_{n=1}^{N}|e_n|, \qquad \text{RMS} = \sqrt{\frac{1}{N}\sum_{n=1}^{N}|e_n^2|},$$

$$\text{MRE} = \frac{1}{N\cdot p_{avg}}\sum_{n=1}^{N}|e_n|\cdot p_n, \qquad \text{MAX} = \max_{1\leqslant n\leqslant N}|e_n|.$$

All calculations are performed on a standard workstation housing two 12-core Intel Xeon Silver 4214 CPU at 2.20GHz.

4.4 Application to core calculation

Five cores are simulated, the Stepanek core [126] and the four Takeda core models [127]. First, comparisons are made with the Monte-Carlo solution provided by [126] and [127].

In the second stage, we use the verification and validation data for APOLLO3® given in [104] to compare NYMO with other APOLLO3® solvers. Table 4.1 shows the solvers used in this comparison.

Table 4.1: Solvers used for core calculations.

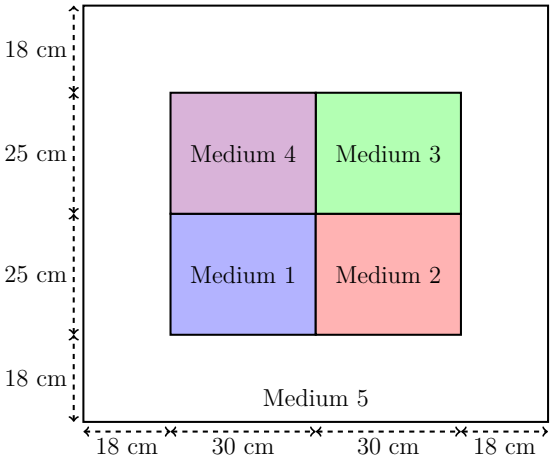
		IDT	Minaret	Minos	Pastis	NYMO
Discretization		S_N MOC	S_N DG	SP_N FEM	P_N FEM	P_N DG
Stepanek		✓	✓	✓	✓	✓
Takeda	M1	✓	✓	✓	✓	✓
	M2	✓	✓	✓	✓	✓
	M3	✓	✓	✓	✓	✓
	M4	-	✓	✓	✓	✓

4.4.1 Stepanek core

Figure 4.3 displays the geometry and cross-sections of the Stepanek Benchmark [126]. It is a small light-water reactor core comprising two fissile media and two absorbing media surrounded by light water. It is a monogroup problem with isotropic scattering and vacuum boundary conditions. The reference eigenvalue provided by TRIPOLI-4® [26] is $k_{\text{eff}} = 1.00888$ [104].

A uniform Cartesian mesh is used, with each cell measuring 1 cm per side. The discretization chosen is of order 2 in angle and linear $\mathbb{P}^1$ in space. The eigenvalue obtained

is $k_{\text{eff}} = 1.00885$, so the error with respect to the reference value is -3 pcm. It is of the same order as that obtained by the IDT, Minaret, Minos and Pastis deterministic solvers in [104], see Table 4.2. The calculation time is 4 seconds.



Medium	σ_t	σ_s	$\nu\sigma_f$	χ
1	0.6	0.53	0.079	1
2	0.46	0.2		
3	0.7	0.66	0.043	1
4	0.65	0.5		
5	0.9	0.89		

Figure 4.3: Geometry, mediums and cross-sections of the Stepanek benchmark.

Table 4.2: Eigenvalue error relative to TRIPOLI-4® for the Stepanek core.

	IDT (S_4)	Minaret (S_4)	Minos (SP_5)	Pastis (P_5)	NYMO (P_2)
error (pcm)	-3	-2	7	-3	-3

4.4.2 The Takeda benchmark suite

The Takeda benchmark suite [127] comprises four different reactor models with two or three distinct cases per model, this amounts to 10 cases in total. The reference Monte-Carlo solutions are taken from [127] and [104]. All the calculations are performed using linear polynomials $\mathbb{P}^1$ in space.

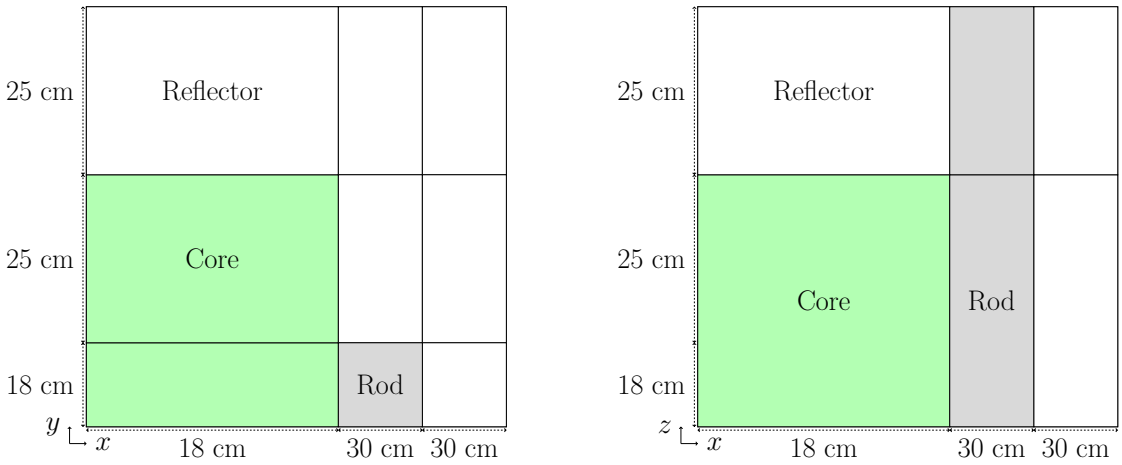
4.4.3 Takeda model 1: small Light Water Reactor core

The first model is a small Light Water Reactor (LWR), see Figure 4.4. The cross-sections are provided with two energy groups. Two cases are considered: in the first, the position of the control rod is empty (void), and in the second, the rod is inserted.

In the first case, the reference eigenvalue given in [127] is $k_{\text{eff}} = 0.9780$ and that given in [104] is $k_{\text{eff}} = 0.97729$. The eigenvalue obtained by NYMO is $k_{\text{eff}} = 0.978678$, so the relative error is 69 pcm with respect to [127] and 142 pcm with respect to [104]. The simulation time is 22 seconds, and there are 12 outer iterations.

In the second case, the reference eigenvalue given by both [127] and [104] is $k_{\text{eff}} = 0.96240$. The eigenvalue obtained by NYMO is $k_{\text{eff}} = 0.962647$, so the relative error is 25 pcm. The simulation time is 10 seconds, and there are 12 outer iterations.

The table shows the errors relative to TRIPOLI-4® as given in [104]. The larger error observed for case 1, where the control rod is replaced by an empty medium, suggests that NYMO has more difficulty dealing with vacuum. We can assume that this is due to the term $1/\sigma$ used in the variational formulation (3.7).



(a) Radial view (b) Axial view

Figure 4.4: Takeda model 1 is a light water reactor with Cartesian geometry.

Table 4.3: Eigenvalue error (in pcm) relative to TRIPOLI-4® [104] for the Takeda model 1 core.

	IDT	Minaret	Minos	Pastis	NYMO
Case 1	30 (S_4)	-16 (S_8)	233 (SP_5)	-189 (P_5)	142 (P_4)
Case 2	25 (S_6)	-10 (S_8)	164 (SP_5)	-35 (P_5)	25 (P_4)

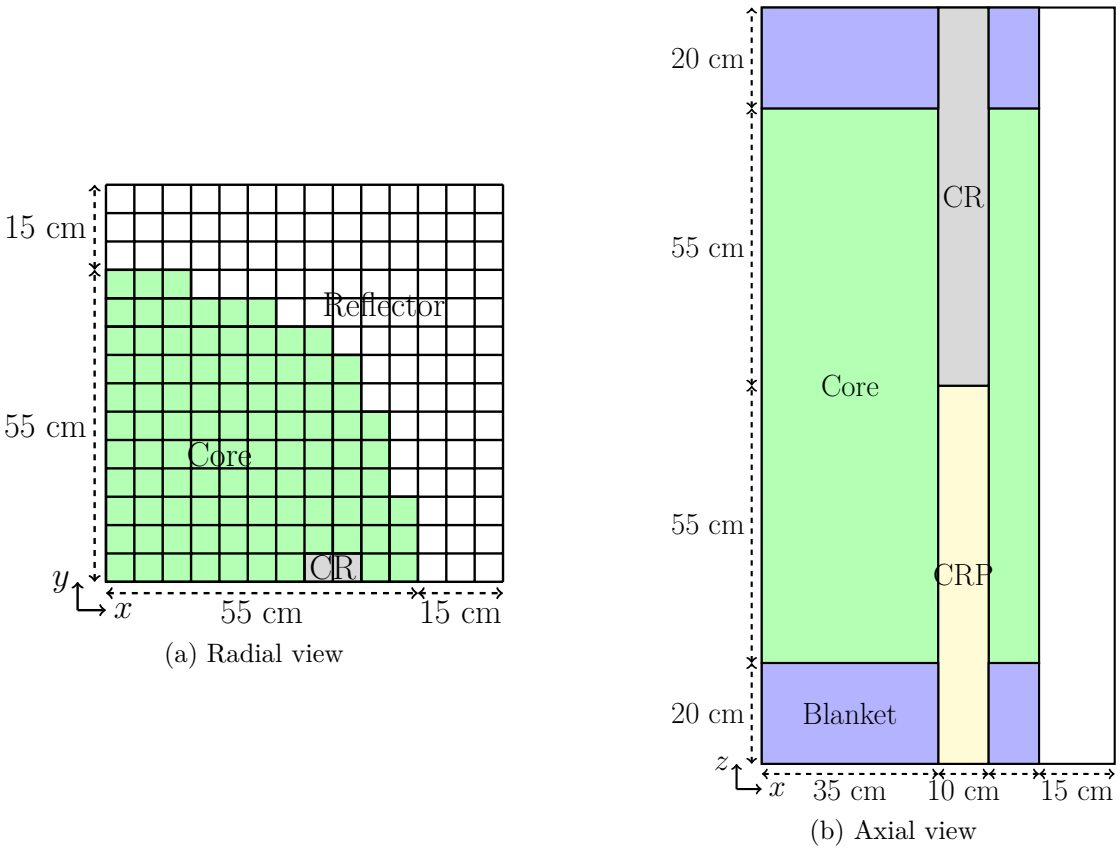
4.4.4 Takeda model 2: small Fast Breeder Reactor core

Model 2 is a breeder reactor with axial and radial blankets. The quarter core geometry is shown in Figure 4.5. The cross-sections are provided with four energy groups and two cases are considered.

In the first case, the control rod is removed and its position is filled with Sodium. The reference eigenvalue from [127] is $k_{\text{eff}} = 0.9732$ and the one given in [104] is $k_{\text{eff}} = 0.97368$. A simulation using NYMO with P_2 order in angle and linear polynomial $\mathbb{P}^1$ in space gives an eigenvalue of $k_{\text{eff}} = 0.973752$. The error is therefore 56 pcm with respect to [127] and 7 pcm with respect to TRIPOLI-4® [104]. The calculation takes 5 seconds and is completed in 16 outer iterations.

In the second case, the control rod is half-inserted. The eigenvalue from [127] is $k_{\text{eff}} = 0.9594$ and the TRIPOLI-4® eigenvalue from [104] is $k_{\text{eff}} = 0.95968$. A simulation with NYMO using P_2 and linear polynomial in space gives $k_{\text{eff}} = 0.959720$. The error is therefore 33 pcm with respect to [127] and 4 pcm with respect to TRIPOLI-4® [104]. The computation takes 11 seconds and is finished in 24 outer iterations.

Table 4.4 shows a comparison with other APOLLO3® solvers based on data published in [104]. The eigenvalue error obtained is in line with the errors of IDT, Minaret and Pastis. Computation times are well below those provided in [104], at 143 s for IDT, 774 s for Minaret and around 2000 s for Pastis.



(a) Radial view

(b) Axial view

Figure 4.5: Takeda model 2 is a fast breeder reactor. Thanks to the symmetries, we model a quarter of the radial geometry and half of the axial geometry when the control rod is fully inserted.

Table 4.4: Eigenvalue error (in pcm) relative to TRIPOLI-4® [104] for the Takeda model 2 core.

	IDT	Minaret	Minos	Pastis	NYMO
Case 1	-4 (S_4)	-11 (S_4)	-29 (SP_3)	-2 (P_3)	7 (P_2)
Case 2	-11 (S_4)	-20 (S_4)	-47 (SP_3)	-2 (P_3)	4 (P_2)

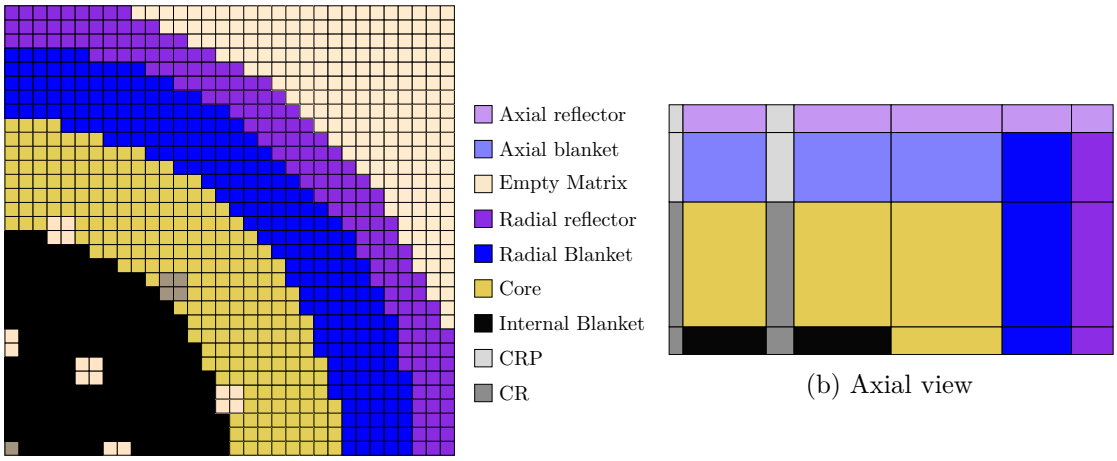
4.4.5 Takeda model 3: heterogeneous Fast Breeder Reactor

This model is a Fast Breeder Reactor core that has a reflector and internal blanket region, see Figure 4.6. There are three different cases depending on the material inserted in the control rod positions: rods, fuel or blanket. The cross-sections are provided with four

energy groups. The core has 1/4 radial and 1/2 axial symmetry. A uniform Cartesian mesh is used and the cell size $\Delta x = \Delta y = \Delta z = 5$ cm.

The reference values provided by [127] are shown in Table 4.5. The simulation is performed using NYMO with P_4 order in angle and linear polynomial $\mathbb{P}^1$ in space and the maximum error obtained is 38 pcm in less than one minute. A comparison with other APOLLO3® solvers based on [104] data is made in Section 4.4.7.

Finally, Figure 4.7 shows the asymptotic behavior of the solver for the k_{eff} according to the angular discretization. We note that a discretization of order 2 in angle is sufficient to obtain an error of less than 100 pcm.



(a) Radial view

(b) Axial view

Figure 4.6: Takeda model 3 is an axially heterogeneous fast breeder reactor. Thanks to the symmetries, a quarter of the radial geometry and half of the axial geometry are modeled.

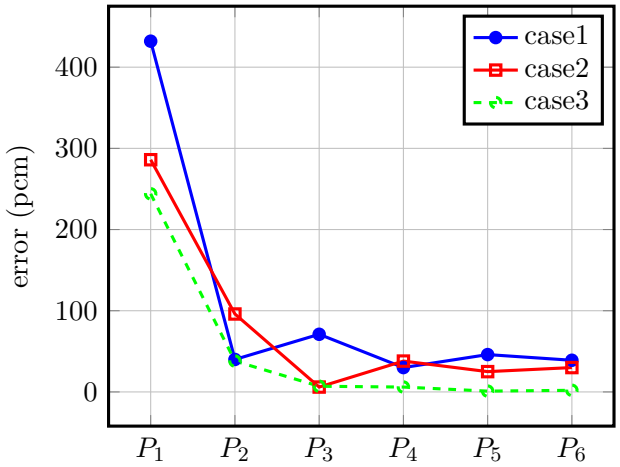
Table 4.5: Relative error (pcm) and time (s) obtained with P_4 and $\mathbb{P}^1$ for Takeda model 3. The reference eigenvalues are from [127].

	Reference	NYMO	error	time (s)
Case 1	0.97090	0.97060	-30	49
Case 2	1.00050	1.00089	38	49
Case 3	1.02140	1.02147	6	51

4.4.6 Takeda model 4: hexagonal Fast Breeder Reactor

Model 4 is a Fast Breeder Reactor core and has prismatic hexagonal geometry, see Figure 4.8. Four group cross-sections are provided and there are three cases with different control rod positions: withdrawn (case 1), half-inserted (case 2) and fully inserted rods (case 3).

The presence of the control rods creates a discontinuity in the cross-sections, and the mesh has been refined to take this into account. All hexagons are meshed as equilateral



	Case 1	Case 2	Case 3
P_1	432	286	244
P_2	40	96	38
P_3	71	6	7
P_4	30	38	6
P_5	46	25	1
P_6	39	30	2

Figure 4.7: Convergence of the eigenvalue according to the angular discretization for Takeda model 3 core. The errors (in pcm) are given in absolute values.

triangles, so the 3D mesh is made up of triangular prisms. There are three different refinements of hexagon mesh, as described in Table 4.6. Refinement 1 uses six triangles, refinement 2 uses 24, and refinement 3 uses 54. The axial mesh is uniform, with a step size of $\Delta z = 5$ cm. All the calculations are performed using linear polynomials $\mathbb{P}^1$ in space.

Table 4.7 shows the eigenvalues obtained and their comparison with the reference eigenvalues of [127]. For case 1, refinement 2 is used, for cases 2 and 3, refinement 3 is used. The maximum error obtained is 22 pcm. A comparison with other APOLLO3® solvers based on [104] data is made in Section 4.4.7.

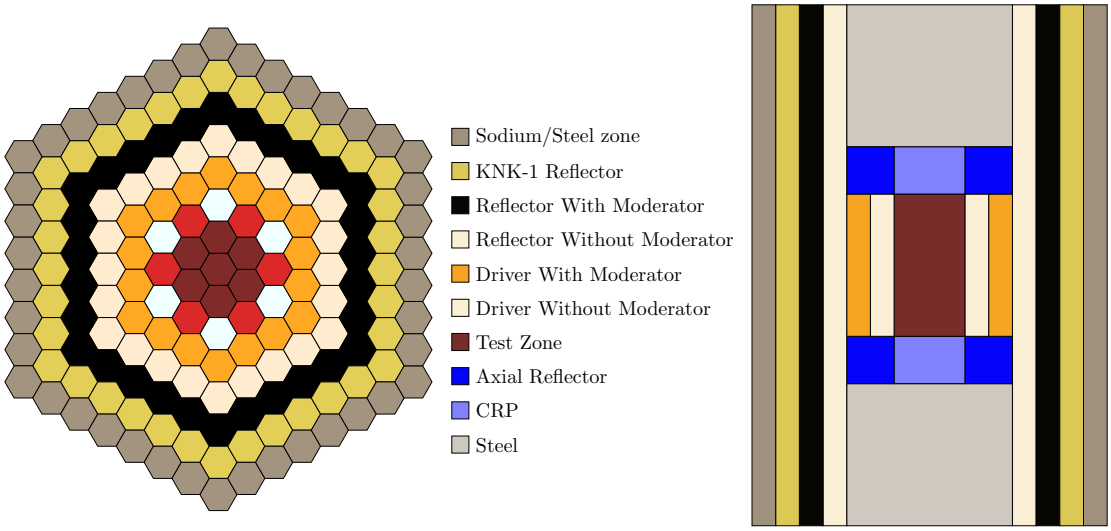
For the convergence study, refinement 2 is used in all cases, each side of a hexagon is cut into two, and then the hexagons are meshed by triangles. Figure 4.9 show the asymptotic behavior of the solver for the k_{eff} according to the angular discretization.

4.4.7 Comparison of Apollo3 core solvers

Based on the data provided in [104], we compare NYMO with the IDT, Minaret, Minos and Pastis solvers for core calculation. The reference solutions used are provided by TRIPOLI-4® and we calculate NYMO's errors according to them.

Table 4.8 and 4.9 are extension of [104, Table 2], comparing different APOLLO3® solvers for core calculation.

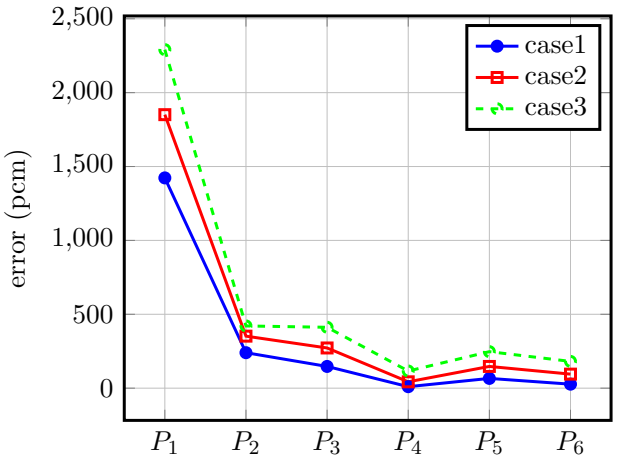
Except for model 1 case 1, the relative errors compared to reference eigenvalue are below 50 pcm, with in particular four pcm obtained for case 2 of model 2. For model 1 case 1, it is possible to refine the mesh to reduce the error, as was done for Minaret, for example. Models 1, 2 and 3 eigenvalue calculations are performed in less than a minute. These calculation times are lower than those given in [104] for the other solvers.



(a) Radial view

(b) Axial view

Figure 4.8: Takeda model 4 is a small fast breeder reactor with hexagonal geometry.



	Case 1	Case 2	Case 3
P_1	1423	1851	2291
P_2	240	352	421
P_3	147	272	412
P_4	10	43	116
P_5	66	147	247
P_6	27	95	181

Figure 4.9: Convergence of the eigenvalue according to the angular discretization order. The mesh of type 2 is used, each side of a hexagon is cut into 2, then the hexagons are meshed into 24 triangles. The errors (in pcm) are given in absolute values.

4.5 Application to heterogeneous core calculation: C5G7 core

The 3-D C5G7 MOx fuel assembly benchmark [89] is a problem designed to evaluate the ability of deterministic transport codes to handle reactor problems without spatial homogenization. The objectives are to calculate the effective multiplication factor and the normalized fission rate distributions (pin power) in three axial slices of the core. It should be noted that in this benchmark, the cladding surrounding the fuel rod is smeared with the fuel pellet.

Table 4.6: Refinement of hexagons into triangles.

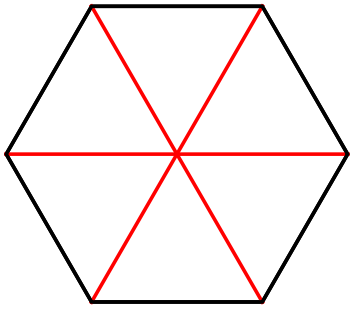
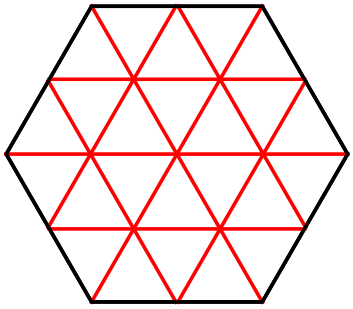
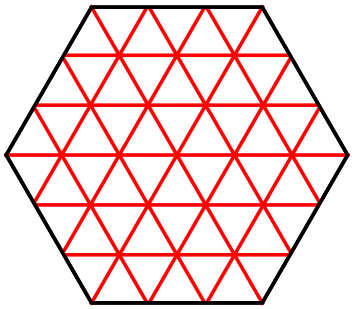
Refine Edge	Nbr. of triangle	Mesh
1	6	
2	24	
3	54	

Table 4.7: Relative error (pcm) and time (s) obtained with P_4 and $\mathbb{P}^1$ for Takeda model 4. The reference eigenvalues are from [127]. In all 3 cases, the hexagons are subdivided into triangles. For case 1, all hexagons are divided into 24 triangles and for case 2 and 3 in 54 triangles.

	Reference	NYMO	error	cpu time
Case 1	1.09510	1.09521	10	474
Case 2	0.98330	0.98352	22	2636
Case 3	0.87990	0.87994	4	2349

4.5.1 Benchmark description and space-angular discretization

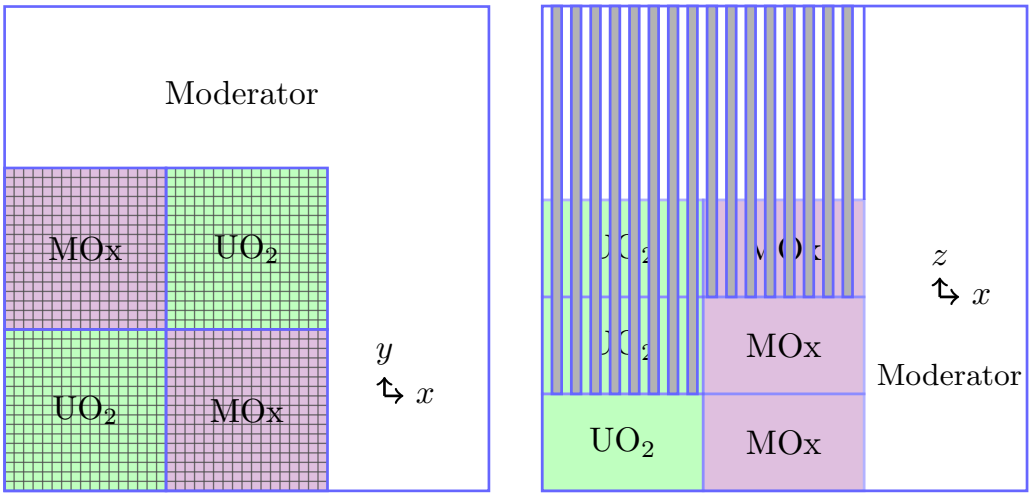
Figure 4.10 shows the geometry of this Pressurized Water Reactor (PWR) core, measuring 128.52 cm on each side. The core is made up of four by four fuel assemblies, modeled in $1/4$ radial symmetry and surrounded by a water reflector. Each assembly consists of a

Table 4.8: Order of angular discretization of solvers used for core calculations.

	Stepanek	Takeda 1		Takeda 2		Takeda 3			Takeda 4		
Case		C1	C2	C1	C2	C1	C2	C3	C1	C2	C3
IDT	S_4	S_4	S_6	S_4	S_4	S_4	S_4	S_4	-	-	-
Minaret	S_4	S_8	S_8	S_4	S_4	S_4	S_4	S_4	S_4	S_{16}	S_8
Minos	SP_5	SP_5	SP_5	SP_3	SP_3	SP_5	SP_5	SP_5	SP_7	SP_9	SP_9
Pastis	P_5	P_5	P_5	P_3	P_3	P_5	P_5	P_5	P_5	P_5	P_5
NYMO	P_2	P_4	P_4	P_2	P_2	P_2	P_2	P_2	P_4	P_4	P_4

Table 4.9: Extension of [104, Table 2], synthesis of APOLLO3® core flux solvers benchmarking. k_{eff} errors in pcm are relative to TRIPOLI-4®.

	Stepanek	Takeda 1		Takeda 2		Takeda 3			Takeda 4		
Case		C1	C2	C1	C2	C1	C2	C3	C1	C2	C3
IDT	-3	30	25	-4	-11	-71	-20	-22	-	-	-
Minaret	-2	-16	-10	-11	-20	-85	-11	-13	34	20	-36
Minos	7	233	164	-29	-47	-130	-104	-4	-394	-269	7
Pastis	-3	-189	-35	-2	-2	-12	-10	-3	-1	-23	1
NYMO	-3	142	25	7	4	-48	12	-21	34	29	31



(a) Radial view (b) Axial view

Figure 4.10: Radial and axial sections of the Rodded B case of C5G7 core. Thanks to the symmetries, a quarter of the radial geometry and half of the axial geometry are modeled.

17×17 lattice of square pin cells with the cell side equal to 1.26 cm. Fuel pins, control rods, guide tubes, and fission chambers are of circular shape with a 0.54 cm radius. The reflection boundary condition in the $z_{\min}$ plane with inserted control rods makes the problem non-physical, which is explicitly stated in the problem description. Three problems: unrodded, rodded A and rodded B, are considered and correspond to various

levels of control rod insertion. The seven-group cross-sections with isotropic scattering for each material and detailed description are provided in [89].

For all calculations, the radial mesh used is described in Figure 4.11. This radial mesh is the unique one applied to all calculations and it contains 15028 cells. Axial mesh has been varied starting from the coarsest one (denoted as Z_1) that has only one mesh interval within each fuel slice and reflector, and refining it progressively by subdividing these slices into two (Z_2), four (Z_4) and eight (Z_8) intervals. For polynomial discretization space, $\mathbb{P}^k$ refers to piecewise polynomial functions space of total degree at most k. That is $\mathbb{P}^0$ the space of piecewise constant polynomial, $\mathbb{P}^1$ refers to the space of piecewise linear polynomial and $\mathbb{P}^2$ refers to the space of piecewise quadratic polynomial.

We performed a series of calculations by varying the $\mathrm{P_N}$ order from 1 to 6, the polynomial basis used is constant, linear or quadratic. The axial mesh refinement is 1, 2, 4 or 8. This involves 72 simulations for each test case. As before, each case was run on a computer that houses two sockets with 12-core. The computing time is between 10 seconds (P_1, $\mathbb{P}^0$ without axial refinement) and 20 hours (P_6, $\mathbb{P}^2$, Z_8) for the most refined case.

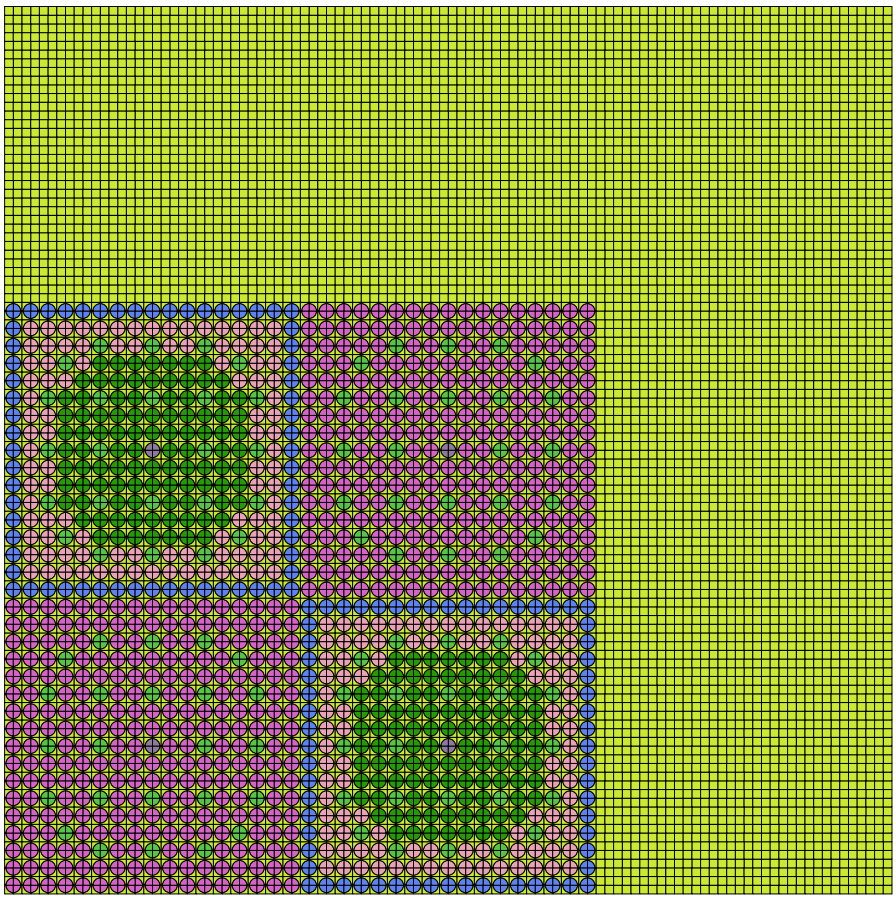


Figure 4.11: C5G7 quarter core mesh used for calculations.

4.5.2 Results of calculation and discussion

The first results show that, for all P_N orders, the error on the $\mathbb{P}^0$ polynomial space is high, so we discard these configurations from further analysis. Moreover, the axial refinement does not seem to significantly improve the results. The differences between Z_4 and Z_8 refinements being very small (between 0 and 4 pcm) we excluded Z_8 from the rest of the analysis. Thus, we may conclude that the flux does not vary much axially and that it is not necessary to refine the mesh in this direction. This assumption is partially confirmed by [125], who studied the influence of the control rods by analyzing the scalar fluxes along control rods. The authors observe that the axial scalar fluxes decrease in the vicinity of the reflector, but vary little in each slice in MOx and UOx outer assemblies. Moreover, the flux variation is more pronounced in the UOx inner assembly when the control rods are inserted (Rodded A and B). In our case, for Rodded A, the best results are still obtained without refinement, while Rodded B gives better results by subdividing each z-slice into two.

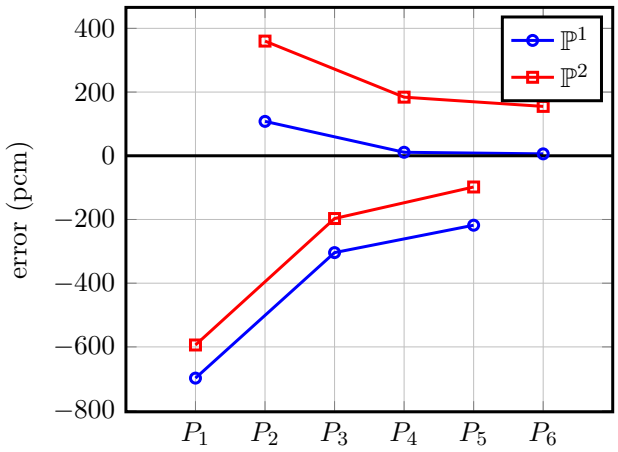
Table 4.10 present for each case, the k_{eff} obtained with one low-order and one high-order discretization. Unrodded and Rodded A cases are without axial refinement. For Rodded B case, each axial slice is divided by two. For the Unrodded and Rodded A cases, the best results are respectively 13 pcm and -29 pcm and obtained with P_5 $\mathbb{P}^2$, while for Rodded B the best solution has 6 pcm error and obtained with P_6 $\mathbb{P}^1$. Considering only the configurations running in less than 3 min (180s) the best result is -48 pcm (P_1 $\mathbb{P}^2$ Z_1) for Unrodded, -80 pcm for Rodded A and -162 pcm Rodded B (the two with P_2 $\mathbb{P}^1$ Z_1).

Table 4.10: Eigenvalue error compared to Monte Carlo reference solutions provided by MCNP for the C5G7 core.

		k_{eff}	error (pcm)	time (s)
Unrodded	MCNP	1.14308	± 6	-
	P_1 $\mathbb{P}^2$	1.14253	-48	157
	P_5 $\mathbb{P}^2$	1.14324	13	5191
Rodded A	MCNP	1.12806	± 6	-
	P_2 $\mathbb{P}^1$	1.12715	-80	142
	P_5 $\mathbb{P}^2$	1.12773	-29	5389
Rodded B	MCNP	1.07777	± 6	-
	P_2 $\mathbb{P}^1$	1.07894	108	322
	P_6 $\mathbb{P}^1$	1.07784	6	6084

Figure 4.12 shows the variation of eigenvalue relative error according to P_N order for each of the polynomial spaces $\mathbb{P}^1$ and $\mathbb{P}^2$. The results are presented with Z_2 refinement. We make several observations here. First, for all polynomial spaces, separating the even (green lines) and odd (red lines) P_N orders, we observe a monotone convergence. In the first two cases however, the even P_N orders stagnate or increase slightly. This behavior is for the moment unexplained. Second, for even (resp. odd) P_N the error obtained is smaller in polynomial space $\mathbb{P}^1$ (resp. $\mathbb{P}^2$).

Finally, we note that the even (resp. odd) P_N converge to the reference solution by overestimating (resp. underestimating) it. This oscillatory approach to the asymptotic regime, but monotonous if one takes separately odd and even orders, has already been observed in previous works on the 2D solutions, in [20] on reactivity and in [130] on the flux shape. In the latter, a different finite element approximation is used, which may suggest that this behavior is due to the properties of spherical harmonic approximation itself, but for the moment it remains unexplained. All these observations deserve further investigation.



	$\mathbb{P}^1$	$\mathbb{P}^2$
P_2	108	360
P_4	11	184
P_6	6	155
P_1	-698	-594
P_3	-304	-197
P_5	-218	-98

Figure 4.12: $k-$eigenvalue relative error in pcm according to the angular and spatial discretization for the 3D-C5G7 Rodded B case. The odd and even angular orders are connected separately.

We next proceed to the pin power calculations with the configurations that provide the best multiplication factor for each case. Table 4.11 shows the maximum (MAX), the average (AVG), root mean square (RMS) and mean relative (MRE) errors on the pin power distribution. The calculations are performed with P_5 $\mathbb{P}^2$ Z_1 for the first two cases and P_6 $\mathbb{P}^1$ Z_2 for Rodded B. For the first two cases, the calculations exhibit AVG, RMS and MRE errors of less than 0.4 % compared to the MCNP solution. In the last case, the AVG, RMS and MRE errors are below 0.8 %. It can be assumed that the error is higher here because of the strong discontinuities introduced by control rod insertion. Overall, these results comply well with the reference values.

Table 4.11: Pin power distribution errors for C5G7 benchmark problem.

Benchmark case		Unrodded	Rodded A	Rodded B
Pin power error (%)	AVG	0.225	0.301	0.743
	RMS	0.280	0.398	0.835
	MRE	0.171	0.249	0.764
	MAX	0.818	1.100	1.708
Assembly power error (%)	UOx Inner	0.014	-0.178	-0.980
	MOx	-0.135	0.041	0.476
	UOx Outer	0.365	0.662	0.866

4.6 Application to lattice calculation: PWR UOx Fuel Assembly

This section is devoted to the eigenvalue and pin power distribution calculation of a 2D PWR UO_2 Fuel Assembly described in [137, §III. 1]. It is a typical 17×17 pin assembly motif containing UO_2 and gadolinium fuel rods and guide tubes. This study aims to present a realistic test case with 281 energy groups taking into account anisotropic scattering. The results obtained are compared with deterministic solvers, here TDT [121, 124, 56], the APOLLO3® characteristics method (MOC) and IDT [96, 95, 15] the method of short characteristics.

4.6.1 Case description and discretization

Figure 4.13 shows a schematic view of the UO_2 fuel assembly geometry. Due to symmetries, it is sufficient to represent one-eighth of the structure. Each pin cell is modeled without homogenization, the thickness of cladding being 0.064 cm. In this study, the original UO_2 properties are replaced with 4.2% enriched fuel with isotopic composition given in Table 4.12. The order of anisotropy is 3, and a self-shielding calculation is performed first to obtain 281-groups cross-sections.

The mesh used for this experiment is displayed in Figure 4.14. The fuel pin is subdivided into six concentric annuli while the gadolinium bearing fuel is subdivided into eleven, accounting for the spatially dependent self-shielding effect and preparing the model for depletion calculation. For NYMO, P_5 angular approximation and constant $\mathbb{P}^0$ spatial discretization yield 16 million degrees of freedom. The step-constant MOC (TDT) is used with 30 azimuthal angles and 12 polar angles. The IDT solver is used with a classical S_8 level symmetric quadrature and linear characteristic option.

Table 4.12: UOX fuel composition.

Isotope	Atomic number density (#/barn/cm)
U234	4.6498E-07
U235	9.7229E-04
U236	2.3051E-07
U238	2.1874E-02
O16	4.5693E-02

4.6.2 Results and discussions

Table 4.13 presents eigenvalue, computation time and overall pin power distribution error when comparing NYMO to TDT and IDT. A k_{eff} of 0.999568 has been obtained, 121 pcm above the TDT solution. The 400 pcm spread compared to IDT is due to the absence of cell sectors in IDT calculations. To conclude the analysis, the pin power distributions are compared. The pin power solutions (Figure 4.15) are normalized such that the total

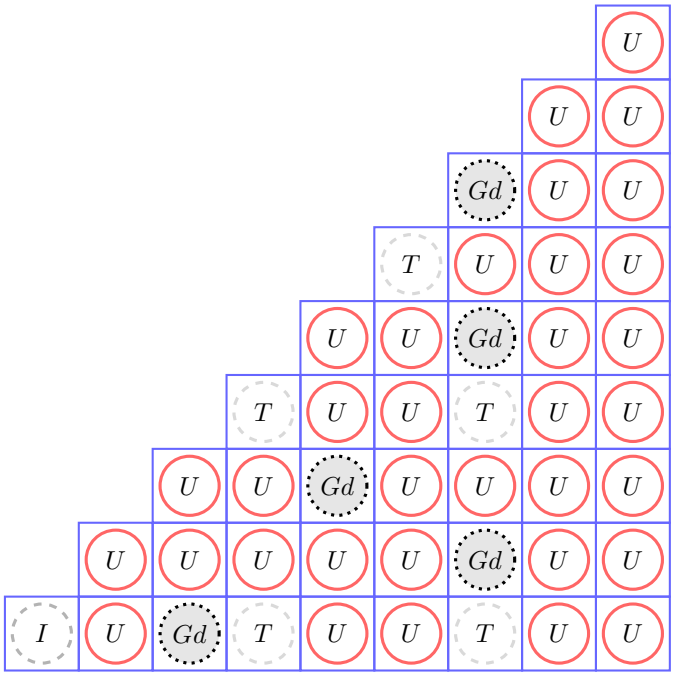


Figure 4.13: PWR UO_2 fuel assembly 1/8th geometry layout, with UO2 fuel rod (U), gadolinium pins (Gd), RCC guide thimble (T) and instrumentation thimble (I).

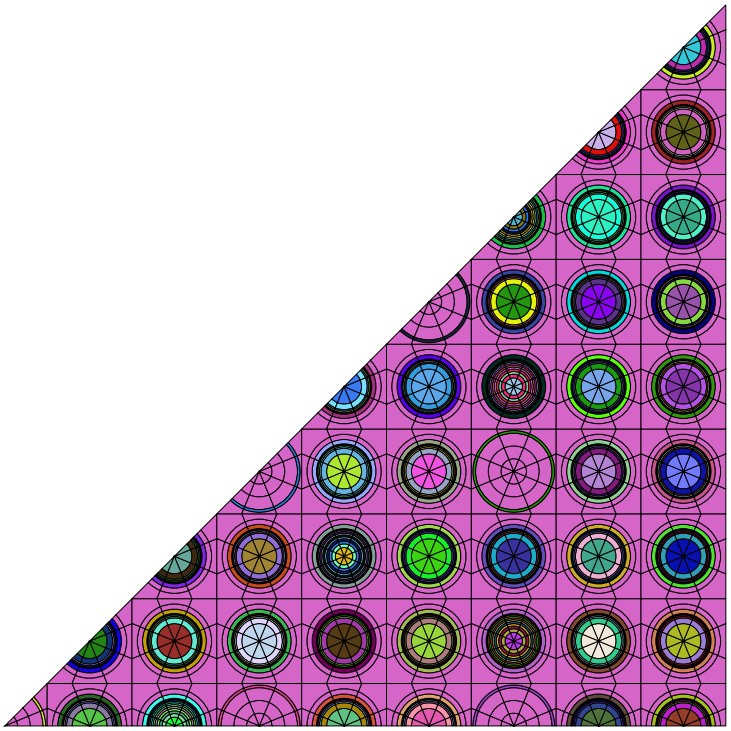


Figure 4.14: Mesh used for PWR UO_2 Fuel Assembly Calculation.

power is equal to the number of fuel pins (264). As shown in Figure 4.16 and 4.17, the

NYMO normalized pin power are all within 1 % of the TDT results and are less than 1.47 % regarding IDT.

The nature of the two approximations being very different, this result is considered to be very satisfactory.

Table 4.13: NYMO eigenvalue and power map compared to TDT and IDT for the PWR UO_2 assembly.

	NYMO	TDT	IDT
k_{eff}	0.999568	0.998360	0.995493
Δk_{eff} (pcm)	-	121	407
Max (%)	-	0.794	1.468
AVG (%)	-	0.314	0.461
RMS (%)	-	0.384	0.607
MRE (%)	-	0.295	0.381
time (s)	142	176	-

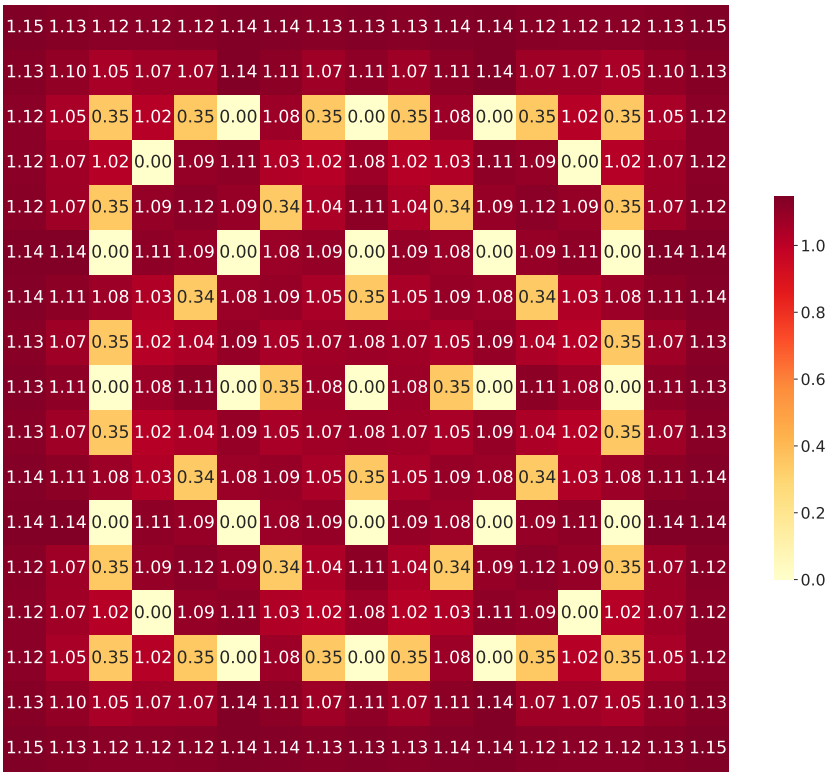


Figure 4.15: NYMO normalized power distribution for the PWR UO_2 fuel assembly.

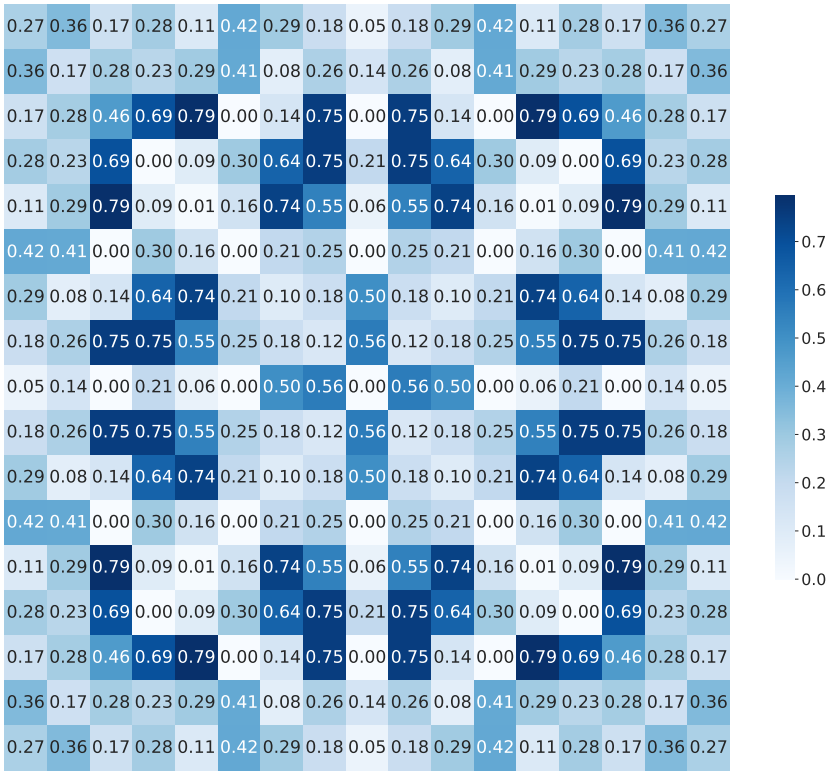


Figure 4.16: NYMO power distribution absolute error (in %) related to TDT.

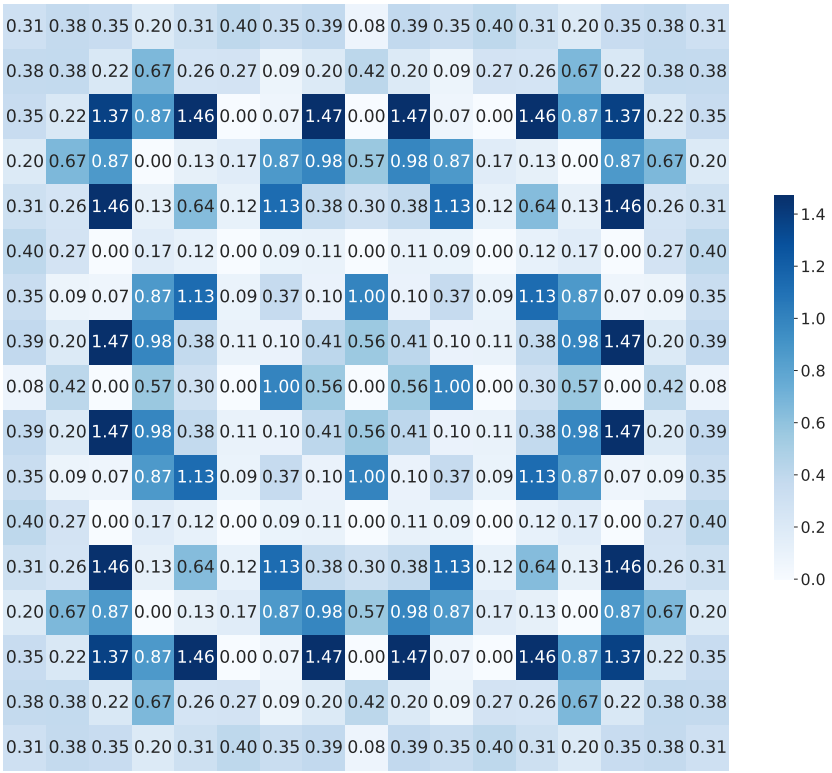


Figure 4.17: NYMO power distribution absolute error (in %) related to IDT.

4.7 Conclusion

The implementation of the three-dimensional extension of the previously developed spherical harmonics – discontinuous Galerkin method has been described. Its capabilities are illustrated on the examples of the well-known Takeda and C5G7 benchmarks. The discretization method can handle the mesh elements of different shapes, in practical applications all geometries describing the fuel elements without any simplification or homogenization. The presented results show that the method is able to attain an accuracy better than 100 pcm in reactivity for a computation time of less than 10 minutes on a desktop machine. The investigated cases show an error smaller than 30 pcm and in some of these of order of one pcm.

Moreover, tests has been conducted on a realistic PWR assembly in 281 energy groups with anisotropic scattering, and the obtained results agreed well with other, characteristic-based transport solvers.

The convergence behavior of odd and even order expansions is different and needs further theoretical analysis. Nevertheless, the method seems to be consistent and stable showing competitive computational times, and it opens the way to its application to high-fidelity simulations of a whole core. This approach needs an implementation of a distributed memory parallel algorithm.

Chapter 5

Analysis of the combined spherical harmonics and discontinuous Galerkin discretization

In [20], a numerical scheme based on a combined spherical harmonics and discontinuous Galerkin finite element method for the resolution of the Boltzmann transport equation is proposed. One of its features is that a streamline weight is added to the test function to obtain the variational formulation. In this chapter, we prove the convergence and provide error estimates of this numerical scheme. To this end, the original variational formulation is restated in a broken functional space. The use of broken functional spaces enables to build a conforming approximation, that is the finite element space is a subspace of the broken functional space. The setting of a conforming approximation simplifies the numerical analysis, in particular the error estimates, for which a Céa's type lemma and standard interpolation estimates are sufficient for our analysis. For our numerical scheme, based on $\mathbb{P}^k$ discontinuous Galerkin finite elements (in space) on a mesh of size h and a spherical harmonics approximation of order N (in the angular variable), the convergence rate is of order $\mathcal{O}\Big(N^{-t} + h^k\Big)$ for a smooth solution which admits partial derivatives of order $k+1$ and t with respect to the spatial and angular variables, respectively. For $k = 0$ (piecewise constant finite elements) we also obtain a convergence result of order $\mathcal{O}\Big(N^{-t} + h^{1/2}\Big)$. Numerical experiments in 1, 2 and 3 dimensions are provided, showing a better convergence behavior for the L^2-norm, typically of one more order, $\mathcal{O}\Big(N^{-t} + h^{k+1}\Big)$.

The preprint version of this work is referenced as [8] K. Assogba, G. Allaire, and L. Bourhrara. *Analysis of a Combined Spherical Harmonics and Discontinuous Galerkin Discretization for the Boltzmann Transport Equation.* (Hal-04196435). Aug. 2023.

Outline of the current chapter

5.1 Introduction

We are interested in the numerical resolution of the linear transport equation. This equation describes the streaming and collisions of neutral particles, for example neutrons, in matter. In the phase space $X = D \times \mathbf{S}^2$, where D is the space domain and $\mathbf{S}^2$ is the 3D unit sphere, the angular flux of particles u, subjected to sources q (collisions, possibly fission and external sources), at any point $(x, \omega) \in X$, is given as the solution of the linear transport equation

$$\begin{aligned} \omega \cdot \nabla u(x,\omega) + \sigma u(x,\omega) &= q(x,\omega) \quad \text{in } X, \\ u(x,\omega) &= f(x,\omega) \quad \text{on } \Gamma_-. \end{aligned}$$

The function f is a given incoming flux on the incoming boundary Γ_-. Here, the scattering is not explicitly modeled and is assumed to be contained in the external sources q. One popular numerical method to solve this equation consists in removing the angular dependence by using a quadrature rule to approximate the angular flux [32]. A different approach is to approximate the angular flux by a truncated series of real spherical harmonics [88, Appendix A], it is the spherical harmonics method or $\mathrm{P_N}$ method [44].

The present work relies on the numerical scheme proposed by [20] for the resolution of the transport equation. This scheme combines the spherical harmonics method with the discontinuous Galerkin finite element method in space. It allows the treatment of 2D unstructured, non-conforming and curved meshes and 3D prismatic meshes. Our previous studies carried out in [10, 11] show that the obtained accuracy is similar to that of high-fidelity solvers such as the method of characteristics [56, 96]. The purpose of this paper is to study the convergence and provide error estimates of this combined spherical harmonics – discontinuous Galerkin approximation.

Discontinuous Galerkin (DG) methods were introduced by Reed and Hill [112] to solve the neutron transport equation without imposing continuity at the interface between two cells. These schemes avoid spurious numerical oscillations when the solution of the transport problem is not smooth enough. In addition, DG methods provide the flexibility to handle unstructured and non-conforming meshes, that is meshes with hanging nodes. Shortly afterwards, Lesaint and Raviart [86, 85] proposed a first estimate of the numerical error of the DG method: for a triangular mesh of size h, an approximation by polynomials of degree k yields convergence of order $\mathcal{O}\left(h^k\right)$ in the L^2 norm. This estimate was further improved in [77, 75], where the error is proved to be actually $\mathcal{O}\left(h^{k+\frac{1}{2}}\right)$. Finally [116] establish a converges rate of $\mathcal{O}\left(h^{k+1}\right)$ on semi-uniform triangulations [116, §3]. However all these authors treated only the linear advection problem, i.e. the ω direction is fixed.

For a more comprehensive overview of DG methods one refers to [49] for Friedrichs' systems, [5] for elliptic problems, [24] for hyperbolic problems and [46] for a big picture.

Concerning the angular dependence of the problem, [44] proposed to approximate the angular flux $u(x,\omega)$ by a truncated series on a real spherical harmonics basis $\{y_n^m;\ 0 \leqslant n \leqslant N,\ -n \leqslant m \leqslant n\}$

$$u(x,\omega) \approx P_N u(x,\omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} u_n^m(x) y_n^m(\omega).$$

Like for Fourier series expansion, the error resulting from the approximation of u by the truncated expansion $P_N u$ is related to the smoothness of u regarding the angular variable. A result for Lipschitz functions is given by [63, Th. 1], and a general result for t-times continuously differentiable functions is provided by [110, Th. 3.3] and summarized in [12, Th. 7.5.10]. There exists a positive constant c such that, for any $v \in C^{t,\gamma}(\mathbf{S}^2)$ a γ-Hölder function t-times continuously differentiable, with $0 < \gamma \leqslant 1$, and for every integer $N \geqslant 1$,

$$\|v - P_N v\|_{L^2(\mathbf{S}^2)} \leqslant \frac{c}{N^{t+\gamma}} \|v\|_{C^{t,\gamma}(\mathbf{S}^2)}.$$

More recently, an alternative result has been proposed by [61, Lemma 3.3] for functions in Sobolev space $H^t(\mathbf{S}^2)$, see [12, §7.5.5]. There exists a positive constant c such that, for any $v \in H^t(\mathbf{S}^2)$, the truncation error satisfies

$$\|v - P_N v\|_{L^2(\mathbf{S}^2)} \leqslant \frac{c}{N^t} \|v\|_{H^t(\mathbf{S}^2)}.$$

In both cases, the constant c is independent of N. A result for the eigenvalue neutron

transport problem in 1D with isotropic scattering is given by [43]: for a layered medium, solving exactly with respect to the space variable, the error on the eigenvalue due to the truncation of the spherical harmonics expansion is found to be $O(\frac{1}{N^2})$.

On the analysis of the coupled angle-space problem, we refer to [107, 78, 6], where the authors use the discrete ordinates method and limit themselves to estimate the scalar flux error. In [91, 113], a scaled least-squares finite element method for the neutron transport problem is proposed to well capture the diffusion limit. The $\mathrm{P_N}$ method is used for the angular discretization and the discretization error is found to be $\mathcal{O}\left(\frac{1}{N} + h^k\right)$ for an approximation by polynomials of degree k. More recent works [136, 61], in the context of radiative transfer, treat the angular flux but do not use the discontinuous Galerkin discretization.

In this chapter we prove that the combined spherical harmonics – discontinuous Galerkin method converges as the discretization parameters tend to zero and we provide an error estimate on the angular flux. For this purpose, in Section 5.2, the linear transport problem and its variational formulation (as proposed in [22]) are recalled. Section 5.3 is devoted to a new variational formulation, so-called broken, where no continuity between subdomains (or mesh cells) is imposed. This broken problem is shown to be equivalent to the original problem. It is then discretized in Section 5.4 by $\mathbb{P}^k$ discontinuous Galerkin finite elements (in space) and a spherical harmonics approximation of order N (in the angular variable), and we prove that the discretized problem admits a unique solution. Finally, in Sections 5.5 and 5.6, we prove that the numerical scheme is convergent and give its convergence rate (see our main results, Theorems 5.5.2 and 5.5.5). Lastly, Section 5.7 provides numerical experiments which support and improve our error estimates.

5.2 Analysis of the original problem

In this section we introduce the linear Boltzmann transport problem used in the context of neutron transport [115, 42, 69]. Then the variational formulation, introduced in [22] and further studied in [20], is recalled. This weak formulation is the starting point of the broken weak formulation presented in section 5.3.

5.2.1 The neutron transport equation

Let us denote by x the spatial variable, which belongs to an open bounded set D in $\mathbb{R}^d$ ($d = 1$, 2 or 3), not necessarily convex with piecewise C^1 boundary ∂D. The direction variable ω belongs to $\mathbf{S}^2$, the unit sphere of $\mathbb{R}^3$. We consider the neutron transport equation in the phase space $X = D \times \mathbf{S}^2$, that is we look for the angular flux $u(x, \omega)$ which is the solution of:

$$\omega \cdot \nabla u + \sigma u = q \quad \text{in } X, \tag{5.1a}$$

$$u = f \quad \text{on } \Gamma_-. \tag{5.1b}$$

The outward unit normal of D is denoted by n, and the incoming Γ_- and outgoing Γ_+ boundaries of X are defined as:

$$\Gamma_\pm = \Gamma_\pm(D) = \{(x,\omega) \in \partial D \times \mathbf{S}^2;\ \pm\omega \cdot n(x) > 0\}.$$

The data of the problem are a given incoming flux f, an external source q and the total macroscopic cross section σ, assumed to be strictly positive. These functions are assumed to be given as:

$$f(x,\omega) \in L^2_- = L^2(\Gamma_-, |\omega \cdot n|\,\mathrm{d}s\,\mathrm{d}\omega), \tag{5.2a}$$
$$q(x,\omega) \in L^2(X), \tag{5.2b}$$
$$\sigma(x) \in L^\infty(D) \quad \text{and} \quad 0 < \sigma_0 \leqslant \sigma(x) \leqslant \sigma_\infty. \tag{5.2c}$$

We also define the space $L^2_+ = L^2(\Gamma_+, (\omega \cdot n)\,\mathrm{d}s\,\mathrm{d}\omega)$. We introduce the spaces V and W defined by:

$$\begin{aligned} V = V(D) &= \left\{v \in L^2(X);\ \omega \cdot \nabla v \in L^2(X)\right\}, \\ W = W(D) &= \left\{v \in V;\ v|_{\Gamma_+} \in L^2_+\right\}, \end{aligned} \tag{5.3}$$

with $v|_{\Gamma_+}$ denoting the restriction of v to Γ_+.

Let us denote $(u,v)_{L^2(X)}$ the standard scalar product in $L^2(X)$ and $(u,v)_{L^2_+} = \int_{\Gamma_+} uv(\omega \cdot n)\,\mathrm{d}s\,\mathrm{d}\omega$ the weighted scalar product of L^2_+, where $\mathrm{d}s$ is the surface measure on ∂D. The space W is then equipped with the scalar product $(u,v)_W$ defined by

$$(u,v)_W = (u,v)_{L^2(X)} + (\omega \cdot \nabla u, \omega \cdot \nabla v)_{L^2(X)} + (u,v)_{L^2_+}.$$

Equipped with the scalar product $(u,v)_W$, W is a Hilbert space. Let us denote by $\|\cdot\|_W$ the associated norm,

$$\|v\|^2_W = \|v\|^2_{L^2(X)} + \|\omega \cdot \nabla v\|^2_{L^2(X)} + \|v\|^2_{L^2_+}. \tag{5.4}$$

Remark 5.2.1. *The space W is required since the trace $v|_{\Gamma_+}$ on Γ_+ of functions in V do not satisfy in general $\int_{\Gamma_+} |v|^2(\omega \cdot n)\,\mathrm{d}s\,\mathrm{d}\omega < \infty$ (the same applies to the traces on Γ_-) [42, p. 219]. Thanks to [31, Proposition 1] it is known that the trace application $\gamma_- : W \to L^2_-$, such that $\gamma_-(u) = u|_{\Gamma_-}$, is continuous. As a consequence, definition (5.3) of the space W is equivalent to $W = \{v \in V;\ v|_{\Gamma_-} \in L^2_-\}$.*

Finally recall the Green formula for functions $(u,v) \in W \times W$ [42, p. 225]:

$$\int_X ((\omega \cdot \nabla u)v + u(\omega \cdot \nabla v))\,\mathrm{d}x\,\mathrm{d}\omega = \int_\Gamma uv(\omega \cdot n)\,\mathrm{d}s\,\mathrm{d}\omega, \tag{5.5}$$

where $\Gamma = \partial D \times \mathbf{S}^2$. The above Green formula will play an important role in what follows. In the sequel, the volume measure $\mathrm{d}x\,\mathrm{d}\omega$ and surface measure $\mathrm{d}s\,\mathrm{d}\omega$ will sometimes be omitted in order to lighten the writing.

To conclude this section, let us recall a classical result of existence and uniqueness [42].

Lemma 5.2.2 ([42]). *Under assumptions (5.2) the transport problem (5.1) admits a unique solution $u \in W$.*

Remark 5.2.3. *We adopt the assumptions of [30, 31] for the existence and uniqueness of solutions to transport problem (5.1), where the convexity of D is not required. However, in Theorem 5.5.2, more regularity is required for the solutions, and this is usually obtained by making the assumption that D is convex, on top of smoothness of the data.*

5.2.2 Original variational formulation

Let $v \in W$ be a test function. Since it is assumed that σ is stritly positive, (5.1a) can be multiplied by $(v + \frac{1}{\sigma}\omega \cdot \nabla v)$ and integrated over the phase space $X = D \times \mathbf{S}^2$,

$$\int_X \left(\frac{1}{\sigma}(\omega \cdot \nabla u)(\omega \cdot \nabla v) + \sigma uv\right) + \int_X \left(u(\omega \cdot \nabla v) + (\omega \cdot \nabla u)v\right) = \int_X q\left(v + \frac{1}{\sigma}(\omega \cdot \nabla v)\right).$$

After using Green's formula (5.5) and boundary condition (5.1b), the resulting variational problem is written:

$$\text{find } u \in W \text{ such that } a(u,v) = L(v) \quad \forall v \in W, \tag{5.6}$$

with

$$a(u,v) = \int_X \left(\frac{1}{\sigma}(\omega \cdot \nabla u)(\omega \cdot \nabla v) + \sigma uv\right) + \int_{\Gamma_+} uv(\omega \cdot n), \tag{5.7}$$

$$L(v) = \int_X q\left(v + \frac{1}{\sigma}(\omega \cdot \nabla v)\right) - \int_{\Gamma_-} fv(\omega \cdot n). \tag{5.8}$$

Proposition 5.2.4 ([22]). *The variational formulation (5.6) admits a unique solution $u \in W$. Furthermore, (5.6) is equivalent to the original transport problem (5.1), that is, if one admits a solution $u \in W$, u is also solution of the other.*

The proof of Proposition 5.2.4 can be found in [22]. It is based on the Lax-Milgram theorem, since the bilinear form a and the linear form L are continuous in W and on the other hand the form a is coercive.

5.3 A broken formulation

The goal of this section is to restate the original variational problem (5.6) on a broken functional space. The new formulation so obtained is called broken weak formulation. The main result is Theorem 5.3.9, stating that the broken problem is well-posed and its solution coincides with the solution to the original problem. In addition, we prove a weaker notion of continuity of $\tilde{a}(u,v)$ by choosing different norms for u and v, see Proposition 5.3.10. The use of a broken functional space enables to build a conforming approximation in section 5.4.

5.3.1 Functional setting

Let us introduce D_h a partition or mesh of the domain D into disjoint regions or mesh elements D_r

$$\overline{D} = \bigcup_{D_r \in D_h} \overline{D}_r. \tag{5.9}$$

Here, h denotes the meshsize, and is defined as the maximum of the diameter h_r of the mesh elements. It should be noted that the mesh D_h in what follows is not necessarily conformal in the sense of classical finite element [2]. We denote by $\mathcal{F}_h^i$ the set of interior faces (or interfaces), that is $F \in \mathcal{F}_h^i$ if there exist two distinct regions D_{r_1} and D_{r_2} such that $F = \partial D_{r_1} \cap \partial D_{r_2}$. Let $\mathcal{F}_h^b$ be the set of boundary faces, i.e $F \in \mathcal{F}_h^b$ if there exists a region D_r such that $F = \partial D_r \cap \partial D$ and $\mathcal{F}_h = \mathcal{F}_h^i \cup \mathcal{F}_h^b$, is the set of all faces. We assume that all faces $F \in \mathcal{F}_h$ are $(d-1)$-dimensional with non-vanishing Lebesgue measure. We define the local phase space $X_r = D_r \times \mathbf{S}^2$, with incoming and outgoing boundaries $\Gamma_\pm^r = \Gamma_\pm(D_r) = \{(x,\omega) \in \partial D_r \times \mathbf{S}^2,\ \pm\omega \cdot n_r(x) > 0\}$, where n_r is the outward unit normal of D_r.

With each element $D_r \in D_h$ we associate the space W_r of functions defined on X_r

$$W_r = W(D_r) = \left\{ v \in L^2(X_r);\ \omega \cdot \nabla v \in L^2(X_r),\ v|_{\Gamma_+^r} \in L^2_{r,+} = L^2(\Gamma_+^r, (\omega \cdot n_r)\,\mathrm{d}s\,\mathrm{d}\omega) \right\}.$$

We then introduce the space $\widetilde{W}$ as the collection of independent spaces W_r over the regions D_r

$$\widetilde{W} = \left\{ v \in L^2(X) \text{ such that } \forall D_r \in D_h,\ v|_{D_r} \in W_r \right\}, \tag{5.10}$$

with $v|_{D_r}$ denoting the restriction of v to D_r. The natural norm on $\widetilde{W}$ (making it a Hilbert space) is

$$\|v\|_{\widetilde{W}}^2 = \sum_r \left(\|v\|_{L^2(X_r)}^2 + \|\omega \cdot \nabla v\|_{L^2(X_r)}^2 + \|v\|_{L^2_{r,+}}^2 \right). \tag{5.11}$$

Remark 5.3.1. *Following Remark 5.2.1, the trace operators $\gamma_\pm^r : W_r \to L^2(\Gamma_\pm^r, |\omega \cdot n_r|\mathrm{d}s\,\mathrm{d}\omega)$ are continuous. However, for two regions D_{r_1} and D_{r_2} sharing the same face F, by denoting $v_r \in W_r$ the restriction of $v \in \widetilde{W}$ to D_r, in full generality $v_{r_1}|_F \neq v_{r_2}|_F$. In other words, the functions of $\widetilde{W}$ are not necessarily continuous through interfaces F.*

Definition 5.3.1. *To each internal face $F \in \mathcal{F}_h^i$ we associate a triple (D_{r_1}, D_{r_2}, n_F), where D_{r_1} and D_{r_2} are the two regions located on either side of the face F and n_F is the unit normal vector to the face F oriented from D_{r_1} to D_{r_2}, where by convention the labels are chosen such that $r_1 < r_2$. The vector n_F is called the face normal vector, which is outgoing from the region D_{r_1} and incoming into the region D_{r_2}. The region D_{r_1} is called the first region of face F and D_{r_2} is called the second region of face F. Furthermore, D_{r_1} and D_{r_2} are said to be adjacent to each other by the face F. Note that the normal vector n_F can be non-constant when the face F is not flat.*

Since functions $v \in \widetilde{W}$ have two traces on an inner face $F \in \mathcal{F}_h^i$, we define their mean

and jump over F:

$$\{\!\{v\}\!\}_F(x,\omega) = \frac{1}{2}\left(v|_{D_{r_1}}(x,\omega) + v|_{D_{r_2}}(x,\omega)\right), \tag{5.12}$$

$$[\![v]\!]_F(x,\omega) = v|_{D_{r_1}}(x,\omega) - v|_{D_{r_2}}(x,\omega), \tag{5.13}$$

where D_{r_1} and D_{r_2} are respectively the first and the second region of the interface F in the sense of Definition 5.3.1. To lighten the notation we omit both the subscript F and variabes (x,ω), and simply write $\{\!\{v\}\!\}$ and $[\![v]\!]$. It is worth noticing that

$$[\![uv]\!] = \{\!\{u\}\!\}[\![v]\!] + [\![u]\!]\{\!\{v\}\!\}. \tag{5.14}$$

Green's formula (5.5) is not valid in general in the full domain D for $\widetilde{W}$-functions, but it is in each region D_r. Therefore we write a variant of (5.5) which is valid for $\widetilde{W}$-functions and will be of great use in what follows:

Lemma 5.3.2. *(Broken Green's formula) For any $(u,v) \in \widetilde{W} \times \widetilde{W}$ we have:*

$$\sum_r \int_{X_r} ((\omega\cdot\nabla u)v + u(\omega\cdot\nabla v))\,\mathrm{d}x\,\mathrm{d}\omega = \int_\Gamma uv(\omega\cdot n)\,\mathrm{d}s\,\mathrm{d}\omega + \sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![uv]\!](\omega\cdot n_F)\,\mathrm{d}s\,\mathrm{d}\omega\,. \tag{5.15}$$

Proof. Let $(u,v) \in \widetilde{W} \times \widetilde{W}$ and use Green's formula (5.5) for each phase space X_r.

$$\begin{aligned}\sum_r \int_{X_r} ((\omega\cdot\nabla u)v + u(\omega\cdot\nabla v))\,\mathrm{d}x\,\mathrm{d}\omega &= \sum_r \int_{\Gamma^r} uv(\omega\cdot n_r)\,\mathrm{d}s\,\mathrm{d}\omega \\ &= \sum_{F\in\mathcal{F}_h^b}\int_{F\times\mathbf{S}^2} uv(\omega\cdot n_F)\,\mathrm{d}s\,\mathrm{d}\omega \\ &+ \sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![uv]\!](\omega\cdot n_F)\,\mathrm{d}s\,\mathrm{d}\omega\,,\end{aligned}$$

where $\Gamma^r = \partial D_r \times \mathbf{S}^2$. Formula (5.15) is obtained by recognizing that

$$\sum_{F\in\mathcal{F}_h^b}\int_{F\times\mathbf{S}^2} uv(\omega\cdot n_F)\,\mathrm{d}s\,\mathrm{d}\omega = \int_\Gamma uv(\omega\cdot n)\,\mathrm{d}s\,\mathrm{d}\omega\,.$$

□

The next lemma gives a sufficient condition for a discontinuous function in $\widetilde{W}$ to belong to W.

Lemma 5.3.3. *A function $u \in \widetilde{W}$ belongs to W if it satisfies*

$$\forall F \in \mathcal{F}_h^i \quad (\omega\cdot n_F)[\![u]\!]_F(x,\omega) = 0 \quad \text{for a.e. } (x,\omega) \in F\times\mathbf{S}^2. \tag{5.16}$$

Proof. This prof uses the ideas developed in [46, §1.2.5] for the broken gradient. We start by defining a broken advection operator acting on the space $\widetilde{W}$, $A_h : \widetilde{W} \to L^2(D\times\mathbf{S}^2)$

is defined by:

$$\text{for all } D_r \in D_h, \quad (A_h v)|_{D_r} := \omega \cdot \nabla(v|_{D_r}).$$

Let $v \in \widetilde{W}$ and $\varphi \in C_c^\infty(D \times \mathbf{S}^2)$, integrating by part element-wise we observe that

$$\begin{aligned}\int_{D\times\mathbf{S}^2} v(\omega\cdot\nabla\varphi) &= \sum_{D_r}\int_{D_r\times\mathbf{S}^2} v(\omega\cdot\nabla\varphi)\\ &= -\sum_{D_r}\int_{D_r\times\mathbf{S}^2} \omega\cdot\nabla(v|_{D_r})\varphi + \sum_{D_r}\int_{\partial D_r\times\mathbf{S}^2} v\varphi(\omega\cdot n)\\ &= -\int_{D\times\mathbf{S}^2}(A_h v)\varphi + \sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![v\varphi]\!](\omega\cdot n_F) + \sum_{F\in\mathcal{F}_h^b}\int_{F\times\mathbf{S}^2} v\varphi(\omega\cdot n)\end{aligned}$$

Since φ is continuous over interfaces and vanishes on ∂D, we obtain

$$\int_{D\times\mathbf{S}^2} v(\omega\cdot\nabla\varphi) = -\int_{D\times\mathbf{S}^2}(A_h v)\varphi + \sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![v]\!]\varphi(\omega\cdot n_F).$$

If for all interfaces $[\![v]\!]_F(\omega\cdot n_F) = 0$, we obtain

$$\int_{D\times\mathbf{S}^2} v(\omega\cdot\nabla\varphi) = -\int_{D\times\mathbf{S}^2}(A_h v)\varphi \quad \forall\varphi \in C_c^\infty(D\times\mathbf{S}^2).$$

This means that the weak advective derivative of v, denoted $\omega\cdot\nabla v$ exists and is equal to $A_h v$ in $L^2(D\times\mathbf{S}^2)$. Noting that $\Gamma_+ \subset \bigcup_{D_r\in D_h}\Gamma_+^r$, we deduce $v|_{\Gamma_+} \in L^2(\Gamma_+, |\omega\cdot n|\,\mathrm{d}s\,\mathrm{d}\omega)$ and conclude that $v \in W$. □

Remark that equality (5.16) says nothing about the jump of a function $u \in \widetilde{W}$ in the directions ω which are tangent to the face F. One may wonder if there is a reciprocal to Lemma 5.3.3, that is, under what conditions a function $u \in W$ belongs to $\widetilde{W}$. The only obstacle is that such a function u should have traces in $L^2(\Gamma_+^r, (\omega\cdot n_r)\,\mathrm{d}s\,\mathrm{d}\omega)$, for any r, which is not obvious. Of course, as soon as u is bounded, it is true.

5.3.2 Local variational formulation

For each region D_r let us define the set of all its faces $\mathcal{F}_r$, its boundary faces $\mathcal{F}_r^b$ and its internal faces $\mathcal{F}_r^i$ by

$$\begin{aligned}\mathcal{F}_r &= \{F \in \mathcal{F}_h;\ F \subset \partial D_r\},\\ \mathcal{F}_r^b &= \{F \in \mathcal{F}_h^b;\ F \subset \partial D_r\},\\ \mathcal{F}_r^i &= \{F \in \mathcal{F}_h^i;\ F \subset \partial D_r\}.\end{aligned}$$

The subset $\mathcal{F}_r^b$ is empty if the region D_r does not intersect the boundary of the domain ∂D. The subset $\mathcal{F}_r^i$ is empty when there are no inner faces, in other words in the case where the mesh is reduced to a single region. Finally, note that $\partial D_r = (\cup_{F\in\mathcal{F}_r} F) = (\cup_{F\in\mathcal{F}_r^b} F) \cup (\cup_{F\in\mathcal{F}_r^i} F)$. For each face $F \in \mathcal{F}_r$, define the subsets $\Gamma_-^r(F)$ and $\Gamma_+^r(F)$ of

Γ^r_- and Γ^r_+, respectively by:

$$\Gamma^r_\pm(F) = (F \times \mathbf{S}^2) \cap \Gamma^r_\pm.$$

The approach proposed in [20] consists in applying the variational formulation (5.6) for each region D_r by imposing as boundary conditions on ∂D_r:

$$u|_{\Gamma^r_-} = \begin{cases} f & \text{on } \Gamma^r_-(F) \quad \forall F \in \mathcal{F}^b_r, \\ u^F_r & \text{on } \Gamma^r_-(F) \quad \forall F \in \mathcal{F}^i_r, \end{cases} \tag{5.17}$$

where u^F_r is the trace on F of the flux in the adjacent region to D_r by the face F (in the sense of Definition 5.3.1). In other words, the incoming flux f, given by the boundary condition (5.1b), is imposed on the boundary faces $F \in \mathcal{F}^b_r$ of D_r and the outgoing flux from the adjacent region to D_r through the face F is imposed on the internal faces $F \in \mathcal{F}^i_r$. The condition $u|_{\Gamma^r_-} = u^F_r$ on the internal faces is called upwind condition in the literature [24].

As suggested in [20], a local variational formulation for each region D_r is then:

$$\text{find } u \in W_r, \quad \text{such that } \; a_r(u,v) = L_r(v) \quad \forall v \in W_r, \tag{5.18}$$

with

$$a_r(u,v) = \int_{X_r} \left(\frac{1}{\sigma}(\omega \cdot \nabla u)(\omega \cdot \nabla v) + \sigma uv \right) + \sum_{F \in \mathcal{F}^b_r} \int_{\Gamma^r_+(F)} uv(\omega \cdot n) + \sum_{F \in \mathcal{F}^i_r} \int_{\Gamma^r_+(F)} uv(\omega \cdot n_r), \tag{5.19}$$

$$L_r(v) = \int_{X_r} q \left(v + \frac{1}{\sigma}(\omega \cdot \nabla v) \right) - \sum_{F \in \mathcal{F}^b_r} \int_{\Gamma^r_-(F)} fv(\omega \cdot n) - \sum_{F \in \mathcal{F}^i_r} \int_{\Gamma^r_-(F)} u^F_r v(\omega \cdot n_r). \tag{5.20}$$

This local variational formulation is nothing but the previous variational formulation (5.6) applied in the region D_r, where the role of the internal or boundary faces is highlighted because of the different type of boundary condition in (5.17).

5.3.3 Global variational formulation

In order to pass from a local variational formulation to a global one (which will be our new broken variational formulation), we simply sum the local problems (5.18) over all regions D_r of the mesh and remark that $\widetilde{W}$ is just the collection of spaces W_r. Before performing this summation, let us give the resulting global or broken variational formulation:

$$\text{find } u \in \widetilde{W}, \quad \text{such that } \; \tilde{a}(u,v) = \tilde{L}(v) \quad \forall v \in \widetilde{W}, \tag{5.21}$$

where

$$\tilde{a}(u,v) = \sum_r \int_{X_r} \left(\frac{1}{\sigma}(\omega\cdot\nabla u)(\omega\cdot\nabla v) + \sigma uv \right) + \int_{\Gamma_+} uv(\omega\cdot n) + \tilde{a}^i(u,v), \tag{5.22}$$

$$\tilde{a}^i(u,v) = \sum_{F\in\mathcal{F}_h^i} \int_{F\times\mathbf{S}^2} \left(\{\!\{u\}\!\}(\omega\cdot n_F) + \frac{1}{2}[\![u]\!]|\omega\cdot n_F| \right) [\![v]\!], \tag{5.23}$$

$$\tilde{L}(v) = \sum_r \int_{X_r} q\left(v + \frac{1}{\sigma}(\omega\cdot\nabla v) \right) - \int_{\Gamma_-} fv(\omega\cdot n), \tag{5.24}$$

and $\{\!\{u\}\!\}, [\![u]\!]$ are defined by (5.12) and (5.13), respectively.

Proposition 5.3.4. *The collection of local variational formulations* (5.18) *is equivalent to the global (or broken) variational formulation* (5.21)*, in the sense that if u is a solution of one of them then u is a solution of the other.*

Proof. The proof is constructive in the sense that we shall obtain (5.21) by summing the local variational formulations (5.18). Doing so leads indeed to (5.21) but with a different formula for $\tilde{a}^i$ which is

$$\tilde{a}^i(u,v) = \sum_r \sum_{F\in\mathcal{F}_r^i} \left(\int_{\Gamma_+^r(F)} u_r v_r(\omega\cdot n_r) + \int_{\Gamma_-^r(F)} u_r^F v_r(\omega\cdot n_r) \right), \tag{5.25}$$

where u_r, v_r are the restrictions to X_r of u, v. Note that $\tilde{a}$, defined by (5.22), is not the sum of the a_r's and similarly $\tilde{L}$, defined by (5.24), is not the sum of the L_r's. Formulas (5.22) and (5.24) are obtained by observing that

$$\sum_r \sum_{F\in\mathcal{F}_r^b} \int_{\Gamma_+^r(F)} uv(\omega\cdot n) = \int_{\Gamma_+} uv(\omega\cdot n),$$

$$\sum_r \sum_{F\in\mathcal{F}_r^b} \int_{\Gamma_-^r(F)} fv(\omega\cdot n) = \int_{\Gamma_-} fv(\omega\cdot n),$$

and passing the integrals on the internal faces in L_r to the bilinear form $\tilde{a}$. The bilinear form $\tilde{a}^i$ gathers all contributions from the internal faces. The proof that (5.25) is equivalent (5.23) is given by Lemma 5.3.5 below. In other words we have just proved that, if u is a solution of the collection of local variational formulations (5.18) for each region D_r, then it is also a solution of (5.21) by the very construction of this global formulation. The converse is obtained by using in the global formulation (5.21) a test function v that vanishes everywhere except in a region D_r and using the expression (5.25) for the bilinear form $\tilde{a}^i(u,v)$. □

Recall the notations for the positive $x^\oplus$ and negative $x^\ominus$ parts of a real number x:

$$x^\oplus = \frac{1}{2}(|x|+x), \quad x^\ominus = \frac{1}{2}(|x|-x),$$

which satisfies $(-x)^\oplus = x^\ominus$ and $(-x)^\ominus = x^\oplus$.

Lemma 5.3.5. *The bilinear form $\tilde{a}^i(u,v)$, defined by (5.25), can be rewritten:*

$$\tilde{a}^i(u,v) = \sum_{F\in\mathcal{F}_h^i} \int_{F\times\mathbf{S}^2} \left(u_{r_1}(\omega\cdot n_F)^{\oplus} - u_{r_2}(\omega\cdot n_F)^{\ominus}\right)[\![v]\!], \tag{5.26}$$

where u_{r_1} and u_{r_2} are the fluxes in the first and second regions D_{r_1} and D_{r_2} of the face F in the sense of Definition 5.3.1. Finally, (5.26), and thus (5.25), is equivalent to (5.23).

Proof. In (5.25) write the double sum $\sum_r \sum_{F\in\mathcal{F}_r^i}$ as a simple sum over all internal faces $F\in\mathcal{F}_h^i$, using for each face $F\in\mathcal{F}_h^i$ its associated triple (D_{r_1}, D_{r_2}, n_F):

$$\begin{aligned}\tilde{a}^i(u,v) = \sum_{F\in\mathcal{F}_h^i} \Bigg(&\int_{\Gamma_+^{r_1}(F)} u_{r_1}v_{r_1}(\omega\cdot n_{r_1}) + \int_{\Gamma_+^{r_2}(F)} u_{r_2}v_{r_2}(\omega\cdot n_{r_2}) + \\ &\int_{\Gamma_-^{r_1}(F)} u_{r_1}^F v_{r_1}(\omega\cdot n_{r_1}) + \int_{\Gamma_-^{r_2}(F)} u_{r_2}^F v_{r_2}(\omega\cdot n_{r_2})\Bigg),\end{aligned}$$

where n_{r_1} is the unit normal vector, outgoing from D_{r_1} and similarly for n_{r_2}. Next we rely on the notations for the positive and negative parts to rewrite all integrals on $F\times\mathbf{S}^2$:

$$\tilde{a}^i(u,v) = \sum_{F\in\mathcal{F}_h^i} \int_{F\times\mathbf{S}^2} \left(u_{r_1}v_{r_1}(\omega\cdot n_{r_1})^{\oplus} + u_{r_2}v_{r_2}(\omega\cdot n_{r_2})^{\oplus} - u_{r_1}^F v_{r_1}(\omega\cdot n_{r_1})^{\ominus} - u_{r_2}^F v_{r_2}(\omega\cdot n_{r_2})^{\ominus}\right),$$

Replacing (n_{r_1}, n_{r_2}) by $(n_F, -n_F)$ and $(u_{r_1}^F, u_{r_2}^F)$ by (u_{r_2}, u_{r_1}), we deduce

$$\tilde{a}^i(u,v) = \sum_{F\in\mathcal{F}_h^i} \int_{F\times\mathbf{S}^2} \left(u_{r_1}v_{r_1}(\omega\cdot n_F)^{\oplus} + u_{r_2}v_{r_2}(\omega\cdot n_F)^{\ominus} - u_{r_2}v_{r_1}(\omega\cdot n_F)^{\ominus} - u_{r_1}v_{r_2}(\omega\cdot n_F)^{\oplus}\right),$$

which yields (5.26) after regrouping terms in u_{r_1} and in u_{r_2}. In order to conclude we use

$$\begin{aligned}u_1(\omega\cdot n_F)^{\oplus} - u_2(\omega\cdot n_F)^{\ominus} &= \frac{1}{2}u_1\left(|\omega\cdot n_F| + (\omega\cdot n_F)\right) - \frac{1}{2}u_2\left(|\omega\cdot n_F| - (\omega\cdot n_F)\right) \\ &= \frac{1}{2}[\![u]\!]\ |\omega\cdot n_F| + \{\!\{u\}\!\}(\omega\cdot n_F),\end{aligned}$$

which, applied to (5.26), leads to (5.23). □

5.3.4 Existence and uniqueness

Proposition 5.3.4 does not say anything on the existence of solutions for the broken variational formulation (5.21) (it is just an equivalence result). The goal of this subsection is to provide an existence and uniqueness result for (5.21) in a quite indirect way. Indeed, we are not able to apply standard results like the Lax-Milgram theorem. This will be clear because the norm (5.11) of $\widetilde{W}$ is too strong to prove coercivity of the bilinear form. Rather, we introduce a new weaker norm on $\widetilde{W}$ which is not equivalent to (5.11) and for which $\widetilde{W}$ is not closed.

Let us define a new norm $\|v\|_{\widetilde{W}^\star}$ on $\widetilde{W}$ by

$$\|v\|^2_{\widetilde{W}^\star} = \sum_r \|v\|^2_{L^2(X_r)} + \sum_r \|\omega\cdot\nabla v\|^2_{L^2(X_r)} + \|v\|^2_{L^2_+} + \sum_{F\in\mathcal{F}_h^i} \int_{F\times\mathbf{S}^2} [\![v]\!]^2 |\omega\cdot n_F|, \tag{5.27}$$

where $L^2_+ = L^2(\Gamma_+, (\omega\cdot n)\,\mathrm{d}s\,\mathrm{d}\omega)$.

Lemma 5.3.6. *There exist $C>0$ such that $\|v\|_{\widetilde{W}^\star} \leqslant C\|v\|_{\widetilde{W}}$ for any $v\in\widetilde{W}$ but the reciprocal inequality does not hold true.*

Proof. The part of the proof that requires further attention is to upper bound the norms of the jumps on the inner faces $\sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![v]\!]^2|\omega\cdot n_F|$.

Let us use a triangular inequality

$$[\![v]\!]^2 = |v_1 - v_2|^2 \leqslant (|v_1| + |v_2|)^2 \leqslant 2\left(|v_1|^2 + |v_2|^2\right)$$

Then

$$\begin{aligned}
\sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![v]\!]^2|\omega\cdot n_F| &\leqslant 2\sum_{F\in\mathcal{F}_h^i}\sum_{D_r\in D_h;\ F\in\partial D_r}\int_{F\times\mathbf{S}^2} v_r^2|\omega\cdot n_F|,\\
&\leqslant 2\sum_r\sum_{F\in\mathcal{F}_r^i}\int_{F\times\mathbf{S}^2} v_r^2|\omega\cdot n_r|,\\
&\leqslant 2\sum_r\sum_{F\in\mathcal{F}_r}\int_{F\times\mathbf{S}^2} v_r^2|\omega\cdot n_r|,\\
&\leqslant 2\sum_r\sum_{F\in\mathcal{F}_r}\left(\int_{F\times\mathbf{S}^2_+} v_r^2|\omega\cdot n_r| + \int_{F\times\mathbf{S}^2_-} v_r^2|\omega\cdot n_r|\right),\\
&\leqslant 2\sum_r \|v\|^2_{L^2_{r,+}} + 2C\sum_r \|v\|^2_{W_r}.
\end{aligned}$$

Using the fact that the trace operator $\gamma^r_- : W_r \to L^2(\Gamma^r_-, |\omega\cdot n_r|\mathrm{d}s\,\mathrm{d}\omega)$ is continuous, with $\|v\|^2_{W_r} = \|v\|^2_{L^2(X_r)} + \|\omega\cdot\nabla v\|^2_{L^2(X_r)} + \|v\|^2_{L^2_{r,+}}$. □

Proposition 5.3.7. *The bilinear form $\tilde{a}$, defined by (5.22), is coercive on $\widetilde{W}$ for the norm (5.27). Namely, for $\alpha = \frac{1}{2}\min(\sigma_0, \sigma_\infty^{-1}, 1)$ (independent of the choice of the mesh), we have*

$$\tilde{a}(v,v) \geqslant \alpha\|v\|^2_{\widetilde{W}^\star} \quad \forall v\in\widetilde{W}. \tag{5.28}$$

Proof. For any $v\in\widetilde{W}$, using Lemma 5.3.5, we have

$$\tilde{a}(v,v) = \sum_r\int_{X_r}\left(\frac{1}{\sigma}(\omega\cdot\nabla v)^2 + \sigma v^2\right) + \int_{\Gamma_+} v^2(\omega\cdot n) + \sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}\{\!\{v\}\!\}[\![v]\!](\omega\cdot n_F) + \frac{1}{2}[\![v]\!]^2|\omega\cdot n_F|. \tag{5.29}$$

The term $\int_{F\times\mathbf{S}^2}\{\!\{v\}\!\}[\![v]\!](\omega\cdot n_F)$, a priori signless, is eliminated by using Green's formula.

For this purpose, first observe that

$$\int_{X_r}\left(\frac{1}{\sigma}(\omega\cdot\nabla v)^2+\sigma v^2\right)=\int_{X_r}\left(\frac{1}{2\sigma}\left((\omega\cdot\nabla v)^2+\sigma^2v^2+(\omega\cdot\nabla v+\sigma v)^2\right)-v(\omega\cdot\nabla v)\right),$$

thus

$$\sum_r\int_{X_r}\left(\frac{1}{\sigma}(\omega\cdot\nabla v)^2+\sigma v^2\right)=\sum_r a_r^\star(v)-\sum_r\int_{X_r}v(\omega\cdot\nabla v),$$

where

$$a_r^\star(v)=\int_{X_r}\frac{1}{2\sigma}\left((\omega\cdot\nabla v)^2+\sigma^2v^2+(\omega\cdot\nabla v+\sigma v)^2\right).$$

On the other hand, owing to Green's formula (5.15)

$$\sum_r\int_{X_r}v(\omega\cdot\nabla v)=\frac{1}{2}\int_\Gamma v^2(\omega\cdot n)+\frac{1}{2}\sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![v^2]\!](\omega\cdot n_F).$$

Next using $[\![v^2]\!]=2\{\!\{v\}\!\}[\![v]\!]$, we obtain

$$\sum_r\int_{X_r}\left(\frac{1}{\sigma}(\omega\cdot\nabla v)^2+\sigma v^2\right)=\sum_r a_r^\star(v)-\frac{1}{2}\int_\Gamma v^2(\omega\cdot n)-\sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}\{\!\{v\}\!\}[\![v]\!](\omega\cdot n_F),$$

thus

$$\tilde{a}(v,v)=\sum_r a_r^\star(v)-\frac{1}{2}\int_\Gamma v^2(\omega\cdot n)+\int_{\Gamma_+}v^2(\omega\cdot n)+\frac{1}{2}\sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![v]\!]^2|\omega\cdot n_F|.$$

On the other hand

$$-\frac{1}{2}\int_\Gamma v^2(\omega\cdot n)+\int_{\Gamma_+}v^2(\omega\cdot n)=\int_\Gamma v^2\left(-\frac{1}{2}(\omega\cdot n)+(\omega\cdot n)^\oplus\right)=\frac{1}{2}\int_\Gamma v^2|\omega\cdot n|.$$

Thereby

$$\begin{aligned}\tilde{a}(v,v)=&\sum_r\int_{X_r}\frac{1}{2\sigma}\left((\omega\cdot\nabla v)^2+\sigma^2v^2+(\omega\cdot\nabla v+\sigma v)^2\right)\\&+\frac{1}{2}\int_\Gamma v^2|\omega\cdot n|+\frac{1}{2}\sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![v]\!]^2|\omega\cdot n_F|.\end{aligned}$$

Since $0<\sigma_0\leqslant\sigma\leqslant\sigma_\infty$, we deduce

$$\tilde{a}(v,v)\geqslant\frac{1}{2}\sum_r\int_{X_r}\left(\sigma_\infty^{-1}(\omega\cdot\nabla v)^2+\sigma_0v^2\right)+\frac{1}{2}\int_{\Gamma_+}v^2|\omega\cdot n|+\frac{1}{2}\sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![v]\!]^2|\omega\cdot n_F|,$$

which implies the desired result (5.28). □

Proposition 5.3.8. *Let $u\in W$ be the unique solution to problem (5.1). If $u\in\widetilde{W}$, then it solves the broken variational formulation (5.21) too.*

Proof. This is immediate by the very construction of the broken variational formulation (5.21). Indeed, let $u \in W$ be the solution of problem (5.1). If $u \in \widetilde{W}$, then its restriction to D_r is obviously a solution in W_r of the local variational formulation (5.18) by virtue of Proposition 5.2.4, applied to the region D_r. Then, by Proposition 5.3.4, u is also a solution of (5.21). The condition $u \in \widetilde{W}$ can be seen as a kind of boundedness assumption since it requires that u has suitable traces in $L^2(\Gamma_+^r, (\omega \cdot n_r)\,\mathrm{d}s\,\mathrm{d}\omega)$, for any subdomain r. □

Theorem 5.3.9. *(well-posedness). Assume that the solution $u \in W$ of (5.1) belongs to $\widetilde{W}$. Then, under assumptions (5.2) on the data f, q and σ, problem (5.21) admits a unique solution in $\widetilde{W}$, which coincides with the solution to (5.1).*

Proof. The uniqueness of the solution follows from the coercivity of the bilinear form $\tilde{a}$ established in Proposition 5.3.7. The existence of a solution was proved in Proposition 5.3.8. □

5.3.5 Upper bound on the bilinear form

We obtain an upper bound on $\tilde{a}(u,v)$ in terms of the norm $\|v\|_{\widetilde{W}^\star}$, defined by (5.27) (the one used for coercivity), and a new stronger norm $\|u\|_{\widetilde{W}^+}$ defined by

$$\|u\|_{\widetilde{W}^+}^2 = \|u\|_{\widetilde{W}^\star}^2 + \sum_{F \in \mathcal{F}_h^i} \|\{\!\{u\}\!\}\|_{L_F^2}^2. \tag{5.30}$$

Here $L_F^2 \coloneqq L^2(F \times \mathbf{S}^2, |\omega \cdot n_F|\,\mathrm{d}s\,\mathrm{d}\omega)$ and

$$\|\{\!\{v\}\!\}\|_{L_F^2}^2 = \int_{F \times \mathbf{S}^2} \{\!\{v\}\!\}^2 |\omega \cdot n_F|.$$

By definition $\|v\|_{\widetilde{W}^\star} \leqslant \|v\|_{\widetilde{W}^+}$, $\forall v \in \widetilde{W}$.

Proposition 5.3.10. *The bilinear form $\tilde{a}$, defined by (5.22), is bounded in the sense that there is $M > 0$ such that*

$$|\tilde{a}(u,v)| \leqslant M \|u\|_{\widetilde{W}^+} \|v\|_{\widetilde{W}^\star} \quad \textit{for all } (u,v) \in \widetilde{W} \times \widetilde{W}, \tag{5.31}$$

with M independent of the choice of the mesh.

Proof. Using $0 < \sigma_0 \leqslant \sigma \leqslant \sigma_\infty$ and Cauchy-Schwarz inequality, we can bound from above $\tilde{a}(u,v)$, defined by (5.22),

$$\begin{aligned} |\tilde{a}(u,v)| \leqslant \frac{1}{\sigma_0} \sum_r \|\omega \cdot \nabla u\|_{L_r^2}\, \|\omega \cdot \nabla v\|_{L_r^2} + \sigma_\infty \sum_r \|u\|_{L_r^2}\, \|v\|_{L_r^2} + \|u\|_{L_+^2} \|v\|_{L_+^2} \\ + \sum_{F \in \mathcal{F}_h^i} \|\{\!\{u\}\!\}\|_{L_F^2} \|[\![v]\!]\|_{L_F^2} + \frac{1}{2} \sum_{F \in \mathcal{F}_h^i} \|[\![u]\!]\|_{L_F^2} \|[\![v]\!]\|_{L_F^2}, \end{aligned} \tag{5.32}$$

where $L^2_r = L^2(X_r)$. We now consider the entire right hand side of (5.32) as a single scalar product $\sum_{i=1}^{I} a_i b_i$ where the vector a collects all norms in u and b all norms in v. Using the inequality

$$\sum_{i=1}^{I} a_i b_i \leqslant \left(\sum_{i=1}^{I} (a_i)^2\right)^{1/2} \left(\sum_{i=1}^{I} (b_i)^2\right)^{1/2},$$

we obtain

$$|\tilde{a}(u,v)| \leqslant \max\left((\sigma_0)^{-1}, \sigma_\infty, 1)\right) \alpha(u)\beta(v),$$

with

$$\alpha(u)^2 = \sum_r \|\omega\cdot\nabla u\|^2_{L^2_r} + \sum_r \|u\|^2_{L^2_r} + \|u\|^2_{L^2_+} + \sum_{F\in\mathcal{F}_h^i} \|\{\!\{u\}\!\}\|^2_{L^2_F} + \sum_{F\in\mathcal{F}_h^i} \|[\![u]\!]\|^2_{L^2_F},$$
$$\beta(v)^2 = \sum_r \|\omega\cdot\nabla v\|^2_{L^2_r} + \sum_r \|v\|^2_{L^2_r} + \|v\|^2_{L^2_+} + 2\sum_{F\in\mathcal{F}_h^i} \|[\![v]\!]\|^2_{L^2_F}.$$

Recalling (5.27) and (5.30), we check that $\alpha(u)^2 = \|u\|^2_{\widetilde{W}^+}$ and $\beta(v)^2 \leqslant 2\|v\|^2_{\widetilde{W}^\star}$, which yields the desired upper bound (5.31). □

5.4 The discrete problem

The goal of this section is to construct a discrete approximation of the problem (5.21). In order to do this, we build a finite-dimensional subspace $\widetilde{W}_{N,k}$ of $\widetilde{W}$ and consider the approximate problem:

$$\text{find } u \in \widetilde{W}_{N,k} \text{ such that } \tilde{a}(u,v) = \tilde{L}(v), \quad \forall v \in \widetilde{W}_{N,k}$$

with the same bilinear form $\tilde{a}$ and linear form $\tilde{L}$ defined respectively in (5.22) and (5.24). The N index is related to the spherical harmonics method $\mathrm{P_N}$ used for the angular discretization and is described in section 5.4.1. In section 5.4.2 we present the Discontinuous Galerkin (DG) method, related to the index k, the maximum degree of the polynomials used in the spatial approximation.

5.4.1 Spherical harmonics method

The unit sphere $\mathbf{S}^2$ of $\mathbb{R}^3$ is parameterized by two angles θ and φ such that $\omega(\theta,\varphi) \in \mathbf{S}^2$ is defined by its components $\omega_x = \sin\theta\cos\varphi$, $\omega_y = \sin\theta\sin\varphi$ and $\omega_z = \cos\theta$, with $\theta \in [0,\pi]$ the colatitude (polar angle), and $\varphi \in [0, 2\pi]$ the longitude. A complete orthonormal basis of $L^2(\mathbf{S}^2)$ is given by the real spherical harmonics $y_n^m(\omega)$ (see e.g [12, §7.5.5], [20]) and leads to the expansion formula of u:

$$u(x,\omega) = \sum_{n=0}^{\infty} \sum_{m=-n}^{n} u_n^m(x) y_n^m(\omega). \tag{5.33}$$

The components $u_n^m(x) = (u, y_n^m)_{L^2(\mathbf{S}^2)}$ of the flux u are called angular flux moments. The spherical harmonics or $\mathrm{P_N}$ method consist in the truncation of (5.33) to term of degree at most N:

$$u^N(x,\omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} u_n^m(x) y_n^m(\omega). \tag{5.34}$$

The operator $P_N : u \mapsto u^N$ defines the orthogonal projection of $L^2(\mathbf{S}^2)$ to $\mathcal{H}_N$ the set of all finite linear combinations of spherical harmonics of degrees $n \leqslant N$, $\mathcal{H}_N = \mathrm{span}\{y_n^m;\ 0 \leqslant n \leqslant N,\ -n \leqslant m \leqslant n\}$. Finally, we introduce the subspace $\widetilde{W}_N$ of $\widetilde{W}$-functions that can be written as a linear combination of spherical harmonics of degree not exceeding N:

$$\widetilde{W}_N = \left\{ u \in \widetilde{W};\ u = \sum_{n=0}^{N} \sum_{m=-n}^{n} u_n^m(x) y_n^m(\omega) \right\}. \tag{5.35}$$

The finite-dimensional space $\widetilde{W}_{N,k}$ will be built as a subspace of $\widetilde{W}_N$.

5.4.2 Spatial approximation

The space discretization amounts to approximate the angular flux moments u_n^m by piecewise polynomial over the mesh D_h. To any $D_r \in D_h$ we associate a finite-dimensional space $\mathbb{P}^k(D_r)$ of d-variate polynomials of total degree at most k on D_r. We introduce the space of piecewise polynomials over D_h

$$\mathbb{P}_h^k = \left\{ v \in L^2(D);\ \forall D_r \in D_h,\ v|_{D_r} \in \mathbb{P}^k(D_r) \right\}. \tag{5.36}$$

Then we consider the finite-dimensional space $\widetilde{W}_{N,k}$ of functions in $\widetilde{W}_N$ that have piecewise polynomials angular moments u_n^m on the mesh D_h:

$$\widetilde{W}_{N,k} = \left\{ u(x,\omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} u_n^m(x) y_n^m(\omega) \text{ such that } u_n^m \in \mathbb{P}_h^k \right\}. \tag{5.37}$$

By construction $\widetilde{W}_{N,k} \subset \widetilde{W}_N \subset \widetilde{W}$.

5.4.3 The full discretization

The fully discretized problem that we consider is then stated as follows:

$$\text{find } u_h^N \in \widetilde{W}_{N,k} \text{ such that } \tilde{a}(u_h^N, v) = \tilde{L}(v),\ \ \forall v \in \widetilde{W}_{N,k}, \tag{5.38}$$

where $\tilde{a}$ and $\tilde{L}$ have been defined in (5.22) and (5.24) respectively.

Proposition 5.4.1. *The discrete problem (5.38) has a unique solution $u_h^N \in \widetilde{W}_{N,k}$.*

Proof. Since $\widetilde{W}_{N,k}$ is of finite dimension, the coercivity of $\tilde{a}$ obtained from Proposition 5.3.7 is sufficient to prove existence and uniqueness of a solution to problem (5.38). □

The P_N method comes in several variants depending on the variational formulation used and the spatial approximation adopted. Some of these variants present problems at the interfaces which are still not completely elucidated, the literature is rich in articles on this subject, see [120] and all the references cited therein. It is in part for this reason that some solvers based on the P_N method only deal with odd orders N. Proposition 5.4.1 shows that the numerical scheme studied in this work is not affected by these interface difficulties. The `NYMO` solver [20] based on this numerical scheme works for odd and even orders N.

Remark 5.4.2. *Once a basis of $\mathbb{P}_h^k$ is chosen, let us call it $(\varphi_j(x))_{1\leqslant j\leqslant J}$, the coefficients $u_{n,j}^m \in \mathbb{R}$ fully determine the discrete flux*

$$u_h^N(x,\omega) = \sum_{n=0}^{N}\sum_{m=-n}^{n}\sum_{j=1}^{J} u_{n,j}^m \varphi_j(x) y_n^m(\omega).$$

The dimension of the discrete space is

$$\dim \widetilde{W}_{N,k} = \dim(\mathbb{P}_h^k)\dim(\mathcal{H}_N) = \operatorname{card}(D_h)\binom{k+d}{d}(N+1)(\frac{d-1}{2}N+1).$$

The writing of the completely discretized problem (5.38) in matrix form reveals integrals over space and angle in the entries of the matrices. These integrals can be calculated exactly for a large class of meshes having plane or cylindrical faces, which can take into account the complexity of the geometries encountered in real applications in nuclear reactor calculations, see [20].

5.5 Error analysis

The goal of this section is to prove the convergence of the above numerical scheme and exhibit a precise order of convergence. If the broken variational formulation (5.21) were to fall in the scope of the Lax-Milgram theorem, the error analysis would be fairly standard. However it is not the case. Indeed, the bilinear form $\tilde{a}$ is coercive for a weaker norm than that of $\widetilde{W}$, as shown in Proposition 5.3.7, and unfortunately, we are unable to prove the continuity of the bilinear form $\tilde{a}$ with the same norm, see Proposition 5.3.10. As a first step, using the coercivity result of Proposition 5.3.7 and upper bound on $\tilde{a}$ in Proposition 5.3.10, we obtain a Céa's Lemma 5.5.1. In a second step, using this Céa's lemma, we obtain an upper bound on the numerical error.

5.5.1 Céa's lemma

The goal is to derive an upper bound for the approximation error $u - u_h^N$, where u solves the original problem (5.1) and u_h^N is the numerical solution of (5.38).

Lemma 5.5.1. *Let $u \in W$ be the unique solution of (5.1) and assume that $u \in \widetilde{W}$. Let $u_h^N \in \widetilde{W}_{N,k}$ be the discrete solution of (5.38). There exists a constant C independent of*

k, h and N such that

$$\left\|u-u_h^N\right\|_{\widetilde{W}^\star} \leqslant C \inf_{v_h^N\in\widetilde{W}_{N,k}} \left\|u-v_h^N\right\|_{\widetilde{W}^+}. \tag{5.39}$$

Proof. Let $v_h^N \in \widetilde{W}_{N,k}$. Since $u \in \widetilde{W}$, from the continuous and discrete variational formulations (5.21) and (5.38), we infer $\tilde{a}(u_h^N, w_h^N) = \tilde{a}(u, w_h^N)$ for any $w_h^N \in \widetilde{W}_{N,k}$. So it is easy to verify $\tilde{a}(u_h^N - v_h^N, u_h^N - v_h^N) = \tilde{a}(u - v_h^N, u_h^N - v_h^N)$. Thus from the coercivity result of Proposition 5.3.7 and upper bound on $\tilde{a}$ in Proposition 5.3.10, we obtain

$$\alpha\left\|u_h^N - v_h^N\right\|^2_{\widetilde{W}^\star} \leqslant \tilde{a}(u_h^N - v_h^N, u_h^N - v_h^N) = \tilde{a}(u - v_h^N, u_h^N - v_h^N) \leqslant M\left\|u-v_h^N\right\|_{\widetilde{W}^+}\left\|u_h^N - v_h^N\right\|_{\widetilde{W}^\star},$$

thus, we have $\left\|u_h^N - v_h^N\right\|_{\widetilde{W}^\star} \leqslant \alpha^{-1}M\left\|u-v_h^N\right\|_{\widetilde{W}^+}$. To conclude, we write

$$\left\|u-u_h^N\right\|_{\widetilde{W}^\star} \leqslant \left\|u-v_h^N\right\|_{\widetilde{W}^\star} + \left\|v_h^N-u_h^N\right\|_{\widetilde{W}^\star} \leqslant \left(1+\alpha^{-1}M\right)\left\|u-v_h^N\right\|_{\widetilde{W}^+},$$

from which we deduce (5.39), with $C = 1 + \alpha^{-1}M$, since v_h^N is arbitrary in $\widetilde{W}_{N,k}$. □

5.5.2 Error estimate

From now on it is assumed that $(D_h)_{h>0}$ is a sequence of admissible meshes [46, Definition 1.57] with mesh-width h going to zero. As is usual for obtaining error estimates of a numerical scheme, the solution u of the transport equation (5.1) is assumed to be smooth (and therefore obviously belongs to $\widetilde{W}$), more precisely to belong to some Sobolev space. Following the notations of [61], we introduce Sobolev spaces with mixed smoothness order. Let $s \in \mathbb{N}$ and $t \in \mathbb{N}$, two positive integers, $\alpha \in \mathbb{N}^d$ and $\beta \in \mathbb{N}^{d-1}$ two multi-indices, with $|\alpha| = \alpha_1 + \cdots + \alpha_d$. Let us denote by ∂_x^α the α-th weak derivative with respect to x and ∂_ω^β the β-th weak derivative with respect to ω. We then define

$$H^{s,t}(D\times\mathbf{S}^2) = \left\{u \in L^2(D\times\mathbf{S}^2);\ \partial_x^\alpha\partial_\omega^\beta u \in L^2(D\times\mathbf{S}^2),\ |\alpha|\leqslant s,\ |\beta|\leqslant t\right\}.$$

The space $H^{s,t}(D\times\mathbf{S}^2)$ is equipped with the norm

$$\|u\|^2_{H^{s,t}(D\times\mathbf{S}^2)} = \sum_{\substack{|\alpha|\leqslant s\\ |\beta|\leqslant t}} \left\|\partial_x^\alpha\partial_\omega^\beta u\right\|^2_{L^2(D\times\mathbf{S}^2)}.$$

We shall also used the semi-norm

$$|u|^2_{H^{s,t}(D\times\mathbf{S}^2)} = \sum_{\substack{|\alpha|= s\\ |\beta|= t}} \left\|\partial_x^\alpha\partial_\omega^\beta u\right\|^2_{L^2(D\times\mathbf{S}^2)}.$$

Our first main result is the following convergence theorem.

Theorem 5.5.2. *Assume that the unique solution u of the transport equation (5.1) belongs to $H^{k+1,t}(D\times\mathbf{S}^2)$. For $N \geqslant 1$ and $k \geqslant 1$, let u_h^N be the discrete solution of*

(5.38). *The following error estimate holds true*

$$\left\|u-u_h^N\right\|_{\widetilde{W}^\star} \leqslant \frac{c}{N^t}\|u\|_{H^{1,t}(D\times\mathbf{S}^2)} + ch^k\,|u|_{H^{k+1,0}(D\times\mathbf{S}^2)}\,.$$

As a consequence

$$\left\|u-u_h^N\right\|_{\widetilde{W}^\star} \leqslant c\|u\|_{H^{k+1,t}(D\times\mathbf{S}^2)}\left(\frac{1}{N^t}+h^k\right).$$

Remark 5.5.3. *Theorem 5.5.2 does not prove convergence for $k=0$. Furthermore, for $k\geqslant 1$ it is suboptimal since we expect an order of convergence of the type $\mathcal{O}\left(N^{-t}+h^{k+1}\right)$ for the $L^2(D\times\mathbf{S}^2)$-norm. The numerical experiments of section 5.7 confirm that a better rate of convergence holds true.*

Remark 5.5.4. *It is worth noting that the classical variational formulation obtained by multiplying equation (5.1a) by v (instead of $(v+\frac{1}{\sigma}\omega\cdot\nabla v)$),*

$$\text{find } u\in W \text{ such that } \int_X(-u(\omega\cdot\nabla v)+\sigma uv)+\int_{\Gamma_+}uv(\omega\cdot n)=\int_X qv-\int_{\Gamma_-}fv(\omega\cdot n)\quad\forall v\in W, \tag{5.40}$$

takes advantage of the term $u(\omega\cdot\nabla v)$ to obtain the classical order of convergence in space for DG methods $\mathcal{O}\left(h^{k+\frac{1}{2}}\right)$ [75]. Using the Cauchy-Schwarz inequality, we obtain $\int_{X_r}u(\omega\cdot\nabla v)\leqslant\left(h_r^{-1}\int_{X_r}u^2\right)^{\frac{1}{2}}\left(h_r\int_{X_r}(\omega\cdot\nabla v)^2\right)^{\frac{1}{2}}$ which implies $\int_{X_r}u(\omega\cdot\nabla v)\leqslant h^{-1}\|u\|^2_{L^2(X_r)}+h\|\omega\cdot\nabla v\|^2_{L^2(X_r)}$. It is then sufficient to weight the terms of the norm $\|\cdot\|_{\widetilde{W}^+}$ (5.30) by weights in h [46, §2.3]. In the variational formulation used in this work (5.6), the term $u(\omega\cdot\nabla v)$ is replaced by $(\omega\cdot\nabla u)(\omega\cdot\nabla v)$, preventing the use of this trick.

In the case $k=0$ of a piecewise constant approximation (which corresponds to the finite volume method), Theorem 5.5.2 can be improved because the streamline derivative term $\int_{X_r}(\omega\cdot\nabla u)(\omega\cdot\nabla v)$ vanishes in the bilinear form $\tilde{a}$ when one of its two arguments, u or v is piecewise constant in each element. Let us introduce a weaker norm $\|\cdot\|_{\widetilde{W}_0^\star}$ than $\|\cdot\|_{\widetilde{W}^\star}$, which is precisely defined by (5.27) but without the streamline derivative term, namely

$$\|u\|^2_{\widetilde{W}_0^\star}=\sum_r\|u\|^2_{L^2(X_r)}+\|u\|^2_{L^2_+}+\sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![u]\!]^2|\omega\cdot n_F|. \tag{5.41}$$

Our second main result is the following convergence theorem.

Theorem 5.5.5. *For $k=0$, assume that the unique solution u of the transport equation (5.1) belongs to $H^{1,t}(D\times\mathbf{S}^2)$. For $N\geqslant 1$, let $u_h^N\in\widetilde{W}_{N,0}$ be the discrete solution of (5.38). The following error estimate holds true*

$$\left\|u-u_h^N\right\|_{\widetilde{W}_0^\star} \leqslant c\|u\|_{H^{1,t}(D\times\mathbf{S}^2)}\left(\frac{1}{N^t}+h^{\frac{1}{2}}\right).$$

The proofs of both theorems are given in Section 5.6.

5.6 Proof of the error estimates

5.6.1 Strategy of the proof of Theorem 5.5.2

The proof of Theorem 5.5.2 starts from inequality (5.39) in Céa's lemma with the special choice $u_h^N = \Pi_h^k P_N u$

$$\left\|u - u_h^N\right\|_{\widetilde{W}^\star} \leqslant C\left\|u - \Pi_h^k P_N u\right\|_{\widetilde{W}^+},$$

where P_N is the angular projection operator, defined by $P_N u = u^N$ in (5.34) and Π_h^k is the spatial projection operator, defined as the L^2-orthogonal projection of $L^2(D)$ onto $\mathbb{P}_h^k$ defined by (5.36),

$$\Pi_h^k P_N u = \sum_{n,m}^{N} (\pi_h^k u_n^m) y_n^m. \tag{5.42}$$

Using the triangle inequality leads to

$$\left\|u - u_h^N\right\|_{\widetilde{W}^\star} \leqslant C\left(\|e_N\|_{\widetilde{W}^+} + \|e_k\|_{\widetilde{W}^+}\right), \tag{5.43}$$

where $e_N = u - P_N u$ is the angular truncation error and $e_k = P_N u - \Pi_h^k P_N u$ is the spatial approximation error inside the space $\widetilde{W}_N$. The aim is therefore to obtain separate estimates of $\|e_N\|_{\widetilde{W}^+}$ and $\|e_k\|_{\widetilde{W}^+}$.

5.6.2 Interpolation estimates

In this subsection we recall polynomial interpolation inequalities in Lemma 5.6.1 and a spherical harmonics approximation error in Lemma 5.6.2, as well as a bound on a partial sum of $|u_n^m|^2_{H^{k+1}(D_r)}$ in Lemma 5.6.3.

Lemma 5.6.1. *([46, §1.4.4] and [76, §4.3]) Let D_r be a cell of the mesh and F a face of the mesh. There exists a positive constant c, independent of D_r, F and h, such that the L^2-orthogonal projection operator π_h^k satisfies*

$$\left\|v - \pi_h^k v\right\|_{L^2(D_r)} \leqslant c h^{k+1} |v|_{H^{k+1}(D_r)} \qquad \forall v \in H^{k+1}(D_r), \tag{5.44}$$

$$\left|v - \pi_h^k v\right|_{H^1(D_r)} \leqslant c h^{k} |v|_{H^{k+1}(D_r)} \qquad \forall v \in H^{k+1}(D_r), \tag{5.45}$$

$$\left\|v - \pi_h^k v\right\|_{L^2(F)} \leqslant c h^{k+\frac{1}{2}} |v|_{H^{k+1}(D_r)} \qquad \forall v \in H^{k+1}(D_r). \tag{5.46}$$

Lemma 5.6.2. *([61, Lemma 3.2]) There exists a constant c, independent of N, such that, for any $g \in H^t(\mathbf{S}^2)$, the Sobolev space on the unit sphere of order $t \in \mathbb{N}$, the truncation error satisfies*

$$\|g - P_N g\|_{L^2(\mathbf{S}^2)} \leqslant \frac{c}{N^t} \|g\|_{H^t(\mathbf{S}^2)}.$$

Finally we recall a classical result. In the sequel, we use the notation $\sum_{n,m}^{N} = \sum_{n=0}^{N} \sum_{m=-n}^{m=n}$.

Lemma 5.6.3. *Under the assumption that the exact solution u belongs to $H^{k+1,0}(D\times\mathbf{S}^2)$, the following inequality holds*

$$\sum_r \sum_{n,m}^{N} |u_n^m|^2_{H^{k+1}(D_r)} \leqslant |u|^2_{H^{k+1,0}(D\times\mathbf{S}^2)}\,,$$

where u_n^m is the spherical harmonic decomposition of u, defined by (5.33).

Proof. Let $\alpha \in \mathbb{N}^d$ a multi-index with $|\alpha| = \sum_{i=1}^d \alpha_i$. By applying the derivation operator ∂_x^α to (5.33), it is easy to see

$$\int_{D_r\times\mathbf{S}^2} (\partial_x^\alpha u)^2 = \int_{D_r\times\mathbf{S}^2} \left(\sum_{n,m}^{\infty} (\partial_x^\alpha u_n^m) y_n^m\right)^2,$$

as the family $(y_n^m)_{n,m}$ form an orthonormal basis of $L^2(\mathbf{S}^2)$, we obtain

$$\|\partial_x^\alpha u\|^2_{L^2(D_r\times\mathbf{S}^2)} = \sum_{n,m}^{\infty} \|\partial_x^\alpha u_n^m\|^2_{L^2(D_r)},$$

which gives by summation over all α such that $|\alpha| = k+1$

$$|u|^2_{H^{k+1,0}(D_r\times\mathbf{S}^2)} = \sum_{n,m}^{\infty} |u_n^m|^2_{H^{k+1}(D_r)}\,.$$

The result announced in the lemma is obtained by summation over all regions of the mesh. □

5.6.3 End of the proof of Theorem 5.5.2

From inequality (5.43) it remains to provide upper bounds for $\|e_N\|_{\widetilde{W}^+}$ and $\|e_k\|_{\widetilde{W}^+}$. First, we rewrite

$$e_k = P_N u - \Pi_h^k P_N u = \sum_{n,m}^{N} (u_n^m - \pi_h^k u_n^m) y_n^m,$$

and use the fact that $(y_n^m)_{n,m}$ is an orthonormal basis of $L^2(\mathbf{S}^2)$. We separate each term in the definition of the norm $\|e_k\|_{\widetilde{W}^+}$.

(i) Upper bound on $\|e_k\|_{L^2(X_r)}$

By orthonormality of $(y_n^m)_{n,m}$ we obtain

$$\|e_k\|^2_{L^2(X_r)} = \sum_{n,m}^{N} \left\|u_n^m - \pi_h^k u_n^m\right\|^2_{L^2(D_r)}.$$

Using the interpolation inequality (5.44) leads to

$$\left\|u_n^m - \pi_h^k u_n^m\right\|_{L^2(D_r)} \leqslant ch^{k+1} \left|u_n^m\right|_{H^{k+1}(D_r)}.$$

It follows that,

$$\sum_r \|e_k\|^2_{L^2(X_r)} \leqslant (ch^{k+1})^2 \sum_r \sum_{n,m}^N \left|u_n^m\right|^2_{H^{k+1}(D_r)}.$$

Using Lemma 5.6.3 we conclude that

$$\sum_r \|e_k\|^2_{L^2(X_r)} \leqslant c(h^{k+1})^2 \left|u\right|^2_{H^{k+1,0}(D\times\mathbf{S}^2)}.$$

The estimates of the remaining terms follow the same kind of proof.

(ii) Upper bound on $\|\omega\cdot\nabla e_k\|_{L^2(X_r)}$

For the convective term, since $|\omega|=1$, we have

$$\|\omega\cdot\nabla e_k\|^2_{L^2(X_r)} \leqslant \int_{D_r\times\mathbf{S}^2} \sum_{i=1}^d |\partial_{x_i} e_k|^2 \leqslant \sum_{n,m}^N \sum_{i=1}^d \int_{D_r} \left|\partial_{x_i}(u_n^m - \pi_h^k u_n^m)\right|^2 \leqslant \sum_{n,m}^N \left|u_n^m - \pi_h^k u_n^m\right|^2_{H^1(D_r)}.$$

Using the interpolation inequality (5.45) yields

$$\left|u_n^m - \pi_h^k u_n^m\right|_{H^1(D_r)} \leqslant ch^k \left|u_n^m\right|_{H^{k+1}(D_r)},$$

which, after summation, leads to

$$\sum_r \|\omega\cdot\nabla e_k\|^2_{L^2(X_r)} \leqslant ch^{2k} \left|u\right|^2_{H^{k+1,0}(D\times\mathbf{S}^2)}.$$

(iii) Upper bound on $\|e_k\|_{L^2_+}$

For the boundary term on ∂D, we rely on the interpolation inequality (5.46)

$$\|e_k\|^2_{L^2_+} = \sum_{F\in\mathcal{F}_h^b} \int_{F\times\mathbf{S}^2} e_k^2 (\omega\cdot n)^{\oplus} \leqslant \sum_{F\in\mathcal{F}_h^b} \int_{F\times\mathbf{S}^2} e_k^2 \leqslant \sum_{n,m}^N \sum_{F\in\mathcal{F}_h^b} \int_F (u_n^m - \pi_h^k u_n^m)^2.$$

Inequality (5.46) implies

$$\left\|u_n^m - \pi_h^k u_n^m\right\|_{L^2(F)} \leqslant ch^{k+\frac{1}{2}} \left|u_n^m\right|_{H^{k+1}(D_r)},$$

thus

$$\|e_k\|^2_{L^2_+} \leqslant c(h^{k+\frac{1}{2}})^2 \left|u\right|^2_{H^{k+1,0}(D\times\mathbf{S}^2)}.$$

(iv) Upper bound on the jump $\|[\![e_k]\!]\|$ and mean $\|\{\!\{e_k\}\!\}\|$

A completely similar calculation leads to the following upper bound for the terms on the inner faces

$$\sum_{F\in\mathcal{F}_h^i}\int_{F\times\mathbf{S}^2}[\![e_k]\!]^2|\omega\cdot n_F|+\int_{F\times\mathbf{S}^2}\{\!\{e_k\}\!\}^2|\omega\cdot n_F|\leqslant c(h^{k+\frac{1}{2}})^2\,|u|^2_{H^{k+1,0}(D\times\mathbf{S}^2)}\,.$$

(v) Upper bound on $\|e_k\|_{\widetilde{W}^+}$

Therefore, summing up all terms, we obtain

$$\|e_k\|_{\widetilde{W}^+}\leqslant ch^k\,|u|_{H^{k+1,0}(D\times\mathbf{S}^2)}\,. \tag{5.47}$$

(vi) Upper bound on $\|e_N\|_{\widetilde{W}^+}$

We now turn to the estimate of the angular approximation $e_N=u-P_Nu$. By virtue of Lemma 5.6.2 we obtain

$$\sum_r\int_{D_r\times\mathbf{S}^2}|u-P_Nu|^2=\sum_r\int_{D_r}\|u-P_Nu\|^2_{L^2(\mathbf{S}^2)}\leqslant\frac{c}{N^{2t}}\sum_r\int_{D_r}\|u\|^2_{H^t(\mathbf{S}^2)}=\frac{c}{N^{2t}}\|u\|^2_{H^{0,t}(D\times\mathbf{S}^2)}.$$

We have $|\omega\cdot\nabla(e_N)|\leqslant|\nabla e_N|$, since $|\omega|\leqslant 1$. Then,

$$\begin{aligned}\sum_r\int_{D_r\times\mathbf{S}^2}|\omega\cdot\nabla(u-P_Nu)|^2&\leqslant\sum_r\int_{D_r\times\mathbf{S}^2}|\nabla(u-P_Nu)|^2\\&\leqslant\sum_r\int_{\mathbf{S}^2}|u-P_Nu|^2_{H^1(D_r)}\leqslant\frac{c}{N^{2t}}\|u\|^2_{H^{1,t}(D\times\mathbf{S}^2)}.\end{aligned}$$

The other boundary terms are bounded from above in a similar fashion, leading to

$$\|e_N\|_{\widetilde{W}^+}\leqslant\frac{c}{N^t}\|u\|_{H^{1,t}(D\times\mathbf{S}^2)}. \tag{5.48}$$

Finally, summing inequalities (5.47) and (5.48) leads to the desired result.

Proof of Theorem 5.5.5

In this subsection the polynomial order is $k=0$. In a first step, we revisit the proof of Céa's Lemma 5.5.1, using the fact that the streamline derivative of piecewise constant functions is zero. Recall that, by its very definition (5.41), the norm $\|\cdot\|_{\widetilde{W}_0^\star}$ satisfy $\|v\|_{\widetilde{W}_0^\star}\leqslant\|v\|_{\widetilde{W}^\star}$ for any $v\in\widetilde{W}$. Similarly, one can define a new norm $\|\cdot\|_{\widetilde{W}_0^+}$ from $\|\cdot\|_{\widetilde{W}^+}$ (see (5.30)) by removing the streamline derivative term, namely

$$\|u\|^2_{\widetilde{W}_0^+}=\|u\|^2_{\widetilde{W}_0^\star}+\sum_{F\in\mathcal{F}_h^i}\|\{\!\{u\}\!\}\|^2_{L^2_F}.$$

When $k = 0$ the result of Céa's lemma (with the same proof) reads

$$\left\|u - u_h^N\right\|_{\widetilde{W}_0^\star} \leqslant C \inf_{v_h^N \in \widetilde{W}_{N,k}} \left\|u - v_h^N\right\|_{\widetilde{W}_0^+}. \tag{5.49}$$

Indeed, coercivity of the bilinear form $\tilde{a}$ holds for the weaker norm $\|\cdot\|_{\widetilde{W}_0^\star}$. Furthermore, since there are no streamline derivatives in the formula for $\tilde{a}(u, v)$ if one of the functions u or v is piecewise constant, the upper bound on $\tilde{a}$ (coming from Proposition 5.3.10) is valid for the two norms $\|\cdot\|_{\widetilde{W}_0^\star}$ and $\|\cdot\|_{\widetilde{W}_0^+}$.

Eventually, the desired estimate is deduced from (5.49) by using the interpolation errors (5.44) and (5.46).

5.7 Numerical experiments

This section is devoted to some numerical tests performed with the `NYMO` software [20, 10, 11, 9], which is a P_N-transport solver of the CEA[1] reactor physics platform `APOLLO3`® [122, 98]. The goal is to compare the actual numerical errors with the theoretical error estimates obtained in Section 5.5. To this end, we design numerical experiments of escalating complexity.

We are interested in the transport source problem, i.e. the incoming flux f is considered to be zero in (5.1). We prescribe a manufactured solution u, from which we deduce the corresponding source q such that equation (5.1) is satisfied. The manufactured solutions are constructed with a finite number of angular moments. The angular approximation orders are then chosen to be sufficiently high, so that the exact solution is in the approximation space. There is therefore no error in angle, and only the error in space is studied. The solution is chosen such that it vanishes on the domain boundary. The chosen flux moments are polynomial, and when calculating the source q, their derivatives are calculated exactly.

From now on, the numerical solution is simply noted u_h. Then, we compute the L^2-error on the angular flux $\|u(x,\omega) - u_h(x,\omega)\|_{L^2(X)}$ and the L^2-error on the derivative in the streamline direction $\|\omega \cdot \nabla(u - u_h)\|_{L^2(X)}$. In practice, we merely compute the error projected in the space $\widetilde{W}_{N,k}$ [2, §6.2.4]. For two successive calculations where the mesh-width is divided by n, the order of convergence is calculated as $p = \log_n(\frac{e_h}{e_{\frac{h}{n}}})$.

For all experiments below, the calculation are performed in double-precision and linear systems are solved by GMRES with a 10^{-10} tolerance and a Jacobi preconditioner is used. Coordinates in 2 and 3 dimensions are denoted by $x = (x_1, x_2)$ and $x = (x_1, x_2, x_3)$.

5.7.1 1D homogeneous

The spatial domain is the interval $D = [0, 1]$. It is homogeneous in the sense that the total cross-section is constant on the domain, given by $\sigma = 0.48$. The transport source

[1] Commissariat à l'Énergie Atomique et aux Énergies Alternatives

problem in 1D is written

$$\mu\frac{\partial u}{\partial x}+\sigma u=q \quad \text{in }]0,1[\times[-1,1]\ni(x,\mu), \qquad u=0 \quad \text{on } \Gamma_-.$$

Recall that $y_n(\mu)$ are the normalized Legendre polynomials. The manufactured solution is chosen of order 2 in μ

$$u(x,\mu)=\sum_{n=0}^{N=2} u_n(x)y_n(\mu), \tag{5.50}$$

with $u_n(x)=x(1-x)$ for all n. It is necessary to use at least a P_3 approximation in angle in order to represent the source q. The mesh of D is uniformly refined and the errors on the angular flux $\|u(x,\mu)-u_h(x,\mu)\|_{L^2(X)}$ and on the derivative of the angular flux $\|\mu\partial_x(u(x,\mu)-u_h(x,\mu))\|_{L^2(X)}$ are displayed in Table 5.1.

Table 5.1: Convergence orders for the scalar flux and its derivative for the 1D problem.

Nbr. of elements	$\|u-u_h\|_{L^2(X)}$	Order (angular flux)	$\|\mu\partial_x(u-u_h)\|_{L^2(X)}$	Order (derivative)
$k=0$ spatial approximation, $N=3$ angular approximation				
2	1.910e-01	-	7.164e-01	-
20	2.423e-02	0.90	8.262e-01	-
200	2.463e-03	0.99	8.273e-01	-
2000	2.466e-04	1.00	8.273e-01	-
$k=1$ spatial approximation, $N=3$ angular approximation				
2	4.797e-02	-	4.137e-01	-
20	5.411e-04	1.95	4.136e-02	1.00
200	5.484e-06	1.99	4.136e-03	1.00
2000	5.492e-08	2.00	4.136e-04	1.00
$k=2$ spatial approximation, $N=3$ angular approximation				
2	2.213e-09	-	7.796e-09	-
20	2.721e-10	-	8.145e-09	-
200	3.191e-11	-	8.179e-09	-
2000	1.088e-11	-	8.182e-09	-

The numerical convergence orders are better than those given by Theorems 5.5.2 and 5.5.5. More precisely, we obtain a convergence rate of order $\mathcal{O}(k+1)$ for the angular flux in the L^2-norm (thus better by one order for $k\geqslant 1$ and by one half order for $k=0$) and a convergence rate of order $\mathcal{O}(k)$ for the derivative in the L^2-norm (exactly as predicted by our theoretical analysis). Moreover the error is zero (up to the GMRES tolerance) when the approximation order reaches the one of the manufactured solution. Furthermore, in the $k=1$ spatial approximation, $N=3$ angular approximation case, we note that the error on the streamline derivative is divided almost exactly by 10, resulting in a perfect slope of 1. The exact solution being continuous and polynomial, a quick calculation shows that in this case, the error depends solely on an integral on its derivative. This integral over a polynomial is calculated exactly.

5.7.2 2D homogeneous

Consider a square domain $D = [0,1]^2$. It is homogeneous in the sense that the total cross-section is constant on the domain, given by $\sigma = 0.48$. The manufactured solution is chosen as

$$u(x,\omega) = u_0^0(x)y_0^0(\omega) + u_1^{-1}(x)y_1^{-1}(\omega) + u_1^1(x)y_1^1(\omega),$$

with $u_0^0(x) = u_1^{-1}(x) = u_1^1(x) = x_1(1-x_1)x_2(1-x_2)$, and $y_n^m(\omega)$ being the real spherical harmonics. Here, the solution u is of order $N = 1$ regarding the angular variable, hence it is necessary to go at least to order $N = 2$ in angle in order to represent the source q. The mesh of D is uniformly refined as displayed on Figure 5.1. The error on the angular flux $\|u - u_h\|_{L^2(X)}$, the error on the derivative in the streamline direction $\|\omega \cdot \nabla(u - u_h)\|_{L^2(X)}$ and the error on the full gradient $\|\nabla(u - u_h)\|_{[L^2(X)]^2}$ are reported in Table 5.2.

The same remark on the convergence orders, as in the 1D case, are in order. The numerical errors match the theoretical error for the L^2-norm of the derivative, but is 1 less (for $k \geqslant 1$) or 1/2 less (for $k = 0$) for the L^2-norm of the angular flux. In addition, the rates of convergence for the streamline derivative and for the full gradient are the same. Consequently, only the error on the streamline derivative will be presented in the rest of this section. Finally, for $N = 2$ in angle and $k = 4$, the exact solution is included in the approximation space, therefore the errors reach the requested GMRES tolerance.

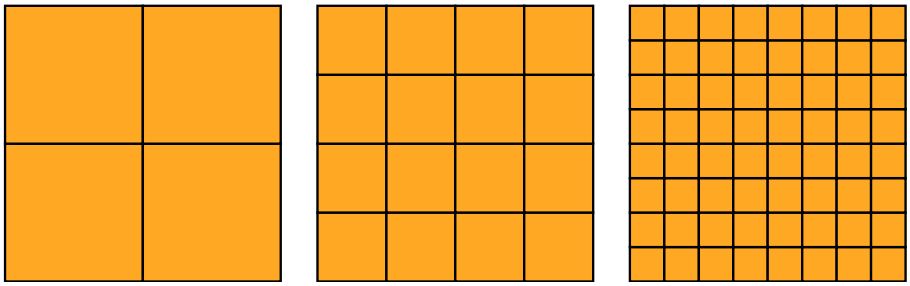


Figure 5.1: Uniform mesh refinement for the 2D problem.

5.7.3 3D homogeneous and heterogeneous

Consider a cubic domain $D = [0,1]^3$. The mesh is uniformly refined in each direction as shown in Figure 5.2. The manufactured solution is chosen as

$$u(x,\omega) = \sum_{n=0}^{N=1} \sum_{m=-n}^{m=n} u_n^m(x)y_n^m(\omega),$$

with

$$u_n^m(x) = \frac{m+2}{n+1} x_1^{n+1}(1-x_1)x_2^{m+2}(1-x_2)x_3(1-x_3),$$

for $n \in \{0,1\}$ and $m \in \{-n,\dots,n\}$. The solution u is P_1 in angle and it is necessary to use at least order 2 in angle in order to represent the source q.

Two different experiments are performed. In Table 5.3 the medium is homogeneous,

Table 5.2: Convergence orders for the angular flux and its derivatives for the 2D problem.

Nbr. of elements	$\|u-u_h\|_{L^2(X)}$	Order	$\|\omega\cdot\nabla(u-u_h)\|_{L^2(X)}$	Order	$\|\nabla(u-u_h)\|_{[L^2(X)]^2}$	Order
$k=1$ spatial approximation, $N=2$ angular approximation						
4	1.039e-02	-	9.062e-02	-	1.511e-01	-
16	2.790e-03	1.90	5.147e-02	0.82	8.529e-02	0.83
64	6.878e-04	2.02	2.657e-02	0.95	4.373e-02	0.96
256	1.674e-04	2.04	1.339e-02	0.99	2.195e-02	0.99
$k=2$ spatial approximation, $N=2$ angular approximation						
4	1.348e-03	-	3.441e-02	-	5.699e-02	-
16	2.008e-04	2.75	9.608e-03	1.84	1.609e-02	1.82
64	2.700e-05	2.90	2.460e-03	1.97	4.154e-03	1.95
256	3.472e-06	2.96	6.187e-04	1.99	1.049e-03	1.99
$k=4$ spatial approximation, $N=2$ angular approximation						
4	3.306e-10	-	1.392e-09	-	2.113e-09	-
16	2.841e-10	-	1.433e-09	-	5.272e-09	-
64	3.232e-10	-	1.449e-09	-	1.364e-08	-
256	5.329e-11	-	1.446e-09	-	1.961e-09	-

with a constant total cross-section $\sigma = 0.48$. In Table 5.4 the medium is heterogeneous and the total cross-sections are piecewise constant, given on Figure 5.2. As in 1D and 2D, the numerical convergence order for the derivative is in accordance with the theoretical one provided by Theorem 5.5.2, while it is one degree higher for the angular flux.

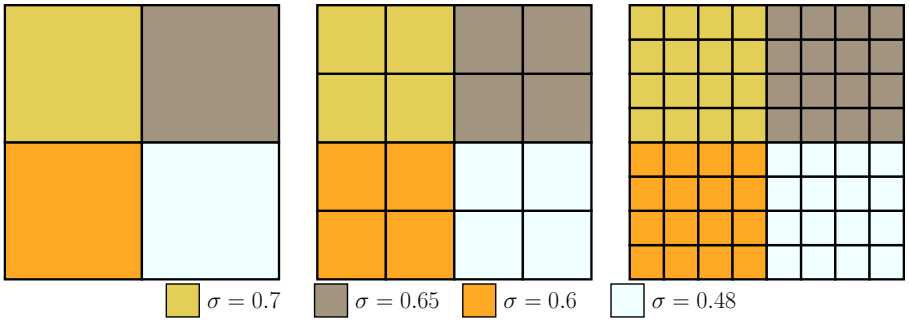


Figure 5.2: Radial section and uniform mesh refinement for the heterogeneous 3D problem.

Table 5.3: Convergence orders for the angular flux and its derivative for the homogeneous 3D problem.

Nbr. of elements	$\|u-u_h\|_{L^2(X)}$	Order (angular flux)	$\|\omega\cdot\nabla(u-u_h)\|_{L^2(X)}$	Order (derivative)
$k=3$ spatial approximation, $N=2$ angular approximation				
8	1.651e-04	-	3.866e-03	-
64	1.051e-05	3.97	6.520e-04	2.57
512	6.554e-07	4.00	8.771e-05	2.89
4096	4.117e-08	3.99	1.116e-05	2.97
$k=4$ spatial approximation, $N=2$ angular approximation				
8	4.019e-05	-	1.294e-03	-
64	1.489e-06	4.75	1.043e-04	3.63
512	4.896e-08	4.93	6.923e-06	3.91
4096	1.555e-09	4.98	4.391e-07	3.97
$k=5$ spatial approximation, $N=2$ angular approximation				
8	9.888e-06	-	3.308e-04	-
64	1.867e-07	5.73	1.226e-05	4.75
512	3.085e-09	5.92	3.992e-07	4.94
4096	4.962e-11	5.96	1.260e-08	4.99

Table 5.4: Convergence orders for the angular flux and its derivative for the heterogeneous 3D problem.

Nbr. of elements	$\|u-u_h\|_{L^2(X)}$	Order (angular flux)	$\|\omega\cdot\nabla(u-u_h)\|_{L^2(X)}$	Order (derivative)
$k=3$ spatial approximation, $N=2$ angular approximation				
8	1.619e-04	-	3.869e-03	-
64	1.041e-05	3.96	6.521e-04	2.57
512	6.496e-07	4.00	8.772e-05	2.89
4096	4.073e-08	4.00	1.116e-05	2.97
$k=4$ spatial approximation, $N=2$ angular approximation				
8	4.028e-05	-	1.295e-03	-
64	1.485e-06	4.76	1.043e-04	3.63
512	4.880e-08	4.93	6.924e-06	3.91
4096	1.551e-09	4.97	4.391e-07	3.98
$k=5$ spatial approximation, $N=2$ angular approximation				
8	9.808e-06	-	3.309e-04	-
64	1.853e-07	5.73	1.227e-05	4.75
512	3.066e-09	5.92	3.992e-07	4.94
4096	4.911e-11	5.96	1.260e-08	4.99

5.8 Conclusion

We proved an error estimate for the fully discrete linear Boltzmann transport equation using a discontinuous Galerkin – spherical harmonics method. More precisely, for $H^{k+1,t}(D \times \mathbf{S}^2)$ smooth solutions, we proved that an approximation of order $k \geqslant 1$ in space and N in angle converges at rate $\mathcal{O}\Big(N^{-t} + h^k\Big)$. For $k = 0$ we obtained that the convergence rate is $\mathcal{O}\Big(N^{-t} + h^{1/2}\Big)$. Numerical experiments in 1, 2 and 3 dimensions show that our theoretical error estimate is optimal for the derivative but is pessimistic for the angular flux. Actually, the numerical error estimate is $\mathcal{O}\Big(N^{-t} + h^{k+1}\Big)$ for the L^2-norm of the angular flux.

Therefore an obvious perspective and future work is to improve our theoretical error estimate. So far, our efforts have not been successful although we tried some classical tricks like using weighted norms in terms of h as in [75], [46, §2]. The use of a weighted test function $\Big(v + \frac{1}{\sigma}(\omega \cdot \nabla v)\Big)$ is similar to T-coercivity theory [34]. The use of T-coercivity could thus help in the analysis of the eigenvalue problem (criticity). Concerning other aspects not addressed by this study, taking into account explicitly scattering and fission in the transport model, especially anisotropic scattering would be very relevant, as well as a posteriori error estimates [135, 61, 62, 40, 51, 41, 35].

Chapter 6

Parallel neutron transport through domain decomposition

In this chapter, we target large-scale neutron transport simulations using the combined discontinuous Galerkin (DG) – spherical harmonics approximation. We leverage the benefits of the discretization to wrap the previously developed solver, in a domain decomposition framework. The developed parallel solver still handles unstructured, curved and non-conforming meshes with vacuum and reflection as boundary conditions. Robustness, strong and weak scalability experiments have been conducted on a petaflop cluster. We reach and maintain a strong scaling efficiency of 100 % up to 4096 cores and 80 % up to 8192 cores. In particular, a calculation with 913 million degrees of freedom is performed in 101 seconds. Asynchronous communications were very effective, allowing for near-perfect overlap between communications and computations. Moreover, the solver is robust, with variations in the obtained solutions compared to the sequential solver being close to the machine precision.

A subset of this chapter has been published with the reference [9] K. Assogba and L. Bourhrara. "The PN Form of the Neutron Transport Problem Achieves Linear Scalability Through Domain Decomposition". *Proceedings of International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2023)*. Niagara Falls, Ontario, Canada, Aug. 2023.

Outline of the current chapter

6.1 Introduction

This chapter focuses on large-scale neutron transport simulation using the combined discontinuous Galerkin - spherical harmonics approximation, as introduced in chapters 3 and 4. The targeted machine architecture is any supercomputer comprising independent CPU computing nodes with distributed memory. Since each computing node only contains a small part of the original data, even very large problems (with billions of unknowns, for example) can be solved. However, to achieve the desired efficiency, it is crucial to carefully design the data communication between processes.

We choose to implement a domain decomposition method in the existing `NYMO` code. The ultimate goal is to provide a scalable transport solver on exascale supercomputers. As `APOLLO3`® is an industrial platform, we attach great importance to the user experience. From the user's perspective, the use of the distributed program is identical to that of the sequential one, and no modification of user data sets is required.

6.1.1 Overview of parallel neutron transport solvers

Two experiments are used to evaluate the efficiency of a parallel algorithm. Strong scalability refers to the ability of a program to maintain or improve its performance as the number of computing units is increased while keeping the problem size constant. In other words, if a program is strongly scalable, doubling the resources should roughly halve the execution time for a fixed-size problem. Weak scalability, on the other hand, measures a program's ability to maintain or improve its performance as both the workload and the number of computing units are increased proportionally. A program exhibits weak

scalability if doubling the resources allows it to double the problem size while keeping the execution time roughly constant.

A significant portion of the literature on high-performance computing in reactor physics is devoted to the transport equation discretized in angle by the S_N method. In the 1990s, the Koch-Baker-Alcouffe (KBA) [80] algorithm was proposed as an efficient way to perform S_N sweeps. The Denovo S_N code [52], which uses this algorithm, exhibits a strong scalability efficiency of around 90 % on 144 cores and approximately 70 % on 3600 cores when using the S_{16} approximation. The efficiency of weak scalability is greater than 100 % when using up to 40,000 cores as compared to the time taken for computation on 4096 cores. Recently, [133] proposed a parallel sweeping that can handle meshes with cyclic dependencies. The solver Chi-Tech has achieved a weak scaling efficiency of over 80 % on more than 100,000 processes, solving a problem with 87.7 trillion unknowns. When it comes to strong scaling on a mesh with cyclic dependencies, the efficiency is either 58 % or 34 % depending on the partitioning strategy used at 2048 processes.

A concomitant approach has been developed, using Schwarz domain decomposition. In [74], a Schwarz iterative algorithm with Robin interface conditions is used in Minos. This method achieves an efficiency of around 80 % on up to 128 cores in the strong scalability experiment. In [103], the authors propose a domain decomposition method based on a block-Jacobi algorithm for Minaret, the APOLLO3® S_N transport core solver. However, this method suffers from convergence penalty in terms of both computing time and number of iterations, as mentioned in [102]. For a 2D core with 675,000 mesh cells, the efficiency is between 32 and 37 % for the calculations performed with MPI.

On the P_N neutron transport side, in the mid-1990s, [45] described a block domain decomposition method. More recently, Parafish [131], a neutron code based on the finite-element method, uses an algebraic domain decomposition to solve the second-order even-parity form of the Boltzmann equation. In a benchmark with P_7 approximation and a total of 3,150,000 unknowns, the efficiency of Parafish is around 28 % when running on 125 cores.

6.1.2 Chapter outline

In this chapter, we will cover the following topics. First, in Sections 6.2 and 6.3, we will provide a brief overview of the problem and the discretization method used. Then, we will discuss the parallelization strategy in Sections 6.4 and 6.6. Finally, in Section 6.7, we will evaluate the performance of the parallelization method by analyzing its robustness, weak and strong scalability behavior.

6.2 Model

The neutron transport equation describes the neutron distribution in a medium, taking into account scattering, fission and external sources. One wishes to determine the multigroup neutron flux $u^g(x, \omega)$ at any point (x, ω) of the phase space $X = D \times \mathbf{S}^2$. The space variable is denoted x and D is the spatial domain. The angular direction is denoted ω and $\mathbf{S}^2$ is the angular domain. The subscript g is the energy group, and unless

necessary is omitted in the following. In the phase space X with an incoming flux f through the incoming boundary Γ_-, the problem is written,

$$\omega \cdot \nabla u + \sigma u = q \quad \text{in} \quad X \tag{6.1a}$$

$$u = f \quad \text{on} \quad \Gamma_- \tag{6.1b}$$

where σ denotes the macroscopic total cross section and q the neutron source taking into account scattering (Hu) and possibly fission (Fu) and external neutron sources (q_{ext}). The eigenvalue problem consists in finding the eigenvalue λ associated with u when there is no external source. Further details are provided in Chapter 3.

We recall that the variational space (3.5) is denoted by W and $v \in W$ denotes a test function. The outgoing boundary is denoted Γ_+. We multiply the equation (6.1a) by $(v + \frac{1}{\sigma}\omega \cdot \nabla v)$ and then integrate over the phase space. After using Green's formula (3.6) and boundary condition (6.1b), one obtain the variational problem:

$$\text{find } u \in W \text{ such that } a(u, v) = L(v), \quad \forall v \in W, \tag{6.2}$$

with

$$a(u, v) = \int_X \left(\frac{1}{\sigma}(\omega \cdot \nabla u)(\omega \cdot \nabla v) + \sigma uv \right) + \int_{\Gamma_+} uv(\omega \cdot n), \tag{6.3}$$

$$L(v) = \int_X q \left(v + \frac{1}{\sigma}(\omega \cdot \nabla v) \right) - \int_{\Gamma_-} fv(\omega \cdot n). \tag{6.4}$$

Under assumptions about data f, q and σ, this problem is well posed, it admits a unique solution and this solution depends continuously on the data of the problem [22].

6.3 Discretization

The discontinuous Galerkin – spherical harmonics numerical scheme for the linear Boltzmann equation has been introduced in the Chapters 3 and 4. Here the key elements we need for the parallel implementation of the solver are recalled. The spherical harmonics method or $\mathrm{P_N}$ consists in developing with truncation to the order N, the angular flux on the real spherical harmonics basis $y_n^m(\omega)$,

$$u^N(x, \omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} u_n^m(x) y_n^m(\omega). \tag{6.5}$$

The unknown resulting from this development u_n^m, called the flux moment, is approached by piecewise polynomials of degree at most k, which we note in the following $\mathbb{P}^k = \text{span}\{\varphi_1, \ldots, \varphi_J\}$. In the end,

$$u_h^N(x, \omega) = \sum_{n=0}^{N} \sum_{m=-n}^{n} \sum_{j=1}^{J} u_{n,j}^m \varphi_j(x) y_n^m(\omega). \tag{6.6}$$

The approximation space thus constructed is $W_h = \text{span}(\varphi_j y_n^m)$. The discrete problem is:

$$\text{find } u \in W_h \text{ such that } a(u,v) = L(v), \quad \forall v \in W_h. \tag{6.7}$$

The incoming flux and the sources are developed in the same way. The test function is replaced by $v = \varphi_i(x) y_l^k(\omega)$. In each mesh cell D_r, the result is a local linear system of unknowns $\mathbf{u_r} = \left(u_{n,j}^m\right)\Big|_{D_r}$

$$A_r \mathbf{u_r} = \mathbf{q_r}, \tag{6.8}$$

with

$$A_r \mathbf{u_r} = \left(\frac{1}{\sigma} A_r^0 + \sigma A_r^1 + A_r^+\right)\mathbf{u_r} + \sum_{F \in \partial D_r} A_F^- u_F^{\uparrow}, \tag{6.9}$$

$$\mathbf{q_r} = \left(A_r^1 + \frac{1}{\sigma} A_r^2\right) q, \tag{6.10}$$

and $u_F^{\uparrow}$ is either the boundary flux f or the flux coming from the neighboring regions of D_r through the internal face F. The matrices A_r^0, A_r^1, A_r^+, A_F^- and A_r^2 are defined in the Section 3.4

6.3.1 Mesh equivalence class

The spatial basis functions $(\varphi_j)_{j=1,\cdots,J}$ are described in the Chapter 3. In each mesh cell, they are rescaled to the cell's origin. As a result, the elementary matrices (6.8) are the same for the mesh cells that are identical up a translation. These cells are grouped into equivalence classes (Section 3.6.2) and the equivalence classes of a mesh are shown in Figure 6.1. By grouping all the regions by packets where each packet contains only equivalent regions, the calculation of the elementary matrices is done only once per packet and the solver is implemented in a matrix assembly-free fashion. This results in substantial gains in storage and simulation time due to efficient cache fetching.

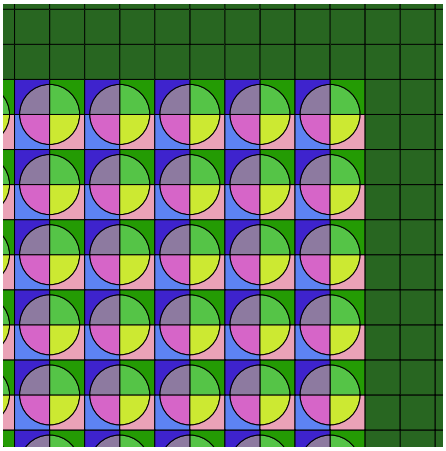


Figure 6.1: Equivalence classes of a mesh. The cells of the same color are identical by translation and thus belong to the same equivalence class.

6.4 Domain decomposition

The domain decomposition has been originally proposed in [123] to prove the existence and uniqueness of the Laplace equation $\Delta u = 0$ with Dirichlet boundary condition on an arbitrary domain. The original example is a domain made up of the union of a disk and a square, Figure 6.2. An explicit solution to Laplace's equation is known on each of these subdomains using Fourier series. Schwarz then shows that the method of alternately solving this problem on each of these subdomains converges.

With the advent of parallel computers, the idea of using this method to undertake parallel numerical simulations developed. A review of these methods can be found in [47].

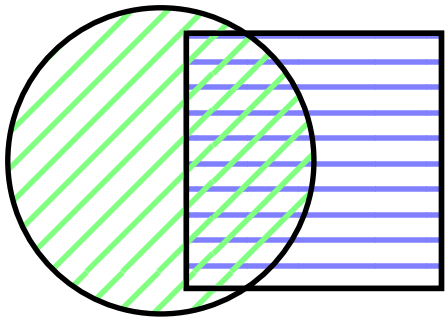


Figure 6.2: A domain formed by the union of a disk and a square, used to describe Schwarz's alternating algorithm.

6.4.1 A simple non-overlapping approach

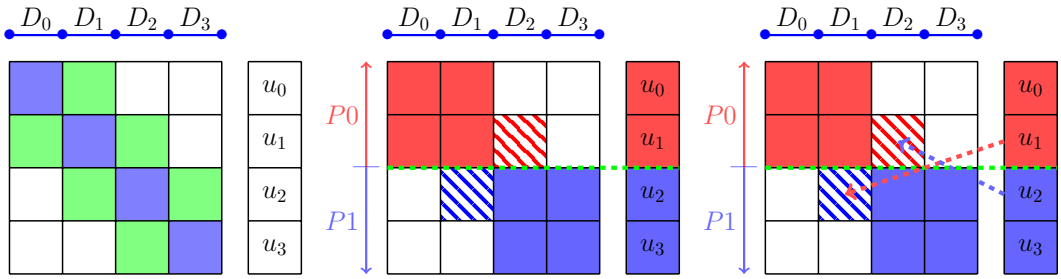
In this work, we keep only the essence of this idea, the domain is partitioned into two or more subdomains. We then proceed to solve the linear system resulting from the discretization.

A core component of this approach is the block structure of the matrices arising from the discretization. The off-diagonal blocks describe the interactions between adjacent mesh cells. From an algebraic point of view, partitioning the mesh induces a row-wise splitting of the matrices and vectors of the linear system, Figure 6.3. Each subsystem can thus be handled by independent computing processes. However, the off-diagonal part of the system requires portions of the flux from other processes. These other processes correspond to neighboring sub-domains in the mesh, and the fluxes exchanged correspond to upwind fluxes. This coupling is carried out inside the matrix-vector product using MPI. The MPI communications are made in a non-blocking way, thus enabling the overlap of communications and calculations.

By positioning at the algebraic scale, non-conforming and unstructured meshes with curved faces can be handled without additional work. We overcome the problem of partitioning these types of meshes by implementing a partitioning strategy based on Hilbert space-filling curves [72], along with a naive partitioning, using mesh numbering. Finally, the implementation is minimally intrusive and the parallel execution requires no

additional effort from the end-users.

To summarize, three basic components are needed: a mesh partitioner, a distributed sparse matrix-vector product and a distributed dot product. In the next, we present each ingredient and the recipe to combine them.



(a) Linear system from discretization. (b) Decomposition of the problem on two processes. (c) Communications required for resolution.

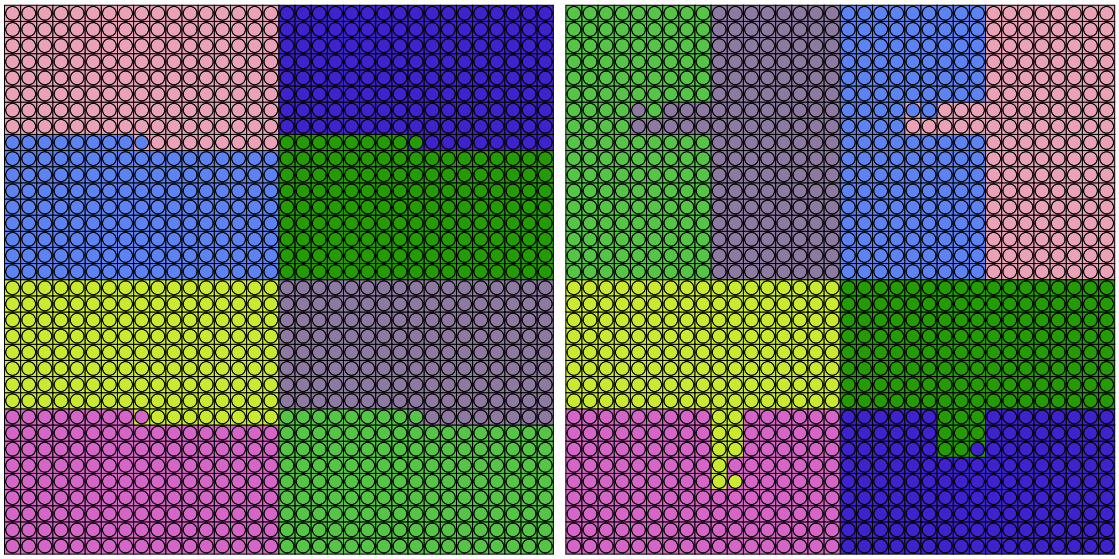
Figure 6.3: Example of a distributed calculation on a 1D mesh. (a) The matrix of the assembled problem without domain decomposition. The off-diagonal blocks describes the interactions between neighboring cells. (b) The problem is partitioned in a balanced way across 2 independent computing units $P0$ and $P1$. (c) The resolution involves fluxes communication between the processes.

6.4.2 Mesh partitioning

Let us denote by n_d the number of MPI processes in the global communicator. The parallel resolution starts by partitioning the initial mesh D_h into n_d subdomains. One needs to guarantee a load balancing between the processes while minimizing the communication between adjacent subdomains. The load balancing criterion is ensured by imposing that the number of elements inside each subdomain is the same (plus or minus one). Two widely used mesh partitioners are Metis [79] and Scotch [33]. However, they do not provide the capabilities to handle non-conforming meshes and curved elements. We have therefore implemented two partitioning strategies. The first one is based on mesh numbering and is called *simple* partitioning in the following. The second one is based on Hilbert space-filling curves [72] and is referred to as *geometric* partitioning.

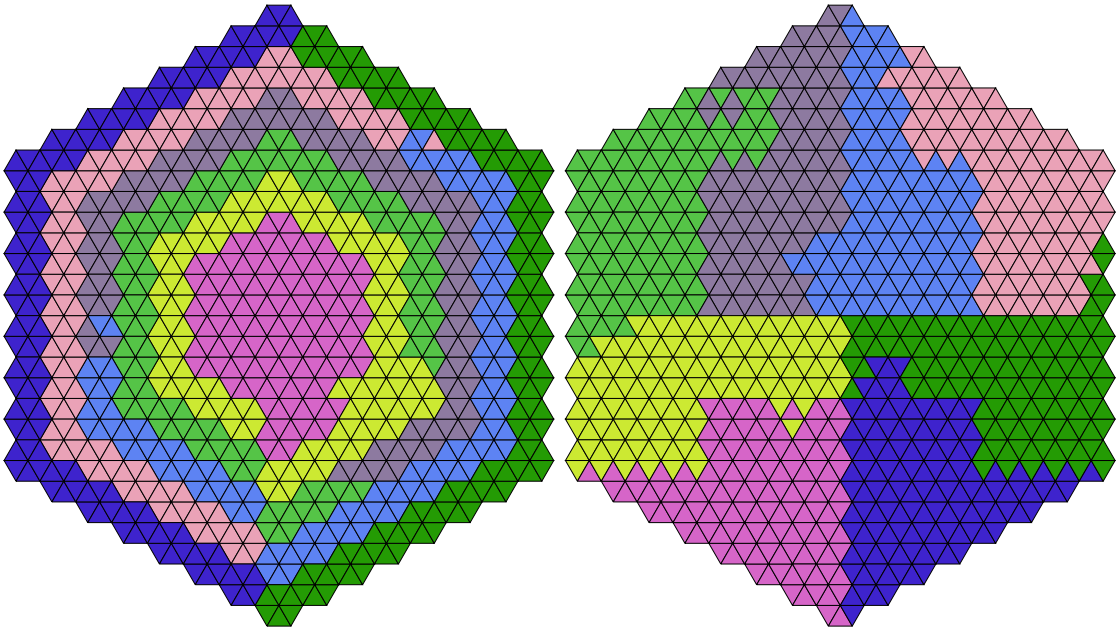
6.4.3 Simple partitioning

The principle of simple partitioning is to divide the mesh into blocks of cells with continuous identifiers. Figure 6.4a shows the partitioning of a set of four 17-by-17 pin cells assemblies using the simple partitioning. Let us note here that the numbering is contiguous within an assembly. Thus this naive partitioning still produces a satisfactory result for the shape of the subdomains. Figure 6.5a shows the simple partitioning of a hexagonal reactor where each hexagonal cell is cut into a triangle. In this case, simple partitioning leads to nested partitions, which is not ideal for communications.



(a) Simple partitioning (b) Geometric partitioning

Figure 6.4: Simple (a) and geometric (b) partitioning of 4 17-by-17 assemblies into 8 subdomains.

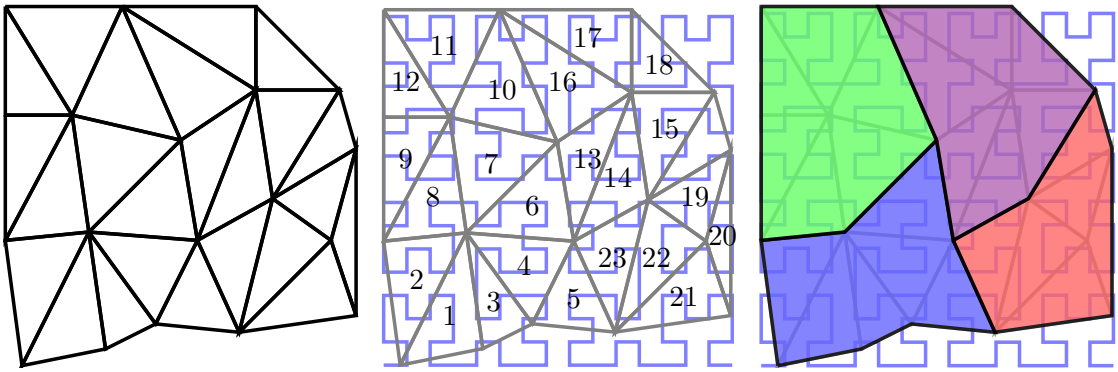


(a) Simple partitioning (b) Geometric partitioning

Figure 6.5: Hexagonal mesh where each hexagon is cut into a triangle (see Table 4.6). In this case, simple partitioning (a) leads to nested partitions, which is not ideal for communications.

6.4.4 Geometric partitioning

The simple partitioning is dependent on mesh numbering. To overcome this dependency, we use space-filling curves. One of the first occurrences in the literature is due to Peano, who in [105] defined a curve that fills an entire square. A little later, Hilbert proposed a geometrical construction of a curve similar to that envisaged by Peano [72]. An interesting feature of this curve is that it preserves the locality of the data. In other words, two points close together on the curve are also close together in the plane. The mesh cells are labeled according to their intersection with the trajectory followed by the curve, see Figure 6.6. The mesh is then partitioned so that each domain has the same number of elements (± 1). Figure 6.4b shows the partitioning of a set of 4 assemblies while Figure 6.5b shows the partitioning of a hexagonal reactor where each hexagonal cell is cut into a triangle.



(a) Mesh (b) Hilbert curve (c) Final partitioning

Figure 6.6: Geometric partitioning using Hilbert space-filling curve.

6.4.5 Post partitioning

After the partitioning, each process owns one domain and uses the corresponding macroscopic cross-sections. The cells that have one of their faces at the boundary of a subdomain, are referred to as halo cells and are labeled according. The rest of the solver proceeds by substituting the global data (mesh and cross-sections) with the local data. These global data can then be freed to save memory space. Finally, each process computes its sub-matrices A_r^0, A_r^1, A_r^+, A_F^- and A_r^2 defined in Section 3.4 on the local mesh and a communication matrix A_{comm}^- is generated from the halo mesh. All matrices are built in an assembly-free fashion.

6.5 Linear system resolution

6.5.1 Sparse linear solvers

While all the numerical experiments in Chapters 3, 4 and 5 are performed with GMRes [118], in this chapter we use BiCGStab [132]. In the GMRes algorithm, a set of House-

holder transformations must be constructed and applied. The construction of Householder transformations requires knowledge of all the blocks in the matrix A in equation (6.8). So in the case where the matrix blocks are distributed, it is necessary to adapt the algorithm for constructing and applying Householder transformations by adding global communications, typically Broadcast, and All_reduce. This could introduce bottlenecks. Hence the choice of BiCGStab, which is entirely local, except matrix-vector and scalar products.

6.5.2 Sparse matrix-vector product

When solving the linear system (6.8) inside the power iteration algorithm, all the processes work concurrently. The neighboring domains are coupled and exchange upwind flux on their common interfaces. At the algebraic level, this exchange is done at each sparse matrix-vector product (SpMV). The use of non-blocking communications allows communication time to be covered by computing time.

The asynchronous matrix-vector product attached to the Krylov solver takes place in three main stages and is described in Algorithm 4 and Figure 6.7. We start by posting an `MPI_Irecv` request to start receiving the incoming flux $x^{\uparrow}$ from neighboring domains. Then for each neighboring domain, the outgoing portion of the flux is copied into a buffer x_{out} and a `MPI_Isend` request is posted. The calculation starts and the part of the product involving only local degrees of freedom is calculated. Then, one waits for the incoming and outgoing communications to finish. The overlap works very well and the waiting time is usually negligible. Finally, one uses the received flux to compute the off-diagonal part of the product.

The copy step is needed because the halo cell data are not necessarily contiguous inside the flux vector. This operation is accelerated using OpenMP and the overhead is negligible. Two other possibilities are (i) to send the data element by elements or (ii) to define an MPI datatype that will handle the stride inside the flux vector. These two options have been tested and are slower than the one chosen. Waiting for outgoing communications to stop is necessary to deallocate the request handler and avoid memory leaks.

Algorithm 4 : General Matrix-Vector product from the point of view of a subdomain.

Input : A_{loc}, x_{loc}, y_{loc}

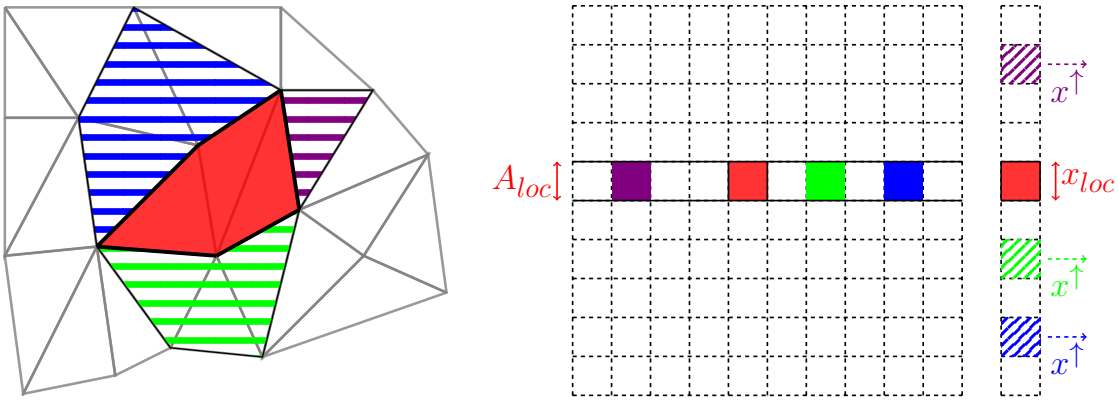
foreach $i_d \in domain_{neighbors}$ **do**
- $reqs_{in} \leftarrow$ MPI_Irecv($x^{\uparrow}$) // async
- Copy outgoing part of x_{loc} in the x_{out} buffer
- $reqs_{out} \leftarrow$ MPI_Isend(x_{out}) // async

$y \leftarrow y + A_{|\text{diag}} x$ // $A_{|\text{diag}} = A^0 + A^1 + A^+ + A^-_{\text{loc}}$

MPI_Waitall($reqs_{in}, reqs_{out}$)

$y \leftarrow y + A_{|\text{offd}} x^{\uparrow}$ // $A_{|\text{offd}} = A^-_{\text{comm}}$

To close this part, it is useful to note that, scalar products involve a collective reduction operation between all domains.



(a) Domain decomposition with a red subdomain highlighted.

(b) Matrix-vector product from the highlighted subdomain perspective.

Figure 6.7: Domain decomposition from the point of view of the red subdomain. Neighboring subdomains are hatched. The local matrix of the red domain is seen as a subset of the global matrix. Each process owns the matrix rows corresponding to the cells it holds. To perform a matrix-vector product, it needs the incoming fluxes from its neighboring domains.

6.6 Distributed parallel programming

MPI or Message Passing Interface [99, 65], is a standard for developing parallel programs in a distributed computing environment. In an MPI program, multiple processes communicate with each other through either individual messages (point-to-point communications) or as a group (collective communications). There are four modes of point-to-point communications: standard, synchronous, buffered, and ready. These communications can be either blocking or non-blocking.

The blocking functions `MPI_Send` and `MPI_Recv` will not return until the communication is complete. In contrast, non-blocking functions `MPI_Isend` and `MPI_Irecv` return immediately even if the communication has not been completed. The functions `MPI_Wait` and `MPI_Test` are used to check if the communication is complete. Additionally, since the MPI-2 standard, one-sided and persistent communications are also available. In one-sided communication, each process exposes some of its memory to other processes, and other processes can read from or write to this memory directly. Finally, the MPI-3 standard offers non-blocking collective communications.

There are several implementations of the standard, including OpenMPI [55], MPICH [64], and MPC [106], the CEA in-house solution. In this work, we use IntelMPI in its 2019 version.

MPI has become a widely used tool in scientific computing, and many legacy computing codes are being updated to integrate MPI as part of the transition to distributed computing architectures. However, integrating MPI communications with an existing computational code can pose challenges to the readability and maintainability of the code. This is due in part to the difficulty of reading a combination of physics or mathematics code with communication code. A possible solution to effectively integrate MPI into existing code

could be to encapsulate MPI functions in a class with a straightforward interface.

6.6.1 A lightweight object-oriented interface around MPI

To simplify the process of developing parallel programs and overcome the challenges mentioned earlier, we have developed an object-oriented interface that wraps around the MPI library. This interface offers a straightforward and elegant means to compose MPI messages, taking care of the more intricate details of the library, and making it less complicated to write parallel programs.

The interface automates the detection and conversion of the data type communicated into an `MPI_Datatype`. Additionally, it automatically resizes the reception buffer. Each function call within this class is timed with `MPI_Wtime`, ensuring that it returns an `MPI_SUCCESS` or triggers an error handler if an error occurs. Finally, upon completion of execution, we obtain a table displaying the number of calls and the execution time for each function.

A collective communication example

Let's consider that a process, here numbered `root`, wants to send a message to all the other processes in the communicator `comm`. Algorithm 6 shows how this procedure works using the `NYMO.Comm` submodule. When the method `bcast` is called, two operations take place in the background. First, the size of the message is broadcasted and the non-root processes resize the message container to the size received. Then the message content is broadcasted. The equivalent in pure MPI would be Algorithm 5. The line `comm.bcast(msg)` of Algorithm 6 turns into at least four lines of code in Algorithm 5.

It's worth noting that the preliminary send of the message size can be avoided. In this case, we can consider using an `MPI_Probe` followed by an `MPI_Get_count`. This saves one communication, but in a multi-threaded environment, the pairing of `MPI_Probe` and `MPI_Recv` is not thread-safe [60, 73]. In addition, it adds function calls and complexity to implementations. Let us illustrate this in the next section.

Algorithm 5 : Broadcast with pure MPI

Input : `MPI_Comm` comm; int root

```
1 int rank, size ;
2 MPI_Comm_rank(comm, &rank) ;
3 MPI_Comm_size(comm, &size) ;
4 std::string msg ;
5 if (rank == root) msg = "Hello" ;
6 int msg_size = msg.size() ;
7 MPI_Bcast(&msg_size, 1, MPI_INT, root, comm) ;
8 if (rank != root) msg.resize(msg_size) ;
9 MPI_Bcast(const_cast<char*>(msg.c_str()), msg_size, MPI_BYTE, root, comm) ;
```

Algorithm 6 : Broadcast with NYMO.Comm

```
Input : NYMO_Comm comm; int root
auto rank = comm.rank() ;
auto size = comm.size() ;
std::string msg ;
if (rank == root) msg = "Hello" ;
comm.bcast(msg) ;
```

A point-to-point communication example

Here, let us consider that there are two processes in the communicator `comm`. The root process sends a message of random length to its neighbor. Algorithm 7 is the pure MPI version and Algorithm 8 is the one using NYMO.Comm. This exchange takes just two lines of code using `NYMO.Comm` submodule.

Algorithm 7 : Point-to-point communication with pure MPI

```
Input : MPI_Comm comm; int rank, root, dest, source, tag
if rank == root then
    std::vector<int> buffer(rand()) ;
    MPI_Send(buffer.data(), buffer.size(), MPI_INT, dest, tag, comm) ;
else
    MPI_Status status ;
    MPI_Probe(dest, tag, comm, &status) ;
    int data_size ;
    MPI_Get_count(&status, MPI_INT, &data_size) ;
    std::vector<int> buffer(data_size) ;
    MPI_Recv(buffer.data(), data_size, MPI_INT, source, tag, comm, &status) ;
```

Algorithm 8 : Point-to-point communication with NYMO.Comm

```
Input : NYMO_Comm comm; int rank, root, dest, source, tag
if rank == root then
    std::vector<int> buffer(rand()) ;
    comm.send(buffer, dest, tag) ;
else
    std::vector<int> buffer ;
    comm.recv(buffer, source, tag) ;
```

6.6.2 Pre-processing: broadcast of mesh, cross-sections and solver options

Due to the Primary-Secondary architecture of parallelism in APOLLO3®, only the root process (the process of rank 0 by default) owns the data. These include the mesh, cross-sections and solver options, which must be supplied to the solver's input.

To send an object instance, or in general, non-contiguous data in memory with MPI, there are three main options:

(a) Serialize the object into a byte string and send the result of this serialization. This involves converting the object into a binary format that can be reconstructed elsewhere.
(b) Register the object as an MPI datatype. This defines how the object should be laid out in memory so MPI knows how to send its constituent parts.
(c) Send each attribute of the object separately and reassemble it at the receiving end.

We have opted for option (c), which is easy to use and implement. So each of the classes concerned implements a `distribute` method that describes how to exchange its attributes. When starting a calculation, the `NYMO` API is called and native `APOLLO3`® objects are used to build internal `NYMO` objects on the root process. At this stage, we send the attributes of newly constructed objects to the other processes. The other processes use the data received to rebuild the objects in their local storage.

At the end of pre-processing, `NYMO` operates internally according to the Simple Program Multiple Data (SPMD) model: all processes are on the same level, and the same program is run by all of them with different data.

6.6.3 Post-processing: aggregate the flux

At the end of a distributed calculation, the flux is dispatched between the different processes. It should be gathered on the main process for post-processing by the other components of `APOLLO3`®. Post-processing includes, for example, the output of reaction rates as HDF[1] files, the generation of multi-parameter cross-section libraries for two-step core-lattice calculations, flux homogenization and visualization.

To this end, local fluxes and local degree-of-freedom managers are transferred to the main process. The main process then uses the data received to reconstruct the complete flux.

6.7 Numerical experiments

The numerical method described in the Section 6.3 have been applied to a wide variety of reactor cores and lattice calculation [20, 10, 11]. The C5G7 benchmark [89] is used here to assess the performance of the distributed implementation.

The C5G7 core is a reduced Pressurized Water Reactor designed to evaluate the ability of transport solvers to perform core calculations without spatial homogenization. Presented in Figure 6.8, the quarter-core is made up of 4 assemblies surrounded by a reflector. Each assembly is made of a grid of 17 by 17 pins. There are 4 axial slices, including 3 fuel slices and in the case presented here, Rodded B, the control rods are inserted $^2/_3$ in the UOx and $^1/_3$ in the MOx. The results of shared memory calculations are presented in Chapter 4 and [11], here we focus on the distributed solver.

[1] Hierarchical Data Format

For all the experiments carried out, the discretization chosen is of order P_6 in angle and linear $\mathbb{P}^1$ in space. The Krylov solver used is BiCGStab [132] with a tolerance (if not specified) of 10^{-4} and the point Jacobi preconditioner. The tolerance on the fission source and on the k_{eff} during outer iterations is 10^{-5}. The flux is calculated and stored in single precision, and the k_{eff} in double precision.

Three experiments are conducted: robustness, strong scalability and weak scalability. All calculations are performed on the Milan partition of the CEA's petaflops cluster Topaze. Milan has 864 nodes, each housing two AMD EPYC 7763 2.45 GHz sockets, equipped with 64 cores each. The nodes are linked together by a high-performance InfiniBand HDR-100 network. The shared-memory-only calculations are performed on one socket. For hybrid MPI + OpenMP jobs, each MPI process is binded to one socket.

The software toolchain consists of Intel C++ Compiler and Intel MPI in their versions 2019.5.281. All developments are available in `APOLLO3®`.

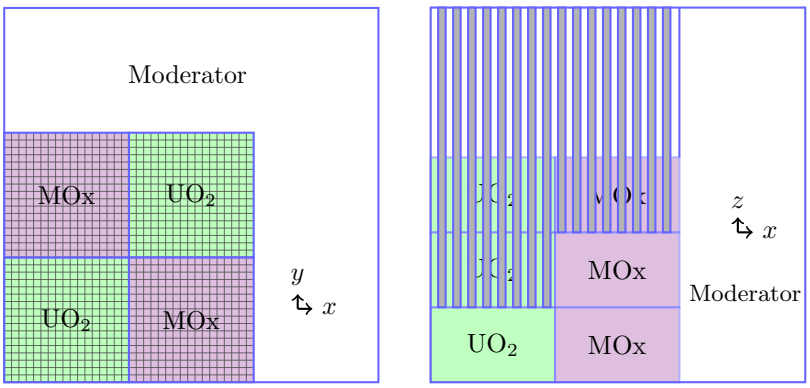


Figure 6.8: Radial (left) and axial (right) sections of the C5G7 core.

6.7.1 Robustness

The robustness analysis is done in two steps. First, we check that the `NYMO` calculation without decomposition is consistent with the Monte Carlo reference solution provided in [89]. These results are provided in Table 6.1. Here, each of the 3 fuel axial slices is divided into 8, and the top layer (moderator) is divided into 16, bringing the number of cells to 832,320. All the tolerances (on outer iterations, k_{eff}, thermals, and matrix solver) are fixed at 10^{-5}. The average (avg), root mean square (rms) and mean relative (mre) pin power errors are defined in[11, §IV]. The maximum error on the power is 1.7%, it is consistent with what is expected when comparing a deterministic code to a Monte Carlo code.

In the second stage, we evaluate the variation on the k_{eff}, the pin power and the number of outer iterations when increasing the number of subdomains. The objective is to verify that the error on the eigenvalue and on the flux in parallel is negligible compared to the sequential solution. The simple partitioning is used. For the eigenvalue, the calculated error is the relative error with respect to the solution without domain decomposition $\delta = {}^{(\lambda_{n_d}-\lambda_{n_0})}/_{\lambda_{n_0}}$ in pcm (10^{-5}) .

Table 6.2 shows the results of the robustness experiment. We observe that there is no variation on the k_{eff} and on the number of iterations. The maximum bias on the fission rate is 4 pcm. Since the flux is computed and stored in simple precision, these spreads are negligible. However, they can be justified by the fact that in parallel, the same operations are performed as in sequential, but in a different order. As machine arithmetic is not associative, these small variations are observed. The method is robust, in the sense that the variations on the solution obtained when the number of subdomains is increased are of the order of machine precision. This offers strong guarantees on the behavior of the parallel solver, which is essential for industrial use.

Table 6.1: Eigenvalue and pin power distribution error for Rodded B case of C5G7 problem.

	k_{eff}	error (pcm)	avg (%)	rms (%)	mre (%)	max (%)	#outer
MCNP	1.07777	-	-	-	-	-	-
NYMO	1.07864	80	0.351	0.447	0.340	1.752	31

Table 6.2: Robustness experiment. The eigenvalue and fission rates obtained with domain decomposition are compared with a reference calculation without decomposition.

n_d	k_{eff}	error (pcm)	avg (%)	rms (%)	mre (%)	max (%)	#outer
0[2]	1.07864	-	-	-	-	-	31
2	1.07864	0	0.0006	0.0008	0.0005	0.0038	31
4	1.07864	0	0.0008	0.0009	0.0007	0.0025	31
8	1.07864	0	0.0010	0.0013	0.0009	0.0040	31

6.7.2 Strong scalability

The objective of strong scalability is to evaluate the potential reduction in time when more computing resources are available. Thus, the number of computing units is increased while the problem size remains constant. We measure speedup $s(d) = {}^{t(0)}/_{t(d)}$ and efficiency $e(d) = {}^{s(d)}/_{n_d}$, $t(d)$ being the elapsed time for n_d subdomains. We use this test to compare the *simple* and *geometric* partitioning. Each axial slice is divided into 4, bringing the number of cells in the mesh to 665,856 and the total number of degrees of freedom to 913,554,432.

Figure 6.9 and Table 6.3 shows the results of this experiment. We go up to 128 domains, which corresponds to 8192 cores. We first notice that whatever the chosen partitioning, the measured time decreases linearly. In addition, there is no real difference in performance between simple and geometric partitioning here. The efficiency is almost perfect up to 64 domains, then decreases a little at 128 domains. At 128 domains, each domain owns 5 216 mesh cells, it can be reasonably assumed that the workload is low in this case. In addition, communication time represents around 50% of the total

[1] 0 means shared-memory calculation without domain decomposition.

time elapsed. The bulk of the communication time is dominated by global reduction operations.

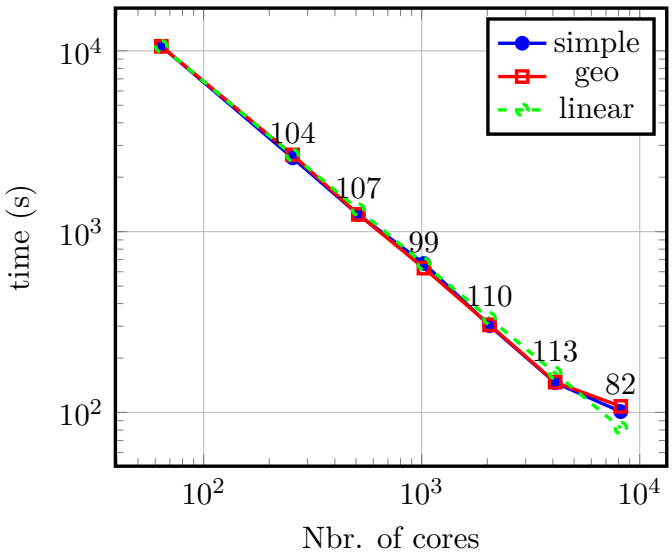


Figure 6.9: Strong scaling experiment. The simple and geometric partitioning are compared to linear scaling. The obtained efficiency for simple partitioning is displayed along the curve.

Table 6.3: Strong scaling experiment on up to 128 domains using a total of 8192 CPU-cores.

		Simple partitioning			Geometric partitioning		
n_d	#core	time (s)	speedup	efficiency (%)	time (s)	speedup	efficiency (%)
0^1	64	10597	-	-	10597	-	-
4	256	2557	4.1	104	2658	4	100
8	512	1242	8.5	107	1245	8.5	106
16	1024	667	15.9	99	628	16.9	106
32	2048	302	35.1	110	305	34.7	109
64	4096	146	72.6	113	147	72.1	113
128	8192	101	104.9	82	108	98.1	77

6.7.3 Comparison to other solvers

Efficiency being a good common metric, Table 6.4 summarizes strong scaling results for Minos, Minaret and Parafish. These results are compared to those obtained by `NYMO` on a problem with 913 million unknowns. `NYMO` maintains near-perfect efficiency with ease.

Table 6.4: Comparison of strong scalability of parallel neutronics finite element solvers. The data reported are from the best-published results for each solver.

	Efficiency % (#core)			Model
Parafish [131, §3.3]	53 (5)	37 (25)	28 (125)	Transport P_N
Minaret [103, §3.3]	64 (13)	32 (27)	37 (39)	Transport S_N
Minos [74, §7.4]	88 (32)	88 (64)	85 (128)	Diffusion SP_N
`NYMO`	110 (2048)	113 (4096)	82 (8192)	Transport P_N

6.7.4 Weak scalability

Here, the number of subdomains is increased from 16 to 32 then 64, while keeping the problem size per process constant. To do this, the mesh is refined axially. Starting from the unrefined mesh, each plane is divided into 2 then into 4. The performance metric used is the weak scaling efficiency, which is defined as the ratio of the execution time on 16 processes to the execution time on n_d processes. The goal is to achieve an efficiency close to one, which indicates that the algorithm scales effectively as more processes are added.

Table 6.5 presents the results of this experiment. The efficiency obtained being close to 100%, these results are excellent. Added to those of the strong scalability, they demonstrate that our parallel implementation is very efficient and versatile.

Table 6.5: Weak scaling experiment.

			Simple partitioning		Geometric partitioning	
n_d	#core	#dof	time (s)	efficiency (%)	time (s)	efficiency (%)
16	1024	228,388,608	145	-	142	-
32	2048	456,777,216	154	94	157	90
64	4096	913,554,432	146	99	147	96

6.8 Conclusion

In this chapter, we have presented a domain decomposition method that is suitable for solving the transport equation in a distributed memory environment. We have also introduced a mesh partitioning strategy based on Hilbert space-filling curves. This method is well-suited for DG-based transport discretizations and is minimally intrusive in terms of implementation.

Numerical experiments demonstrate that the method is highly robust, showing little variation in the solutions obtained as the number of computational processes increases. Additionally, it has excellent weak and strong scalability, which is beneficial to end-users as it requires no additional effort compared to the sequential version.

Looking forward, we aim to evaluate this approach on larger test cases, particularly for the direct calculation of heterogeneous whole cores. Additionally, we plan to integrate distributed and scalable preconditioners and acceleration strategies.

Finally, exascale implies GPUs. It is therefore necessary to adapt current codes to target these architectures. The first idea would be to port the assembly and resolution of the linear system to the GPU. This is the most costly stage of the solver. For this, we could use linear algebra libraries already ported to GPUs, such as Gingko [4] or Petsc [14], or even Kokkos kernels [129]. However, we will have to be very careful about memory movements.

Appendix A

Spherical harmonics

This appendix completes Section 2.3.1 with useful properties on Legendre polynomials, associated Legendre functions and spherical harmonics. For a closer look at the elements presented here, we refer to [90, 12].

Outline of the current chapter

A.1 Legendre polynomials

Definition A.1.1. *The Legendre polynomial of degree n, P_n are defined by,*

$$P_0(\mu) = 1, \tag{A.1}$$

$$P_n(\mu) = \frac{1}{2^n n!}\frac{\mathrm{d}^n}{\mathrm{d}\mu^n}(\mu^2 - 1), \quad n \geqslant 1. \tag{A.2}$$

Figure A.1 shows Legendre polynomials up to degree 5.

Proposition A.1.1. *The Legendre polynomials have the following properties [90, 12].*

Parity: $P_n(\mu)$ is a real polynomial in μ of degree n and of the same parity as n,

$$P_n(-\mu) = (-1)^n P_n(\mu), \qquad \forall n \geqslant 0.$$

Let $\delta_{n,l}$ denote the Kronecker symbol. The Legendre polynomials are orthogonal on the interval $[-1, 1]$,

$$\int_{-1}^{1} P_n(\mu)P_l(\mu)\,\mathrm{d}\mu = \frac{2}{2n+1}\delta_{n,l}, \qquad \forall n, l \geqslant 0.$$

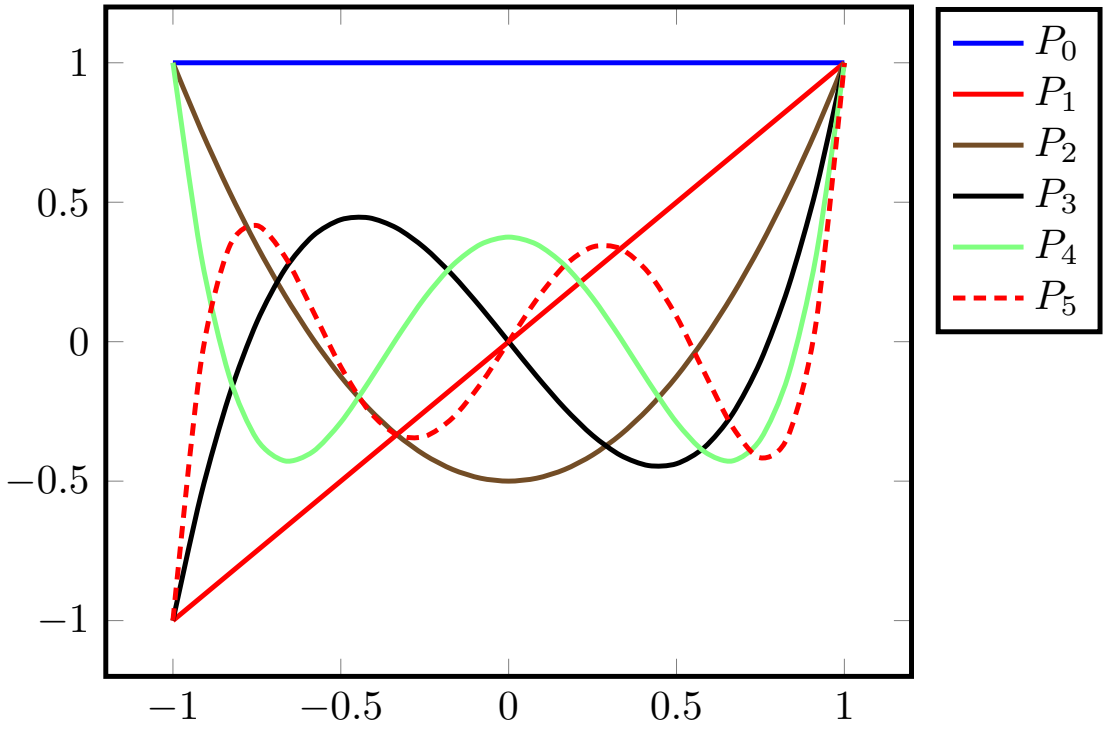


Figure A.1: Legendre polynomials up to degree 5.

Recursion formula,

$$(2n+1)\mu P_n(\mu) = (n+1)P_{n+1}(\mu) + nP_{n-1}(\mu), \qquad \forall n \geqslant 1.$$

This formula, along with $P_0(\mu) = 1$ and $P_1(\mu) = \mu$, can be used to obtain other Legendre polynomials recursively.

A.2 Associated Legendre functions

Definition A.2.1. *The associated Legendre functions are denoted P_n^m,*

$$P_n^0(\mu) = P_n(\mu), \qquad n \geqslant 0, \tag{A.3}$$

$$P_n^m(\mu) = (-1)^m(1-\mu)^{\frac{m}{2}}\frac{\mathrm{d}^m}{\mathrm{d}\mu^m}P_n(\mu), \qquad n \geqslant 0;\ m \in [\![1, n]\!]. \tag{A.4}$$

Proposition A.2.1. *The associated Legendre functions satisfy the following properties [90, 12].*

Parity,

$$P_n^m(-\mu) = (-1)^{n+m}P_n^m(\mu), \qquad \forall n, m \geqslant 0.$$

The associated Legendre functions are orthogonal on the interval $[-1, 1]$,

$$\int_{-1}^{1} P_n^m(\mu) P_l^m(\mu)\,\mathrm{d}\mu = \frac{2}{C_n^m}\delta_{n,l}, \qquad \forall n, m, l \geqslant 0,$$

with

$$C_n^m = (2n+1)\frac{(n-m)!}{(n+m)!}. \tag{A.5}$$

Recursion formula,

$$(2n+1)\mu P_n^m(\mu) = (n-m+1)P_{n+1}^m(\mu) + (n+m)P_{n-1}^m(\mu), \quad \forall n \geqslant 1;\ 0 \leqslant m \leqslant n.$$

A.3 List of spherical harmonics

Spherical harmonics are defined at Definition 2.3.1. Table A.1 shows the spherical harmonics up to degree $n = 2$.

Table A.1: List of spherical harmonics up to degree 2. We use the variable change $x = \sin\theta\cos\varphi$, $y = \sin\theta\sin\varphi$ and $z = \cos\theta$.

n\m	-2	-1	0	1	2
0			1		
1		$\sqrt{3}y$	$\sqrt{3}z$	$\sqrt{3}x$	
2	$\sqrt{15}xy$	$\sqrt{15}yz$	$3\sqrt{5}/2\ z^2$	$\sqrt{15}xz$	$\sqrt{15}/2(x^2-y^2)$

Appendix B

Error estimate addendum

This appendix complements Chapter 5. We describe the calculation of a source in Section B.1 and a numerical application in 1D, with a solution independent of the angular variable is presented in Section B.2.

Outline of the current chapter

B.1 Calculation of a source from a manufactured solution

The $L^2(E)$-scalar product is denoted by $(u, v)_E$. Algorithm 9 describes the calculation of the source q corresponding to a chosen solution u of angular order n_c. As described in Chapter 3 the polynomial basis φ_i^r is computed and stored in a local frame of reference for each cell, hence the need of the change of variables $\mathcal{T}$.

B.2 Another 1D example

One problem with a solution independent of the angular variable is considered here.

$$u(x, \mu) = u_0(x)y_0(\mu). \tag{B.1}$$

Indeed, $y_0(\mu) = 1$ and one chooses $u_0(x) = x(1 - x)$. It is necessary to use at least a P_1 approximation in angle in order to represent the source q.

The mesh is uniformly refined, and the error on the angular flux $\|u(x, \mu) - u_h(x, \mu)\|_{L^2(X)}$ and the error on the derivative of the angular flux $\|\mu\partial_x(u(x, \mu) - u_h(x, \mu))\|_{L^2(X)}$ are shows in Table B.1.

We obtain a convergence rate of order $\mathcal{O}(k + 1)$ for the angular flux in the L^2-norm (thus better by one order for $k \geqslant 1$ and by one half order for $k = 0$) and a

Algorithm 9 : Calculation of a source q from a manufactured solution u.

Input : $u(x,\omega) = \sum_{n=0}^{n=n_c} \sum_{m=-n}^{m=n} u_n^m(x) y_n^m(\omega)$

for $D_r \in D_h$ **do**
- $x_0 = \text{origin}(D_r)$
- Change of variables $\mathcal{T} : x \to x + x_0$
- **for** $n \in \{0, \dots, n_c\}$; $m \in \{-n, \dots, n\}$ **do**
  - $u_{n,r}^m(x) = u_n^m \circ \mathcal{T}(x)$
  - $\left(u_{n,r}^m\right)'(x) = \frac{\mathrm{d}u_{n,r}^m}{\mathrm{d}x}(x)$
- **for** $ii = (i,n,m) \in dof(u|_{D_r})$ **do**
  - $U[r,ii] = (\varphi_i^r, u_{n,r}^m)_{D_r}$ // store dof of u
  - $U_{dx}[r,ii] = (\varphi_i^r, \left(u_{n,r}^m\right)')_{D_r}$ // store dof of u'
- **for** $ii = (i,n,m) \in dof(u|_{D_r})$ **do**
  - **for** $jj = (j,l,k) \in dof(u|_{D_r})$ **do**
    - $q[r,ii] \mathrel{+}= \left\{U_{dx}[r,ii]\,(\omega_0 y_n^m, y_l^k)_{\mathbf{S}^2} + \sigma U[r,ii]\,(y_n^m, y_l^k)_{\mathbf{S}^2}\right\}(\varphi_i, \varphi_j)_{D_r}$

convergence rate of order $\mathcal{O}(k)$ for the derivative in the L^2-norm (exactly as predicted by our theoretical analysis). Moreover the error is zero (up to the GMRES tolerance) when the approximation order reaches the one of the manufactured solution.

Furthermore, in the $k = 1$ spatial approximation, $N = 1$ angular approximation case, we note that the error on the streamline derivative is divided almost exactly by 10, resulting in a perfect slope of 1. The exact solution being continuous and polynomial, a quick calculation shows that in this case, the error depends solely on an integral on its derivative. This integral over a polynomial is calculated exactly.

Number of elements	$\|u - u_h\|_{L^2(X)}$	Order of convergence in angular flux	$\|\mu\partial_x(u - u_h)\|_{L^2(X)}$	Order of convergence on derivative
$k = 0$ spatial approximation, $N = 1$ angular approximation				
2	3.868e-02	-	2.886e-01	-
20	3.288e-03	1.07	3.329e-01	-
200	3.140e-04	1.02	3.333e-01	-
2000	3.124e-05	1.00	3.333e-01	-
$k = 1$ spatial approximation, $N = 1$ angular approximation				
2	2.249e-02	-	1.667e-01	-
20	2.609e-04	1.94	1.666e-02	1.00
200	2.656e-06	1.99	1.666e-03	1.00
2000	2.661e-08	2.00	1.666e-04	1.00
$k = 2$ spatial approximation, $N = 1$ angular approximation				
2	6.218e-10	-	3.322e-09	-
20	9.357e-11	-	3.504e-09	-
200	7.728e-12	-	3.520e-09	-
2000	2.952e-12	-	3.523e-09	-

Table B.1: Convergence orders for the angular flux and its derivative for the 1D problem with the manufactured solution (B.1).

Bibliography

[1] R. T. Ackroyd. *Finite Element Methods for Particle Transport: Applications to Reactor and Radiation Physics*. Research Studies in Particle and Nuclear Technology 6. Taunton, Somerset, England : New York: Research Studies Press ; J. Wiley, 1997.

[2] G. Allaire. *Numerical Analysis and Optimization: An Introduction to Mathematical Modelling and Numerical Simulation*. Numerical Mathematics and Scientific Computation. Oxford: Oxford University Press, 2007.

[3] G. Allaire, X. Blanc, B. Després, and F. Golse. *Transport et diffusion*. Palaiseau: Editions de l'École polytechnique, 2018.

[4] H. Anzt et al. "Ginkgo : A Modern Linear Operator Algebra Framework for High Performance Computing". *ACM Transactions on Mathematical Software* 48.1 (Mar. 2022), pp. 1–33.

[5] D. N. Arnold, F. Brezzi, B. Cockburn, and L. D. Marini. "Unified Analysis of Discontinuous Galerkin Methods for Elliptic Problems". *SIAM Journal on Numerical Analysis* 39.5 (Jan. 2002), pp. 1749–1779.

[6] M. Asadzadeh. "Analysis of a Fully Discrete Scheme for Neutron Transport in Two-Dimensional Geometry". *SIAM Journal on Numerical Analysis* 23.3 (June 1986), pp. 543–561.

[7] M. Asadzadeh. "L2-Error Estimates for the Discrete Ordinates Method for Three-Dimensional Neutron Transport". *Transport Theory and Statistical Physics* 17 (1988), pp. 1–24.

[8] K. Assogba, G. Allaire, and L. Bourhrara. *Analysis of a Combined Spherical Harmonics and Discontinuous Galerkin Discretization for the Boltzmann Transport Equation*. (Hal-04196435). Aug. 2023.

[9] K. Assogba and L. Bourhrara. "The PN Form of the Neutron Transport Problem Achieves Linear Scalability Through Domain Decomposition". *Proceedings of International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2023)*. Niagara Falls, Ontario, Canada, Aug. 2023.

[10] K. Assogba, L. Bourhrara, I. Zmijarevic, and G. Allaire. "Precise 3D Reactor Core Calculation Using Spherical Harmonics and Discontinuous Galerkin Finite Element Methods". *Proceedings of International Conference on Physics of Reactors 2022 (PHYSOR 2022)*. Pittsburgh, PA, United States: American Nuclear Society, May 2022, pp. 1224–1233.

[11] K. Assogba, L. Bourhrara, I. Zmijarevic, G. Allaire, and A. Galia. "Spherical Harmonics and Discontinuous Galerkin Finite Element Methods for the Three-Dimensional Neutron Transport Equation: Application to Core and Lattice Calculation". *Nuclear Science and Engineering* 197.8 (Aug. 2023), pp. 1584–1599.

[12] K. E. Atkinson and W. Han. *Theoretical Numerical Analysis: A Functional Analysis Framework*. 2nd ed. Texts in Applied Mathematics 39. New York: Springer, 2005.

[13] G. Bal. "Couplage d'équations et homogénéisation en transport neutronique". These de doctorat. Paris 6, Jan. 1997.

[14] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. "Efficient Management of Parallelism in Object-Oriented Numerical Software Libraries". *Modern Software Tools for Scientific Computing*. Ed. by E. Arge, A. M. Bruaset, and H. P. Langtangen. Boston, MA: Birkhäuser, 1997, pp. 163–202.

[15] Y. S. Ban, E. Masiello, R. Lenain, H. G. Joo, and R. Sanchez. "Code-to-Code Comparisons on Spatial Solution Capabilities and Performances between nTRACER and the Standalone IDT Solver of APOLLO3®". *Annals of Nuclear Energy* 115 (May 2018), pp. 573–594.

[16] A.-M. Baudron, J. J. Lautard, Y. Maday, and O. Mula. "MINARET: Towards a Time-Dependent Neutron Transport Parallel Solver". *SNA + MC 2013 - Joint International Conference on Supercomputing in Nuclear Applications + Monte Carlo*. EDP Sciences, 2014, p. 04103.

[17] A.-M. Baudron and J.-J. Lautard. "MINOS: A Simplified Pn Solver for Core Calculation". *Nuclear Science and Engineering* 155.2 (Feb. 2007), pp. 250–263.

[18] G. I. Bell and S. Glasstone. *Nuclear Reactor Theory*. New York, United States: Van Nostrand Reinhold Company, 1970.

[19] A. S. Bielen. *Spherical Harmonics Solutions to Second Order Forms of the Boltzmann Transport Equation Using Particle Transport Code SCEPTRE*. Tech. rep. Thesis Master of Science, Pennsylvania State University. Departement of Mechanical and Nuclear Engineering., 2008.

[20] L. Bourhrara. "A New Numerical Method for Solving the Boltzmann Transport Equation Using the PN Method and the Discontinuous Finite Elements on Unstructured and Curved Meshes". *Journal of Computational Physics* 397 (Nov. 2019), p. 108801.

[21] L. Bourhrara. "H1 Approximations of the Neutron Transport Equation and Associated Diffusion Equations". *Transport Theory and Statistical Physics* 35.3-4 (Aug. 2006), pp. 89–108.

[22] L. Bourhrara. "New Variational Formulations for the Neutron Transport Equation". *Transport Theory and Statistical Physics* 33.2 (Jan. 2004), pp. 93–124.

[23] L. Bourhrara. "W N Approximations Of Neutron Transport Equation". *Transport Theory and Statistical Physics* 38.4 (Oct. 2009), pp. 195–227.

[24] F. Brezzi, L. D. Marini, and E. Süli. "Discontinuous Galerkin Methods for First-Order Hyperbolic Problems". *Mathematical Models and Methods in Applied Sciences* 14.12 (Dec. 2004), pp. 1893–1903.

[25] D. A. Brown et al. “ENDF/B-VIII.0: The 8th Major Release of the Nuclear Reaction Data Library with CIELO-project Cross Sections, New Standards and Thermal Scattering Data”. *Nuclear Data Sheets*. Special Issue on Nuclear Reaction Data 148 (Feb. 2018), pp. 1–142.

[26] E. Brun et al. “TRIPOLI-4®, CEA, EDF and AREVA Reference Monte Carlo Code”. *Annals of Nuclear Energy* 82 (Aug. 2015), pp. 151–160.

[27] A. G. Buchan, C. C. Pain, M. D. Eaton, R. P. Smedley-Stevenson, and A. J. H. Goddard. “Linear and Quadratic Octahedral Wavelets on the Sphere for Angular Discretisations of the Boltzmann Transport Equation”. *Annals of Nuclear Energy* 32.11 (July 2005), pp. 1224–1273.

[28] C. B. Carrico, E. E. Lewis, and G. Palmiotti. “Three Dimensional Variational Nodal Transport Methods for Cartesian, Triangular, and Hexagonal Criticality Calculations”. *Nuclear Science and Engineering* 111.2 (1992), pp. 168–179.

[29] CEA. *La neutronique*. Le Moniteur. Monographies de la DEN. Gif-sur-Yvette, 2013.

[30] M. Cessenat. “Théorèmes de trace Lp pour des espaces de fonctions de la neutronique”. *C. R. Acad. Sci. Paris* 299.16 (1984), pp. 831–834.

[31] M. Cessenat. “Théorèmes de trace pour des espaces de fonctions de la neutronique”. *C. R. Acad. Sci. Paris* 300.1 (1985), pp. 89–92.

[32] S. Chandrasekhar. “On the Radiative Equilibrium of a Stellar Atmosphere. II.” *The Astrophysical Journal* 100 (1944), p. 76.

[33] C. Chevalier and F. Pellegrini. “PT-Scotch: A Tool for Efficient Parallel Graph Ordering”. *Parallel Computing*. Parallel Matrix Algorithms and Applications 34.6 (July 2008), pp. 318–331.

[34] P. Ciarlet. “T-Coercivity: Application to the Discretization of Helmholtz-like Problems”. *Computers & Mathematics with Applications* 64.1 (July 2012), pp. 22–34.

[35] P. Ciarlet, M. H. Do, and F. Madiot. “A Posteriori Error Estimates for Mixed Finite Element Discretizations of the Neutron Diffusion Equations”. *ESAIM: Mathematical Modelling and Numerical Analysis* 57.1 (Jan. 2023), pp. 1–27.

[36] P. G. Ciarlet. *The Finite Element Method for Elliptic Problems*. Society for Industrial and Applied Mathematics, Apr. 2002.

[37] B. Cockburn, G. E. Karniadakis, and C.-W. Shu. *Discontinuous Galerkin Methods: Theory, Computation and Applications*. Springer Science & Business Media, Dec. 2012.

[38] P. Coppolani, N. Hassenboelher, J. Joseph, J.-F. Petetrot, J.-P. Py, and J.-S. Zampa. *La chaudière des réacteurs à eau sous pression*. EDP SCIENCES, Nov. 2004.

[39] M. Coste-Delclaux. “Modélisation Du Phénomène d’autoprotection Dans Le Code de Transport Multigroupe APOLLO2”. These de Doctorat. Paris, CNAM, Jan. 2006.

[40] W. Dahmen, F. Gruber, and O. Mula. “An Adaptive Nested Source Term Iteration for Radiative Transfer Equations”. *Mathematics of Computation* 89.324 (Jan. 2020), pp. 1605–1646.

[41] W. Dahmen and O. Mula. *Accuracy Controlled Schemes for the Eigenvalue Problem of the Radiative Transfer Equation*. July 2023. arXiv: `2307.07780 [cs, math]`.

[42] R. Dautray and J.-L. Lions. *Mathematical Analysis and Numerical Methods for Science and Technology*. Vol. 6. Berlin, Heidelberg: Springer Berlin Heidelberg, 1988.

[43] B. Davison. “On the Rate of Convergence of the Spherical Harmonics Method: (For the Plane Case, Isotropic Scattering)”. *Canadian Journal of Physics* 38.11 (Nov. 1960), pp. 1526–1545.

[44] B. Davison. “Spherical-Harmonics Method for Neutron Transport Theory Problems with Incomplete Symmetry”. *Canadian Journal of Physics* 36.4 (Apr. 1958), pp. 462–475.

[45] C. R. E. de Oliveira, C. C. Pain, and A. J. H. Goddard. “Parallel Domain Decomposition Methods for Large-Scale Finite Element Transport Modelling”. *Int. Conf. Math. Comp. Reactor Physics and Environmental Analysis of Nuclear System*. Portland, Oregon: American Nuclear Society, Dec. 1995.

[46] D. A. Di Pietro and A. Ern. *Mathematical Aspects of Discontinuous Galerkin Methods*. Mathématiques et Applications 69. Berlin ; New York: Springer, 2012.

[47] V. Dolean, P. Jolivet, and F. Nataf. *An Introduction to Domain Decomposition Methods: Algorithms, Theory, and Parallel Implementation*. SIAM, 2015.

[48] J. Duderstadt and W. Martin. *Transport Theory*. New York: John Wiley and Sons, 1979.

[49] A. Ern and J. L. Guermond. “Discontinuous Galerkin Methods for Friedrichs’ Systems. I. General Theory”. *SIAM Journal on Numerical Analysis* 44.2 (Jan. 2006), pp. 753–778.

[50] A. Ern and J.-L. Guermond. *Theory and Practice of Finite Elements*. Ed. by S. S. Antman, J. E. Marsden, and L. Sirovich. Vol. 159. Applied Mathematical Sciences. New York: Springer New York, 2004.

[51] A. Ern, M. Vohralík, and M. Zakerzadeh. “Guaranteed and Robust L^2 -Norm *a Posteriori* Error Estimates for 1D Linear Advection Problems”. *ESAIM: Mathematical Modelling and Numerical Analysis* 55 (2021), S447–S474.

[52] T. M. Evans, A. S. Stafford, R. N. Slaybaugh, and K. T. Clarno. “Denovo: A New Three-Dimensional Parallel Discrete Ordinates Code in SCALE”. *Nuclear Technology* 171.2 (Aug. 2010), pp. 171–200.

[53] B. Fontaine et al. “Description and Preliminary Results of PHENIX Core Flowering Test”. *Nuclear Engineering and Design* 241.10 (Oct. 2011), pp. 4143–4151.

[54] D. Fournier, R. Herbin, and R. Le Tellier. “Discontinuous Galerkin Discretization and Hp-Refinement for the Resolution of the Neutron Transport Equation”. *SIAM Journal on Scientific Computing* 35 (Mar. 2013), A936–A956.

[55] E. Gabriel et al. “Open MPI: Goals, Concept, and Design of a Next Generation MPI Implementation”. *Recent Advances in Parallel Virtual Machine and Message Passing Interface*. Ed. by D. Kranzlmüller, P. Kacsuk, and J. Dongarra. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2004, pp. 97–104.

[56] A. Gammicchia, S. Santandrea, and S. Dulla. “Cross Sections Polynomial Axial Expansion within the APOLLO3® 3D Characteristics Method”. *Annals of Nuclear Energy* 165 (Jan. 2022), p. 108673.

[57] E. M. Gelbard. *Applications of Spherical Harmonics Method to Reactor Problems*. Tech. rep. WAPD-BT-20. Westinghouse Electric Corp. Bettis Atomic Power Lab, Sept. 1960.

[58] E. M. Gelbard. *Applications Of The Simplified Spherical Harmonics Equations In Spherical Geometry*. Tech. rep. WAPD-TM-294. Westinghouse Electric Corp. Bettis Atomic Power Lab, Apr. 1962.

[59] E. M. Gelbard. *Simplified Spherical Harmonics Equations And Their Use In Shielding Problems*. Tech. rep. WAPD-T-1182(Rev.1). Westinghouse Electric Corp. Bettis Atomic Power Lab, Feb. 1961.

[60] D. Gregor, T. Hoefler, B. Barrett, and A. Lumsdaine. *Fixing Probe for Multi-Threaded MPI Applications (Revision 4)*. Tech. rep. Technical report, Indiana University (January 2009), 2009.

[61] K. Grella and C. Schwab. “Sparse Tensor Spherical Harmonics Approximation in Radiative Transfer”. *Journal of Computational Physics* 230.23 (Sept. 2011), pp. 8452–8473.

[62] K. Grella. “Sparse Tensor Phase Space Galerkin Approximation for Radiative Transport”. *SpringerPlus* 3.1 (Dec. 2014), p. 230.

[63] T. H. Gronwall. “On the Degree of Convergence of Laplace’s Series”. *Transactions of the American Mathematical Society* 15.1 (1914), pp. 1–30.

[64] W. Gropp, E. Lusk, N. Doss, and A. Skjellum. “A High-Performance, Portable Implementation of the MPI Message Passing Interface Standard”. *Parallel Computing* 22.6 (Sept. 1996), pp. 789–828.

[65] W. Gropp, E. Lusk, and A. Skjellum. *Using MPI: Portable Parallel Programming with the Message-passing Interface*. MIT Press, 1999.

[66] J.-L. Guermond and G. Kanschat. “Asymptotic Analysis of Upwind Discontinuous Galerkin Approximation of the Radiative Transport Equation in the Diffusive Limit”. *SIAM Journal on Numerical Analysis* 48.1 (Jan. 2010), pp. 53–78.

[67] J.-L. Guermond, G. Kanschat, and J. C. Ragusa. “Discontinuous Galerkin for the Radiative Transport Equation”. *Recent Developments in Discontinuous Galerkin Finite Element Methods for Partial Differential Equations: 2012 John H Barrett Memorial Lectures*. Ed. by X. Feng, O. Karakashian, and Y. Xing. The IMA Volumes in Mathematics and Its Applications. Springer International Publishing, 2014, pp. 181–193.

[68] E. Hall, P. Houston, and S. Murphy. “Hp-Adaptive Discontinuous Galerkin Methods for Neutron Transport Criticality Problems”. *SIAM Journal on Scientific Computing* 39.5 (Jan. 2017), B916–B942.

[69] A. Hébert. *Applied Reactor Physics*. Presses internationales Polytechnique, 2016.

[70] J. S. Hesthaven and T. Warburton. *Nodal Discontinuous Galerkin Methods: Algorithms, Analysis, and Applications*. Springer Science & Business Media, Dec. 2007.

[71] N. Hfaiedh and A. Santamarina. "Determination of the Optimized SHEM Mesh for Neutron Transport Calculations" (July 2005).

[72] D. Hilbert. "Ueber die stetige Abbildung einer Line auf ein Flächenstück". *Mathematische Annalen* 38.3 (Sept. 1891), pp. 459–460.

[73] T. Hoefler, G. Bronevetsky, B. Barrett, B. R. de Supinski, and A. Lumsdaine. "Efficient MPI Support for Advanced Hybrid Programming Models". *Recent Advances in the Message Passing Interface*. Ed. by R. Keller, E. Gabriel, M. Resch, and J. Dongarra. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2010, pp. 50–61.

[74] E. Jamelot and P. Ciarlet. "Fast Non-Overlapping Schwarz Domain Decomposition Methods for Solving the Neutron Diffusion Equation". *Journal of Computational Physics* 241 (May 2013), pp. 445–463.

[75] C. Johnson and J. Pitkäranta. "An Analysis of the Discontinuous Galerkin Method for a Scalar Hyperbolic Equation". *Mathematics of Computation* 46.173 (1986), pp. 1–26.

[76] C. Johnson. *Numerical Solution of Partial Differential Equations by the Finite Element Method*. Cambridge University Press. Cambridge: Cambridge University Press, 1987.

[77] C. Johnson, U. Nävert, and J. Pitkäranta. "Finite Element Methods for Linear Hyperbolic Problems". *Computer Methods in Applied Mechanics and Engineering* 45.1 (Sept. 1984), pp. 285–312.

[78] C. Johnson and J. Pitkäranta. "Convergence of a Fully Discrete Scheme for Two-Dimensional Neutron Transport". *SIAM Journal on Numerical Analysis* 20.5 (Oct. 1983), pp. 951–966.

[79] G. Karypis and V. Kumar. "A Fast and High Quality Multilevel Scheme for Partitioning Irregular Graphs". *SIAM Journal of Scientific Computing* 20.1 (1998), pp. 359–392.

[80] K. Koch, R. Baker, and R. Alcouffe. "Solution of the First-Order Form of the Three-Dimensional Discrete Ordinates Equation on a Massively Parallel Machine". *Trans. Am. Nucl. Soc.* 65.198 (1992).

[81] K. D. Lathrop. "Ray Effects in Discrete Ordinates Equations". *Nuclear Science and Engineering* 32.3 (June 1968), pp. 357–369.

[82] J.-J. Lautard, Y. Maday, and O. Mula. "MINARET or the Quest towards the Use of Time-Dependent Neutron Transport Solvers for Nuclear Core Calculations on a Regular Basis". *SNA + MC 2013-Joint International Conference on Supercomputing in Nuclear Applications + Monte Carlo*. 2013.

[83] J.-J. Lautard and J.-Y. Moller. *MINARET, a Deterministic Neutron Transport Solver for Nuclear Core Calculations*. Dec. 2010.

[84] R. Lenain, E. Masiello, F. Damian, and R. Sanchez. “Domain Decomposition Method for 2D and 3D Transport Calculations Using Hybrid MPI/OPENMP Parallelism”. *MC2015 - Joint International Conference on Mathematics and Computation (M&C), Supercomputing in Nuclear Applications (SNA) and the Monte Carlo (MC) Method*. ANS, 2015.

[85] P. Lesaint. “Sur la résolution des systèmes hyperboliques du premier ordre par des méthodes d’éléments finis”. PhD thesis. Université Paris VI, 1975.

[86] P. Lesaint and P. A. Raviart. “On a Finite Element Method for Solving the Neutron Transport Equation”. *Mathematical Aspects of Finite Elements in Partial Differential Equations*. Ed. by C. de Boor. Academic Press, Jan. 1974, pp. 89–123.

[87] E. E. Lewis and I. Dilber. “Finite Element, Nodal and Response Matrix Methods: A Variational Synthesis for Neutron Transport”. *Progress in Nuclear Energy*. Finite Element and Allied Methods 18.1 (Jan. 1986), pp. 63–74.

[88] E. E. Lewis and W. F. Miller. *Computational Methods of Neutron Transport*. New York: Wiley, 1984.

[89] E. E. Lewis, G. Palmiotti, T. A. Taiwo, R. N. Blomquist, M. A. Smith, and N. Tsoulfanidis. “Benchmark Specifications for Deterministic MOX Fuel Assembly Transport Calculations without Spatial Homogenization (3-D Extension C5G7 MOX)”. *OECD’s Nuclear Energy Agency* 6 (2003).

[90] T. M. MacRobert and I. N. Sneddon. *Spherical Harmonics: An Elementary Treatise on Harmonic Functions with Applications*. Oxford: Pergamon, 1967.

[91] T. A. Manteuffel and K. J. Ressel. “Least-Squares Finite-Element Solution of the Neutron Transport Equation in Diffusive Regimes”. *SIAM Journal on Numerical Analysis* 35.2 (Jan. 1998), pp. 806–835.

[92] S. Marguet. *The Physics of Nuclear Reactors*. Cham: Springer International Publishing, 2017.

[93] J. C. Mark. *The Spherical Harmonic Method*. National Research Council of Canada. Chalk River, Ont.: National Research Council of Canada, Atomic Energy Project, Division of Research, 1947.

[94] R. E. Marshak. “Note on the Spherical Harmonic Method As Applied to the Milne Problem for a Sphere”. *Physical Review* 71.7 (Apr. 1947), pp. 443–446.

[95] E. Masiello, R. Lenain, and W. Ford. “3D Heterogeneous Cartesian Cells for Transport-Based Core Simulations”. *Annals of Nuclear Energy* 142 (July 2020), p. 107364.

[96] E. Masiello, R. Sanchez, and I. Zmijarevic. “New Numerical Solution with the Method of Short Characteristics for 2-D Heterogeneous Cartesian Cells in the APOLLO2 Code: Numerical Analysis and Tests”. *Nuclear Science and Engineering* 161.3 (Mar. 2009), pp. 257–278.

[97] J. E. Morel and J. M. McGhee. “A Self-Adjoint Angular Flux Equation”. *Nuclear Science and Engineering* 132.3 (July 1999), pp. 312–325.

[98] P. Mosca et al. “APOLLO3®: Overview of the New Code Capabilities for Reactor Physics Analysis”. *To Appear in Proceedings of International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C 2023)*. 2023.

[99] MPI. *MPI: A Message-Passing Interface Standard*. Technical Report. USA: University of Tennessee, 1994.

[100] O. Mula Hernandez. “Quelques contributions vers la simulation parallèle de la cinétique neutronique et la prise en compte de données observées en temps réel”. These de doctorat. Paris 6, Sept. 2014.

[101] S. Murphy. “Methods for Solving Discontinuous-Galerkin Finite Element Equations with Application to Neutron Transport”. PhD thesis. Aug. 2015.

[102] N. Odry, J.-J. Lautard, J.-F. Vidal, and G. Rimpault. “Coarse Mesh Rebalance Acceleration Applied to an Iterative Domain Decomposition Method on Unstructured Mesh”. *Nuclear Science and Engineering* 187.3 (Sept. 2017), pp. 240–253.

[103] N. Odry, J.-F. Vidal, G. Rimpault, J.-J. Lautard, and A.-M. Baudron. “Performance Study of a Parallel Domain Decomposition Method”. *PHYSOR 2016 - International Conference on the Physics of Reactors: Unifying Theory and Experiments in the 21st Century*. May 2016.

[104] J.-M. Palau et al. “Recent Advances in the V and V of the New French Cea APOLLO3"® Neutron Transport Code. Benchmarks Analysis of the Flux Solvers”. JAEA-CONF–2014-003 (2015).

[105] G. Peano. “Sur une courbe, qui remplit toute une aire plane”. *Arbeiten zur Analysis und zur mathematischen Logik*. Ed. by G. Peano. Teubner-Archiv zur Mathematik. Vienna: Springer, 1990, pp. 71–75.

[106] M. Pérache, H. Jourdren, and R. Namyst. “MPC: A Unified Parallel Runtime for Clusters of NUMA Machines”. *Proceedings of the 14th International Euro-Par Conference on Parallel Processing*. Euro-Par ’08. Las Palmas de Gran Canaria, Spain: Springer-Verlag, 2008, pp. 78–88.

[107] J. Pitkäranta and L. R. Scott. “Error Estimates for the Combined Spatial and Angular Approximations of the Transport Equation for Slab Geometry”. *SIAM Journal on Numerical Analysis* 20.5 (Oct. 1983), pp. 922–950.

[108] J. Planchard. *Méthodes Mathématiques En Neutronique*. Eyrolles, 1995.

[109] G. C. Pomraning. “Asymptotic and Variational Derivations of the Simplified PN Equations”. *Annals of Nuclear Energy* 20.9 (Sept. 1993), pp. 623–637.

[110] D. L. Ragozin. “Constructive Polynomial Approximation on Spheres and Projective Spaces.” *Transactions of the American Mathematical Society* 162 (1971), pp. 157–170.

[111] J. C. Ragusa, J. L. Guermond, and G. Kanschat. “A Robust SN-DG-approximation for Radiation Transport in Optically Thick and Diffusive Regimes”. *Journal of Computational Physics* 231.4 (Feb. 2012), pp. 1947–1962.

[112] W. Reed and T. Hill. *Triangular Mesh Methods for the Neutron Transport Equation*. Technical Report LA-UR-73-479. United States: Los Alamos Scientific Laboratory, 1973, p. 23.

[113] K. J. Ressel. “Least-Squares Finite-Element Solution of the Neutron Transport Equation in Diffusive Regimes”. PhD thesis. University of Colorado at Denver, 1994.

[114] P. Reuss. *Précis de neutronique*. EDP Sciences, 2003.

[115] P. Reuss and J. Bussac. *Traité de neutronique*. Paris: Hermann, 1978.

[116] G. R. Richter. “An Optimal-Order Error Estimate for the Discontinuous Galerkin Method”. *Mathematics of Computation* 50.181 (1988), pp. 75–88.

[117] E. Rosier. “Study, Development and Evaluation of the Subgroup Methods Based on the Physical Probability Tables in APOLLO3® for Thermal Reactor Calculations”. These de Doctorat. université Paris-Saclay, Dec. 2022.

[118] Y. Saad and M. H. Schultz. “GMRES: A Generalized Minimal Residual Algorithm for Solving Nonsymmetric Linear Systems”. *SIAM Journal on Scientific and Statistical Computing* 7.3 (July 1986), pp. 856–869.

[119] R. Sanchez and N. J. McCormick. “A Review of Neutron Transport Approximations”. *Nuclear Science and Engineering* 80.4 (1982), pp. 481–535.

[120] R. Sanchez. “On PN Interface and Boundary Conditions”. *Nuclear Science and Engineering* 177.1 (May 2014), pp. 19–34.

[121] S. Santandrea and R. Sanchez. “Analysis and Improvements of the DPN Acceleration Technique for the Method of Characteristics in Unstructured Meshes”. *Annals of Nuclear Energy* 32.2 (Jan. 2005), pp. 163–193.

[122] D. Schneider, F. Dolci, F. Gabriel, J.-M. Palau, M. Guillo, and B. Pothet. “APOLLO3® CEA/DEN Deterministic Multi-Purpose Code for Reactor Physics Analysis”. *PHYSOR 2016 – Unifying Theory and Experiments in the 21st Century*. Sun Valley, United States, May 2016.

[123] H. A. Schwarz. “Ueber einen Grenzübergang durch alternirendes Verfahren”. *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich* 15 (1870), pp. 272–286.

[124] D. Sciannandrone, S. Santandrea, and R. Sanchez. “Optimized Tracking Strategies for Step MOC Calculations in Extruded 3D Axial Geometries”. *Annals of Nuclear Energy*. Special Issue of The 3rd International Conference on Physics and Technology of Reactors and Application 87 (Jan. 2016), pp. 49–60.

[125] A. Seubert, W. Zwermann, and S. Langenbuch. “Solution of the C5G7 3-D Extension Benchmark by the SN Code TORT”. *Progress in Nuclear Energy*. OECD/NEA C5G7 MOX Benchmark: 3-D Extension Case 48.5 (July 2006), pp. 432–438.

[126] J. Stepanek, T. Auerbach, and W. Haelg. *Calculation of Four Thermal Reactor Benchmark Problems in X-Y Geometry*. Tech. rep. Switzerland, 1982, p. 22.

[127] T. Takeda and H. Ikeda. “3-D Neutron Transport Benchmarks”. *Journal of Nuclear Science and Technology* 28.7 (1991), pp. 656–669.

[128] D. Tomatis, F. Bidault, A. Bruneton, and Z. Stankovski. “Overview of SERMA’s Graphical User Interfaces for Lattice Transport Calculations”. *Energies* 15.4 (Jan. 2022), p. 1417.

[129] C. R. Trott et al. “Kokkos 3: Programming Model Extensions for the Exascale Era”. *IEEE Transactions on Parallel and Distributed Systems* 33.4 (Apr. 2022), pp. 805–817.

[130] S. Van Criekingen, E. E. Lewis, and R. Beauwens. “Mixed-Hybrid Transport Discretization Using Even and Odd PN Expansions”. *Nuclear Science and Engineering* 152.2 (Feb. 2006), pp. 149–163.

[131] S. Van Criekingen, F. Nataf, and P. Havé. “Parafish: A Parallel FE–PN Neutron Transport Solver Based on Domain Decomposition”. *Annals of Nuclear Energy* 38.1 (Jan. 2011), pp. 145–150.

[132] H. A. van der Vorst. “Bi-CGSTAB: A Fast and Smoothly Converging Variant of Bi-CG for the Solution of Nonsymmetric Linear Systems”. *SIAM Journal on Scientific and Statistical Computing* 13.2 (Mar. 1992), pp. 631–644.

[133] J. I. C. Vermaak, J. C. Ragusa, M. L. Adams, and J. E. Morel. “Massively Parallel Transport Sweeps on Meshes with Cyclic Dependencies”. *Journal of Computational Physics* 425 (Jan. 2021), p. 109892.

[134] Y. Wang and J. C. Ragusa. “A High-Order Discontinuous Galerkin Method for the SN Transport Equations on 2D Unstructured Triangular Meshes”. *Annals of Nuclear Energy* 36.7 (July 2009), pp. 931–939.

[135] Y. Wang and J. C. Ragusa. “Standard and Goal-Oriented Adaptive Mesh Refinement Applied to Radiation Transport on 2D Unstructured Triangular Meshes”. *Journal of Computational Physics* 230.3 (Feb. 2011), pp. 763–788.

[136] G. Widmer, R. Hiptmair, and C. Schwab. “Sparse Adaptive Finite Elements for Radiative Transfer”. *Journal of Computational Physics* 227.12 (June 2008), pp. 6071–6105.

[137] A. Yamamoto, T. Ikehara, T. Ito, and E. Saji. “Benchmark Problem Suite for Reactor Physics Study of LWR Next Generation Fuels”. *Journal of Nuclear Science and Technology* 39.8 (Aug. 2002), pp. 900–912.

ECOLE
DOCTORALE
DE MATHEMATIQUES
HADAMARD

Titre : Résolution numérique de l'équation du transport de neutrons par la méthode des harmoniques sphériques et une méthode de Galerkin discontinue

Mots clés : équation de transport de Boltzmann, harmoniques sphériques, éléments finis discontinus, maillages non structurés

Résumé : Dans cette thèse, nous étudions un schéma numérique pour la résolution de l'équation du transport de neutrons. Notre étude s'applique dans le domaine de la physique des réacteurs nucléaires pour la simulation numérique de cœurs de réacteurs et d'assemblages combustible. Le schéma numérique étudié est basé sur la méthode des harmoniques sphériques pour la variable angulaire et celle des éléments finis discontinus pour la variable spatiale. Ce schéma numérique permet de traiter les maillages non-structurés, non-conformes avec des faces 2D courbes, typiquement des cercles et des arcs de cercle. La prise en compte de maillages courbes permet de représenter de façon exacte la géométrie des crayons combustibles de réacteurs à eau pressurisée. Par ailleurs, tirant profit de la périodicité des motifs dans un cœur, les produits matrice-vecteur sont optimisés et parallélisés.
Le solveur de flux qui en résulte a un large éventail d'applications. Afin d'évaluer sa précision et ses performances, il a été appliqué à des calculs de cœurs de réacteurs et d'assemblages de combustible en une, deux et trois dimensions. Les solutions obtenues sont conformes aux standards de l'industrie nucléaire et les temps de calcul compétitifs, même dans le cas de géométries complexes de cœurs et d'assemblages.
D'autre part, nous prouvons la convergence et fournissons des estimations d'erreur de ce schéma numérique.
Enfin, nous développons une méthode de décomposition de domaine afin d'effectuer des calculs parallèles en mémoire distribuée. Les tests numériques menés montrent que la méthode permet d'obtenir un passage à l'échelle quasi-linéaire. Nous maintenons une efficacité en scalabilité forte de 100 % jusqu'à 4096 cœurs de calcul et de 80 % jusqu'à 8192 cœurs. Le solveur est en outre robuste, les variations sur les solutions obtenues par rapport au solveur séquentiel étant proches de la précision machine.

Title : On a numerical scheme to solve the neutron transport equation using spherical harmonics and a discontinuous Galerkin method

Keywords : Boltzmann transport equation, spherical harmonics, discontinuous finite element, unstructured meshes

Abstract : In this thesis, we carry out the mathematical analysis and performance study of a numerical scheme for solving the Boltzmann model of neutral particle transport. Our study applies in the field of nuclear reactor physics to the numerical simulation of reactor cores and fuel assemblies. The studied numerical scheme is based on combined spherical harmonics and discontinuous finite element methods, respectively for the angular and space variables. It handles a large class of unstructured prismatic meshes, allowing all the geometries describing fuel cells to be processed without any simplification or homogenization. Taking advantage of the similarity of some cells, our matrix-vector product algorithm is highly optimized and parallelized.
The resulting transport numerical solver has a wide range of applications, it can be used for a core calculation as well as for a precise 281 energy groups lattice calculation accounting for anisotropic scattering. To assess the accuracy and performance of this numerical scheme, it was applied to one, two and three dimensional reactor core and fuel assembly calculations. In practice it produces accurate solutions even in the case of complex core and assemblies geometries.
On the other hand, we prove the convergence and provide error estimates of this numerical scheme.
Finally, we leverage the benefits of the discretization to wrap the proposed solver, in a domain decomposition framework. We conduct robustness, strong, and weak scalability experiments on a petaflop cluster. We reach and maintain a strong scaling efficiency of 100 % up to 4096 CPU cores and 80 % up to 8192 CPU cores. Moreover, the parallel solver is robust, with variations in the obtained solutions compared to the sequential solver being close to the machine precision.

Institut Polytechnique de Paris
91120 Palaiseau, France