Development of a hybrid methodology coupling deterministic and stochastic neutronic calculations : Applications to heterogeneous PWR and SFR calculations

Fiona Desplats

THÈSE

Pour obtenir le grade de

DOCTEUR DE L'UNIVERSITÉ GRENOBLE ALPES

École doctorale : I-MEP2 - Ingénierie - Matériaux, Mécanique, Environnement, Energétique, Procédés, Production
Spécialité : MEP : Mécanique des fluides Energétique, Procédés
Unité de recherche : CEA Cadarache

Développement d'une méthode hybride couplant des calculs neutroniques déterministes et stochastiques : applications aux calculs de curs hétérogènes de type REP et RNR

Development of a hybrid methodology coupling deterministic and stochastic neutronic calculations: Applications to heterogeneous PWR and SFR calculations

Présentée par :

Fiona DESPLATS

Direction de thèse :

Jean-François VIDAL Directeur de Recherche CEA-E6, Université Grenoble Alpes	Directeur de thèse
Jean-Marc PALAU	Co-encadrant de thèse
Pascal ARCHIER	Co-encadrant de thèse

Rapporteurs :

Tony LELIEVRE
Ingénieur des Ponts et Chaussées, Ecole des Ponts ParisTech
Oscar CABELLOS
PROFESSEUR, Universidad Politécnica de Madrid

Thèse soutenue publiquement le **12 octobre 2022**, devant le jury composé de :

Tony LELIEVRE Ingénieur des Ponts et Chaussées, Ecole des Ponts ParisTech	Rapporteur
Oscar CABELLOS PROFESSEUR, Universidad Politécnica de Madrid	Rapporteur
Adrien BIDAUD PROFESSEUR DES UNIVERSITES, Grenoble INP	Président
Alain HEBERT PROFESSEUR, Ecole Polytechnique de Montréal	Examinateur
Alain MAZZOLO INGENIEUR HDR, CEA centre de Paris-Saclay	Examinateur
Claire VAGLIO-GAUDARD INGENIEUR HDR, CEA centre de Cadarache	Examinatrice

Invités :

Pascal Archier
INGENIEUR DOCTEUR, CEA Cadarache
Jean-Marc Palau
INGENIEUR HDR, Université Aix-Marseille

ACKNOWLEDGEMENTS

I would like to start by expressing my gratitude to Dr. Pascal Archier, Dr. Jean-Marc Palau, and Dr. Jean-François Vidal for creating this PhD proposal and for their support over the past three years. I want to thank all three of them for trusting me with this subject and for their advice on nuclear physics and simulation methods. Pascal, thank you for being available to discuss anything that I was unsure about.

I would also like to thank the adjudicators, and particularly the reviewers Tony Lelièvre and Oscar Cabellos. Thank you for taking part in the defense, and for the time spent reviewing the manuscript.

A big thank you to Véronique Bellanger-Villard for welcoming me to the lab and for always keeping her door open. I also thank my other lab mates without whom my experience at SPRC would not have been as enriching. Pierre T for your help with stochastic methods and general discussions on numerical methods. Elias and Giorgio who took the time to help teach me stochastic and deterministic methodologies. I will always remember all my fellow lab mates for the fun time we spent together, for the stimulating discussions, and for being a great bunch of people both in and out of the lab. Thanks to those I have not named directly for the good times we had together.

I would like to thank my friends, lab mates, colleagues and research team for helping me also see the human adventure of the last three years. Elias, Martin, Giorgio, Fanny, Kevin, Alex, Axel, Pierre, and Valentin, to name but a few. Pierre and Valentin, we started and finished our PhDs together; it was a real pleasure to share this adventure with you and to help each other face some of the highs and lows of the last three years. Thank you all for the laughter, friendship, and for contributing to such a positive experience. I want to wish good luck to the next bunch of students, François, Sidi, Maxime, Claire, you're going to do great!

Thank you to all of my friends, for being there for me and for all the adventures we have had together.

Maxime, a huge thanks for all of your support, your patience, and for making me take a break, discover this part of France and enjoy nature.

Finally, I am grateful to my parents and my siblings for their moral support, encouragement, and motivation to accomplish my goals. They have always supported me, believed in me, and made me who I am today.

ABSTRACT

Demonstrating the viability and safety of a project is an essential step in the licensing process of new nuclear plants. Simulation makes it possible to explore and predict how complex core concepts will behave in different situations. Improving the performances of numerical methods is essential for innovation.

Methods to model nuclear reactor cores are split into deterministic and stochastic methods. Deterministic methods discretize the phase space to solve the neutron transport equation. Approximations are an integral part of this method. Stochastic methods rely on random sampling of the neutron population. Their probabilistic nature can hinder their calculation costs. While at first, it might seem that stochastic methods are reference methods and deterministic methods allow for quicker but less precise calculations there are situations where neither method has led to good results. To take advantage of both methods and overcome some of their respective limitations, this work focuses on the development of numerical methods coupling both.

IDT is a discrete ordinates deterministic solver developed within the APOLLO3® platform with a non-overlapping domain decomposition method (DDM) implemented. LAST is a stochastic mini-app. The goal is to develop a method, making it possible to differentiate the treatment of the spatial zone from the rest of the problem, to obtain fine information on local neutronic quantities of interest and a better description of local heterogeneities.

To move towards a coupled approach, two different hybrid methods were developed and studied. The first is an energetic deterministic-deterministic hybrid method. This method is a first step towards the development of a hybrid method using DDM. IDT's DDM scheme was modified to accommodate different energy grids. Treating subdomains with different energy grids allows the user to treat resonances for each subdomain with refined energy meshes and/or advanced self-shielding methods. This method leads to more accurate results compared to a coarse-energy mesh calculation for shorter calculation times than the fine-energy mesh calculation.

The second method is a partially coupled deterministic-stochastic hybrid method. The boundary fluxes resulting from the deterministic code were used as input for the LAST simulation. The deterministic calculation is unaffected by the stochastic calculation. A first study is performed with the coarser and quicker IDT solver and then LAST is used on a small portion of the problem to obtain more precise data on a region of interest.

Keywords: Hybrid method, Spatial Domain Decomposition Method, Deterministic / Monte Carlo Coupling

RÉSUMÉ

La démonstration de la viabilité et de la sûreté d'un projet est une étape essentielle du processus de développement des nouvelles centrales nucléaires. La simulation permet d'explorer et de prévoir comment des concepts de réacteurs complexes se comporteraient dans différentes situations. L'amélioration des performances des méthodes numériques est essentielle à l'innovation.

Les méthodes de modélisation en neutronique se divisent en deux catégories, les méthodes déterministes et stochastiques. Les méthodes déterministes discrétisent l'espace des phases pour résoudre l'équation de transport des neutrons. Les approximations font partie intégrante de cette méthode. Les méthodes stochastiques reposent sur un échantillonnage aléatoire de la population de neutrons. Leur nature probabiliste peut nuire à leur coût de calcul. À première vue, il semble que les méthodes stochastiques soient des méthodes de référence, tandis que les méthodes déterministes permettent des calculs plus rapides, mais moins précis. Dans certaines situations, aucune des deux méthodes ne conduit à des résultats satisfaisants. Pour tirer parti des deux méthodes et surmonter certaines de leurs limites respectives, ce travail se concentre sur le développement de méthodes numériques couplant les deux.

IDT est un solveur déterministe aux ordonnées discrètes de la plate-forme de calcul APOLLO3® ayant déjà une méthode de décomposition de domaines (DDM) sans recouvrement implémentée. La maquette stochastique LAST a été utilisée pour implémenter cette méthode hybride. L'objectif est de réaliser des calculs de réacteurs où des zones spatiales sont singularisées à l'aide de méthodes de décomposition de domaines spatiale. Cette méthode permet de différencier le traitement de la zone spatiale du reste du problème, d'obtenir des informations fines sur des grandeurs neutroniques locales et une meilleure description des hétérogénéités locales.

Pour avancer vers une approche couplée, deux méthodes hybrides différentes ont été développées et étudiées. La première est une méthode hybride déterministe-déterministe énergétique. Cette méthode est une première étape dans le développement d'une méthode hybride utilisant la décomposition de domaine. Le schéma DDM d'IDT a été modifié pour prendre en compte différentes grilles d'énergie. Le traitement des sous-domaines avec différentes grilles d'énergie permet à l'utilisateur de traiter les résonances de chaque sous-domaine avec des mailles d'énergie raffinées et/ou des méthodes avancées d'auto-autoprotection. Cette méthode permet d'obtenir des résultats plus précis qu'avec un calcul à maillage énergétique grossier pour des temps de calcul plus courts qu'avec un calcul à maillage énergétique fin.

La deuxième méthode est une méthode hybride déterministe-stochastique partiellement couplée. Les flux aux limites résultant du code déterministe ont été utilisés comme entrée pour la simulation LAST. Le calcul déterministe n'est pas affecté par le calcul stochastique. Une première étude est réalisée avec le solveur IDT plus grossier et plus rapide, puis LAST est utilisé sur une petite partie du problème pour obtenir des données plus précises sur une région d'intérêt.

Mots clés : Méthode hybride, Décomposition de domaine spatiale, Couplage déterministe / Monte-Carlo

RÉSUMÉ ÉTENDU

Les méthodes numériques jouent un rôle majeur dans le développement de nouveaux concepts de réacteurs, mais aussi pour la prolongation de la durée de vie grâce aux démonstrations de sûreté produites. En raison du coût élevé de la construction d'une installation nucléaire, et des exigences de sécurité, il est nécessaire de pouvoir démontrer la viabilité d'un projet en utilisant des critères prédéfinis, comme la disponibilité des matériaux, la capacité de recyclage du combustible usé, la compétitivité économique, la durée de vie prévue, etc. En France, le gouvernement exige une démonstration de la sûreté du projet de centrale pour pouvoir autoriser un projet [1]. L'article 3.8 de [2] précise que la démonstration de sûreté doit utiliser des outils de calcul vérifiés.

Ces outils sont appelés "Outils de Calcul Scientifique" (OCS) et sont définis par l'Autorité de Sûreté Nucléaire (ASN), l'autorité française régulant la sûreté nucléaire et la radioprotection. L'ASN a défini un guide, le "guide 28" [3], dans lequel elle émet des recommandations concernant les outils de calcul afin de s'assurer qu'ils répondent aux besoins de l'exploitant de l'installation.

Il existe deux grandes familles de modélisation en neutronique : les méthodes déterministes et les méthodes stochastiques. Les méthodes déterministes discrétisent l'espace des phases pour résoudre l'équation du transport de neutrons. Des approximations et traitements sont nécessaires pour résoudre un problème. Un solveur déterministe correspond à une combinaison de traitements spatial, énergétique, et angulaire. Les différentes méthodes pouvant être utilisées sont présentées en Fig. 1.

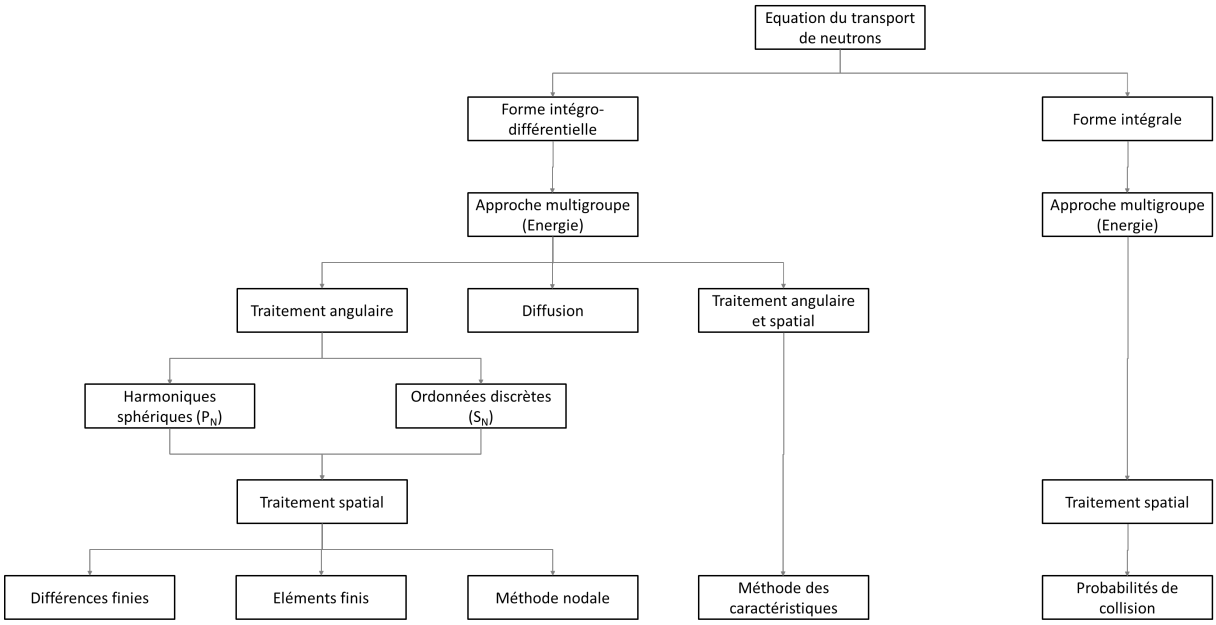


Fig. 1: Différentes méthodes de discrétisation pouvant être utilisées dans une simulation déterministe

Dans ce travail nous avons utilisé IDT. IDT est un solveur déterministe aux ordonnées discrètes de la plate-forme de calcul APOLLO3® ayant déjà une méthode de décomposition de domaines (DDM) sans recouvrement implémentée.

Les méthodes stochastiques utilisent peu d'approximation, mais reposent sur l'échantillonnage aléatoire de la population de neutron pour obtenir des grandeurs d'intérêts grâce à des estimateurs. Cependant, ces résultats sont entachés d'une incertitude statistique et peuvent demander de simuler beaucoup de particules pour être représentatives de la situation. Cette méthode est plus précise que les méthodes déterministes, et est utilisée comme référence, notamment pour la validation de nouveaux solveurs. La maquette stochastique LAST a été utilisée pour les calculs probabilistes.

Ces deux méthodes sont complémentaires, les méthodes stochastiques sont plus adaptées aux calculs de référence, mais les méthodes déterministes permettent de réaliser des calculs moins couteux et sont plus faciles à coupler avec des calculs thermomécaniques ou thermohydrauliques. Malgré cette complémentarité des deux branches de méthodes numériques, il y a des problèmes qui sont difficiles à

traiter avec les deux méthodes. Les expériences d'irradiation GEDEON [5] réalisées dans le réacteur MELUSINE [6] en sont un exemple. Ces expériences ont été conçues pour étudier l'UO2 empoisonné par le gadolinium dans les REP. Les cellules de gadolinium sont un mélange d'UO2 et de Gd2O3 et sont placées dans un réseau avec un rapport de modération de 1,8. Les prédictions déterministes sont moins bonnes à faible taux de combustion, en partie à cause du manque d'informations sur la position des barres de contrôle pendant l'irradiation. Le mouvement pendant l'irradiation augmente le coût d'une analyse Monte-Carlo complète de ces expériences.

Pour essayer de contourner les limitations respectives de chaque méthode, nous allons travailler sur une méthode hybride combinant celles-ci. Ayant des coûts de calculs plus faibles, la méthode déterministe serait utilisée pour réaliser le calcul global et la méthode stochastique pourrait ainsi se focaliser sur une zone d'intérêt. On pourrait ainsi avoir plus de précision sur une zone difficile à traiter en déterministe, sans devoir faire le calcul stochastique sur le problème complet.

Cette thèse explore les différents couplages déterministe-stochastique, et propose une méthode utilisant la décomposition de domaine spatiale déjà implémentée dans le solveur déterministe IDT. L'objectif est d'utiliser la décomposition de domaine spatiale pour singulariser une zone. Cette zone pourra donc être traitée avec la méthode stochastique, ce qui permettrait d'obtenir une information plus fine sur les grandeurs d'intérêt et une meilleure description des hétérogénéités spatiales. Le but de ce travail est d'implémenter cette méthode hybride.

Afin d'avoir une vision plus complète des méthodes déterministes et stochastiques actuelles et des problèmes à traiter, le comportement d'un neutron d'un cœur de réacteur sera étudié. Cette partie permettra de présenter les algorithmes de résolutions d'IDT et LAST. Ceci sera suivi par une analyse bibliographique des différentes méthodes hybrides existantes.

Les méthodes combinant deux méthodes déterministes (ou stochastiques) peuvent aussi être appelées hybrides. Ces méthodes permettent d'obtenir des résultats intéressants, mais peuvent toujours être sujettes aux limitations des méthodes déterministes (ou stochastiques) pures. Les hybrides déterministes-stochastiques peuvent rentrer dans trois catégories principales :

- Les techniques de réduction de variance. Ces méthodes biaisent la simulation stochastique en modifiant les poids des particules, ce qui permet de simuler moins de particules, tout en s'assurant des résultats satisfaisants autour du détecteur. Elles ont été étendues pour aussi traiter des problèmes globaux. Les problèmes de protection sont leur application principale.
- Les méthodes chaînées. On choisit de parler de chaînage puisque l'interaction entre les deux codes est unidirectionnelle, c'est-à-dire qu'un calcul fournit des informations au calcul suivant, mais il n'y a pas de retour. Ces méthodes peuvent être appliquées à des problèmes complexes, comme des études de protection dans des installations nucléaires. Un calcul déterministe peut réaliser un premier calcul pour déterminer la distribution des sources au bord du cœur. Ces sources peuvent ensuite être propagées dans le reste du bâtiment. Ces méthodes permettent aussi d'obtenir des gains en précision et temps de calcul.
- Les méthodes couplées. Cette fois-ci les deux méthodes échangent des informations pour mettre à jour le calcul hybride complet. Les calculs stochastiques et déterministes doivent être lancés ensemble pour pouvoir réaliser le calcul couplé. Ce type de méthode peut être très avantageux pour les problèmes où les deux zones étudiées sont dépendantes l'une de l'autre. Implémenter une méthode hybride couplée rajoute des difficultés à gérer dans le contrôle de la simulation, mais conduit à des bénéfices importants quand le problème entier dépend des résultats de la zone d'intérêt.

Les méthodes de décomposition de domaine (DDM) permettent de séparer un problème en plusieurs sous-problèmes et résolvent le problème global en mettant à jour les conditions aux limites des sous-domaines à chaque itération. Elles paraissent donc adaptées pour le développement de méthodes couplées, du moment que les conditions aux limites entre les sous-domaines sont compatibles. Elles ont déjà été utilisées pour développer des méthodes déterministes avec des traitements spatiaux différents pour chaque sous-domaine notamment. Ces méthodes se prêtent donc à la singularisation d'une région d'intérêt et adapter le traitement de celle-ci.

Avant de s'attaquer au développement de la méthode hybride, il est important de mieux comprendre le fonctionnement et l'implémentation de la DDM dans IDT. La première étape de cette thèse a été de modifier l'algorithme de DDM d'IDT pour permettre l'utilisation de maillages énergétiques différents entre les sous-domaines. Ce travail est nécessaire pour se confronter à l'impact sur la DDM des échanges de données déséquilibrés. En effet, nous allons avoir des sous-domaines plus raffinés que d'autres et devoir créer des méthodes permettant ces échanges inégaux. Cette étape permettra aussi de prendre en main le code déterministe en vue des futures modifications.

Pour reconstruire l'information venant des sous-domaines grossiers vers les sous-domaines raffinés, trois méthodes de reconstruction ont été testées. Les méthodes sont présentées ci-dessous, dans le Tableau 1.

	Reconstruction élémentaire (BRM)	**Reconstruction par le courant (CRM)**	**Reconstruction par le flux (FRM)**
Facteur	$\frac{\Delta u^g}{\Delta u^G}$	$\frac{J_1^+(x,g)}{\sum_{g\in G} J_1^+(x,g)}$	$\frac{\Psi_1^+(x,-\vec{\Omega}_d,g)}{\sum_{g\in G} \Psi_1^+(x,-\vec{\Omega}_d,g)}$

Tableau 1: Méthodes de reconstruction étudiées

Une modélisation adéquate du réflecteur est essentielle pour obtenir une bonne estimation du flux en périphérie du cœur. C'est aussi une donnée pour les calculs de structure. Nous avons donc choisi de tester notre méthode de reconstruction sur un problème cœur-réflecteur. Le problème étudié est représenté en Fig. 2.

Fig. 2: Quart de cœur étudié dans ce travail

Nous avons choisi de nous concentrer sur un réflecteur de type EPR, constitués d'une vingtaine de centimètres d'acier inoxydable. Les résonances du fer compliquent la modélisation de ces réflecteurs comparée à la modélisation d'un réflecteur en eau.

Les trois méthodes ont permis d'obtenir des résultats intermédiaires en termes de k_{eff} et temps de calcul. Cependant, une analyse des taux d'absorption permettra d'identifier les gains locaux apportés par celles-ci. Dans ce travail, nous avons considéré un maillage fin et un maillage grossier avec des bornes énergétiques communes. Cependant, ces contraintes ne sont pas compatibles avec la généralisation de cette méthode. Pour généraliser cette méthode, il faudrait tout d'abord pouvoir utiliser des raffinements locaux dans chaque sous-domaine pour s'adapter au problème énergétique du sous-domaine. Pour rendre cette méthode plus facile à utiliser par un futur utilisateur, il faudrait pouvoir utiliser des maillages sans bornes énergétiques communes, et donc implémenter de nouvelles méthodes de reconstruction adaptées à des bornes non-communes. Les maillages énergétiques vers lesquels il faudrait tendre dans cette nouvelle méthode sont illustrés en Fig. 3.

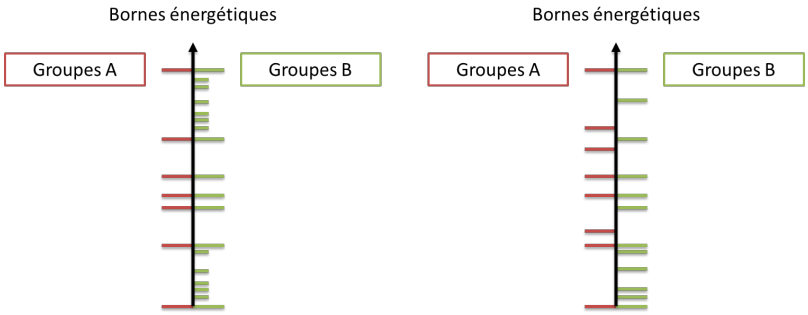


Fig. 3: Illustration des maillages énergétiques dans la méthode actuelle (gauche) et une généralisation de cette méthode (droite)

Cette méthode de reconstruction énergétique est un premier pas vers le développement d'une future méthode hybride déterministe-déterministe, cette méthode fait toujours face aux problèmes des méthodes déterministes. Malgré les gains que l'on pourrait obtenir, cette méthode sera toujours impactée par les problèmes d'autoprotection en 3D, qui ne peut être réalisée dans APOLLO3®. Pour pouvoir s'affranchir de cette contrainte, nous faisons le choix de passer au développement d'une méthode hybride dans la suite du travail.

Une première méthode chainée est étudiée par la suite, pour travailler sur l'échange d'information du déterministe vers le stochastique. Cette méthode permettra d'étudier les gains qu'un calcul stochastique peut apporter sur une petite zone du problème déterministe. Le calcul déterministe ne prend pas de retour d'information par le calcul LAST. Cette étape est essentielle avant de réaliser le développement de la méthode couplée, puisqu'elle permet de quantifier pour un problème donné, l'impact de l'initialisation déterministe sur le calcul stochastique.

La décomposition de domaines d'IDT échange les flux aux bords des sous-domaines pour mettre à jour les conditions aux limites. Nous avons choisi d'utiliser ces flux aux bords pour initialiser le calcul LAST dans notre méthode hybride pour être cohérents avec la procédure de DDM déjà implémentée. Cependant, LAST utilise une distribution de sources pour initialiser ses calculs. Nous avons donc besoin de transformer des flux aux bords en sources stochastiques. Les flux aux bords sont définis sur les arêtes des sous-domaines, donc nous allons créer une distribution de sources surfaciques pour le calcul LAST. Pour transformer ces flux en sources, nous allons échantillonner la distribution de flux aux bords pour obtenir des sources surfaciques. Cette méthode est présentée en Algorithme 1.

Algorithme d'échantillonnage de la source stochastique

Stocker les flux aux bords d'IDT pour les lire dans le calcul LAST

Créer la distribution des poids à partir des flux angulaires d'IDT $w_d^g(x) = \Psi_d^g(x)\vec{\Omega}_d.\vec{n}$

Échantillonner N particules

- Utiliser la distribution des poids comme distribution discrète et tirer un index
  - Utiliser l'index pour déterminer la maille spatiale dans IDT et la convertir en une position (x, y, z)
  - Utiliser l'index pour déterminer le groupe énergétique dans IDT et le convertir en une énergie E
  - Utiliser l'index pour déterminer la direction dans IDT et trouver les coordonnées associées $\{\mu, \eta, \xi\}$

Algorithme 1: Procédure de transformation des flux au bord des sous-domaines d'IDT en une distribution de sources surfaciques pour LAST

Pour tester cette méthode, nous avons commencé à regarder un problème 2D simplifié avec une traverse de cellules combustible et de réflecteur eau, illustré en Fig. 4.

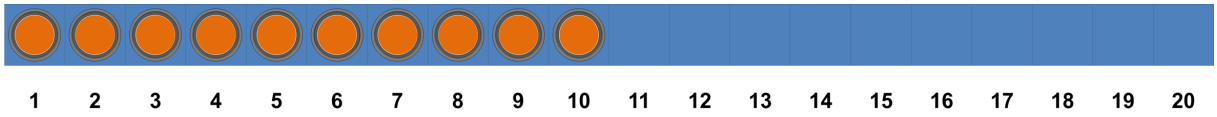


Fig. 4: Illustration du problème traverse qui sera étudiée

Nous réalisons un premier calcul avec IDT sur toute la géométrie, et utilisons les flux à l'interface entre le combustible et le réflecteur pour initialiser le calcul stochastique. À chaque batch du calcul stochastique, nous refixerons les sources au bord gauche de la traverse en eau. Cette méthode permet de bien reproduire les flux dans chaque cellule d'eau. Nous allons ensuite dégrader les distributions énergétiques, angulaires et spatiales du calcul initial IDT pour voir l'impact sur le calcul chainé. Sur ce problème transverse, nous obtenons de bons résultats sur les flux dans la traverse en eau.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF ALGORITHMS

LIST OF ACRONYMS AND ABBREVIATIONS

APOLLO3®	Multi-purpose deterministic neutronic code under development at CEA/Saclay
ASN	Autorité de Sûreté Nucléaire – Nuclear Safety Authority, it is tasked with regulating nuclear safety, on behalf of the state, and informing citizens
BRM	Basic Reconstruction Method
CRM	Current Reconstruction Method
DDM	Domain Decomposition Methods
FR	Fast Reactor
FRM	Flux Reconstruction Method
IDT	Transport solver of APOLLO3® that can perform 2D and 3D lattice simulations, and 2D and 3D domain decomposition simulations
k_{eff}	k-effective or effective multiplication factor
LAST	A prototype for Monte Carlo transport code, developed at CEA/Cadarache
LWR	Light Water Reactor
MC	Monte Carlo
MOC	Method of Characteristics
MOSC	Method of Short Characteristics
MOX	Mixed Oxide
OCS	Outil de Calcul Scientifique – Scientific computing tools
P_N	Spherical Harmonics
PWR	Pressurized Water Reactor
S_N	Discrete Ordinates method
TRIPOLI-4®	Fourth generation of the continuous-energy radiation transport Monte Carlo code developed at CEA/Saclay
UOX	Uranium Oxide
ZR4	Zircaloy-4

1. INTRODUCTION

Numerical methods play an important role in the demonstration of both the viability of a new reactor concept and the safety demonstration of power plant life extension. Due to the high cost of building a nuclear facility, as well as the safety requirements, it is necessary to be able to demonstrate the viability of a project using predefined criteria, such as material availability, used fuel recycling capability, economic competitiveness, expected lifetime, etc. In France, the government requires a safety demonstration of the plant project to license the project [1]. Article 3.8 of [2] specifies that the safety demonstration must use verified computational tools.

The Autorité de Sûreté Nucléaire (ASN) has defined a guideline, “guide 28” [3], in which it issues recommendations regarding the computational tools to ensure that they satisfy the plant operator’s needs. ASN is the French authority regulating nuclear safety and radiation protection. Such tools are called “Outils de Calcul Scientifique” OCS in French, are scientific computing tools, which are defined by the CEA and its industrial partners to fulfill ASN’s requirements for each field of application. These tools can notably be used to study core physics, shielding, or dismantling.

In this context this work aims to develop innovative numerical methods to improve neutronic simulations. While computational tools rely on a model to simulate neutron behavior in a core, they also require input data and a calculation scheme to compute parameters / quantities of interest. For neutronic calculation tools, the four main elements are:

- Model: neutron transport equation;
- Input data: neutron cross sections, material composition, geometry;
- Calculation scheme: methods used to compute the quantities of interest.
- Quantities of interest: reactor power map, effective multiplication factor k_{eff}, reaction rates, etc.

As it will be detailed later in the next paragraph, existing methods are not suited to some complex situations such as new reactor concepts. Exploring new neutronic calculation schemes is therefore necessary.

1.1 CURRENT METHODS IN NEUTRONICS AND CHALLENGES

Calculation schemes modeling nuclear reactor cores can be separated into two categories: deterministic and stochastic methods. These methods are used in other fields, but this work will only describe the theory and processes in neutronics.

Deterministic methods solve the neutron transport equation by discretizing the phase-space, which means that approximations are necessary to solve the equation. While this method leads to an approximate solution, this solution is still provided over the whole problem.

The discretization of the neutron transport equation for deterministic methods will be explained further in §2.2.1. The focus here is the number of mesh cells necessary to discretize a full-core light water reactor (LWR), based on the order of magnitudes presented in [4].

Figure 1 represents the spatial dimensions of the different levels of a reactor core where core height can be considered to be approximately 4m.

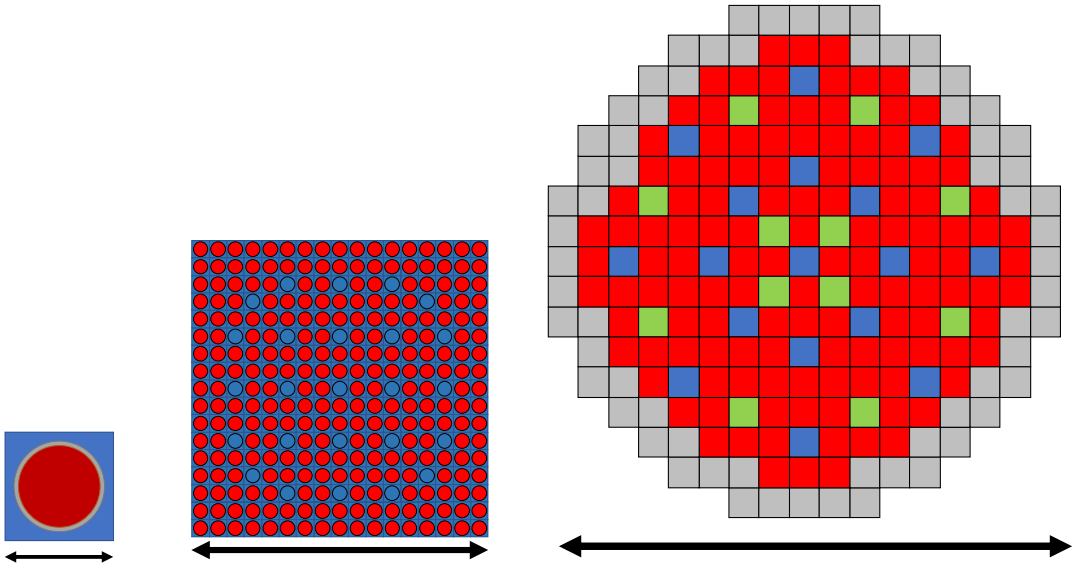


Figure 1: Radial views of a 900 MWe LWR core

Based on the dimensions presented in Figure 1, it is possible to determine the minimum number of mesh cells necessary to fully represent each spatial level of a LWR core. This information is condensed in Table 1, in which the minimum number of unknowns necessary for each phase-space component to perform a critical LWR deterministic calculation is represented. Because the LWR core is critical, it is possible to neglect the time dependency, and for this reason, the discretization is only for the phase-space excluding time.

	Energy	**Direction / angle**	**Space**		
			Cell	**Assembly**	**Core**
Minimum number of mesh cells	100 000	250	5 radial 40 axial	281 cells per assembly	150 assemblies
Total per phase space component	100 000	250	$8{,}42 \times 10^{6}$		
Total	$2{,}2 \times 10^{14}$				

Table 1: Minimum number of unknowns for a deterministic calculation of a LWR

To perform a whole core LWR deterministic calculation without approximations requires at least $2{,}2 \times 10^{14}$ unknowns. This means that at every step of the calculation $2{,}2 \times 10^{14}$ equations must be solved. Performing such a calculation requires large computational resources, which is not currently feasible. It is necessary to simplify the problem with approximations to be able to perform a deterministic core calculation.

Stochastic methods, on the other hand, do not rely on a discretization of the phase-space. Stochastic methods are instead probabilistic (Monte Carlo type) methods, and rely on random sampling of the neutron population. Trajectories, interactions with matter, and induced reactions are all governed by probabilities of occurrence. Due to the probabilistic nature of these methods, a statistical error is associated with each result. The process is repeated until the statistical error is reduced to a satisfactory level. No or very few approximations of the physics and geometry of the problem are necessary to perform stochastic calculations, rendering this method more precise than deterministic methods. However, this precision requires longer calculation times and more computational resources.

The two branches of numerical methods named are complementary. At first glance, stochastic methods seem better suited for reference calculation, whereas deterministic calculations would be quicker but less precise.

While these two methods are complementary in most cases, there are still situations where neither method performs well. The irradiation experiments GEDEON [5] performed in the MELUSINE reactor [6] are such examples. The experiments were designed to study gadolinium poisoned UO_2 in PWRs. The gadolinium cells are a mixture of UO_2 and Gd_2O_3 and are placed in a lattice with a moderation ratio of 1.8. Deterministic predictions are worse at lower burnup, in part due to the lack of information on the position of the control rods during irradiation. The movement during irradiation increases the cost of a full Monte Carlo analysis of these experiments.

Using the complementary nature of the two methods, it seems that combining the two into a hybrid method would help overcome of their limitations. The deterministic calculation is better suited for the global calculation with control rod movement. The stochastic calculation is better suited for the gadolinium cell rate calculation. To be able to treat such problems, we chose to study them with the development of a hybrid method.

1.2 OBJECTIVES OF THIS WORK

This Ph.D. thesis explores the possibilities of deterministic - stochastic coupling and proposes a method using spatial domain decomposition already implemented in the deterministic solver IDT. IDT is a discrete ordinates deterministic solver of the APOLLO3® platform that can perform 2D and 3D computations using a non-overlapping spatial domain decomposition method. The stochastic prototype LAST, developed at CEA Cadarache, has been used to implement this hybrid method. The objective is to perform calculations of reactor cores where spatial areas are singularized using spatial domain decomposition methods (DDM). This hybrid method would make it possible to differentiate the treatment of the spatial zone from the rest of the problem, to obtain fine information on local neutronic quantities of interest and a better description of local heterogeneities. The goal of this work is to implement a 2-way coupled hybrid method using DDM.

1.3 MANUSCRIPT LAYOUT

After this brief introduction to the types of methods used for neutron core calculations, it is time to dive in deeper and provide solutions to the problem statement. Chapter 2 explains the mathematical formalism of the neutron transport equation. Once the neutron transport derivation is complete, a derivation of the mathematical formalism of the two branches of numerical methods will be performed. After comparing both deterministic and stochastic methods, an overview of the different types of hybrid methods will complete chapter 2.

Chapter 1 presents the analysis of a deterministic-deterministic energetic hybrid method. It introduces some of the challenges in the boundary exchange for hybrid methods when meshes are different on each side of the interface. It also presents the changes in the DDM calculation methodology to accommodate different meshes.

Chapter 1 proposes a deterministic-stochastic hybrid method where the two codes are not directly linked, and where the deterministic simulation provides additional information to the stochastic calculation. The methodology relies on a converged deterministic calculation as input for the stochastic code.

Conclusions on the work accomplished, the challenges faced and recommendations for future work developing this hybrid method are presented in our last section.

2. STATE-OF-THE-ART HYBRID METHODS TO SOLVE TRANSPORT PROBLEMS

This chapter is aimed at giving the reader key elements of neutronics and an understanding of the current challenges in neutronics studies. The chapter first focuses on neutron behavior in a reactor core, then moves on to the derivation of said behavior (§2.1). We then look at deterministic (§2.2) and stochastic (§2.3) resolution schemes, and also review different hybrid methods developed (§2.4) to overcome challenges faced by purely deterministic or stochastic analysis.

Contents

2.1 DERIVING THE BOLTZMANN EQUATION

Neutron behavior in a reactor core can be described by a distribution in space, direction, energy, and time, which obeys the neutron transport equation. This is a linear form of the Boltzmann transport equation [7] [8] [9] [10], in which it is possible to consider a "gas of neutrons deploying throughout a gas of nuclides" [4]. The particle distribution is defined in the phase-space, which is constituted of position $\vec{r}$, particle direction $\vec{\Omega}$, and energy E. The neutron angular flux $\Psi(\vec{r}, \vec{\Omega}, E, t)$ is obtained by solving the neutron transport equation and is used to calculate quantities of interest such as reaction rates.

2.1.1 Hypotheses and approximations to study nuclear reactors

In this work, we will look at the Boltzmann equation applied to nuclear reactors. Before deriving the equation to solve, it is necessary to establish the context in which it will be solved. The following hypotheses are made to set the environment:

- In a nuclear reactor, the maximum kinetic energy (20 MeV) does not justify a relativist correction. For this reason, neutrons can only be considered as classical particles, and the polarizing effect is neglected. In this context, a neutron can be completely characterized by its position, direction and speed (or energy).
- Mean free path – a few centimeters in a LWR and a few tens of cm in a fast reactor (FR) – is significantly larger than a nucleus – about 10^{-15} m. This means that collision can be viewed as a local phenomenon in time and space. We assume that neutrons move in straight lines between two collisions. We neglect any other force acting on neutrons other than the one responsible for collisions.
- β^- disintegration's half-life (10 min) is notably longer than the average neutron lifetime in a reactor (from 10^{-3} to 10^{-7}s), rendering it possible to neglect this phenomenon.
- Neutron density in a nuclear reactor (10^{15} neutrons/cm^3) is appreciably smaller than nuclei density (10^{23} nuclei/cm^3). Neutron-neutron interactions are much less probable than neutron-nuclei interactions. As a result, neutron-neutron collisions are neglected.
- From the neutron perspective, materials in a reactor core can be considered as isotropic.
- Delayed neutrons (those produced by radioactive decay of fission products) are not distinguished from prompt neutrons. The effect of the delay is noticeable for small time scales (10^{-5} s for thermal reactors) and not in a stationary case. Stationarity will be explained further down in §2.1.5.

2.1.2 Phase-space

The neutron population is described at instant t in the phase-space Γ by its position $\vec{r}$, direction $\vec{\Omega}$, and energy E. The phase-space $\Gamma = (\vec{r}, \vec{\Omega}, E)$ is a seven-dimensional space.

2.1.3 Cross-sections

$\sigma_{r,is}$ is the microscopic cross-section, which can be considered as the surface of the r-type interaction, of a neutron with the isotope is. It is typically expressed in barns, where $1b = 10^{-24} cm^2$. As previously expressed in §2.1.1, materials are considered isotropic yielding rotationally invariant microscopic cross-sections $\sigma_{r,is}(\vec{r}, E)$.

Σ_r is the macroscopic cross-section and is the probability of the interaction r of a neutron by unit of distance. Writing the macroscopic cross-section as such might be problematic when dealing with a sum.

To avoid confusion, the sum is indicated in **bold**. For instance, the macroscopic cross-section is the sum of the microscopic cross-sections for all isotopes multiplied by their respective concentrations:

$$\Sigma_r(\vec{r}, E, t) = \sum_{\boldsymbol{is}} N_{is}(\vec{r}, t)\sigma_{r,is}(\vec{r}, E)$$

The reader can turn to references [11] and [10] for more information regarding neutron interactions with matter, nuclear physics, and cross-sections.

2.1.4 Particle balance

The statistical study of the evolution of a particle population was conducted at the end of the 19th century by Boltzmann, who then derived the Boltzmann transport equation. As mentioned in §2.1.1, to determine parameters of interest for nuclear physics, a simplified version of this equation is solved.

Using the hypotheses defined above, and the methodology defined in [9], it is possible to derive the neutron transport equation through a particle balance on the volume $d^3\vec{r}d^2\vec{\Omega}dE$ in the phase-space, between the instants t and $t + dt$. We derive the transport equation by looking at the particle balance in the phase-space, using the number of particles at time t, in $\vec{r}$ within d^3r with a velocity of $\vec{v}$ within d^3v, contained in the $d\Gamma = d^3\vec{r}d^2\vec{\Omega}dE$ volume at the instant t. In the absence of collisions, these same particles occupy volume $d\Gamma' = d^3\vec{r'}d^2\vec{\Omega'}dE'$ at instant $t + dt$. However, the external source of neutrons and collisions also need to be accounted for.

Neutrons appearing between instants of t and $t + dt$ can be attributed to both a neutron source and collisions. Neutrons disappearing during this interval of time, can, in turn, be attributed to collisions. With this information, it is possible to synthesize the particle balance as an equation:

$$time\ variation + disappearances = collisions + source$$

The particles of interest, in this case, are neutrons, so the neutron balance can be expressed as follows:

$$\frac{\partial n}{\partial t}(\vec{r}, \vec{\Omega}, E, t) + v\vec{\Omega}.\vec{\nabla}n(\vec{r}, \vec{\Omega}, E, t) = s_{collisions}(\vec{r}, \vec{\Omega}, E, t) + S(\vec{r}, \vec{\Omega}, E, t) \qquad (2.1)$$

With $n(\vec{r}, \vec{\Omega}, E, t)$ neutron angular density, v speed, and $S(\vec{r}, \vec{\Omega}, E, t)$ source term. The source term can be separated into two components: external source and neutrons induced by fission.

- **Angular flux**

The product of v and the neutron angular density is called the neutron angular flux:

$$\Psi(\vec{r}, \vec{\Omega}, E, t) = vn(\vec{r}, \vec{\Omega}, E, t)$$

With this definition, it is possible to rewrite the temporal variation term as so:

$$\frac{1}{v}\frac{\partial \Psi}{\partial t}(\vec{r}, \vec{\Omega}, E, t) = \frac{\partial n}{\partial t}(\vec{r}, \vec{\Omega}, E, t)$$

- **Collision term**

As mentioned previously, collisions are responsible for neutrons both appearing and disappearing. The probability of a neutron disappearing through collisions can be expressed with $\Sigma_t(\vec{r}, E, t)$ the probability that a neutron in $(\vec{r}, \vec{\Omega}, E)$ collides with a nucleus.

On the contrary, the probability for neutron in $(\vec{r}, \vec{\Omega}, E)$ to collide with a nucleus and be remitted in $(\vec{r}, \vec{\Omega'}, E')$, depends on the scattering cross-section $\Sigma_s(\vec{r}, \vec{\Omega}, E \to \vec{\Omega}', E', t)$.

$$s_{collision} = -\Sigma_t(\vec{r}, E, t)\Psi(\vec{r}, \vec{\Omega}, E, t) + \int_0^\infty \int_{4\pi} \Sigma_s(\vec{r}, \vec{\Omega}, E \to \vec{\Omega}', E', t)\Psi(\vec{r}, \vec{\Omega}', E', t)d\vec{\Omega'}^2 dE'$$

- **External source**

Neutrons emitted by an independent external source Q, are one part of the neutron source term. This source includes different production methods, all independent of the flux. This could be a reactor startup source, spallation or spontaneous fission.

- **Neutron source**

The other part of the neutron source term is made up of neutrons induced by fission. This source is dependent on the flux. A fraction of the neutrons produced are instantly emitted (prompt neutrons) while others are emitted with a delay (delayed neutrons). The total number of neutrons induced by fission is made up both of these terms. The number of prompt neutrons produced is:

$$\mathbf{F_p}\Psi(\vec{r},\vec{\Omega},E,t) = \sum_{is}^{Nf}\frac{1}{4\pi}\int_0^{\infty}\nu_{p,is}(E')\chi_{p,is}(E'\to E)\Sigma_{f,is}(\vec{r},E',t)\Psi(\vec{r},E',t)dE' \qquad (2.2)$$

Fission products are grouped in 6 to 8 precursor families depending on their half-life. C_k the concentration of precursor family k, λ_k its decay constant and $\chi_{d,k}$ the delayed neutron spectrum. The number of delayed neutrons is:

$$\mathbf{F_d}\Psi(\vec{r},\vec{\Omega},E,t) = \frac{1}{4\pi}\sum_k \chi_{d,k}(E)\lambda_k C_k(\vec{r},t) \qquad (2.3)$$

The evolution of the precursor's concentration reveals this terms dependence to the flux.

$$\forall k,\qquad \frac{dC_k}{dt}(\vec{r},t) + \lambda_k C_k(\vec{r},t) = \sum_{is}^{Nf}\int_0^{\infty}\chi_{d,k,is}(E')\,\Sigma_{f,is}(\vec{r},E',t)\Psi(\vec{r},E',t)dE' \qquad (2.4)$$

The production cross-section $\nu\Sigma_f(\vec{r},\vec{\Omega},E\to\vec{\Omega}',E',t)$ is defined for steady-state situations according to the hypotheses made in §2.1.1.The total source term in steady-state conditions can be expressed by combining the external and fission source components:

$$S = Q + \sum_{is}^{Nf}\frac{1}{4\pi}\int_0^{\infty}\int_{4\pi}\nu\Sigma_f(\vec{r},\vec{\Omega},E\to\vec{\Omega}',E',t)\Psi(\vec{r},\vec{\Omega}',E',t)\chi(E\to E')d\vec{\Omega'}^2dE'$$

Using the four previous points, it is possible to rewrite the particle balance (2.1) depending on the angular flux.

$$\begin{aligned}\frac{1}{v}\frac{\partial\Psi}{\partial t}(\vec{r},\vec{\Omega},E,t) + \vec{\Omega}.\vec{\nabla}\Psi(\vec{r},\vec{\Omega},E,t) + \Sigma_t(\vec{r},E,t)\Psi(\vec{r},\vec{\Omega},E,t)\\ = \int_0^{\infty}\int_{4\pi}\Sigma_s(\vec{r},\vec{\Omega},E\to\vec{\Omega}',E',t)\Psi(\vec{r},\vec{\Omega}',E',t)d\vec{\Omega'}^2dE'\\ + \sum_{is}^{Nf}\frac{1}{4\pi}\int_0^{\infty}\int_{4\pi}\nu\Sigma_f(\vec{r},\vec{\Omega},E\to\vec{\Omega}',E',t)\Psi(\vec{r},\vec{\Omega}',E',t)\chi(E\to E')d\vec{\Omega'}^2dE' + Q\end{aligned} \qquad (2.5)$$

The angular flux is simplified to alleviate the neutron transport equation, it now becomes $\Psi = \Psi(\vec{r},\vec{\Omega},E,t)$. It is possible to write this equation in an even simpler form, by using operators. Operators are written in **bold** to differentiate them from other variables.

$$\frac{1}{v}\frac{\partial\Psi}{\partial t} + \boldsymbol{L}\Psi = \boldsymbol{H}\Psi + \boldsymbol{F}\Psi + Q \qquad (2.6)$$

L is the leakage operator, **H** scattering operator, and **F** fission operator. The leakage operator includes both geometrical leakage and disappearances from collisions.

It is important to note that without boundary conditions, the neutron transport problem is not closed. The existence and uniqueness of the solution are confirmed by spatial boundary conditions and a temporal initial condition.

$$\begin{cases} \Psi(\vec{r},\vec{\Omega},E,t) = \Psi_0(\vec{r},\vec{\Omega},E) & \text{à } t = t_0 \\ \Psi(\vec{r},\vec{\Omega},E,t) = \beta\left(\overrightarrow{r'} \to \vec{r}, \vec{\Omega} \to \vec{\Omega}, E, t\right)\Psi\left(\overrightarrow{r'},\overrightarrow{\Omega'},E,t\right) & \vec{r},\overrightarrow{r'} \in \vec{r}_{bord}, \vec{\Omega}.\vec{n}_{ext} < 0 \end{cases} \quad (2.7)$$

β is called albedo and expresses the ratio between outgoing and ingoing fluxes at the domain interface. If there is a reflective boundary condition then $\beta = 1$ and $(\vec{r},\vec{\Omega}) = \left(\overrightarrow{r'},\overrightarrow{\Omega'} - 2\overrightarrow{\Omega'}.\vec{n}_{ext}\right)$ but if there is a vacuum boundary condition then $\beta = 0$.

2.1.5 Stationary approximation

Materials exposed to irradiation see their properties vary due to neutronic reactions. This effect is directly observable on macroscopic cross-sections.

The inventory variation of nuclides under irradiation can be described by the Bateman equation. For each fission product or actinide, the concentration variation in fuel is dependent on neutron flux. This means that studying a reactor core over time requires iteratively solving the Boltzmann and Bateman equations.

However, in a nuclear reactor, cross-sections slowly vary with time when compared to the average neutron lifespan (§2.1.1). The stationary hypothesis simplifies the equation to be solved. Under this hypothesis, it is possible to neglect the time-dependent component of the neutron transport equation (2.6). Furthermore, for a critical nuclear power reactors, a source is used during startup to get the subcritical core to reach criticality. Once the reactor is critical, the flux level produced by fissions is much more significant than the startup source's. From this point on, we neglect the flux produced by this source. The neutron transport problem can then be written as an eigenvalue problem (2.8), introducing the eigenvalue λ.

$$\boldsymbol{L}\Psi = \boldsymbol{H}\Psi + \lambda\boldsymbol{F}\Psi \quad (2.8)$$

Correcting the fission operator is convenient because the eigenvalue now coincides with the inverse of the effective multiplication factor k_{eff} of the system. It is possible to rewrite the (2.8) eigenvalue problem depending on k_{eff}.

$$\boldsymbol{L}\Psi = \boldsymbol{H}\Psi + \frac{1}{k_{eff}}\boldsymbol{F}\Psi \quad (2.9)$$

This multiplication factor represents the number of neutrons from one iteration to the next. This factor makes it possible to predict reactor behavior:

- $k_{eff} > 1$: supercritical state of the system, in which the population of neutrons increases with each generation;
- $k_{eff} = 1$: critical state where the chain reaction is self-sustaining, meaning there is no change in the neutron population over time;
- $k_{eff} < 1$: subcritical state then the number of neutrons decreases with time.

The multiplication factor shows the difference between the current reactor state and the critical state. Reactivity is defined using the multiplication factor $\rho = 1 - \frac{1}{k_{eff}}$. It is important to note that solving the eigenvalue problem is not sufficient to directly obtain the angular flux Ψ, because it also means that $k \times \Psi$ is as well, with $k \in \mathbb{R}$. To be able to determine the flux, one can normalize the result using the global power of the core.

In the rest of this manuscript, the phase-space will refer to the six-dimension space excluding time.

2.1.6 Scattering operator

The stationary approximation renders the transfer cross-section $\Sigma_s(\vec{r}, \vec{\Omega}, E \to \vec{\Omega}', E')$ independent of time. This is not the only hypothesis made to fit the nuclear reactor context, meaning it is possible to simplify this expression even further.

As expressed §2.1.1, materials are isotropic, making cross-sections rotationally invariant and solely dependent on the deviation angle $\mu = \vec{\Omega}.\vec{\Omega}'$ of the neutron after a collision. Under this hypothesis, the transfer cross-section can be rewritten as:

$$\Sigma_s(\vec{r}, \vec{\Omega}, E \to \vec{\Omega}', E') = \frac{1}{2\pi} \Sigma_s(\vec{r}, E \to E', \mu)$$

By using the separation of variables method, the angular dependence of the transfer cross-section can be developed by a Legendre polynomial expansion. P_k said polynomials.

$$\Sigma_s(\vec{r}, E \to E', \mu) = \frac{1}{2\pi} \sum_{k=0}^{\infty} \frac{(2k+1)}{2} \Sigma_{s,k}(\vec{r}, E \to E') P_k(\mu) \qquad (2.10)$$

$$\Sigma_{s,k}(\vec{r}, E \to E') = \int_{4\pi} \Sigma_s(\vec{r}, E \to E', \mu) P_k(\mu) d\mu$$

$\Sigma_{s,k}(\vec{r}, E \to E')$, the k-order moment of the transfer cross-section.

Furthermore, the addition theorem can be applied to Legendre polynomials, resulting in the development of the scattering operator on real spherical harmonics [11] [12].

$$P_k(\mu) = \frac{1}{2k+1} \sum_{l=-k}^{k} A_{kl}(\vec{\Omega}) A_{kl}(\vec{\Omega}')$$

By injecting this information into the angular flux expression, it is possible to obtain:

$$\Psi(\vec{r}, \vec{\Omega}, E) = \frac{1}{4\pi} \sum_{k=0}^{\infty} \sum_{l=-k}^{k} \Phi_{kl}(\vec{r}, E) A_{kl}(\vec{\Omega})$$

$$\Phi_{kl}(\vec{r}, E) = \int_{4\pi} A_{kl}\left(\vec{\Omega'}\right) \Psi\left(\vec{r}, \vec{\Omega'}, E\right) d\vec{\Omega'}$$

It is possible to simplify the scattering operator **H** by using the polynomial expansions of the angular flux and transfer cross-section.

$$\boldsymbol{H}\Psi(\vec{r}, \vec{\Omega}, E) = \frac{1}{4\pi} \sum_{k=0}^{\infty} \sum_{l=-k}^{k} A_{kl}(\vec{\Omega}) \int_{0}^{\infty} \Sigma_{s,k}(\vec{r}, E' \to E) \Phi_{kl}(\vec{r}, E') dE' \qquad (2.11)$$

Numerical methods describing critical nuclear reactors are the focus of this document, meaning that the eigenvalue problem needs to be resolved.

This part of the review describes the main methods and approximations necessary to numerically solve the neutron transport equation.

As a reminder, there exist two main types of methods to obtain the solution of the neutron transport equation.

- Stochastic methods rely on few approximations, the solution is instead obtained by looking at the simulated average neutron life, creating a statistical error attached to the result. The main principles of this method are explained in §2.3.
- Deterministic methods, however, do rely on approximations and the discretization of the phase-space. The initial problem is transformed into a system of equations. The steps to solve the neutron transport equation by deterministic methods are detailed in the following paragraph.

2.2 DETERMINISTIC METHODS

Deterministic methods rely on discretization of the phase-space, as such different methods can be used to discretize energy, angle, and space. Figure 2, which is inspired by a figure in reference [4], synthesizes different discretization methods.

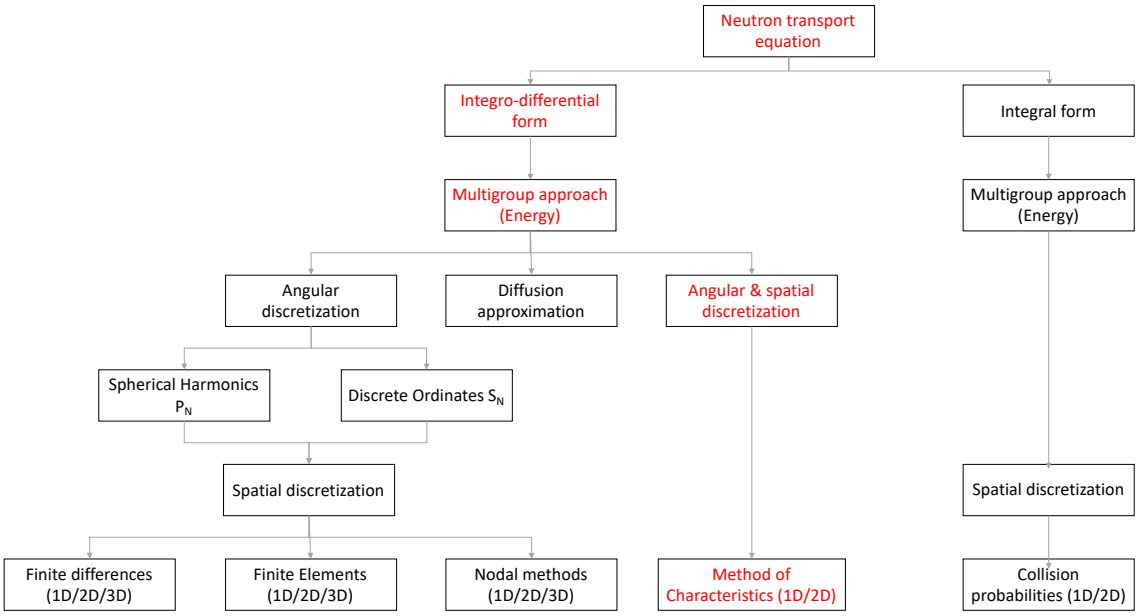


Figure 2: Different discretization methods used in a deterministic simulation

Methods used by IDT are in red, which is the deterministic solver chosen in this thesis. The following paragraphs derive the equations implemented in IDT, relying on references [13] [14] [15] and [16].

APOLLO3® is a platform in which different solvers can be used; IDT is one of these solvers. A more comprehensive explanation of the project around the APOLLO3® multi-purpose code can be found in [17] and [18]. Deterministic solvers are a combination of angular and spatial discretization. The following list illustrates how different discretization methods are used in APOLLO3® solvers.

- TDT [19]: where a coupled spatial/angular discretization is implemented. It is traditionally used for 2D problems using either collision probabilities (TDT-P_{ij}) or method of characteristics (TDT-MOC). Both methods lead to good results in terms of precisions; TDT-P_{ij} is better suited to smaller geometries. TDT-MOC is one of the most precise deterministic methods available.
- MINOS [20]: can treat Cartesian geometries using Raviart-Thomas finite elements. The transport operator is simplified by either using the diffusion approximation or a simplified spherical harmonics angular method. For these reasons, this method is usually preferred for 3D core calculation where the precision/calculation-time balance favors calculation time.
- MINARET [21] [22]: is also a discrete ordinates method, the spatial treatment is performed using discontinuous Galerkin finite elements. This method can treat non-structured radial geometries, but leads to increased calculation costs compared to IDT.
- IDT [14] [15] [23] [24] [25]: the Integro-Differential Transport (IDT) code is designed to solve the multigroup time-independent equation for neutrons in 2D and 3D Cartesian geometries. IDT was originally developed as a stand-alone discrete ordinates (S_N) flux solver for 2D use in APOLLO2 [23] and is currently under development for 3D use in the CEA code APOLLO3® [17]. A non-overlapping spatial domain decomposition method (DDM) is implemented in IDT [25].

The reader can refer to the following references, [4] [26] [21] [27] and [28], for a larger overview of deterministic methods.

2.2.1 Solving neutron transport problems

2.2.1.1 Energy discretization

The multigroup approximation is used to discretize the energy spectrum in most deterministic solvers. It consists of replacing continuous-energy variables with discrete variables by dividing the energy spectrum into subdomains called groups. In Figure 3 the energy spectrum is divided into N_g groups, which are conventionally numbered decreasingly with energy: $E_0 = E_{max}$ and $E_{N_g+1} = E_{min}$.

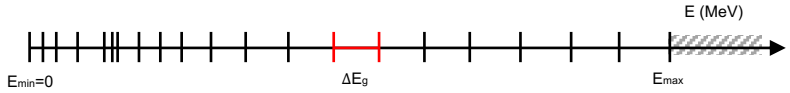


Figure 3: Generic multigroup representation

Furthermore, high-energy groups are called fast groups, whereas low energy ones are called thermal groups.

In the multigroup approximation, cross-sections and flux are discretized into a finite number N_g of energy groups, and are constant and averaged in each of these groups.

Once the energy spectrum is divided into groups, the neutron transport equation is integrated over each group. To ensure the correct resolution of the neutron transport equation, it is necessary to conserve reaction rates. For this reason, cross-sections are defined as follows:

$$\Sigma^g(\vec{r},\vec{\Omega}) = \frac{\int_{E_{g-1}}^{E_g} \Sigma(\vec{r},E)\Psi(\vec{r},\vec{\Omega},E)dE}{\int_{E_{g-1}}^{E_g} \Psi(\vec{r},\vec{\Omega},E)dE}$$

Unfortunately, this expression of the multigroup cross-section illustrates difficulties in solving the neutron transport equation. There is now an angular dependence of the macroscopic cross-section, even though materials are isotropic. The macroscopic cross-section is now dependent on the angular flux, which is itself determined by solving the neutron transport equation.

In cases where the isotopes' energy dependence is weak, the flux can be replaced by an analytic spectrum representative of a type of reactor. However, for resonant isotopes another method needs to be used, this process is called self-shielding [7] [28] This generates self-shielded cross-sections, specific to each problem.

The multigroup angular flux can be expressed as:

$$\Psi(\vec{r},\vec{\Omega},E) = \{\Psi^g(\vec{r},\vec{\Omega}) \qquad avec\ g \in [\![1,N_g]\!]\}$$

As mentioned previously, the multigroup approximation transforms the eigenvalue problem (2.8) as a system of N_g angular-flux-dependent monokinetic equations over intervals $[E_g; E_{g-1}]$.

$$L^g\Psi^g = H^{g'\to g}\Psi^g + \frac{1}{k_{eff}}F^g\Psi^g \qquad (2.12)$$

The operators are defined in the multigroup approximation as follows.

$$\boldsymbol{L}^g = \vec{\Omega}.\vec{\nabla} + \Sigma_t^g$$

$$\mathbf{H}^{g'\to g}\Psi^g(\vec{r},\vec{\Omega}) = \frac{1}{4\pi}\sum_{k=0}^{\infty}\sum_{l=-k}^{k}\Sigma_{s,k}^{g'\to g}(\vec{r})\Phi_{kl}^{g'}(\vec{r})Y_{kl}(\vec{\Omega})$$

$$\boldsymbol{F}^{g}\Psi^{g}(\vec{r}) = \frac{1}{4\pi}\sum_{j=1}^{N_{fiss}} \chi_j^g N_j(\vec{r}) \sum_{g'=1}^{N_g} \nu\Sigma_{f,j}^{g'}(\vec{r})\Phi^{g'}(\vec{r})$$

$$\Phi_{kl}^{g}(\vec{r}) \int_{4\pi} Y_{kl}(\vec{\Omega})\Psi^{g}(\vec{r},\vec{\Omega})d\vec{\Omega}$$

Neutrons tend to lose energy with each collision, this phenomenon is called down-scattering. The increase in energy by collision, also called up-scattering, is negligible at high energy. This makes it possible to separate the scattering operator into three components.

$$\sum_{g'=0}^{N_g+1} \boldsymbol{H}^{g'\to g} = \boldsymbol{H}^{g\to g} + \sum_{g'<g} \boldsymbol{H}_{down}^{g'\to g} + \sum_{g'>g} \boldsymbol{H}_{up}^{g'\to g}$$

Because up-scattering is negligible at high energy, it is possible to solve the multigroup equations starting with high energy groups and using the previously calculated flux moments to determine the slowing-down operator. This is called the Gauss-Siedel iteration.

When using this method, caution is recommended for groups in which up-scattering can no longer be neglected. In this case, it is necessary to iterate the calculation within the group to ensure convergence.

The fission source is estimated at the beginning of every iteration using the scalar fluxes from the previous one. The effective multiplication coefficient k_{eff} is calculated at the end of each iteration.

2.2.1.2 Angular discretization

The multigroup approximation is insufficient to discretize the whole phase-space, it is also important to discretize the angular component. Three methods are traditionally used to discretize the angular variable in transport codes [28].

- Discrete ordinates S_N: considers a finite number of directions $\{\vec{\Omega}_d\}_{d\in[\![1,N_d]\!]}$ belonging to the unit sphere.
- Spherical harmonics P_N: projects the angular flux on a spherical harmonic basis.
- Simplified spherical harmonics SP_N: is, as the name suggests, a simplified version of the P_N method, making it more suited to core calculations.

Discrete ordinates are the basis of the method of characteristics (MOC), and IDT discretizes the phase-space using a variation of this method (MOSC). This method provides a system of N_d differential flux-dependent equations for the angular variable.

To simplify the discretization of the neutron transport equation, group indices will no longer be written for either fluxes or cross-sections.

$$\Psi(\vec{r},\vec{\Omega}) \equiv \Psi^{g}(\vec{r},\vec{\Omega})$$

The system of equations is obtained by considering a set of directions $\{\vec{\Omega}_d\}_{d\in[\![1,N_d]\!]}$, transforming the multigroup equation (2.12) as:

$$\left[\vec{\Omega}_d.\vec{\nabla} + \Sigma_t(\vec{r})\right]\Psi(\vec{r},\vec{\Omega}_d) = \frac{1}{4\pi}\sum_{k=0}^{\infty}\sum_{l=-k}^{k} \Sigma_{s,k}(\vec{r})\Phi_{kl}(\vec{r})Y_{kl}(\vec{\Omega}_d) + \frac{1}{4\pi}\sum_{is=1}^{N_{fiss}} \chi_{is}(\vec{r})N_{is}(\vec{r}) \sum_{g'=1}^{N_g} \nu\Sigma_{f,is}(\vec{r})\Phi(\vec{r}) \quad (2.13)$$

The previous equation can be simplified by rewriting the angular flux as follows.

$$\Psi_d(\vec{r}) \equiv \Psi(\vec{r},\vec{\Omega}_d) \quad (2.14)$$

2.2.1.3 Spatial discretization

As shown previously in Figure 2, spatial discretization methods in neutron transport codes are similar to those used in other fields. Spatial discretization in IDT can be performed by finite differences, the nodal method, or MOSC. This document presents MOC spatial discretization. The reader can turn to references [21] [26] for more information on the methods not presented here.

The MOC variant implemented in IDT is the method of short characteristics (MOSC). This method is based on integrating the neutron transport equation over trajectories in the 2D or 3D domain. Since neutrons are supposed to move in straight lines between collisions, the trajectories modeled are also straight lines. These straight trajectories are called characteristics. Masiello [29] chose to implement the MOSC in IDT since it was less memory-expensive than MOC. MOC stores all the points of impact of each trajectory to the interfaces between subdomains, while MOSC only stores projections on constant or linear functions. MOSC trajectories are limited to a single cell and uses MOC to calculate the flux on these trajectories. Having trajectories limited on a single cell means that the matrixes connecting spatial flux moments only need to be calculated once for each type of cell.

Figure 4 represents trajectories to discretize a fuel cell under the MOSC.

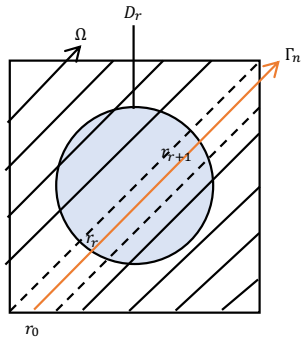


Figure 4: MOSC discretization for a fuel cell

MOSC relies on the angular flux integration along trajectories, making it more convenient to use a source problem rather than an eigenvalue problem. The (2.13) equation becomes the equation below, with q_d the source term, containing both collision and fission sources.

$$\left[\vec{\Omega}_d.\vec{\nabla} + \Sigma_t(\vec{r})\right]\Psi_d(\vec{r}) = q_d \qquad (2.15)$$

Before integrating the flux along the trajectory Γ_n, it is important to define the optical path τ. The distance traveled $s = |\vec{r} - \vec{r}_0|$ by the particle is central to this definition.

$$\tau\left(s, \vec{r}, \vec{\Omega}\right) = \int_0^s \Sigma_t\left(\vec{r} - s'\vec{\Omega}\right)ds'$$

By integrating the angular flux along the trajectory Γ_n, the source problem becomes an expression of the angular flux.

$$\Psi_d(\vec{r}) = \Psi_d(\vec{r}_0)e^{-\tau(s,\vec{r},\vec{\Omega}_d)} + \int_0^s q_d\left(\vec{r} - s'\vec{\Omega}_d\right)e^{-\tau(s',\vec{r},\vec{\Omega}_d)}ds' \qquad (2.16)$$

Introducing operators $\boldsymbol{T}$ and $\boldsymbol{K}$, the equation (2.16) can be written under operator form.

$$\Psi_d(\vec{r}) = \boldsymbol{T}\left(\vec{r}, \vec{\Omega}_d\right)\Psi_d(\vec{r}_0) + [\boldsymbol{K}q_d]\left(\vec{r}, \vec{\Omega}_d\right) \qquad (2.17)$$

In practice, the integrals in equation (2.16) are obtained by dividing the domain D into discrete regions D_r. In this case, the source and cross-sections are projected into regions using domain functions. Over the considered regions cross-sections Σ_r and source q_r are constant and averaged over r.

$$\forall\, \vec{r} \in D_r, \forall\, \vec{\Omega} \in S \begin{cases} \Sigma(\vec{r}) \cong \Sigma_r f_r(\vec{r}) \\ q(\vec{r}, \vec{\Omega}) \cong q_r(\vec{\Omega}) f_r(\vec{r}) \end{cases}$$

With this information the integration in equation (2.16) can be performed over the interval $I = [\vec{r}_r; \vec{r}_{r+1}] = \Gamma_n \cap D_r$.

$$\Psi_d(\vec{r}_{r+1}) = \Psi_d(\vec{r}_r) e^{-\tau(s,\vec{r},\vec{\Omega}_d)} + \frac{1 - e^{-\tau(s,\vec{r},\vec{\Omega}_d)}}{\Sigma_r} q_r(\vec{\Omega}_d) \qquad (2.18)$$

The new definition of the angular flux (2.18), provides the information necessary to define the operator $\boldsymbol{T}$ and $\boldsymbol{K}$, from equation (2.17). $\boldsymbol{T}$ is the transmission operator, and $\boldsymbol{K}$ is the leakage operator.

$$\boldsymbol{T}(\vec{r}, \vec{\Omega}_d) = e^{-\tau(s,\vec{r},\vec{\Omega}_d)} \qquad \boldsymbol{K}(\vec{r}, \vec{\Omega}_d) = \frac{1 - e^{-\tau(s,\vec{r},\vec{\Omega}_d)}}{\Sigma_r}$$

The goal is to solve the discretized neutron transport equation (2.15). To do so, the averaged neutron flux per region r is defined. This is a piecewise-constant definition of the flux.

$$\Psi_r(\vec{\Omega}) = \frac{1}{V_r} \int_{D_r} \Psi(\vec{r}, \vec{\Omega}) d^3\vec{r}$$

In practice, the flux integral over the domain D_r can be used to reconstruct the region-averaged angular flux.

$$\Psi_r(\vec{\Omega}) = \frac{1}{\Sigma_r} \left(q_r(\vec{\Omega}) - \Delta J_r(\vec{\Omega}) \right) \qquad (2.19)$$

ΔJ_r is the difference between outgoing $(\vec{n}.\vec{\Omega} > 0)$ and incoming currents. Following this definition, the difference can be expressed using the region's surface S_r.

$$\Delta J_r(\vec{\Omega}) = \frac{1}{V_r} \int_{S_r} |\vec{n}.\vec{\Omega}| \Psi(\vec{r}, \vec{\Omega}) d^2\vec{r}$$

This difference is typically approximated by a quadrature formula, using a set of characteristics parallel to $\vec{\Omega}_d$ and intersecting the cell. To each of these trajectories, a weight $w_{r,n}$ is associated, resulting in ΔJ_r being expressed as a sum of the different characteristics I.

$$\Delta J_r(\vec{\Omega}) \cong \sum_I w_{r,n}(\vec{\Omega}) \left(\Psi(\vec{r}_{r+1}, \vec{\Omega}) - \Psi(\vec{r}_r, \vec{\Omega}) \right)$$

Now that the derivation of the discretized neutron transport equation is completed, we can look at how this is implemented in IDT. The sequential algorithm implemented in IDT is shown below, in Algorithm 1.

Numerical Steps in IDT to solve the multigroup problem

- Loop over external iterations — e=1,…,N_e
  - Update the local fission source from $\Phi^{(e-1)}$
  - Multigroup approximation – Scattering source calculation — g=1,…,N_G
    - Update local scattering sources ($\forall g' \neq g$) from $\Psi^{(e),g'<g}$ and $\Psi^{(e-1),g'>g}$
    - Internal iterations — i=1,…,N_i
      - Solve the spatial sub-problem: sweep over the mesh and propagate from boundary conditions
      - Update the local self-scattering source $\boldsymbol{H}^{gg}\Psi^{(e,i),g}$
  - Update k_{eff}
  - Convergence test on k_{eff} and power distribution

Algorithm 1: Algorithm implemented in IDT to solve a multigroup problem

To conclude the derivation of the discretized neutron transport equation, the main properties of IDT are summarized in Table 2.

Transport equation form	Integro-differential
Energetic discretization	Multigroup approximation
Angular discretization	S_N method
Spatial discretization	Finite differences Nodal method MOSC
Anisotropy	Arbitrary
Applications	Assemblies and small cores

Table 2: Main properties of the IDT solver

The previous section focused on the sequential deterministic scheme implemented in IDT. As mentioned at the beginning of this section, a non-overlapping spatial DDM is already implemented in IDT. The following section is aimed at providing the reader with a better grasp of how DDM works and impacts the multigroup scheme.

2.2.2 Domain Decomposition Methods

The development of parallel architectures over the last decades has led to an increase in the use of domain decomposition methods (DDM) to numerically solve problems. As this continues, algorithms need to be suited to parallel calculations. A large amount of literature exists, as evidenced by the proceedings of the annual conference on DDM [30]. This document only refers to DDM applications to neutronics codes.

The first definition of DDM is attributed to the mathematician H.A. Schwarz in the 1870s. Schwarz proposed an iterative method to analytically solve elliptical problems in complex geometries [31]. The equations of this problem rely on $\boldsymbol{L}$ a differential operator and u the unknown.

$$\begin{cases} \boldsymbol{L}u = f & in\ \Omega \\ u = g & over\ \Gamma = \partial\Omega \end{cases} \qquad (2.20)$$

Such methods made a comeback in the 1990s, in part due to Lions' work [32], and with the development of parallel architectures.

2.2.2.1 Overlapping DDM

Schwarz's idea was to separate the global domain into two overlapping subdomains $\Omega = \Omega_1 \cap \Omega_2$. This method iteratively solves the Laplace problem with a Dirichlet boundary condition (BC), taking into account the presence of the other subdomain. The Laplace problem is similar to the initial problem.

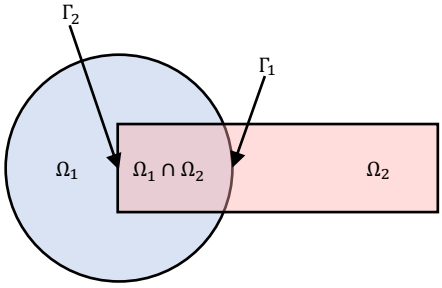


Figure 5: Original problem studied by Schwarz in 1870

It is possible to write the problem (2.11) as two iterative problems, one for each subdomain.

Problem 1

$$\begin{aligned} \boldsymbol{L}u_1^{n+1} &= f \quad in\ \Omega_1 \\ u_1^{n+1} &= g \quad over\ \partial\Omega_1 \setminus \Gamma_1 \\ u_1^{n+1} &= u_2^n \quad over\ \Gamma_1 = \partial\Omega_2 \cap \Omega_2 \end{aligned}$$

Problem 2

$$\begin{aligned} \boldsymbol{L}u_2^{n+1} &= f \quad in\ \Omega_2 \\ u_2^{n+1} &= g \quad over\ \partial\Omega_2 \setminus \Gamma_2 \\ u_2^{n+1} &= u_1^n \quad over\ \Gamma_1 = \partial\Omega_1 \cap \Omega_1 \end{aligned}$$

Each "half-problem" is solved at every iteration and updates the solutions (u_1^n, u_2^n). Schwarz established that the convergence of the iterative process tends toward the solution of (2.20). This process is called alternating Schwarz method. When the global domain is separated into two subdomains, there are two variants of the alternating Schwarz method called additive and multiplicative Schwarz methods. Algorithm 2 compares the two algorithms, and the differences are written in red.

Additive Schwarz method	**Multiplicative Schwarz method**
Problem 1 $\boldsymbol{L}u_1^n = f_1 \quad in\ \Omega_1$ $u_1^n = g \quad over\ \partial\Omega_1 \setminus \Gamma_1$ $u_1^n = u_2^{n-1} \quad over\ \Gamma_1 = \partial\Omega_2 \cap \Omega_2$ Problem 2 $\boldsymbol{L}u_2^n = f_2 \quad in\ \Omega_2$ $u_2^n = g \quad over\ \partial\Omega_2 \setminus \Gamma_2$ $u_2^n = u_1^{n-1} \quad over\ \Gamma_1 = \partial\Omega_1 \cap \Omega_1$	Problem 1 $\boldsymbol{L}u_1^n = f_1 \quad in\ \Omega_1$ $u_1^n = g \quad over\ \partial\Omega_1 \setminus \Gamma_1$ $u_1^n = u_2^{n-1} \quad over\ \Gamma_1 = \partial\Omega_2 \cap \Omega_2$ Problem 2 $\boldsymbol{L}u_2^n = f_2 \quad in\ \Omega_2$ $u_2^n = g \quad over\ \partial\Omega_2 \setminus \Gamma_2$ $u_2^n = u_1^n \quad over\ \Gamma_1 = \partial\Omega_1 \cap \Omega_1$

Algorithm 2: Algorithm comparison between additive and multiplicative Schwarz methods

The way information is exchanged between subdomains at iteration n is the main difference between the two methods, as shown in Algorithm 2. Information is propagated as soon as it is available in the multiplicative method, while in the additive one, information from the previous iteration is used.

It is to be noted that the additive method's convergence is slower than the multiplicative one. This can be understood by the fact that the additive method requires more iterations since it uses the information from the previous generation.

2.2.2.2 Non-overlapping DDM

With this approach, the idea is to separate the global domain into S non-overlapping subdomains, as represented in Figure 6. The information is now solely propagated on the interfaces between subdomains.

$$\Omega = \bigcup_{i=1}^{S} \Omega_i \quad \Omega_i \cap \Omega_j = \emptyset \tag{2.21}$$

The difference between these two types of DDM is seen at the interface between subdomains.

$$dim\left(\bigcup_i \bigcup_j (\overline{\Omega}_i \cap \overline{\Omega}_j)\right) = \begin{cases} dim(\Omega) - 1 & non - overlapping \\ dim(\Omega) & overlapping \end{cases} \tag{2.22}$$

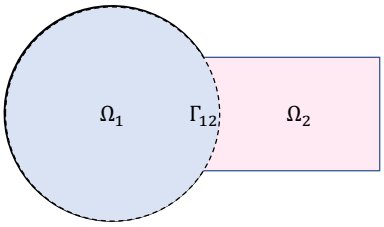


Figure 6: Non-overlapping DDM of the geometry of the original Schwarz problem

The following problem is solved in each subdomain. The goal is to determine u on the interface $\Gamma_i \cap \Gamma_j$.

$$\begin{cases} L_i u_i = f_i & in\ \Omega \\ u_i = g & over\ \ \Gamma_i \cap \Gamma \\ u_i = u_\Gamma & over\ \Gamma_i \cap \Gamma_j \end{cases} \tag{2.23}$$

System (2.23) can be written as a linear system by starting with the unknowns from subdomain 1, then numbering those from subdomain 2, and finally, those belonging to the interface.

$$\begin{bmatrix} A_1 & 0 & -B_{1\Gamma} \\ 0 & A_2 & -B_{2\Gamma} \\ -D_{\Gamma 1} & -D_{\Gamma 2} & C_{\Gamma_{12}} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_{\Gamma_{12}} \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ g \end{bmatrix} \tag{2.24}$$

System (2.24) relies on the following definitions.

- A_i the discretized form restricted to Ω_i, of the ***L*** operator ;
- $C_{\Gamma_{12}}$ the discretized form restricted to $\Gamma_i \cap \Gamma_j$, of the ***L*** operator ;
- $B_{i\Gamma}$ and $D_{\Gamma \mathrm{i}}$ matrices linking the unknowns of a subdomain to the boundary unknown ;
- u_i the discretized form restricted to Ω_i, of the unknown u ;
- $u_{\Gamma_{12}}$ the discretized form restricted to $\Gamma_i \cap \Gamma_j$, of the unknown u ;
- f_i the discretized form restricted to Ω_i, of f.

To highlight the bloc structure of (2.24) matrices of the same type can be grouped.

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} u \\ u_\Gamma \end{bmatrix} = \begin{bmatrix} f \\ g \end{bmatrix} \tag{2.25}$$

It is possible to transform this system into a linear problem depending on the boundary unknowns.

$$Su_\Gamma = b \tag{2.26}$$

where $S = C - DA^{-1}B$ is the Schur complement and $b = g - DA^{-1}f$. It is also possible to obtain the solution of subdomains, by solving the system on the boundary surfaces (2.26). This system can be solved by both direct and iterative methods and is typically performed with preconditioned Krylov subspace methods.

$$u = A^{-1}(f - Bu_\Gamma) \tag{2.27}$$

Building the Schur complement requires the inversion of matrix A. This matrix is diagonal by blocs, making the inversion of the matrix *A*, is equivalent to inverting the local problem for each subdomain.

2.2.2.3 Applications to neutron transport

Spatial DDM have increasingly been used in the last few years in neutronic calculation schemes, particularly in deterministic methods. It is important to note that the interest in spatial DDM is complementary to the development of parallel architectures.

DDM have been applied to deterministic schemes with two main objectives. On the one hand, they can be used to decompose the phase-space to solve in parallel the decomposed problem.

Table 3 shows the parallelization of the phase-space using deterministic methods.

Energy	Angle	Space
Each processor solves a monokinetic problem. This is equivalent to solving the multigroup problem by an iterative bloc Gauss-Jacobi	It depends on the method used. Sn methods can be parallelized because they use a set of discrete and independent directions. Each direction can be treated by a different processor.	No particular parallelization

Table 3: Parallelization of the phase-space in deterministic methods

This parallel step is necessary to calculate the (2.8) collision source and to solve the (2.5) monokinetic equation for a given direction. Once this step is finished, data is synchronized sequentially using moments of the angular flux in equation (2.6).

2.2.3 Modification of the deterministic scheme to accommodate DDM

The implementation of a spatial DDM in IDT does not change its goal: to solve the neutron transport equation. The formulation of the problem is written for a domain D. Domains are now called D to avoid confusion with angular directions.

$$\begin{cases} \boldsymbol{A}\Psi = \boldsymbol{Q} & in\ D \\ \Psi^{-} = 0 & over\ \Gamma \end{cases} \qquad (2.28)$$

Ψ is the angular flux within the domain D, Ψ^{-} is the ingoing flux at the boundary Γ, $\boldsymbol{A}$ is the transport operator, and $\boldsymbol{Q} = \frac{1}{\lambda}\boldsymbol{F}\Psi$ the normalized fission source. $\boldsymbol{F}$ the fission operator and λ the problem's eigenvalue. The phase space is represented in X.

$$\begin{aligned} X &\equiv \left(\vec{r} \in D, \vec{\Omega} \in S^2, E \in \mathbb{R}^+\right) \\ \Gamma &\equiv \left(\vec{r} \in \partial D, \vec{\Omega} \in S^2, E \in \mathbb{R}^+\right) \\ \Gamma^{\pm} &\equiv \left(\vec{r} \in \partial D, \vec{\Omega} \in S^2 : \vec{\Omega}.\overrightarrow{n_+(\vec{r})} \lessgtr 0, E \in \mathbb{R}^+\right) \end{aligned} \qquad (2.29)$$

As a reminder ∂D is the domain's border, $\overrightarrow{n_+(\vec{r})}$ is the outgoing normal at a point $\vec{r} \in \partial D$ and Γ^+ and Γ^- are the outgoing and incoming boundaries of the phase space X.

A non-overlapping domain decomposition method is implemented in IDT as shown in Figure 7. In his Ph.D thesis, Lenain [13] shows that a non-overlapping method was chosen to optimize memory costs.

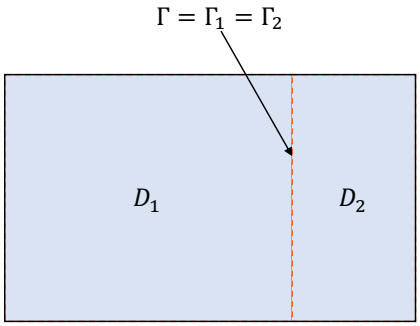


Figure 7: Example of non-overlapping domain decomposition

For each subdomain the phase space can be written as follows:

$$\begin{aligned} X_i &\equiv \left(\vec{r} \in D_i, \vec{\Omega} \in S^2, E \in \mathbb{R}^+\right) \\ \Gamma_i &\equiv \left(\vec{r} \in \partial D_i, \vec{\Omega} \in S^2, E \in \mathbb{R}^+\right) \\ \Gamma_i^{\pm} &\equiv \left(\vec{r} \in \partial D_i, \vec{\Omega} \in S^2 : \vec{\Omega}.\overrightarrow{n_+(\vec{r})} \lessgtr 0, E \in \mathbb{R}^+\right) \end{aligned} \qquad (2.30)$$

For each subdomain, the following problem can be solved. Ψ_i is the multigroup angular flux for the i-th subdomain and $\Psi_{\Gamma 12}$ is the angular flux at the interface between subdomains 1 and 2.

$$\begin{cases} \boldsymbol{A}_i \Psi_i = \boldsymbol{Q_i} & in\ X_i \\ \Psi_i^- = 0 & over\ \Gamma_i \cap \Gamma \\ \Psi_i = \Psi_{\Gamma 12} & over\ \Gamma_i \cap \Gamma_j \end{cases} \qquad (2.31)$$

The vacuum boundary condition is ensured by the second line of equations (2.31), where the incoming boundary flux is null. The global solution is obtained by line 3 of equations (2.31). This line shows that the continuity of the angular flux at the interface is imposed. By proceeding as shown in §2.2.2.2, the following system is obtained.

$$\begin{bmatrix} A_1 & 0 & -B_{1\Gamma} \\ 0 & A_2 & -B_{2\Gamma} \\ -D_{\Gamma 1} & -D_{\Gamma 2} & C_{\Gamma 12} \end{bmatrix} . \begin{bmatrix} \Psi_1 \\ \Psi_2 \\ \Psi_{\Gamma 12} \end{bmatrix} = \begin{bmatrix} Q_1 \\ Q_2 \\ 0 \end{bmatrix} \qquad (2.32)$$

As a reminder, $B_{i\Gamma}$ and $D_{\Gamma i}$ are the projection operators, while $C_{\Gamma 12}$ is the coupling operator. $B_{i\Gamma}$ projects the flux into the subdomain, $D_{\Gamma i}$ projects the flux to the interface, and $C_{\Gamma 12}$ couples the interface fluxes. To continue and solve the problem, the interface flux needs to be expressed as.

$$\Psi_{\Gamma 12} = \begin{cases} \Psi_{\Gamma 1} = \begin{cases} \Psi_{\Gamma 1+}, & \vec{\Omega} \in S^2 : \vec{\Omega}.\overrightarrow{n_{1+}(\vec{r})} > 0 \\ \Psi_{\Gamma 1-}, & \vec{\Omega} \in S^2 : \vec{\Omega}.\overrightarrow{n_{1+}(\vec{r})} < 0 \end{cases} \\ \Psi_{\Gamma 2} = \begin{cases} \Psi_{\Gamma 2+}, & \vec{\Omega} \in S^2 : \vec{\Omega}.\overrightarrow{n_{2+}(\vec{r})} > 0 \\ \Psi_{\Gamma 2-}, & \vec{\Omega} \in S^2 : \vec{\Omega}.\overrightarrow{n_{2+}(\vec{r})} < 0 \end{cases} \end{cases} \qquad (2.33)$$

System (2.32) becomes:

$$\left[\begin{array}{cc|cccc} A_1 & 0 & 0 & 0 & -\tilde{B}_{1\Gamma} & 0 \\ 0 & A_2 & 0 & 0 & 0 & -\tilde{B}_{2\Gamma} \\ \hline -\widetilde{D}_{\Gamma 1} & 0 & I & 0 & -\tilde{C}_1 & 0 \\ 0 & -\widetilde{D}_{\Gamma 1} & 0 & I & 0 & \tilde{C}_2 \\ 0 & 0 & 0 & -I & I & 0 \\ 0 & 0 & -I & 0 & 0 & I \end{array}\right] . \left[\begin{array}{c} \Psi_1 \\ \Psi_2 \\ \hline \Psi_{\Gamma 1+} \\ \Psi_{\Gamma 2+} \\ \Psi_{\Gamma 1-} \\ \Psi_{\Gamma 2-} \end{array}\right] = \left[\begin{array}{c} Q_1 \\ Q_2 \\ \hline 0 \\ 0 \\ 0 \\ 0 \end{array}\right] \qquad (2.34)$$

During his Ph.D., Lenain [13] found this system particularly interesting because of the first two lines. These lines show the restrictions of the transport equation to each subdomain to be independent, thus making it possible to run each subdomain calculation in parallel. The communication procedure is

represented by the last two lines of the system. This is the reason he implemented an additive Schwarz scheme to solve the transport equation.

For each iteration, the incoming multigroup angular flux is imposed as the boundary condition for each subdomain. This incoming flux is nothing more than the outgoing flux from neighboring subdomains obtained at the previous iteration by inversing the multigroup transport problem in each subdomain. Boundary fluxes are updated at the end of each external iteration.

The implementation of these updates to accommodate multiple subdomains in IDT is shown in Algorithm 3. It can be noted that, compared to the sequential scheme shown in Algorithm 1, a loop over subdomains is added above the scattering source calculation.

Numerical Steps in IDT to solve the multigroup problem using DDM

- Loop over external iterations — e=1,…,N_e
  - Loop over subdomains — s=1,…,N_s
    - Update the local fission source from $\Phi_s^{(e-1)}$
    - Multigroup approximation – Scattering source calculation — g=1,…,N_G
      - Update local scattering sources ($\forall g' \neq g$) from $\Psi_s^{(e),g'<g}$ and $\Psi_s^{(e-1),g'>g}$
      - Internal iterations — i=1,…,N_i
        - Solve the spatial sub-problem: sweep over the mesh and propagate from boundary conditions
        - Update the local self-scattering source $\boldsymbol{H}^{gg}\Psi_s^{(e,i),g}$
    - Update the boundary flux and exchange boundary fluxes between subdomains

$$\forall\, subdo, \forall g, \forall \vec{\Omega} \begin{cases} \Psi^{(e)}_{\Gamma\, subdo,in}(\vec{r},\vec{\Omega},g) = \Psi^{(e)}_{\Gamma\, neighbor,out}(\vec{r},\vec{\Omega},g), & \vec{\Omega}.\vec{n}_{ext} < 0 \\ \Psi^{(e)}_{\Gamma\, neighbour,in}(\vec{r},\vec{\Omega},g) = \Psi^{(e)}_{\Gamma\, subdo,out}(\vec{r},\vec{\Omega},g), & \vec{\Omega}.\vec{n}_{ext} \geq 0 \end{cases}$$

    - Update k_{eff}
    - Convergence test on k_{eff} and power distribution
  - Reconstruction distribution at the core scale

Algorithm 3: Algorithm implemented in IDT to solve a multigroup problem using multiple subdomains

Domain decomposition methods make it possible to separate a global calculation into sub-problems. A non-overlapping DDM is implemented in IDT enabling the user to separate the core to be studied into different subdomains. Subdomains are defined and set by the user. In Figure 8, the user separated a quarter of a core into nine subdomains, each subdomain corresponding to the size of a subassembly.

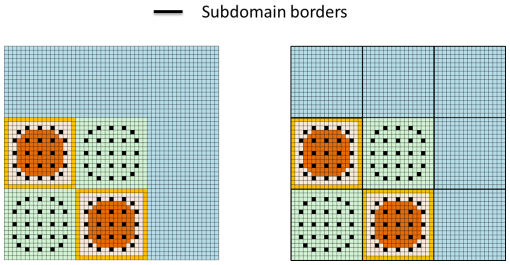


Figure 8: Illustration of how a global domain (left) can be separated into nine subdomains (right)

In the current implementation of the DDM in IDT, subdomains do not need to be the same size. This can be interesting from a computational point of view when certain subdomains are much more

expensive to calculate than others. To simplify boundary exchanges between subdomains, the phase space is treated the same way for each subdomain. This means that the spatial treatment of each subdomain is the same for all subdomains.

Future work to develop the hybrid method will use the DDM already implemented in IDT. The reason for this will be explained in §2.5.

2.2.4 Advantages and disadvantages of deterministic methods

In §1.1, Monte Carlo methods were shown to be better suited as reference methods than deterministic methods. Deterministic methods seemed more interesting for design purposes thanks to their comparatively quicker calculation time. To have a better understanding of the limitations of deterministic methods, we synthesized the advantages and disadvantages in Table 4.

Advantages	Disadvantages
Provides information in all phase-space of the problem	Increasing the phase-space discretization will increase the calculation time
Depletion calculations	
Coupling with thermo-hydraulic or thermo-mechanic codes	
	Discretization error
	Self-shielding procedure and associated error

Table 4: Main advantages and disadvantages of deterministic methods

2.3 STOCHASTIC METHODS

In this paper, stochastic methods are indifferently being referred to as stochastic, probabilistic, or Monte Carlo methods.

The Boltzmann equation offers a statistical description of a neutron population in the phase-space. Monte Carlo methods rely on random sampling to obtain a numerical solution of the problem, naturally suited to obtain a numerical solution of the Boltzmann equation.

To do so, neutron lives are simulated, from their birth up to their disappearance from the system by either leakage or absorption. Figure 9, represents the different lives a neutron can lead, as well as the reactions considered in a Monte Carlo simulation.

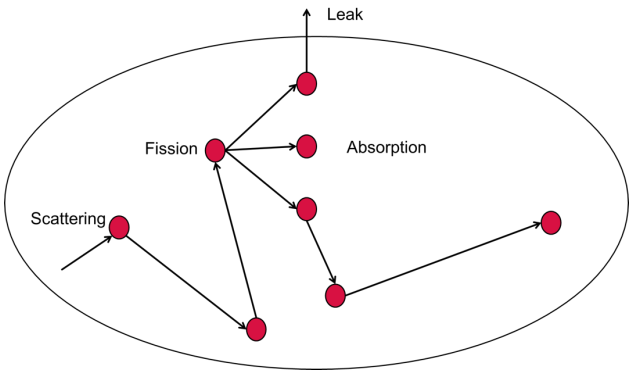


Figure 9: Representation of the different possible reactions considered in a Monte Carlo simulation

Due to the probabilistic nature of the simulation, a discretization of the phase-space is not necessary to solve the problem. This leads to fewer approximations, making stochastic methods better suited to reference calculations. However, to ensure convergence of such a simulation a large number of particle histories must be simulated, requiring large calculation times and computational resources for some calculations.

Before going over the Monte Carlo algorithm, it is necessary to define key notions of stochastic methods. This type of simulation relies on the law of large numbers and the central limit theorem.

2.3.1 Neutron transport equation derivation for stochastic methods

Contrary to deterministic methods used in IDT, stochastic methods rely on an integral form of the neutron transport equation. Figure 10 is used to derive the stationary and integral form of the neutron transport equation, using the methodology presented in references [4] [33].

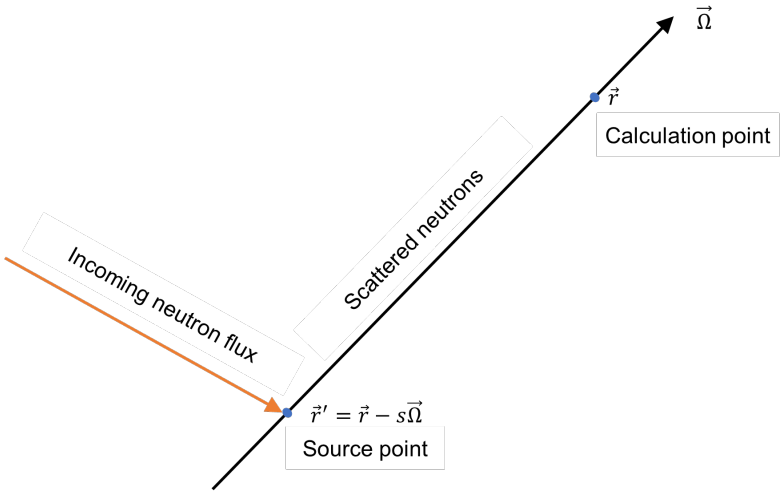


Figure 10: Balance representation to obtain the integral of the neutron transport equation

Figure 10 shows that the angular flux at the point of calculation $\Psi(\vec{r},\vec{\Omega},E)$ is the result of two contributions.

- The first part is composed of neutrons emitted by the source $S(\vec{r}',\vec{\Omega},E)$. These neutrons only reach the point of calculation if they do not collide. This contribution is made up of all the source points.

$$\int_0^{\infty} exp\left(-\int_0^{s}\Sigma(\vec{r}-s'\vec{\Omega},E)ds'\right)S(\vec{r}-s\vec{\Omega},\vec{\Omega},E)ds$$

- The second contribution is made up of neutrons arriving at $\vec{r}' = \vec{r} - s\vec{\Omega}$, with an energy E', instead of E and in a different direction. These neutrons contribute to the angular flux at the point of calculation if they collide with matter resulting in an energy E and direction $\vec{\Omega}$. Once they are at the desired energy and direction, they must not collide with anything else over the remaining distance.

$$\int_0^{\infty} exp\left(-\int_0^{s}\Sigma(\vec{r}-s'\vec{\Omega},E)ds'\right)\int_0^{\infty}\int_{4\pi}\Sigma_{tr}(\vec{r}-s\vec{\Omega},\vec{\Omega}',E'\rightarrow\vec{\Omega},E)\Psi(\vec{r}-s\vec{\Omega},\vec{\Omega},E)d\vec{\Omega}'\,dE'\,ds$$

By summing these two contributions, it is possible to obtain the integral form of the neutron transport equation, as written below.

$$\begin{aligned}\Psi(\vec{r},\vec{\Omega},E) = &\int_0^{\infty} exp\left(-\int_0^{s}\Sigma(\vec{r}-s'\vec{\Omega},E)ds'\right)\int_0^{\infty}\int_{4\pi}\Sigma_{tr}(\vec{r}-s\vec{\Omega},\vec{\Omega}',E'\\ &\rightarrow\vec{\Omega},E)\Psi(\vec{r}-s\vec{\Omega},\vec{\Omega},E)d\vec{\Omega}'\,dE'\,ds\\ &+\int_0^{\infty} exp\left(-\int_0^{s}\Sigma(\vec{r}-s'\vec{\Omega},E)ds'\right)S(\vec{r}-s\vec{\Omega},\vec{\Omega},E)ds\end{aligned} \qquad (2.35)$$

It is possible to rewrite the transport equation (2.35) as a function of the collision density $\xi(\vec{r},\vec{\Omega},E) = \Sigma(\vec{r},E)\Psi(\vec{r},\vec{\Omega},E)$.

$$\xi(\vec{r},\vec{\Omega},E) = \Sigma(\vec{r},E)\left[\int_0^s exp\left(-\int_0^s \Sigma(\vec{r}-s'\vec{\Omega},E)ds'\right)S(\vec{r}-s\vec{\Omega},\vec{\Omega},E)ds + \int_0^\infty exp\left(-\int_0^s \Sigma(\vec{r}-s'\vec{\Omega},E)ds'\right)\int_0^\infty\int_{4\pi}\frac{\Sigma_{tr}(\vec{r}-s\vec{\Omega},\vec{\Omega}',E'\to\vec{\Omega},E)}{\Sigma(\vec{r},E)}\xi(\vec{r}-s\vec{\Omega},\vec{\Omega},E)d\vec{\Omega}'\,dE'\,ds\right] \quad (2.36)$$

With this new expression, it is possible to define the following operators. **T** is the transfer operator, while **C** is called the collision operator.

$$\boldsymbol{T}\left(\vec{r'}\to\vec{r},\vec{\Omega},E\right) = \Sigma(\vec{r},E)\int_0^\infty exp\left(-\int_0^s \Sigma(\vec{r}-s'\vec{\Omega},E)ds'\right)ds$$

$$\boldsymbol{C}(\vec{r},\vec{\Omega}',E'\to\vec{\Omega},E) = \int_0^\infty\int_{4\pi}\frac{\Sigma_{tr}(\vec{r}-s\vec{\Omega},\vec{\Omega}',E'\to\vec{\Omega},E)}{\Sigma(\vec{r},E)}d\vec{\Omega}'\,dE'$$

Equation (2.36) can be written under operator form using the previously defined operators **T** and **C**.

$$\xi = \boldsymbol{TC}\xi + \boldsymbol{T}S \quad (2.37)$$

By defining the transport operator $\boldsymbol{K}\left(\vec{r'},\vec{\Omega}',E'\to\vec{r},\vec{\Omega},E\right) = \boldsymbol{T}\left(\vec{r'}\to\vec{r},\vec{\Omega},E\right)\boldsymbol{C}(\vec{r},\vec{\Omega}',E'\to\vec{\Omega},E)$, equation (2.37) becomes:

$$\xi = \boldsymbol{K}\xi + \boldsymbol{T}S \quad (2.38)$$

2.3.2 Monte Carlo scheme

The following paragraph reviews the different steps of a Monte Carlo simulation.

(1) Statistical process parameters definition. These are necessary to generate particle histories.
(2) Associating the random variable X to a quantity of interest R. Every time a neutron reaches the counting area, a score w is attributed to X.
(3) Probability density definition. To do so, the following equation must be satisfied in the region of interest Z: $E[X(Z)] = R(Z)$. Under this condition, defining probability densities is equivalent to randomly choosing the particles, their interactions and the distance traveled, and following the laws of physics.
(4) Repeating N times the simulation. N cycles, also called batches, of M particles, are simulated, after which it is possible to obtain $\bar{X}_n(Z) = \frac{1}{M}\sum_{m=1}^{M} w_{n,m}(Z)$. $w_{n,m}$ the score associated with history n of batch m.
(5) Law of large numbers to obtain R. According to the law of large numbers, it is possible to obtain $R(Z)$ with the average value $\bar{X}_N(Z) = \frac{1}{N}\sum_{n=1}^{N}\bar{X}_n(Z)$.

$$\lim_{N\to\infty}\overline{X}_N(Z) = R(Z)$$

This is possible with the central limit theorem, according to which $\bar{X}_N(Z)$ are distributed along a Gaussian of variance σ^2.

$$\sigma^2 = \frac{1}{U(U-1)}\sum_{v=1}^{U}\left(X_v(Z) - \overline{X}_U(Z)\right)^2$$

A confidence interval $c(e)$ with the central limit theorem can be calculated for a Gaussian probability density P, and e a positive real number. In practice, e is a multiple of σ^2.

$$c(e) = P\left(\overline{X}_U(Z) - e \le X(Z) \le \overline{X}_U(Z) + e\right)$$

Figure 11 represents the particle history simulation of step (3) and is inspired by a similar scheme in reference [4].

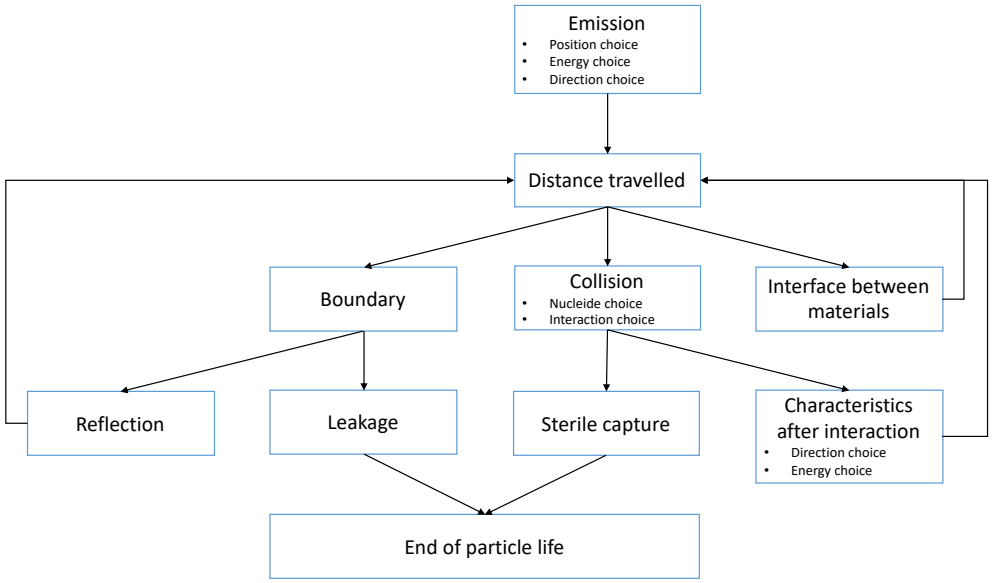


Figure 11: Schematic representation of a particle's life simulated by a stochastic method

To summarize, deterministic methods discretize the phase-space and require approximations to solve the neutron transport equation, while stochastic methods rely on random sampling and deal with statistical processes. IDT's MOSC is limited by the number of characteristics and directions considered, while stochastic methods' uses are limited by computational resources.

The processes described so far are performed sequentially. However, stochastic methods are naturally suited to parallel processes, as shown in step (4). The development of parallel architectures has improved the speed of a stochastic simulation.

2.3.3 Estimators

The goal of a stochastic simulation is to estimate quantities of interest. Given the probabilistic nature of the simulation, estimators need to deal with statistical data.

Neutron flux, current, reaction rates, and k_{eff} can all be obtained with estimators. Table 5 synthesizes the main estimators used in a Monte Carlo simulation to obtain the previously mentioned quantities of interest.

		Volume	Surface	Point
Flux	**Track length**	wl		
	Collision	$\frac{w}{\Sigma(\vec{r},E)}$		
	Other		$\frac{w}{\vec{\Omega}.\vec{n}}$	$w\frac{\Sigma_s(\vec{r},\vec{\Omega}',E'\rightarrow\vec{\Omega},E)}{\Sigma(\vec{r},E')}\frac{e^{-\tilde{\Sigma}}}{s^2}$
Current			w	
Reaction rates	**Track length**	$\Sigma_r(\vec{r},E)wl$		
	Collision	$\Sigma_r(\vec{r},E)\frac{w}{\Sigma(\vec{r},E)}$		
	Other			$\Sigma_r(\vec{r},E)w\frac{\Sigma_s(\vec{r},\vec{\Omega}',E'\rightarrow\vec{\Omega},E)}{\Sigma(\vec{r},E')}\frac{e^{-\tau}}{s^2}$

Table 5: Main estimators used to obtain flux, current, or reaction rates

Table 5 relies on the following physical parameters.

- w neutron weight before colliding;
- l distance traveled by the neutron between two collisions;
- Σ total macroscopic cross-section;
- Σ_s scattering macroscopic cross-section;
- Σ_r macroscopic cross-section of the reaction of interest;
- $\vec{n}$ normal direction to the considered surface;
- τ optical path between the collision point and the point of calculation.

Neutron weight is different in analog and non-analog simulations. Analog simulations aims to represent natural laws, meaning that each particle is simulated exactly and the weight is set to 1 for each event. This method has led to low scores for certain configurations, such as too few particles reaching a detector after an absorbing slab with a source on the other side. To limit the number of histories to be run, biasing methods are developed. These biased calculations are non-analog. The neutron weight is adjusted after each collision and tends to zero. In this simulation, the neutron weight can be seen as its probability to contribute to the desired result.

Table 5 highlights the fact that estimators cannot be used arbitrarily for all situations. For instance, the collision estimator cannot be used in vacuum cases because it is not defined for $\Sigma = 0$. Point kernel estimators are better suited to vacuum conditions. Convergence speed also varies with the estimator and the problem to treat.

k_{eff} can be obtained either using reactions or with specific estimators, two of which are given in Table 6.

Track length	$wl\sum_j \nu_j N_j \sigma_{fj}$
Collision	$\frac{w}{\Sigma}\sum_j \nu_j N_j \sigma_{fj}$

Table 6: k_{eff} estimators

Stochastic methods do not yield results over the whole geometry unless estimators are explicitly defined to do so. If flux variations are to be observed in a water traverse, for instance, the user needs to define cells in that traverse instead of one block of water. Creating additional tally volumes by superimposing a mesh over the problem geometry can be necessary for such problems but for evolution problems, this

is not sufficient. In such problems, additional volumes need to be created to account for the changes through time in the volumes of interest. However, by discretizing the whole problem, deterministic methods naturally produce results over the whole phase-space. This can be a problematic aspect when using stochastic methods.

2.3.4 Stochastic code used

Many different stochastic codes have been developed using the method presented such as TRIPOLI-4® [34] , MCNP [35], OpenMC [36], and SERPENT [37]. In this work, we chose to use LAST [38]. It was developed at CEA Cadarache as a Monte Carlo code for 3D neutron transport to run a simulation with self-evaluated nuclear data. This code was chosen to be used for test purposes. In our case, we chose to work with LAST, to implement a proof of concept of the hybrid method. If this method yields interesting results, it could then be implemented into a reference code.

However, the implementation of functionalities in LAST must be verified with a reference code. TRIPOLI-4® is a continuous-energy Monte Carlo code developed at CEA Saclay capable of performing 3D neutron and photon transport. It is used as a verification tool for the estimators implemented in LAST.

2.3.5 Advantages and disadvantages of stochastic methods

Stochastic methods are continuous in energy, space, and angle, and avoid discretization errors; therefore, they seem to have significant advantages over deterministic methods. To better understand the limitations of both methods, we look at stochastic methods in Table 7.

Advantages	Disadvantages
No discretization required → well suited for complex geometries and used as a reference method	Stochastic uncertainties → Standard deviation decrease as the square root of the number of histories run. Calculations can become very costly for large and complex problems
	Increasing the number of tally volumes, surfaces and points can also be computationally costly
Well suited to low scattering regions	The calculation becomes more costly in highly scattering regions. Neutrons will have to collide and scatter a greater number of times before escaping or dying.

Table 7: Main advantages and disadvantages of stochastic methods

From Table 4 and Table 7, we get a better understanding of the complementary nature of both these two methods. While stochastic methods do not discretize the phase-space and can be used as reference methods, their stochastic nature means that a sufficient number of histories needs to be run for each problem studied. This means that coupling stochastic problems with thermo-hydraulics codes is difficult because feedback from the thermo-hydraulics code means running new histories to lower the uncertainties and can become too computationally costly. Deterministic methods are much better suited for coupled schemes and are often much quicker to run. As discussed in §1.1 some situations are difficult to study with both stochastic and deterministic methods. To treat these problems, different hybrid methods have been developed.

2.4 EXISTING HYBRID METHODS IN REACTOR PHYSICS

Deterministic and stochastic methods each have their strengths and weaknesses. The errors of discretization of the phase-space and errors due to the self-shielding process are the main weaknesses of deterministic methods. Monte Carlo methods can be very computationally intensive because they simulate individual particles and are by nature stochastic. Different responses have been developed to overcome these weaknesses.

2.4.1 Coupling to treat shielding problems

Modeling nuclear facilities can be quite a challenge due to their large dimensions and complex geometries. This can make deep penetration problems all the more difficult to handle for traditional methods.

2.4.1.1 Hybrid calculations separating global problem into multiple problems

Chen found [39] that the main difficulties of shielding design and analysis came from the complex geometry of the target and/or source region. The thickness of the shield around the target/source, and charged particle transport need to be modeled. Chen said that Monte Carlo methods are unable to give satisfactory results for deep penetrating simulations and that discrete-ordinate-deterministic methods are limited in treating complex geometries and cannot handle charge particle transport. This led to the development of a coupled scheme where the shielding problem was separated into a source region treated by the Monte Carlo method and a bulk shield region treated by the deterministic method, which was implemented as follows in Algorithm 4.

Numerical Steps in Chen's MC-DO [39]
Monte Carlo calculation over the source region and store particle tracks crossing the link surface in the binary file
Initialize the deterministic model using the link source as the source input
The interface program reads the link file and defines meshes, energy group structure, and angular quadrature sets
The user-created input file to define the remaining parameters
Discrete ordinates deterministic calculation of the bulk shield region

Algorithm 4: Implementation of Chen's coupled shielding MC-DO method

In this method, the code is separated into two regions to study but the two codes use do not interact much. The shield region uses the source region data to initialize its second calculation but no feedback is implemented. The interaction between the two codes is weak.

2.4.1.2 Hybrid calculations using variance reduction methods

Instead of separating the problem into two regions, it is also possible to use the deterministic method to initialize the Monte Carlo problem. Becker et al. [40] also looked at deep-penetration problems without dividing the problem and implemented a solution as follows in Algorithm 5. Becker et al. used the deterministic scalar flux to define a correction of said flux, the "correcton flux". This allowed them to transform the stochastic problem to solve. Becker et al. showed that correctons propagate to deep parts in fewer collisions and in greater numbers than the usual particles. This allows their method to obtain flux tallies over the whole problem and not only around a detector, as would be the case with traditional weight window Monte Carlo simulations.

Numerical Steps in Becker et al. [40]
Run an inexpensive deterministic global estimate of the forward flux
Transform the original problem of the angular flux into a modified problem of the multiplicative correction to the deterministic flux
Run a Monte Carlo calculation to estimate the multiplicative correction of the deterministic flux (correcton flux)

Algorithm 5: Implementation of the coupled method by Becker et al.

Another way to modify the Monte Carlo simulation using deterministic methods is studied by Guadagni et al. for dose rate calculations [41]. They worked on the implementation of a deterministic estimator to speed-up Monte Carlo calculations. This modified track-length estimator can be seen as a combination of deterministic straight-line transport and a Monte Carlo track-length estimator simulation. Using

deterministic straight-line in the estimators means that particles can contribute to the score as long as “virtual” line of flight intersects the scoring volume, even if the real particle does not cross the scoring volume. The deterministic aspect comes from the path the virtual particle is being sent on and this method is implemented as shown in Algorithm 6.

Numerical Steps in Guadagni et al. [41]

The user sets a bounding sphere to each detection volume

Loop over bounding spheres

- Create a virtual copy of the original particle. The virtual copy is a replica of the original particle
- Sample a random direction in the solid angle. A solid angle subtended by the sphere is calculated based on the bounding radius and the distance between the interaction point and the sphere center.
- Sample a virtual collision for the virtual particle to adjust its weight and assign the previously obtained direction
- Calculate the straight-line distance between the virtual particle and the closest volume frontier
- The particle is moved to the volume frontier and data is stored
- Correct weight of the particle exponentially
- Repeat until the virtual particle reaches a domain boundary
- Check if the particle has crossed the detector volume. If crossed, register the score and then kill the particle
- Kill the virtual particle and initialize transport quantities. Go back to the transport of the real particle

Algorithm 6: Implementation of the hybrid estimator by Guadagni et al.

In analog Monte Carlo simulations, a particle is transported through each encounter and this process is repeated until the desired statistical precision is reached. Some particles may not reach the region of interest and will not contribute to tallies and therefore transporting these particles can be exorbitantly expensive depending on the problem. Variance reduction methods were developed to avoid simulating large quantities of particles and without obtaining the desired precision. These methods modify the weights of the particles to be simulated to bias the simulation.

Two different hybrid methods developed and in use at Oak Ridge National Laboratory [42] are hybrid variance reduction methods. Both CADIS and FW-CADIS use automated variance reduction methods. The importance map used by the weight windows is obtained from a deterministic calculation. The CADIS method is based on the idea that the adjoint function can be considered as the importance of a particle and implemented as shown in Algorithm 7. Sampling the deterministic source distribution means that the particles are obtained in proportion to their expected contribution to the detector response. The CADIS method is aimed at the optimization of localized detector regions while FW-CADIS is aimed at the optimization of distributions.

Numerical Steps in CADIS and FW-CADIS
Run a deterministic calculation to get either forward (FW-CADIS) or adjoint (CADIS) flux distribution
Calculate variance reduction parameters
Run a Monte Carlo simulation

Algorithm 7: Implementation of the CADIS and FW-CADIS method

These new methods have led to faster calculations for both local (CADIS) and global (FW-CADIS) simulations. The forward calculation in FW-CADIS allows the method to evenly distribute the particles across the problem. This methodology can now accurately treat global problems and can be used in large and complex problems.

To deal with large and complex problems, such as shielding problems, different hybrid methods were imagined. Chen developed a 1-way stochastic-deterministic coupling to deal with deep-penetration problems and chose to separate the problem into two regions. Becker et al. did not split the problem, instead, they chose to initialize the Monte Carlo calculation with deterministic data and convert the angular flux problem to a correction flux problem. This way they ensured more particles covered the whole problem. Guadagni et al. chose to only focus on the detector, which lead them to develop a hybrid estimator. This quick overview gives us a glance at the variety of hybrid methods that can be imagined depending on the problem to be solved, the regions of interest, and the codes available among others.

2.4.2 Coupled calculations to treat reactor core problems

When the region of interest is made up of a few subassemblies and accurate high-fidelity results are not always required over the whole core, refining a whole core calculation can be very costly. However, with traditional methods, the accuracy of the model is the same over the whole problem. Hybrid methods are being developed to overcome this issue and to obtain higher fidelity results at a lower cost than traditional methods. The following sections illustrate the different methods imagined and their implementation.

2.4.2.1 Deterministic coupling of angular and/or spatial methods

Wang et al. [43] implemented a hybrid P_N-S_N calculation scheme in Rattlesnake. This angular hybrid scheme made it possible to treat regions of interest with a higher resolution using the discrete ordinates method (S_N) while other regions were treated with a spherical harmonics expansion method (P_N). According to Wang et al. [43], P_N methods yield more accurate results than S_N methods for problems with homogenization using the same number of unknowns.

In his work [26] [44] [45], Girardi uses a non-overlapping spatial domain decomposition method to couple different deterministic methods depending on the geometry to be modeled. In this method, named $M_M M_D$, Girardi defined coupling operators in order to translate the boundary conditions between different subdomains and methods. The goal of the $M_M M_D$ method is to allow for local spatial/angular refinement using domain decomposition methods and is implemented with the steps shown below in Algorithm 8.

Numerical Steps in Girardi's $M_M M_D$ [26]

- Calculate the coupling coefficients
- Initialization of the outer iterations (fission source and k_{eff})
- Outer iteration
  - Loop over groups
    - Coupling iteration
      - Loop over subdomains
        - Calculate the incoming boundary condition
        - Call the monokinetic solver (IDT, VNM, or TDT)
        - Storing the outgoing boundary condition
    - Update the fission source dot product
  - Update k_{eff} and normalize fission sources using k_{eff}
  - Continue until convergence is reached (k_{eff} and fission source)

Algorithm 8: Implementation of the $M_M M_D$ method

2.4.2.2 Deterministic-stochastic coupling to treat the energy spectrum

Lee et al. [46] proposed a method directly coupling deterministic and Monte Carlo methods, where the Monte Carlo method was used to treat the resonance energy range. The method aims to benefit from both the accuracy of the Monte Carlo method over the resonance range and the fast execution of the deterministic method over high and low-energy ranges. The implementation of this method is described in Algorithm 9. The Monte Carlo simulation is a fixed-source simulation and the source is a sum of the down-scattered neutrons from the deterministic fast range and fission sources from the group of interest. In this hybrid method, there is some overlap between the stochastic and deterministic methods. The Monte Carlo method also includes the thermal range to account for upscattering from thermal energies, but tallies are only produced by the resonance range.

Numerical Steps in Lee et al. hybrid method [46]

- Fast region calculation using the deterministic method
- Sample scattering source for the Monte Carlo fixed source calculation
- Resonance region calculation using the Monte Carlo method
- Thermal region calculation using the deterministic method
- Continue until convergence is reached (k_{eff})

Algorithm 9: Implementation of Lee et al. energy hybrid method

2.4.2.3 Deterministic-stochastic coupling to treat subassemblies

During his Ph.D., Baker [47] implemented a coupled Monte Carlo-discrete ordinates scheme using response matrixes. He separated the problem into two regions: a Monte Carlo region and a deterministic one. He expressed the outgoing flux from the Monte Carlo region using a precalculated response matrix and outgoing source term (2.39).

$$\Psi^{out} = \boldsymbol{R}\Psi^{in} + S^{out} \qquad (2.39)$$

S^{out} being the exiting flux from the Monte Carlo region under vacuum boundary conditions. The response matrix is the angular flux leaving the Monte Carlo region for state k due to a unit incident angular flux in

state k'. Ψ^{in} is not generally known and can be solved iteratively by setting the following system for each iteration r.

$$\begin{cases} \Psi^{out(r+1)} = \boldsymbol{R}\Psi^{in(r)} + S^{out} \\ \Psi^{out(1)} = S^{out} \end{cases} \quad (2.40)$$

Using this, it is possible to perform a discrete ordinates calculation and obtain $\Psi^{in(r)}$. This scheme is illustrated in Algorithm 10.

Numerical Steps in R.S. Baker Ph.D. [47]

Define the Monte Carlo cross-sections and initialize data required by Monte Carlo functions

Samples the fixed source to determine S_g^{out} and computes the volumetric sources in the S_N region resulting from the Monte Carlo calculation

Loop over energy groups

> Calculate the response matrix R_{gg}

Initialize S_N cross-sections and functions required by S_N calculation

Loop over energy groups

> Calculate the volumetric source in group g for both Monte Carlo and S_N regions
>
> Determine the contribution to group g of the volumetric sources located in the Monte Carlo region
>
> Initial outgoing boundary flux is calculated
>
> Perform S_N inner iteration to get ${\Psi_g}^{in(r+1,1)}$ and ${\Psi_g}^{out(r+1,2)}$
>
> Calculate maximum relative error between ${\Psi_g}^{in(r+1,1)}$ and ${\Psi_g}^{out(r+1,2)}$
>
> Continue until convergence is reached

Algorithm 10: Implementation of the hybrid method for Baker's work during his Ph.D.

In this method, regions are treated with either a Monte Carlo or deterministic method, the response matrix is used to update the boundary condition with the deterministic calculation. Baker found that precalculating the response matrix reduces the computational cost of the method.

Another way to use response matrixes is shown in COMET [25]. COMET is a hybrid stochastic-deterministic solver capable of treating whole core problems. It uses solutions from local fixed sources problems to generate response functions and iterates the deterministic method to converge the eigenvalue problem, as shown in Algorithm 11. It does not directly solve the multigroup transport problem.

Numerical steps in COMET [48]

- Local calculations to generate response functions (stochastic code)
- Global calculations to converge whole core solution (deterministic solver)
  - Guess initial eigenvalue and partial currents
  - Perform inner iterations to converge on the partial current moments for the eigenvalue computed in the previous step
    - Normalize the outgoing partial current moments
    - Use the external boundary condition or the internal interface boundary condition to update the incoming partial current moments
    - Compute the outgoing partial current moments via the pre-computed response coefficients
    - Repeat until moments are converged
  - Use the global particle balance to update the eigenvalue
  - Repeat until the eigenvalue is converged
- Superposition of local problem solutions to construct the flux distribution in the core

Algorithm 11: Implementation of the hybrid method in COMET

COMET aims to be a high-fidelity whole core solver and obtains comparable accuracy to multigroup stochastic solvers significantly faster [48]. The hybrid method implemented in COMET accelerates the computational speed and fission source convergence for large eigenvalue problems compared to stochastic methods.

2.4.2.4 Stochastic-stochastic coupling to treat the energy spectrum

The hybrid methods thus far considered all used deterministic methods to accelerate the continuous energy Monte Carlo method. However, multi-group Monte Carlo methods are also faster than continuous energy Monte Carlo methods. Kowalski et al. [49] [50] worked on combining these two methods in a single Monte Carlo calculation. This coupled method is implemented in SCONE, which is already supported in both continuous energy and multi-group stochastic calculations. Unlike the previous methods, the only modification necessary is the transition from multi-group to continuous energy.

Multi-Group to continuous energy transition in SCONE [49]

- If the group is a resonant group
  - Draw energy from uniform function in lethargy
  - Determine total cross-section data from sampled energy
  - Repeat until converged
- Otherwise, draw energy from uniform function in lethargy

Algorithm 12: Implementation of the energy selection from multi-group to continuous energy in SCONE

Kowalski and Shwageraus [50] observed that most of the error introduced in the continuous region by the multigroup calculation is located close to the boundary between the two regions.

Hybrid methods open a world of possibilities to solve deep-penetration problems and core problems as shown above. These methods have yielded interesting gains in precision, and particularly in calculation time when dealing with deterministic-stochastic hybrid methods.

2.5 HYBRID METHOD STUDIED IN THIS WORK

Hybrid methods in reactor physics cover a large range of methods from acceleration methods of stochastic codes and protection problems to coupled stochastic-deterministic codes. Hybrid methods can also refer to coupled deterministic codes. From this point on, hybrid methods will refer to deterministic-stochastic coupled schemes for neutron transport.

2.5.1 Choice of hybrid method

The irradiation experiments carried out by the ICARE program [51] in the MELUSINE reactor in Grenoble are examples of whole core configurations for which traditional methods have reached their limits. In the ICARE program, the experimental devices were designed to measure integral capture cross-sections of fission products and actinides in PWR and under-moderated conditions. The experimental subassemblies consist of stacked UO_2 depleted pellets and doped pellets with isotopes of interest. On the one hand, a 3D deterministic analysis of the configuration is complicated by the geometry of the experimental subassembly. A 3D-self-shielding calculation is difficult to perform due to the sequence of stacked pellets. In APOLLO2 and APOLLO3®, there is no 3D-self-shielding method, each plane is self-shielded with a 2D scheme. Resonances perturb the energy spectrum seen by target nuclei. On the other hand, the depletion calculation requiring control rod movement during irradiation makes it extremely costly to perform a full Monte Carlo analysis of these experiments. We chose to focus on studying such problems with the development of our hybrid method.

This work also focuses on hybrid methods using domain decomposition methods (DDM). These methods can allow users to select methods to treat different parts of a problem. The goal of this work is to implement a 2-way coupled hybrid method using DDM. In this method, the core is separated into subdomains and the user can choose to apply either a stochastic or deterministic method. The DDM control the boundary flux exchanges between subdomains. This method does not require any assumptions about the geometry in either region. This could be beneficial to studying the irradiation experiments of the ICARE program. For instance, experimental devices would be treated using a Monte Carlo method, while the rest of the core where the self-shielding problem is not as complicated are studied using a deterministic method. This could also be beneficial for irradiation cycles because the Monte Carlo region would be smaller which should accelerate convergence.

The basic scheme of this 2-way coupled hybrid method would be to start with a problem to study. On the right-hand side of Figure 12, an arbitrary quarter of a core was chosen to illustrate the methodology. Using the deterministic method (IDT in our case), the user would separate the core into subdomains and specify which method to use for each subdomain. In Figure 12, each subdomain corresponds to a subassembly and two subassemblies would be treated using the Monte Carlo method (LAST in our case). We think it is best to limit the number of subdomains to be treated with the Monte Carlo method to ensure that the stochastic region remains small keeping calculation costs down.

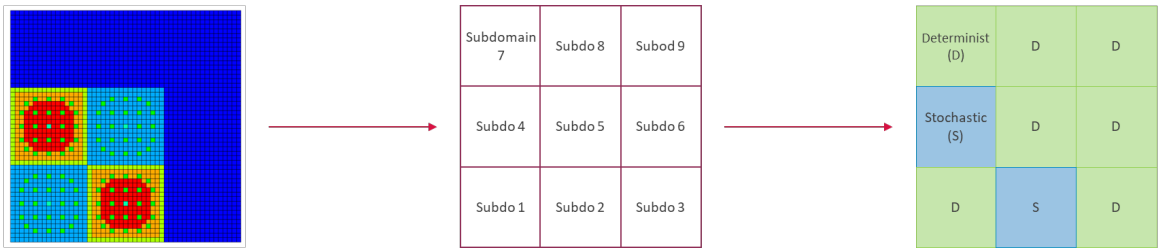


Figure 12: Main steps to separate a core problem in the 2-way coupled hybrid methods – Left: core to study – Middle: Separation into subdomains – Right: Calculation scheme applied to each subdomain

An important aspect of the method to develop is the intention to use the deterministic method's spatial domain decomposition scheme to control our hybrid method. The goal is to use this scheme to exchange data between the two methods. This means that all developments during this work aim to be compatible with the already implemented domain decomposition scheme.

2.5.2 Predicted difficulties and development plan

Developing this coupled scheme is not trivial due to the different nature of these methods. It is assumed that at least the following difficulties will arise:

- Communications between Monte Carlo and S_N methods will have to deal with statistical estimators. This raises the question of how the "noise" from the Monte Carlo simulation will affect the convergence of the deterministic calculation. Relaxing certain convergence criteria may be sufficient.
- Calculation time difference might be avoided by limiting the size and/or the number of the Monte Carlo subdomains. Using deterministic sources to initialize the Monte Carlo problem might also help reduce the stochastic calculation time.
- Monte Carlo results being quasi-continuous means that part of the information sent from the deterministic code will be missing. To overcome this issue, we will have to sample the deterministic distribution.
- Fitting the hybrid scheme in the deterministic DDM already implemented. In IDT, a non-overlapping spatial DDM is implemented. This means that subdomains do not have common cells. Using an overlapping DDM, it would be possible to use the boundary fluxes in the overlapping cells as a function with which to reconstruct a coarse flux, which might facilitate exchanges. This strict separation of subdomains means that some subdomain boundaries might be where our flux needs to be studied. To limit this issue, subdomains should be a few cells larger than the region of interest.

To overcome these difficulties, the development of the hybrid method will be separated into the following steps: deterministic-deterministic hybrid method in IDT to modify DDM, deterministic to stochastic coupled calculation to have a 1-way hybrid, stochastic to deterministic 1-way hybrid, deterministic-stochastic 2-way hybrid using the two previous steps.

2.6 CONCLUSIONS

Neutronic modeling is traditionally performed using either deterministic or stochastic methods. Deterministic codes discretize the phase-space and introduce models to solve the transport equation. Self-shielding can be an important a source of bias. Spatial discretization methods cannot be used to treat any and all geometries. These methods play a great role in multi-physics computational schemes. Monte Carlo methods can be very computationally intensive because they simulate individual particles using random sampling. They are used as reference codes but their cost and problematic statistical convergence for large domains limits their use. In an effort to improve numerical modeling of reactor problems, hybrid methods are being developed. Therefore by combining both deterministic and stochastic methods in the same calculation, as a hybrid method, we can take advantage of the benefits offered by each method.

Deterministic-stochastic hybrid methods today can be split into three categories: variance reduction methods, weakly-interacting-coupled codes and strongly-interacting-coupled codes. Variance reduction methods are mainly used and developed to treat shielding problems. They bias the simulation by changing the particle weights. This makes it possible to simulate fewer particles and ensure their satisfactory results over the detector region. These methods were originally developed for local detectors, but have been adapted to be used over large problems (eg: FW-CADIS [42], and Becker et al [40]). Hybrid methods with no feedback between the two methods are referred to as weakly interacting methods. In Chen's MC-DO method [39], the discrete ordinates calculation is initialized by the Monte Carlo simulation. This shows that in some problems, initializing our problem with one method can lead to important gains in both precision and calculation time. Strongly interacting methods do rely on feedback between the two methods to ensure convergence. Girardi [26] and Lee et al. [46] developed methods where running both methods at the same time in a hybrid method yielded gains in precision. Implementing a hybrid method that runs both deterministic and stochastic calculations can add complexity. It creates new challenges, but in problems where the regions of interest depend on the surrounding problem, it can be very beneficial.

DDM separate a large problem into smaller sub-problems and updates the boundary conditions of each subdomain to account for neighboring phenomena. This means that as long as the boundary conditions

are compatible, subdomains can be treated with different methods. This can lead to a more flexibility to solve complex problems. Using a modified DDM scheme can allow the user to work with different spatial descriptions as was seen in Girardi's work [26]. Therefore, DDM seem to be naturally suited to strongly-interacting hybrid methods.

On key aspect of a deterministic and stochastic hybrid approach is the boundary exchange. The DDM implemented in IDT is currently managing subdomains with different descriptions. The first step is therefore to modify the DDM scheme to allow subdomains to have different energy discretization. This IDT update gives a better sense of how boundary exchanges are impacted when neighboring subdomains have different energetic refinements. This will also help us get a better understanding of the IDT source code for future modifications. This work is presented in the next chapter.

3. DEVELOPMENT OF A MULTIPLE-ENERGY-GRID METHOD USING DDM

Implementing a hybrid deterministic-deterministic energetic scheme serves different purposes. It was an important step in the development of a deterministic-stochastic scheme, giving us time to get a better understanding of IDT and particularly of DDM functions under normal usage. In a deterministic-stochastic scheme, we use a continuous energy stochastic code. A multiple-energy-grid deterministic hybrid makes it possible to anticipate some of the problems that will occur when exchanging coarse energy data with fine (or continuous) energy data. This work on the difficulties of unequal boundary flux exchanges is a first step towards discrete-continuous boundary flux exchanges (§3.1). After this, the study case is presented (§3.2). Three different flux reconstruction methods are implemented in IDT to deal with the unequal boundary data exchange. Section §3.3 compares these methods.

Contents

3.1 METHODOLOGY DESCRIPTION

Self-shielding is an essential step in deterministic schemes, and the number of energy groups chosen by the user can lead to significant differences in terms of precision and calculation times. One of the challenges of studying complex core problems is the wide variety of interfaces present in the problem, such as UOX-MOX or fuel-reflector. The materials present in the different subassemblies each have their specific resonances, which may overlap. This begs the question of choosing appropriate energy meshes and self-shielding methods to describe all these subassemblies. However, with a modified domain decomposition scheme, the user would be able to treat the resonances of each subdomain with refined energy meshes and/or advanced self-shielding methods, which could lead to interesting improvements in precision.

The aim here is to use the already implemented DDM in the deterministic solver IDT and to modify the existing scheme to support subdomains with different energy grids. Being able to apply different energy grids allows the user to obtain finer data over parts of the problem. Figure 13 illustrates how a core can be split into subdomains using IDT. In this example, each subdomain corresponds to a subassembly. The difference with the regular DDM calculation is that the user can specify which energy grid is applied to each subdomain. In Figure 13, the core is split into a core-reflector problem where the energy grids in the fuel and reflector are different.

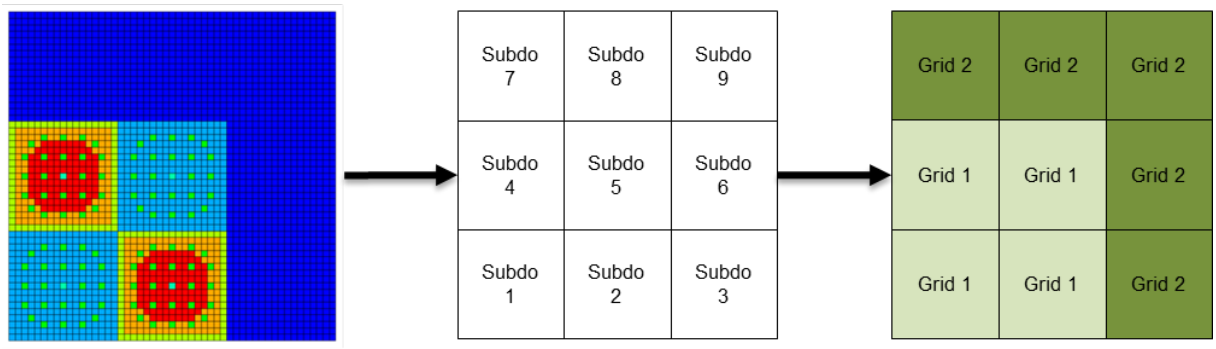


Figure 13: Illustration of how multiple energy grids could be used with DDM – Right: core to study – Middle: Core split unto subdomains – Left: Energy grid applied to each subdomain

By using multiple energy grids in the same problem, some regions will have more data. The boundary-exchange update step will need to be modified accordingly. Before changing the DDM scheme to use multiple energy grids, we start by taking a closer look at the boundary exchanges between subdomains.

3.1.1 Domain Decomposition modeling

The modification of the multigroup problem to accommodate domain decomposition methods was presented in §2.2.2.3 This led to the development of equations (2.29). We will write this system by separating the phase-space into U subspaces in equations (3.1).

$$\begin{gathered} X \equiv \cup_{u=1}^{U} X_u \\ X_u \equiv \left(\vec{r} \in D_u, \vec{\Omega} \in S^2, E \in \mathbb{R}_G^+\right) \\ \partial X_u \equiv \left(\vec{r} \in \Gamma_u^{\pm}(\vec{\Omega}), \vec{\Omega} \in S^2, E \in \mathbb{R}_G^+\right) \\ \Gamma_u^{\pm}(\vec{\Omega}) \equiv \left(\vec{r} \in \partial D_u, \vec{\Omega} \in S^2 : \vec{\Omega}.\overrightarrow{n_{u+}(\vec{r})} \lesseqgtr 0, E \in \mathbb{R}_G^+\right) \end{gathered} \qquad (3.1)$$

We also saw with (2.31) that the solution of multigroup transport equation can be decomposed over the subspaces and can be expressed as follows in (3.2). For this purpose, we simplified the flux for a specific group and direction to $\Psi(x)$.

$$\begin{gathered} \Psi(x) = \sum_u \chi_u(x)\Psi_u(x) \\ \chi_u(x) = \begin{cases} 1 \; if \; x \in X_u \\ 0 \; otherwise \end{cases} \end{gathered} \qquad (3.2)$$

Angular boundary fluxes are exchanged between subdomains to update the boundary condition. This decision was made to ensure the continuity of solution in the DDM scheme implemented in IDT. The continuity condition, shown below, preserves the global solution and particle balance when updating the local fluxes.

$$\Psi_u^+(x) = \Psi_v^-(x) \qquad x \in \partial X_u^+ \cap \partial X_v^- \; for \; all \; u \cap v \qquad (3.3)$$

To better understand how energy groups are exchanged, equation (3.3) is written as follows. With $\vec{\Omega}_d$ being the considered direction and g the considered energy group.

$$\Psi_u^+(x, \vec{\Omega}_d, g) = \Psi_v^-(x, \vec{\Omega}_d, g) \qquad (3.4)$$

Figure 14 is a zoom into Figure 13 to illustrate two neighboring cells and get a closer look at the boundary exchange between subdomains with different energy grids. In this situation, energy grid 1 (cell A) is finer than energy grid 2 (cell B). The spatial and angular discretizations are the same on either side. This means that the angular flux in grid 1 is finer in energy than the angular flux in grid 2. In this illustration grid 1 is made up of g groups while grid 2 is made up of G groups.

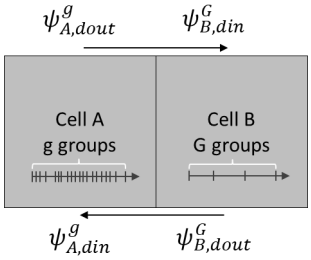


Figure 14: Boundary fluxes between two neighboring cells with different energy grids

From equation (3.4) and Figure 14, the modification of the continuity condition will only affect the energy component of the angular boundary flux. Depending on the boundary update, we can imagine two situations. The first being updating the B-incoming boundary flux, which will use the finer A-outgoing flux. The second being reconstructing the A-incoming boundary flux, which uses the coarser B-outgoing flux. To account for the changes presented above during the boundary flux exchange, we need to update the boundary condition in the scheme presented in Algorithm 3.

3.1.2 Matching energetic bounds

A new scheme is implemented to allow the user to run a calculation with multiple energy grids and is shown in Algorithm 13. In red are the differences with Algorithm 3. The boundary flux exchanges are no longer solely controlled by the continuity condition; there is now a distinction depending on whether the neighboring subdomains have the same energy grids or not.

Numerical Steps in IDT to solve the multigrid problem using DDM

Loop over external iterations — e=1,…,N_e

Loop over subdomains — s=1,…,N_s

Update the local fission source from $\Phi_s^{(e-1)}$

Multigroup approximation – Scattering source calculation — g=1,…,N_G

Update local scattering sources ($\forall g' \neq g$) from $\Psi_s^{(e),g'<g}$ and $\Psi_s^{(e-1),g'>g}$

Internal iterations — i=1,…,N_i

Solve the spatial sub-problem: sweep over the mesh and propagate from boundary conditions

Update the local self-scattering source $\boldsymbol{H}^{gg}\Psi_s^{(e,i),g}$

Update the boundary flux and exchange boundary fluxes between subdomains and check if energy grids different

If grids are the same

$$\forall g, \forall \vec{\Omega} \begin{cases} \Psi^{(e)}_{\Gamma\ subdo,in}(\vec{r},\vec{\Omega},g) = \Psi^{(e)}_{\Gamma\ neighbor,out}(\vec{r},\vec{\Omega},g), & \vec{\Omega}.\vec{n}_{ext} < 0 \\ \Psi^{(e)}_{\Gamma\ neighbour,in}(\vec{r},\vec{\Omega},g) = \Psi^{(e)}_{\Gamma\ subdo,out}(\vec{r},\vec{\Omega},g), & \vec{\Omega}.\vec{n}_{ext} \geq 0 \end{cases}$$

If neighboring grid is different from subdomain grid

$$\forall \vec{\Omega} \begin{cases} \forall G, \Psi^{(e)}_{\Gamma\ subdo,in}(\vec{r},\vec{\Omega},G) = \sum_{g\in G} \Psi^{(e)}_{\Gamma\ neighbor,out}(\vec{r},\vec{\Omega},g), & \vec{\Omega}.\vec{n}_{ext} < 0 \\ \forall g, \Psi^{(e)}_{\Gamma\ neighbour,in}(\vec{r},\vec{\Omega},g) = \alpha_{G\to g}\Psi^{(e)}_{\Gamma\ subdo,out}(\vec{r},\vec{\Omega},G), & \vec{\Omega}.\vec{n}_{ext} \geq 0 \end{cases}$$

Update k_{eff}

Convergence test on k_{eff} and power distribution

Reconstruction distribution at the core scale

Algorithm 13: Algorithm implemented in IDT to solve problems using DDM and multiple energy grids

Figure 14 and Algorithm 13 show that a test must be performed when updating boundary fluxes to check if neighboring grids are the same, or if the neighboring grid is finer, or if the neighboring grid is coarser. To perform these tests, energy meshes need to be defined by the user. During the data exchange between subdomains with different energy grids, it is necessary to distinguish sending fine information to a coarser grid with sending coarse data to a finer grid. The former means condensing information for it to fit. The latter means that some data will be missing and a fine-flux will need to be expanded from the coarse flux. The reconstruction of the fine-flux will not be trivial. Implementing this test means transforming the continuity equation (3.4) into the following system.

$$\begin{cases} \Psi_1^-(x,d,g) = \Psi_2^+(x,d,g) & if\ grid1 = grid2 \\ \Psi_1^-(x,d,G) = \sum_{g\in G} \Psi_2^+(x,d,g) & if\ grid1 < grid2 \\ \Psi_1^-(x,d,g) = \alpha_{G\to g}\Psi_2^+(x,d,G) & if\ grid1 > grid2 \end{cases} \quad (3.5)$$

From the new update boundary flux exchange in (3.5), it is clear that when the outgoing coarse flux from cell B is used to generate the cell A incoming fine flux, additional information must be created to have a complete fine-group description.

An elementary reconstruction model was initially considered in which the fine-flux was reconstructed using equation (3.6). We decided to call this reconstruction method the Basic Reconstruction Method (BRM).

$$\Psi_1^-(x,\vec{\Omega}_d,g) = \frac{\Delta u^g}{\Delta u^G}\Psi_2^+(x,\vec{\Omega}_d,G) \qquad (3.6)$$

An issue with this reconstruction method is that it does not account for the energetic variation of the flux within the coarse group. To overcome this problem, a fine-flux reconstruction using the partial outgoing integrated current $J_1^+(x,g) = \int_{\vec{\Omega}.\vec{n}>0}\Psi_1^+(x,\vec{\Omega},g)d^2\Omega$ is used. We called this the Current Reconstruction Method (CRM). In this method we consider the outgoing current, which is obtained using the outgoing directions.

$$\Psi_1^-(x,\vec{\Omega}_d,g) = \frac{J_1^+(x,g)}{\sum_{g\in G}J_1^+(x,g)}\Psi_2^+(x,\vec{\Omega}_d,G) \qquad (3.7)$$

While equation (3.7) is interesting the energy-angle distribution within a coarse group is hidden through the current. This led us work on this final reconstruction method that will be called the Flux Reconstruction Method (FRM).

$$\Psi_1^-(x,\vec{\Omega}_d,g) = \frac{\Psi_1^+(x,-\vec{\Omega}_d,g)}{\sum_{g\in G}\Psi_1^+(x,-\vec{\Omega}_d,g)}\Psi_2^+(x,\vec{\Omega}_d,G) \qquad (3.8)$$

Equation (3.8) only yielded results for constant MOSC and 0-th angular flux spatial moments. Moments with $m > 0$ are weighed with the 0-th moment factor to avoid null or negative values. Equation (3.8) becomes a system depending on the spatial moment.

$$\begin{cases} \Psi_1^{-,0}(x,\vec{\Omega}_d,g) = \dfrac{\Psi_1^{+,0}(x,-\vec{\Omega}_d,g)}{\sum_{g\in G}\Psi_1^{+,0}(x,-\vec{\Omega}_d,g)}\Psi_2^{+,0}(x,\vec{\Omega}_d,G) \\ \Psi_1^{-,m}(x,\vec{\Omega}_d,g) = \dfrac{\Psi_1^{+,0}(x,-\vec{\Omega}_d,g)}{\Psi_2^{+,0}(x,-\vec{\Omega}_d,G)}\Psi_2^{+,m}(x,\vec{\Omega}_d,G) \end{cases} \qquad (3.9)$$

Given that angular boundary flux are used to update the boundary conditions at each iteration modifications to the IDT solver only affected the boundary flux correspondences. Setting up flux exchanges between subdomains is not a trivial problem and is described with two neighboring cells in Figure 14. When information is transferred from fine to coarse, it needs to be collapsed. The natural and conservative solution is to sum up the fine components belonging to a coarse group. Transferring coarse information to a finer grid requires creating information in the missing cells. The three reconstruction methods in Table 8 were implemented in IDT.

	Basic Reconstruction Method (BRM)	**Current Reconstruction Method (CRM)**	**Flux Reconstruction Method (FRM)**
$\boldsymbol{\alpha_{G\to g}}$	$\dfrac{\Delta u^g}{\Delta u^G}$	$\dfrac{J_1^+(x,g)}{\sum_{g\in G}J_1^+(x,g)}$	$\dfrac{\Psi_1^+(x,-\vec{\Omega}_d,g)}{\sum_{g\in G}\Psi_1^+(x,-\vec{\Omega}_d,g)}$

Table 8: Reconstruction methods tested and the associated expression of the form factor

3.1.3 Choosing energy grids

Now that the boundary flux exchange procedure has been updated to accommodate multiple energy grids, the next step is to decide what rules need to be applied to the energy grids used. To ensure compatibility between the different energy meshes, the energy boundaries of the G-mesh are common with those of the g-mesh. Sharing the same coarse mesh facilitates boundary flux exchanges because no special consideration needs to be added at group boundaries. If energy meshes do not share the

same group boundaries, the reconstruction method might need to distinguish the reconstruction methods in the following situations (Figure 15). In Figure 15, the A energy mesh is coarser than B.

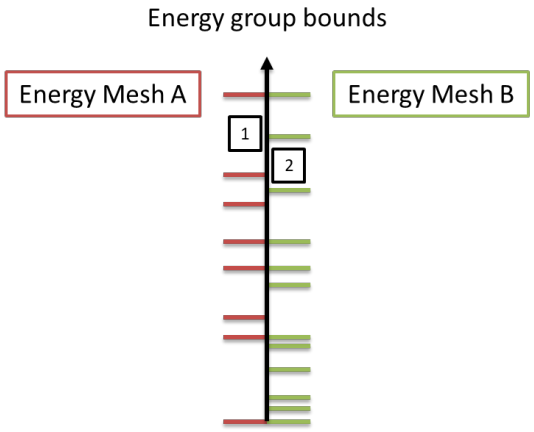


Figure 15: Energy meshes with unequal group boundaries

The first difficult situation to deal with in Figure 15 corresponds to the flux condensation in group 1 of mesh A, using fluxes in mesh B. The difficulty here is that the mesh B's second group does not have a common boundary with the group in which it needs to be condensed. The condensation can be expressed as follows.

$$\Psi_A^-(x,\vec{\Omega}_d,G_1) = \Psi_B^+(x,\vec{\Omega}_d,g_1) + \alpha_{overlap\ with\ mesh\ A}\Psi_A^+(x,\vec{\Omega}_d,g_2) \qquad (3.10)$$

The second challenge comes from a situation where the contribution to B's energy group comes from two of A's. This is situation number 2 in Figure 15. To deal with this situation, we would need to apply a reconstruction method to both energy groups and only consider the part overlapping with B's group.

$$\Psi_B^-(x,\vec{\Omega}_d,g_2) = \alpha_{overlap\ with\ mesh\ B}\alpha_{G\to g}\Psi_A^+(x,\vec{\Omega}_d,G_1) + \alpha_{overlap\ with\ mesh\ B}\alpha_{G\to g}\Psi_A^+(x,\vec{\Omega}_d,G_2) \qquad (3.11)$$

The challenge here is defining the link between the overlap between energy mesh reconstruction factor, and the group reconstruction factor.

Using any energy grid regardless of the common group boundaries, would be much easier for the user in an industrial code. However, using common group boundaries might lead to better results, since fewer distinctions need to be made to the reconstruction method. The goal of our method is to familiarize ourselves with IDT's DDM, so we chose to use energy grids with the same group boundaries.

Cross-sections are generated using the APOLLO3 [17] [25] code using the SHEM-MOC methodology and calculation scheme for 281 [52], 51, 23 and 7 groups. The 281-group cross-sections are self-shielded using Livolant-Jeanpierre methodology [53]. They are then partially homogenized (cell by cell) and collapsed into 51, 23 or 7 energy groups, without using any equivalence.

The 281-group mesh was chosen to be used as a deterministic-reference calculation. The 51- and 23-group meshes were chosen as intermediately coarse grids, while the 7-group mesh was chosen as the coarsest mesh. The 7-group mesh was also chosen in the benchmark presented in the next section. The case study is based on this benchmark as explained in §3.2. The reader can find the 51-, 23-, and 7-group energy meshes descriptions in the appendices.

3.2 2D C5G7 CORE

The 2D C5G7 benchmark [54] was originally developed by the OECD/NEA Expert Group in 2001 as an international benchmark to test the abilities of deterministic codes to treat reactor problems without spatial homogenization. It is a small water-cooled reactor made up of sixteen fuel assemblies: eight uranium oxide assemblies (UOX), and eight mixed oxide assemblies (MOX). These fuel assemblies are surrounded by a light water reflector. Figure 16 represents a quarter of this core with the appropriate boundary conditions. Each subassembly is a square lattice of 17x17 pin cells.

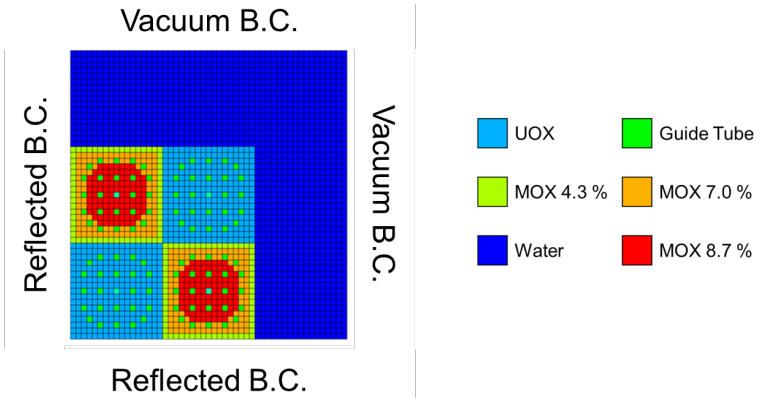


Figure 16: 2D core geometry of the C5G7 benchmark and boundary conditions

Seven-group cross-sections are provided for this benchmark to describe the neutron behavior in each medium. Cross-sections were generated for a fuel-clad mixture, as illustrated in Figure 17. According to the benchmark description [54], the 7-group cross sections were obtained using the densities and dimensions specified by Cathalau et al. [55] using the DRAGON code [56]. Each fuel type was modeled in an infinite-lattice using WIMS-AECL 69-group library. After this, the results were collapsed to seven energy groups.

3.2.1 C5G7 problem description

To test the multiple-energy-grid technique, 2D cases inspired by the C5G7 benchmark were created. Geometry and boundary conditions remain the same, the only difference comes from generating our own cross-sections. As mentioned above, the new cross-sections are generated using the SHEM-MOC calculation scheme. Another residue of the benchmark is the use of a single cross-section set to represent all UOX or all MOX fuel cells in the core, whether they are located in the center of the core or bordering the reflector. The assumptions made in the 2D C5G7 benchmark generate calculation biases in the new cases as well. We chose to produce the cross-sections with the same assumptions but using the Livolant-Jeanpierre self-shielding method implemented in APOLLO3®. The idea was to keep the philosophy of the 2D C5G7 benchmark, knowing that identical results will not be obtained by generating cross-sections with up-to-date methods.

Cell geometry is kept simple as described in the 2D C5G7 benchmark. Fuel cells are described as heterogeneous Cartesian cells [24], while reflector cells are homogeneous as illustrated below in Figure 17. In this document, fuel cells refer to heterogeneous Cartesian cells with either UOX or MOX fuel in them.

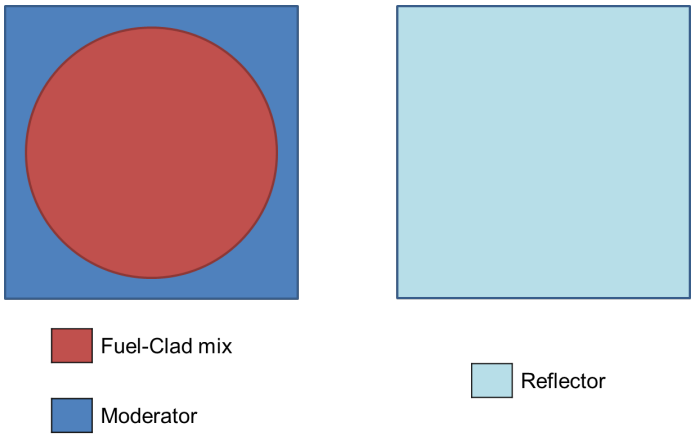


Figure 17: Pin cell geometry – Left: Fuel cell – Right: Reflector cell

In this study we decided to investigate the benefits of a multiple-energy-grid method on core/reflector interfaces. The core was separated into two mesh zones, the first zone in green is made up of all the fuel subassemblies, and the second zone in red is the reflector as shown in Figure 18.

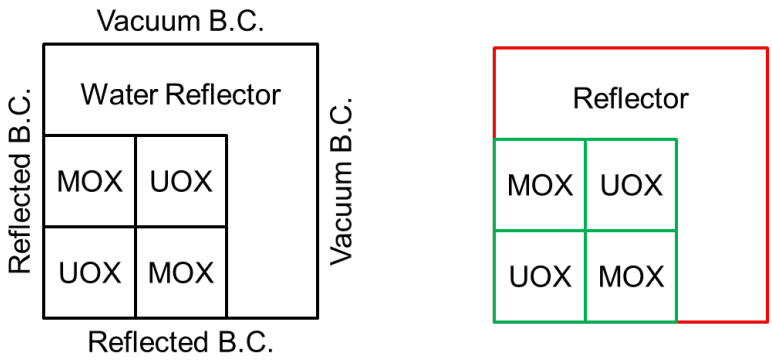


Figure 18: 2D core geometry of the problems to study – Left: boundary conditions – Right: energy mesh separations

The name of the different multiple-energy-grid cases is GX-Y, with X the number of groups in the fuel energy grid, and Y the number of groups in the reflector grid. When the fuel and the reflector grids are the same, cases will be called GX.

An accurate model of the reflector is necessary to ensure a proper estimation of the flux outside the core, and is necessary for vessel integrity studies. To study such problems, three configurations were inspired by the 2D C5G7 benchmark with modified reflector assemblies. In the first configuration, every aspect of the C5G7 geometry was conserved, as shown on the left side of Figure 19. In our second problem, reflector assemblies are inspired by PWR reflector subassemblies. A typical PWR reflector subassembly is approximately made up of 2 cm of stainless steel and 18 cm of water. To approximate this, the first two rows of cells bordering the core are made up of steel coming from the PERLE experiment carried out on the EOLE reactor at CEA Cadarache [57]. This amounts to 2.52 cm of steel and 18.9 cm of water, which mimics the PWR baffle. The last configuration is based on EPR reflector subassemblies. These reflectors are constituted by 20 cm of stainless steel. In our case, the reflector subassemblies contain 17 rows of steel cells (21.42 cm of PERLE experiment steel) as shown on the right in Figure 19.

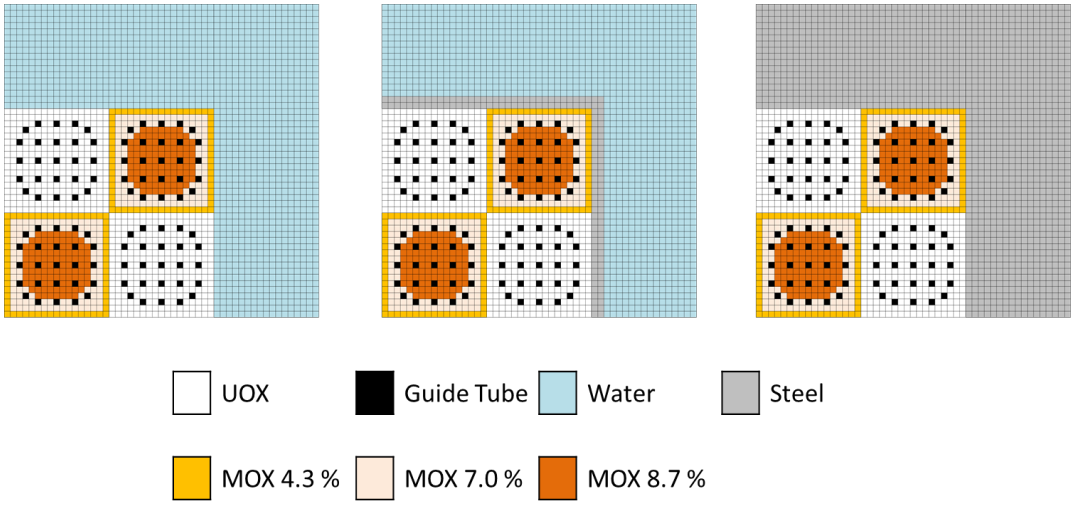


Figure 19: Reflector configurations created – Left: Light water reflector – Middle: Thin steel baffle reflector – Right: Heavy steel baffle

In this work, we will only discuss the results obtained over the heavy steel baffle reflector, which is the configuration on the right of Figure 19 and the most challenging. We chose this configuration because the results were the least satisfactory of the three. The reader can refer to two articles analyzing the other configurations [58] [59]. The first paper focused on comparing the results obtained for the left and middle geometries in Figure 19. It showed that the steel baffle led to poorer results in terms of absorption rates. However, the results obtained with each of the reconstruction methods suited was overall similar.

The second paper presented the illustration on the right of Figure 19, which is the one also discussed in this document.

3.2.2 Reference case

From this point on, the EPR-inspired heavy-steel-baffle reflector configuration will be referred to as the heavy reflector case.

This chapter is an investigation into a multiple-energy-grid scheme and our goal is to get a sense of the behavior and robustness of a locally refined scheme. We will compare the multiple-energy-grid results to single-energy-grid results, to understand the impact on the deterministic domain decomposition calculation.

Pin cell geometries are homogenized and cross-sections are the same regardless of the position of the cell to keep with the procedure of the 2D C5G7 benchmark. The cross-sections can then be condensed to generate the macroscopic cross sections for 51, 23, and 7 groups. No equivalence is performed, which means that the eigenvalues are different in all our single-energy-grid calculations.

Cell geometry is illustrated in Figure 17, and we decided to keep the fuel-clad mix in our calculation. This means that the pincell description is very different to a stochastic calculation, where pincells have to be described according to problem specifications. These differences in problem description will mean that are not able to separate the errors coming from the energy from the rest.

3.2.3 Single energy grid results

Before running the multiple-energy-grid scheme, it was important to quantify the differences between the different single-energy-grid calculations. The only difference in each case is the energy grid used which has an impact on the results.

Table 9 compares the single-energy-grid calculations. Eigenvalue differences to our reference case (G281 case) and calculation times (with and without using OpenMP parallelization) are all shown. OpenMP and MPI parallelization standards are both implemented in IDT. However, to be able to use MPI parallelization we needed to modify many of the domain decomposition functions. To be able to focus on the data exchange, which will be used by the hybrid scheme, we only used OpenMP parallelization in the multiple-energy-grid scheme. The lack of equivalence, when condensing cross-sections from 281 groups to 51, 23, and 7 groups, leads to 64, 105 and 535 pcm of reactivity difference to the reference case.

	Number of groups in each energy mesh			
	281	**51**	**23**	**7**
Eigenvalue difference (pcm)	0	64	105	535
Calculation time	10h57	2h58	1h17	16 min
Calculation time with OpenMP parallelization	3h49	1h02	22 min	5 min

Table 9: Global results obtained for each single energy grid calculation

Figure 20 represents the cell-by-cell difference of absorption rates between the reference and other single-energy-grid cases. Absorption rates are integrated over each cell and normalized by setting the total production rate to 10^5 pcm.

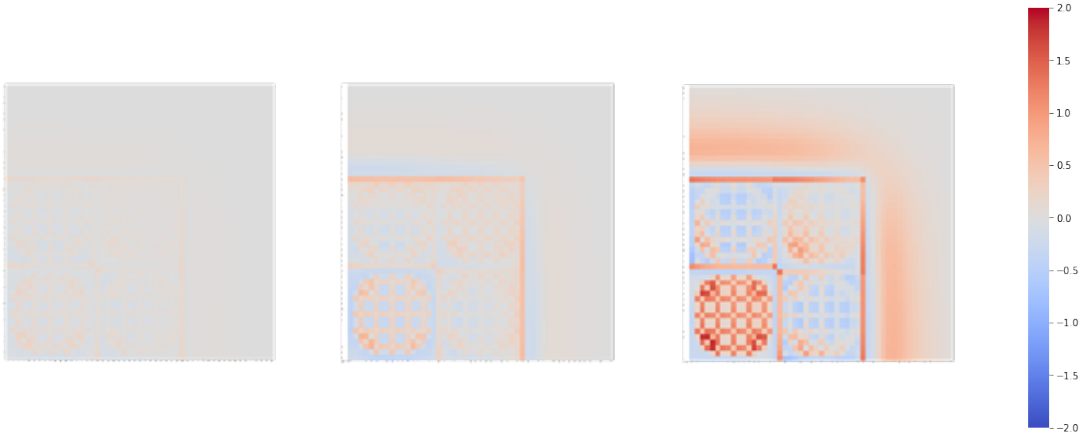


Figure 20: Cell by cell absorption difference with reference case – Left: G51 – Middle: G23 – Right: G7

As expected, the error in absorption rate calculation increases as the number of energy groups decreases. It can be noted that the largest differences are along the interface between the core and the reflector, and in the central UOX subassembly. The error in the UOX subassembly might be a remnant of the C5G7 benchmark cross-sections definition. By using the same cross-sections regardless of the position in the core, errors are introduced in the calculation. As shown in Figure 18, one of the UOX subassemblies is located at the center of the core and is surrounded by MOX fuel subassemblies, while the other UOX subassembly is bordering the reflector. The neutron behavior in these two subassemblies is very different, which leads to the errors obtained in Figure 20.

In Table 10, the root means squared (RMS) error, the maximum and minimum difference with G281 reference case are synthesized.

		Number of groups in each energy mesh							
		281		**51**		**23**		**7**	
RMS	ε_{max}	0	0	0.002	0.26	0.02	0.70	0.18	2.01
	ε_{min}		0		-0.13		-0.39		-0.76

Table 10: Maximum, average and minimum difference in pcm between single-energy-grid cases

Table 9, Figure 20, and Table 10 confirm the expected increase in error and decrease in calculation time with the decrease in the number of groups simulated. In the following section, we will discuss the overall convergence of the multiple-energy-grid cases using “global” results obtained in Table 9. Once the convergence of the multiple-energy-grid method has been shown, we will look at the effect of the scheme on each pincell using absorption rates.

3.3 Analysis of a modified 2D C5G7 core

3.3.1 Basic reconstruction method

Table 11 shows that the multiple-energy-grid eigenvalues are not in-between the two single-energy-grid eigenvalues.

		Number of groups in the reflector region (red)			
		281	**51**	**23**	**7**
Number of groups in the fuel's (green) cross sections	**281**	0	4	-42	20
	51	61	64	29	86
	23	115	120	105	168
	7	498	512	501	535

Table 11: The difference in pcm between the obtained eigenvalue and the reference case using the BRM

From Table 11, it can be seen that most of the eigenvalues are bounded by the corresponding single-energy-grid calculation. The eigenvalue difference in G23-7 (168 pcm) is lower than G7-23 (501 pcm) but are both bounded by the G23 (105 pcm) and G7 (535 pcm). However, this is not the case for G281-23, G51-23, G23-281, and G23-51. Both G23-281 (115 pcm) and G23-51 (120 pcm) are larger than G23 (105 pcm), which should be the higher bound. On the contrary, G281-23 (-42 pcm) and G51-23 (29 pcm) are both lower than G281 (0 pcm) and G51 (64 pcm) respectively by about 35 pcm.

Let us take a look at the calculation time presented in Table 12 to check if calculation times are also bounded.

		Number of groups in the reflector region (red)			
		281	**51**	**23**	**7**
Number of groups in the fuel's (green) cross sections	**281**	3h49	2h11	2h12	2h14
	51	4h30	1h02	52 min	30 min
	23	3h02	50 min	22 min	11 min
	7	2h48	47 min	27 min	5 min

Table 12: Calculation time with OpenMP parallelization (12 threads) using the BRM

Table 12 shows that every calculation time is bounded except G51-281 and G7-23. It takes 4h30 to run, which is much longer than the 3h49 it took to perform the single-energy-grid 281-group calculation. Given these results in Table 11 and Table 12, we want to take a closer look at the number of iterations each calculation took. We noticed in Table 13 that the maximum number of iterations (200) was reach for the G51-281 and G23-281 cases.

		Number of groups in the reflector region (red)			
		281	**51**	**23**	**7**
Number of groups in the fuel's (green) cross sections	**281**	No (91)	No	No	No
	51	Yes	No	No	No
	23	Yes	No	No	No
	7	No	No	No	No

Table 13: Maximum iterations reached using BRM

The previous tables indicated that combining the 23 and 51-group grids did not yield satisfactory results in terms of eigenvalue and calculation time. It seems surprising that a more localized assessment of the BRM would yield improved results. This is why we chose to focus on the combination of the 7 and 23-group meshes to look at the cell-by-cell differences for a local analysis of the BRM. Figure 21 represents

the cell-by-cell difference of absorption rates between the reference case and the multiple-energy-grid cases. The reader can find all the other cell-by-cell differences in appendices.

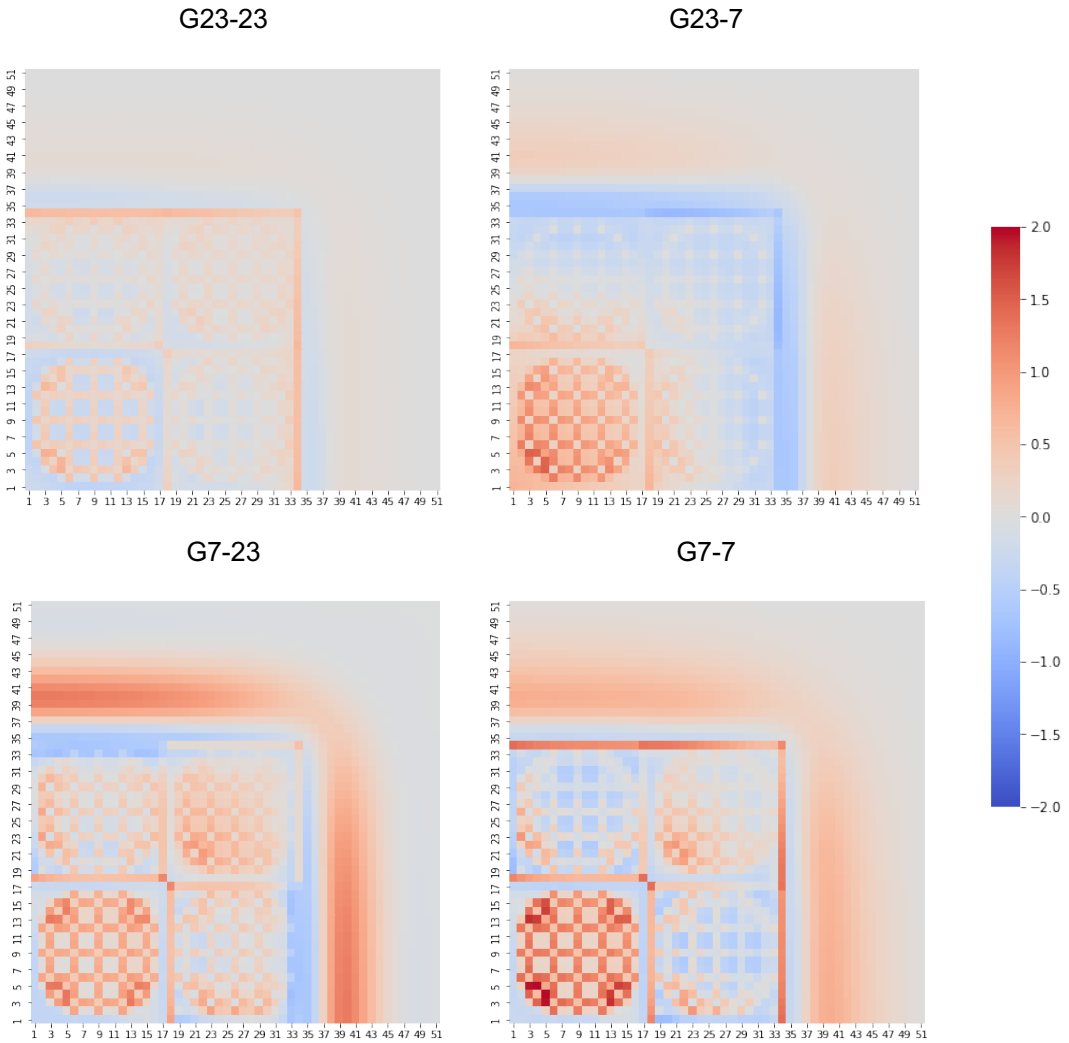


Figure 21: Cell-by-cell absorption-rate differences with the reference case using the BRM

Both the G23-7 and G7-23 calculation seem to have lower errors that the single-energy G7 calculation. This confirms that the BRM method did not increase errors for these cases. The BRM held up locally, absorption-rate differences were bounded by the single-energy-grid calculations, and globally, eigenvalues and calculation times as well. To check if this is the case for all the other calculation, in Table 14, the root mean squared (RMS) error, the maximum and minimum difference with the G281 reference case are synthesized. This table confirms the observations made in Figure 21, and also shows that most of the other calculations increase errors compared the single-energy-grid calculations.

		Number of groups in each energy mesh							
		281		51		23		7	
Number of groups in the fuel (green) region	281	RMS	ε_{max}	0.0006	0.16	0.31	2.90	0.59	3.02
			ε_{min}		-0.10		-2.52		-4.94
	51	0.07	0.37	0.002	0.26	0.17	2.24	0.38	2.13
			-0.19		-0.13		-1.74		-3.98
	23	0.03	0.90	0.17	2.24	0.02	0.70	0.16	1.86
			-0.41		-1.75		-0.39		-1.09
	7	0.42	1.82	0.39	1.94	0.33	1.66	0.18	2.01
			-1.04		-1.37		-0.99		-0.76

Table 14: Maximum, average and minimum difference using the BRM

This BRM does not yield satisfactory results and does not always converge. It shows that the reconstruction performs better for GX>Y calculations. This is why we want to move on to reconstruction methods that account for either the current or flux shape.

3.3.2 Current reconstruction method

Table 15 represents the difference between eigenvalues and the reference case. The eigenvalue difference in G23-7 (195 pcm) is lower than G7-23 (501 pcm) but both are bounded by the G23 (105 pcm) and G7 (535 pcm). It can be noted that the difference in eigenvalue is lower for GX>Y cases than GX<Y cases.

		Number of groups in the reflector region (red)			
		281	51	23	7
Number of groups in the fuel's (green) cross sections	281	0	3	0	90
	51	66	64	60	149
	23	119	115	105	195
	7	485	488	501	535

Table 15: The difference in pcm between the obtained eigenvalue and the reference case using the CRM

Unlike the BRM in §3.3.1, none of the multiple-energy-grid calculations reach the maximum number of iterations and all converge with about the same number of outer iterations. In Table 16, the calculation time using OpenMP parallelization is shown. Table 16 is encouraging as it shows that both eigenvalue differences and calculation times are bounded by the corresponding single-energy-grid calculations.

		Number of groups in the reflector region (red)			
		281	**51**	**23**	**7**
Number of groups in the fuel's (green) cross sections	**281**	3h49	2h12	1h46	1h36
	51	2h39	1h02	36 min	25 min
	23	2h28	49 min	22 min	9 min
	7	2h27	48 min	19 min	5 min

Table 16: Calculation time with OpenMP parallelization (12 threads) using the CRM

From Table 15 and Table 16, we can gather that the iterative process converges for all cross-section combinations using the CRM. It also seems that a local increase in precision results in an overall increase in accuracy. It can be noted that G23-7 is 40% shorter than G7-23. It appears that a coarse-group description of the reflector is beneficial to the calculation scheme while a poorer source description leads to performing more iterations to reach convergence. The difference in calculation time could also be because in our core there are four fuel subassemblies and five reflector subassemblies. This could be a contributing factor to the increase in calculation time.

In Table 16, we noticed that some multiple-energy-grid calculations are particularly long with little benefit to the eigenvalue difference. For instance, G7-281 lasts for 2h27 while the G23 and G51 calculations are respectively 22 min and 1h02 long. This is also the case for G23-281. It seems unrealistic to focus on such cases, when they would not likely be run to study core-reflector problems. Figure 22 represents the cell-by-cell difference of absorption rates between the reference case and the multiple-energy-grid cases. We choose to focus on four cases for readability and to illustrate that most of the error from the multi-grid scheme is located close to the interface and the central UOX subassembly as expected. The reader can find all cell-by-cell differences in the appendices.

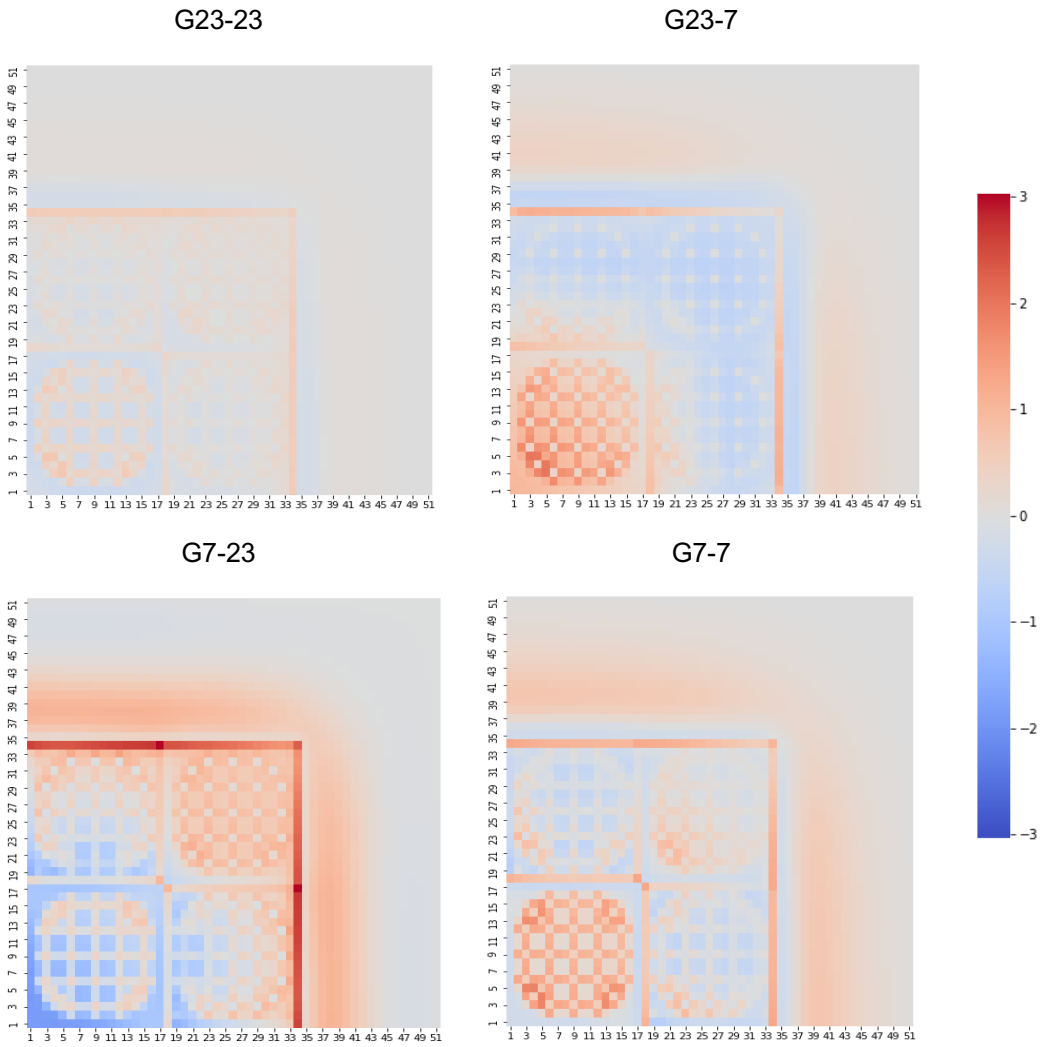


Figure 22: Cell-by-cell absorption-rate differences with the reference case using the CRM

The CRM showed interesting gains in calculation time and eigenvalue, but they do not hold up locally. If the CRM led to improvements, the maximum error should be obtained in the G7 case. The maximum error is obtained in the G7-23 case. This seems to indicate that in the G7-23 case, the CRM propagates and increases errors at the interface. In Table 17, the root mean squared (RMS) error, the maximum and minimum difference with the G281 reference case are synthesized. This table also confirms the observations made in Figure 22.

		Number of groups in each energy mesh							
		281		51		23		7	
Number of groups in the fuel (green) region	281	RMS	ε_{max}	0.0003	0.10	0.01	0.63	0.17	1.26
			ε_{min}		-0.03		-0.24		-0.73
	51	0.006	0.33	0.002	0.26	0.01	0.70	0.16	1.46
			-0.20		-0.13		-0.20		-0.66
	23	0.02	0.71	0.02	0.70	0.02	0.70	0.18	1.84
			-0.39		-0.38		-0.39		-0.67
	7	0.99	4.20	0.52	3.14	0.45	3.03	0.18	2.01
			-2.60		-1.94		-1.87		-0.76

Table 17: Maximum, average and minimum difference using the CRM

The errors in G7-Y cases using the CRM methods are all higher than the errors in the G7 case. Overall, the GX<Y cases all have larger or equal errors to the corresponding GX case using the CRM, while GX>Y all have lower errors. This effect was also noticeable in Table 15 and Table 16, and seems to indicate that the CRM is better suited to problems where the core is more refined than the reflector. The contribution to the source description mainly comes from the fuel subassemblies, which could be why having a better neutron feedback description does not significantly improve the coarse source description.

The CRM leads to better eigenvalues with shorter calculation times, but creates more errors in half the cases when looking at the absorption rates. A possible explanation of this problem might be due to the form factor itself. The simplification of the form factor (using the current) may not provide an appropriate description of neutron feedback.

3.3.3 Flux reconstruction method

Table 18 represents the difference between eigenvalues with the reference case using the FRM.

		Number of groups in the reflector region (red)			
		281	51	23	7
Number of groups in the fuel's (green) cross sections	281	0	3	0	90
	51	66	64	60	149
	23	119	115	105	195
	7	485	488	501	535

Table 18: The difference in pcm between the obtained eigenvalue and the reference case using the FRM

It is interesting to note that G23-7 is closer to G23 than G7-23 in both Table 15 and Table 18. This means that from the eigenvalue standpoint both the FRM and CRM yield better results for cases GX>Y and seems to indicate that this effect might be due to the geometry and problem separation.

Table 19 is also encouraging, it shows that both the multiple-energy-grid eigenvalues and calculation times are in-between the corresponding single-grid calculations.

		Number of groups in the reflector region (red)			
		281	**51**	**23**	**7**
Number of groups in the fuel's (green) cross sections	**281**	3h49	2h15	1h45	1h36
	51	2h39	1h02	36 min	25 min
	23	2h37	49 min	22 min	9 min
	7	2h46	48 min	19 min	5 min

Table 19: Calculation time with OpenMP parallelization (12 threads) using the FRM

The iterative process converges, and multiple-energy-grid results are in-between the corresponding the single-grid results.

Figure 23 represents the cell-by-cell difference in absorption rates between the reference case and the other cases. Only two multiple-energy-grid cases (G23-7 and G7-23) are presented in Figure 23 for the same reasons mentioned in §3.3.2.

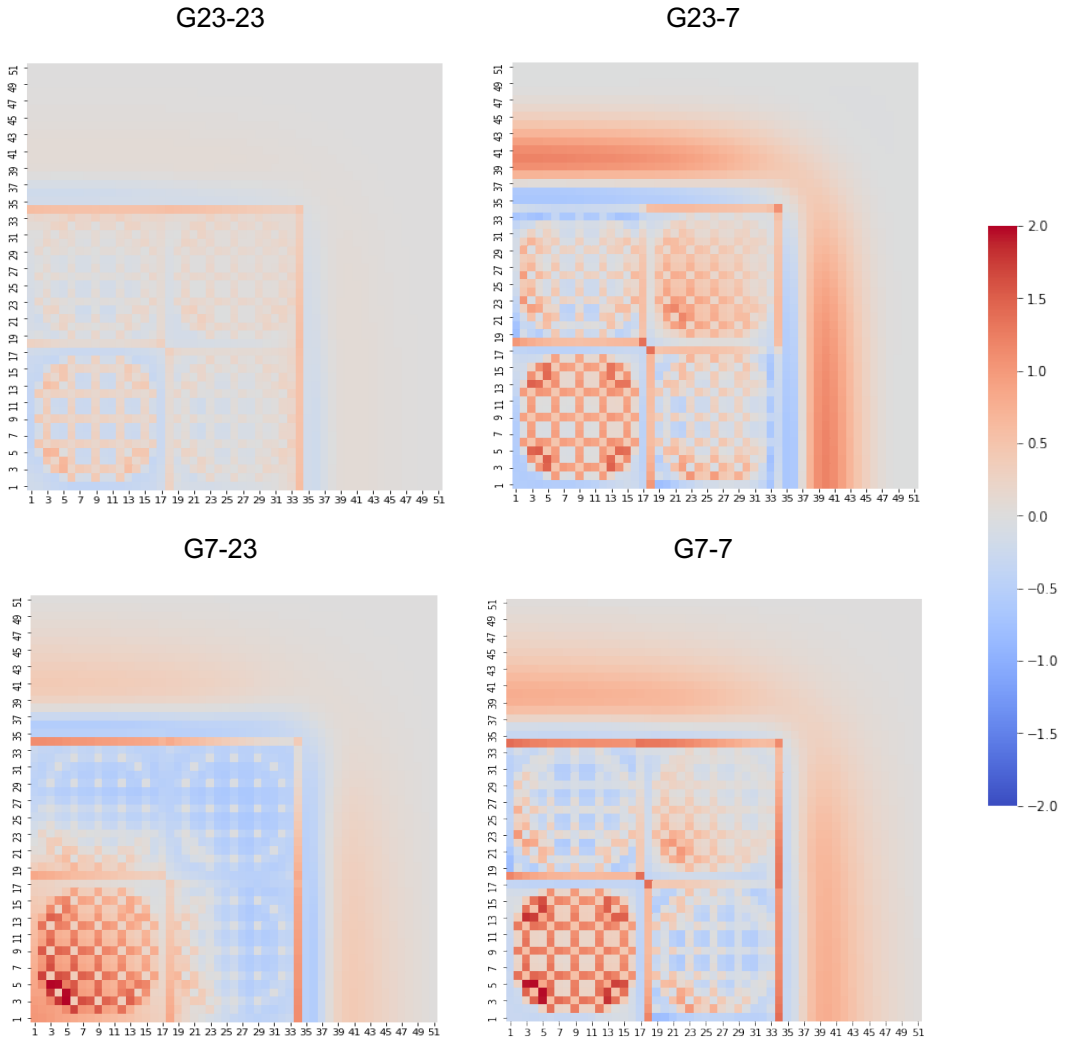


Figure 23: Cell-by-cell absorption-rate differences with the reference case using the FRM

In Figure 23, most of the error is located around the fuel-reflector interface. The maximum error is still obtained in the G7 case. This time the maximum error of the G7-23 case is no longer located along the interface between the two grids, but is now located in the central UOX subassembly. Table 20 confirms that while the maximum difference in absorption rates is obtained in the G7 case, theRMS and minimum

difference of the G7-23 case both exceed the errors obtained in the G7 case. This seems to indicate that the FRM creates more errors on average in the G7-23 case.

		Number of groups in each energy mesh							
		281		51		23		7	
Number of groups in the fuel (green) region	281	RMS	ε_{max}	0.0003	0.10	0.008	0.59	0.17	1.27
			ε_{min}		-0.04		-0.25		-0.72
	51	0.0004	0.10	0.002	0.26	0.011	0.67	0.17	1.48
			-0.06		-0.13		-0.20		-0.66
	23	0.02	0.94	0.02	0.91	0.02	0.70	0.18	2.00
			-0.23		-0.23		-0.39		-0.65
	7	0.37	1.79	0.26	1.41	0.23	1.54	0.18	2.01
			-0.94		-0.91		-0.80		-0.76

Table 20: Maximum, average and minimum difference using the FRM

From Table 20, it seems that the FRM is better suited to GX>Y problems, since it degraded other calculations by increasing the negative and average errors. However, the errors obtained in Table 20 are smaller than those obtained using the CRM in Table 17. The FRM seems to yield better results from the absorption rates point-of-view than the CRM.

The FRM also leads to better eigenvalues and shorter calculation times despite the assumptions made for the definition of the form factor. However, the average error increases when the fuel's energy mesh is coarser than the reflector's.

3.4 Conclusions

Implementing a multiple-energy-grid scheme in IDT makes it possible to treat subdomains with appropriate energy meshes, and leads to more accurate results compared to using the coarser energy grid over the whole core. This threefold approach shows gains in k_{eff} and calculation time. Out of the three methods developed, the flux reconstruction method (FRM) yields better results at the local level than both the current (CRM) and basic reconstruction methods (BRM).

Instead of considering a "fine" and a "coarse" energy mesh as we did in our work, in the future, it would be interesting to update the multiple-energy-grid scheme to handle different energy grids that deal with different resonances. All the grids would then share the same coarse mesh but each grid would be refined in a specific energy range as illustrated in Figure 24.

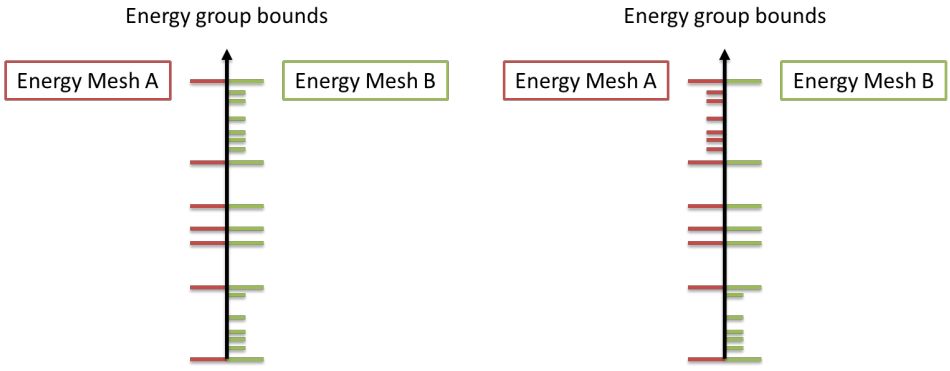


Figure 24: Illustration of energy meshes using the current multiple-energy-grid method (left) and a possible generalization of the method (right)

In this proposed update, reconstruction and condensation would not be a binary test, but would depend on the comparison for each group between the two grids. By only focusing on the boundary exchange step of Algorithm 13, we can illustrate (Algorithm 14) the proposed procedure to update the boundary condition for the multiple-energy-grid scheme on the right side of Figure 24.

Boundary flux exchange only from Algorithm 13

If energy meshes A and B are the same

$$\forall g, \forall \vec{\Omega} \begin{cases} \Psi^{(e)}_{\Gamma\ subdo,in}(\vec{r},\vec{\Omega},g) = \Psi^{(e)}_{\Gamma\ neighbor,out}(\vec{r},\vec{\Omega},g), & \vec{\Omega}.\vec{n}_{ext} < 0 \\ \Psi^{(e)}_{\Gamma\ neighbour,in}(\vec{r},\vec{\Omega},g) = \Psi^{(e)}_{\Gamma\ subdo,out}(\vec{r},\vec{\Omega},g), & \vec{\Omega}.\vec{n}_{ext} \geq 0 \end{cases}$$

If energy meshes A and B are different

Create a fine energy mesh M_g using all the groups from both A and B

Loop over groups in subdomain

If group in fine energy mesh M_g then group is g

Find all the fine groups in the coarse group

$$\forall \vec{\Omega} \begin{cases} \Psi^{(e)}_{\Gamma\ neighbor,in}(\vec{r},\vec{\Omega},G) = \sum_{g \in G} \Psi^{(e)}_{\Gamma\ neighbor,out}(\vec{r},\vec{\Omega},g), & \vec{\Omega}.\vec{n}_{ext} < 0 \\ \Psi^{(e)}_{\Gamma\ subdo,in}(\vec{r},\vec{\Omega},g) = \alpha_{G\to g}\Psi^{(e)}_{\Gamma\ subdo,out}(\vec{r},\vec{\Omega},G), & \vec{\Omega}.\vec{n}_{ext} \geq 0 \end{cases}$$

If group not in fine energy mesh M_g then group is G

$$\forall \vec{\Omega} \begin{cases} \Psi^{(e)}_{\Gamma\ subdo,in}(\vec{r},\vec{\Omega},G) = \sum_{g \in G} \Psi^{(e)}_{\Gamma\ neighbor,out}(\vec{r},\vec{\Omega},g), & \vec{\Omega}.\vec{n}_{ext} < 0 \\ \Psi^{(e)}_{\Gamma\ neighbour,in}(\vec{r},\vec{\Omega},g) = \alpha_{G\to g}\Psi^{(e)}_{\Gamma\ subdo,out}(\vec{r},\vec{\Omega},G), & \vec{\Omega}.\vec{n}_{ext} \geq 0 \end{cases}$$

Otherwise, draw energy from uniform function in lethargy

Algorithm 14: Boundary flux exchange in updated multiple-energy-grid method

The main difference in this proposed procedure with the developed Algorithm 13 is that a fine mesh is created and that each group needs to be checked against this mesh, rather than just comparing mesh sizes. This proposed update to the current multiple-energy-grid scheme would allow the user to refine multiple regions of interest. A better description of the reflector resonances would also likely have a more important impact on the calculation.

Another potential update to the multiple-energy-grid scheme could be to study the boundary flux reconstruction without common group boundaries, as illustrated in Figure 25. The difficulties with such a method were discussed in §3.1.2, but the main challenges would be that no common grid could be used to simplify the test during the boundary flux exchange and different coefficients would be used to reconstruct the flux, and to weigh the group correspondences.

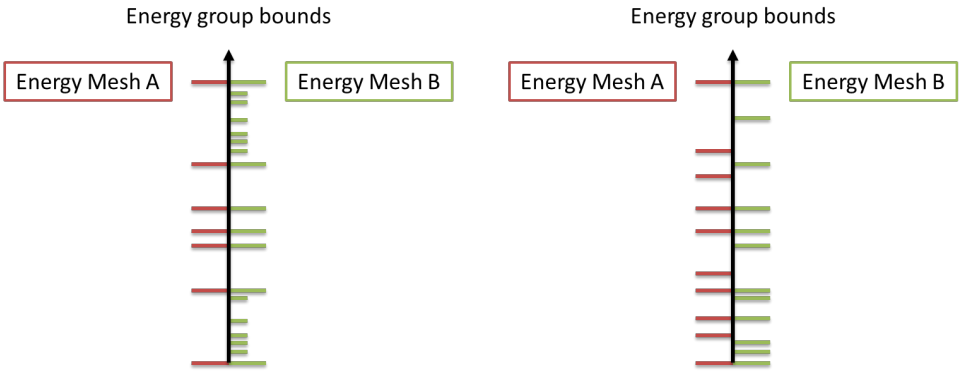


Figure 25: Illustration of energy meshes using the current multiple-energy-grid method (left) and a possible generalization of the method without common energy boundaries (right)

The multiple-energy-grid method can also be seen as a first step towards an adaptive energetic method, where at each step an error condition determines if a region needs to be refined. IDT uses macroscopic cross sections generated from a previous APOLLO3® calculation. To be able to implement such a method in IDT, it would be necessary to precalculate many different energy grids and then switch between them.

To conclude, modifying the DDM to accommodate the multiple-energy-grid scheme helped us understand what data is necessary to update the boundary condition between subdomains: subdomains update the boundary condition using boundary fluxes, an eigenvalue and the fission source for each subdomain. As a result, the stochastic method needs to have surface flux estimators for each group and direction, as well as the fission rate.

While a multiple-energy-grid scheme is a starting point for future deterministic-deterministic hybrid methods, it faces most of the same challenges as traditional deterministic methods such as self-shielding and spatial discretization. For this reason, we decided to move on to the development of a deterministic-stochastic hybrid method. In the next chapter, we look at the implementation of a 1-way coupled deterministic-stochastic scheme.

4. DEVELOPMENT OF A MONTE CARLO-DETERMINISTIC HYBRID SPATIAL METHOD

In the previous chapter, one of the goals of the multiple-energy-grid method was to have a finer energetic description of certain subdomains. This method made it possible to get more information on a part of the core, which led to having more precise reaction rates for the energy-refined subdomains. However it kept the limitations of traditional deterministic methods.

In this chapter, a hybrid deterministic-stochastic method is developed to try to overcome some of the deterministic limitations. A partially coupled method was chosen to study the benefits the stochastic methods can bring to a small section of the deterministic calculation. We consider this method partially coupled because the deterministic calculation is unaffected by the stochastic calculation. The first section of this chapter will explain the hybrid method implemented here (§4.1). After this, the study case is presented (§4.2), followed by a conclusion and perspectives (§4.3).

Contents

4.1 PARTIAL COUPLING PRINCIPLES AND IMPLEMENTATION IN IDT AND LAST

This section focuses on a hybrid method, which uses a deterministic flux distribution as a fixed source for a stochastic calculation. A deterministic calculation is first performed on a large problem. A stochastic calculation is also performed on a small section. The deterministic flux distribution provided by the complete calculation initializes the stochastic calculation and is fixed on its boundaries. The stochastic calculation will lead to increased precision over the region of interest, while the overall calculation will be run using a quicker and coarser method.

4.1.1 Description of 1-way coupling application to core problems

The method developed here starts by performing a whole core deterministic calculation using IDT's DDM. The boundary fluxes resulting from this deterministic calculation are stored in a file and then used by the stochastic calculation. A stochastic calculation is consequently run on a smaller region of interest using the boundary fluxes from the neighboring deterministic subdomains. This process is illustrated in Figure 26.

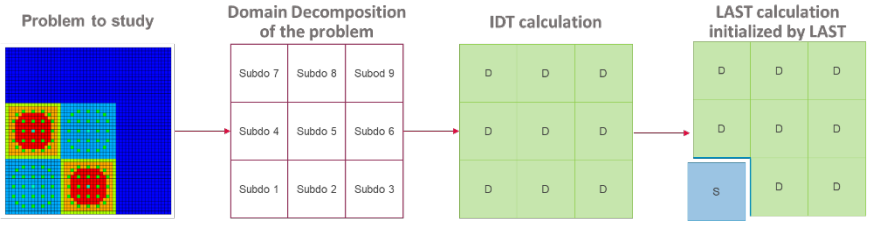


Figure 26: Illustration of the main steps in the 1-way coupled process

We sample the fixed source to determine the boundary source for our simplified Monte Carlo problem. In this method, we chose to deal with a surface source created from the deterministic angular boundary fluxes.

In an industrial hybrid method, using files to transfer information between the two codes will become too memory expensive. Transferring information using files will also slow the calculation down significantly. In such a code, it would be best to transfer information in memory directly. However, in our case this was not possible. IDT and LAST are two independent codes, making it easier to implement and test a hybrid method using files.

4.1.2 Sampling of the incoming boundary fluxes to create a fixed source

As mentioned above, one of the reasons we chose to sample our boundary source was to be able to run calculations with large numbers of particles. To create the discrete distribution to sample, we use the C++ library <random> [60] and the *discrete_distribution* function. The source creation is implemented in C++ in LAST, which is why we used C++ libraries to implement the sampling function that will create the source from the IDT distribution.

In §2.2, we saw that in deterministic methods, the neutron transport equation is split into a system of equations to solve. By solving the neutron transport equation, we obtain the following system.

$$\Psi(x) = \int_{Emin}^{Emax} \int_{0}^{4\pi} \Psi(x,E,\Omega) d\Omega dE \approx \sum_{g} \sum_{d} \Psi_d^g(x)$$
$$\Psi(x,E) = \int_{0}^{4\pi} \Psi(x,E,\Omega) d\Omega \approx \sum_{d} \Psi_d^g(x) \qquad (4.1)$$
$$\Psi(x,E,\Omega) = \Psi_d^g(x)$$

The results of a deterministic calculation are discrete fluxes. Let us consider a deterministic calculation performed using R positions, G groups, and D directions. Using the discrete distribution for each component of the deterministic phase-space, we can sample Monte Carlo coordinates of the phase-space.

The array presented in Table 21 is generated by the boundary flux output procedure in IDT. The Cartesian coordinates at the end of each direction $\{\mu, \eta, \xi\}$ can be obtained from the IDT calculation directly. This data is necessary when converting coordinates in the deterministic phase-space to the stochastic phase-space. The quadrature weight is printed to allow us to weigh boundary flux by the quadrature weight.

Subdomain number	Axis	Sub-edge number x	Group number g	Direction number d	d_x	d_y	d_z	Boundary flux $\Psi_d^g(x)$	Direction weight w

Table 21: Array generated by IDT boundary flux print

The angular distribution, the third line of system (4.1), is the only one printed in Table 21. To be able to sample the deterministic spatial distribution (first line) and energetic distribution, we need to create the remaining distributions from $\Psi_d^g(x)$.

Once each distribution is created, we can start by sampling a surface from the spatial distribution. It is then possible to create the corresponding energetic distribution and sample an energy group. From the spatial mesh and energy group, the angular distribution is created and a direction can be sampled. After this, we need to convert deterministic data into a point of the phase-space for the stochastic calculation. This procedure is illustrated below in Algorithm 15.

Sampling algorithm

- Read boundary fluxes from the IDT output file
- Sum boundary fluxes over directions and energy to create the spatial distribution
- Sample N particles
  - Sample the spatial distribution and get index of spatial mesh
    - Use spatial mesh index to determine energy distribution
    - Sample the energy distribution and get energy group index
      - Use spatial mesh and energy group indexes to determine angular distribution
      - Sample the angular distribution and get direction
  - Convert spatial mesh into a position (x, y, z) by sampling uniformly on sub-edge
  - Sample uniformly energy E within energy group boundaries
  - Use sampled direction number to find the associated $\{\mu, \eta, \xi\}$

Algorithm 15: Sampling algorithm of discrete distributions

To be able to create the deterministic spatial and energetic source distributions, some functions were created: a function determining the number of spatial meshes along each axis, the number of energy groups, and the number of directions. The objects to store the three distributions are then generated. The last function is used to create each distribution from the global data (4.2).

$$\begin{aligned} \Psi[x] &= \sum_g \sum_d \Psi_d^g(x) \\ \Psi[x][g] &= \sum_d \Psi_d^g(x) \\ \Psi[x][g][d] &= \Psi_d^g(x) \end{aligned} \qquad (4.2)$$

These functions require looping over the deterministic data a second time, which slightly slows down the Monte Carlo simulation in the hybrid scheme. To limit the number of functions to run and steps implemented, we wanted to see the effect of sampling over the boundary flux distribution presented in (4.3).

$$\Psi(x,E,\Omega) = \Psi_d^g(x) \qquad (4.3)$$

This new method is presented in Algorithm 16. The simplified sampling method led to the same results as the ones shown in Algorithm 15. To limit the potential for mistakes and limit the functions called during the sampling procedure, we decided to use the sampling algorithm implemented below.

Sample source algorithm

- Loop over parallel threads
  - Read boundary flux from the IDT output file
  - Create distribution of weights using all the IDT data to sample from
  - Sample N particles
    - Use C++ discrete distribution to sample the weights distribution and get index
    - Use index to find the corresponding subsurface on each edge and convert into a position (x, y, z)
    - Use index to find the corresponding energy group and sample energy E within energy group boundaries
    - Use index to find the corresponding direction and associated $\{\mu, \eta, \xi\}$

Algorithm 16: Algorithm implemented in LAST to sample IDT boundary fluxes

It is important to note is that we illustrated this issue for a single interface between neighboring cells. If we had multiple deterministic-stochastic interfaces, we would need a weight distribution associated with each interface. This modifies slightly the sampling algorithm because we would no longer sample N particles but $N \times number\ of\ interfaces$.

4.1.3 Interfacing phase space coordinates

Another aspect of the sampling algorithm is the need to convert surface, group, and direction numbers into stochastic coordinates of the phase-space. We will start by looking at the description of the spatial problem in both calculations. Let us imagine the same interface treated by both a deterministic and a stochastic method. The deterministic calculation is separated into eight subdivisions along each axis. The number of subdivisions is determined by the user for each calculation. This is illustrated using a fuel cell in Figure 27. In red is the interface we will focus on.

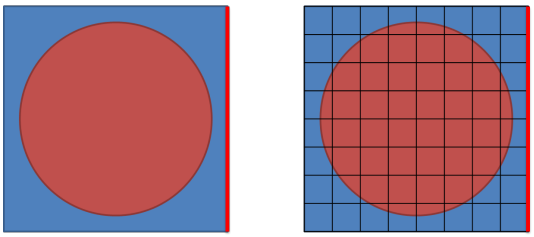


Figure 27: Fuel cell treated with a stochastic method (left) and fuel treated with a deterministic method (right)

The resulting boundary flux for each calculation on this interface is represented in Figure 28. The output of the deterministic calculation will give a value of the boundary flux for each subdivision. This means that the number of subdivisions is also important to determine the shape of the boundary flux.

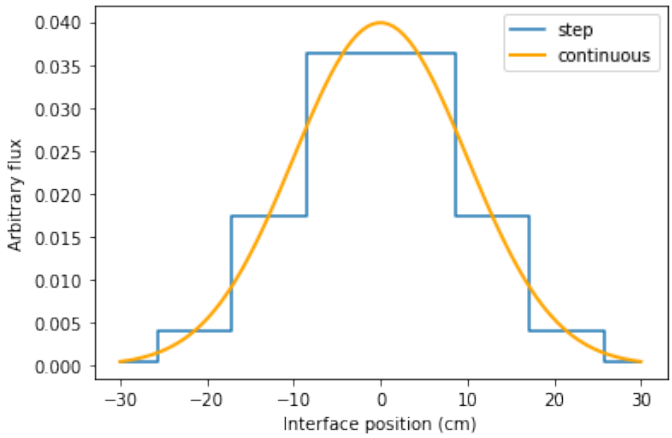


Figure 28: Normalized spatial distribution of boundary flux steps and continuous representations

In a 3D Cartesian geometry, each interface is characterized by two out of three of the coordinates (x, y, z). In our example, let us say that the interface position in Figure 28 is along the y-axis. This means that x and z are known, while y is only known in intervals. To determine the y-position in each interval, we need to the spatial description used in each interval. The spatial description may be determined using polynomials. To simplify the spatial problem, we decided to run constant MOSC calculation. Constant calculations make it possible to uniformelysample the y-position in each subdivision.

One problem with spatial correspondence is that it is problem-dependent. Subdomain numbering is dependent on the number of interfaces between the two calculations, the size of the problem, and the number of subspaces in each cell. This means that a different spatial correspondence method is implemented for each different study case. This problem is in part due to the use of two independent codes, with no common geometry description as is the case between IDT and LAST. LAST uses ROOT geometries [61], while the user needs to use the native IDT formalism to describe the geometry. However, for more practical use, having the user implements spatial correspondences is not realistic. A solution for this problem would either to use the same geometry interpreter and/or generator or to create an interface, which would automatically adapt the coordinates between the two methods. In this second method, the user would have to precisely define the interface between the two methods, and the rest would be automatic. The design software INCA [62] could be a starting point for a future interface. INCA offers the possibility of defining the geometry or calculation scheme for TRIPOLI-4® and APOLLO3®. However, additional work must be done to make the same problem definition compatible for both codes. Further modifications must be done to deal with spatial correspondences while calculations are running as well.

Let us move on to the deterministic energetic description. The deterministic energy distribution is split into groups, which are seen as steps in Figure 29. The problem is the same as above; we need to sample within each group to determine a single energy. We again decided to use more than tens of groups in the deterministic calculations and to sample uniformly within each group, to be able to use the same sampling method regardless of the problem to study. If we wanted to use fewer groups, we might need to change sampling methods with the energy range. The sampling method would also need to be changed with the problem's energy spectrum.

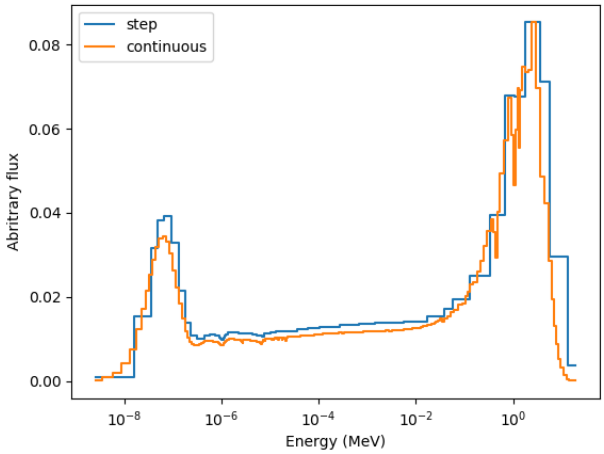


Figure 29: Normalized energetic distribution of boundary flux steps and continuous representations

The angular description of deterministic problems uses the discrete ordinates method as explained in §2.2.1.2. The discrete ordinates are the directions associated with the center of each mesh as shown in Figure 30. In IDT, we are able to obtain the vector corresponding to each direction. We decided to use this vector and not sample within the solid angle. The S_N calculation does not provide the distribution within the solid angle. Sampling in the solid angle would mean assuming a distribution.

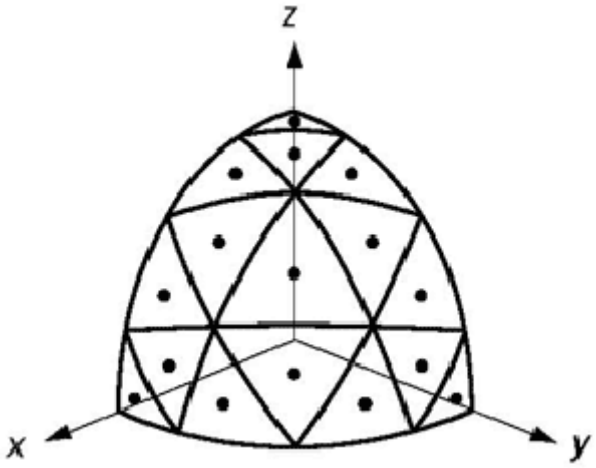


Figure 30: S_N quadrature with discrete directions [63]

Once the sampled point of the IDT distribution is converted into a point in the phase space, its weight is set to one, and it is added to the list of sources for the stochastic calculation. The sampling procedure means that meshes with more important boundary fluxes will have more particles in the stochastic simulation. The number of particles in each mesh will compensate for the fact that the particle weight is set to one.

4.1.4 Computational structure of the method's implementation

With a better understanding of how we convert an IDT boundary flux distribution into sources for LAST, let us have a look at how this first hybrid method is implemented in Algorithm 17. The deterministic calculation is run using the DDM algorithm shown in Algorithm 3. Once the deterministic calculation is converged, subdomain boundary fluxes are stored in an output file and the LAST calculation is run using this file as input sources.

Numerical Steps in IDT to solve a problem using DDM and calling LAST

- Loop over external iterations — e=1,…,N_e
  - Loop over subdomains — s=1,…,N_s
    - Update the local fission source from $\Phi_s^{(e-1)}$
    - Multigroup approximation – Scattering source calculation — g=1,…,N_G
      - Update local scattering sources ($\forall g' \neq g$) from $\Psi_s^{(e),g'<g}$ and $\Psi_s^{(e-1),g'>g}$
      - Internal iterations — i=1,…,N_i
        - Solve the spatial sub-problem: sweep over the mesh and propagate from boundary conditions
        - Update the local self-scattering source $\boldsymbol{H}^{gg}\Psi_s^{(e,i),g}$
    - Update the boundary flux and exchange boundary fluxes between subdomains
    - Update k_{eff}
    - Convergence test on k_{eff} and power distribution
  - Reconstruction distribution at the core scale
  - Print boundary fluxes in output file
  - Call LAST to run a fixed source calculation following Algorithm 16

Algorithm 17: Algorithm implemented in IDT to run a fixed source calculation in LAST with boundary sources from IDT

As in the case of the multiple-energy-grid scheme, the region of interest is isolated from the rest of the problem by its subdomain's boundaries. Creating a subdomain that only contains the region of interest, helps limit the output data to its boundaries. This reduces the amount of information to pass between codes and ensures that the subdomain of interest has the same boundaries as the problem to study with the stochastic method.

4.2 APPLICATION TO A PROPAGATION CASE: CORE-REFLECTOR TRAVERSE

The 1-way coupled scheme is implemented to propagate a fixed source coming from a deterministic calculation. The idea is to propagate this source and use the stochastic estimators to access more information compared to the deterministic calculation. This means that we would not only have access to a flux for instance.

Setting up this scheme requires multiple steps to ensure the viability of the method. We need to look at different problems to develop a more complex methodology. For this problem, we wanted to work with a single interface between the two codes, to simplify the spatial description of our problem. This first study sets the single-interface sampling scheme.

4.2.1 Problem geometry

Having a single interface between our two codes meant that we needed to work on a simplified 2D problem. We then decided to look at a separation between subdomains along the y-axis and to focus on a single row of cells surrounded by reflective boundary conditions. We created a geometry using 10 UOX cells and 10 water cells as depicted below in Figure 31.

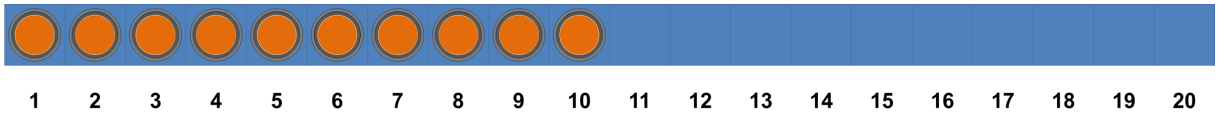


Figure 31: Core-reflector traverse geometry with cell numbering

Each fuel cell is a heterogeneous Cartesian cell represented in Figure 32 while each water cell is homogenous.

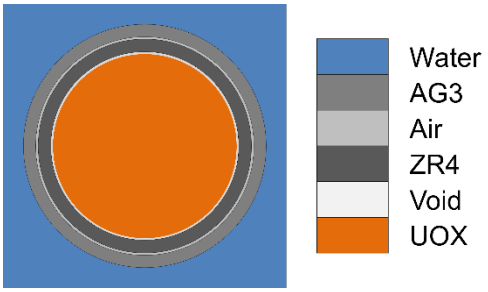


Figure 32: Fuel cell media

Zircaloy-4 (ZR4) is a zirconium-tin-iron-chrome alloy, which is used for cladding for some fuel pins in LWRs [64] [65]. AG3 is an aluminum-magnesium-iron alloy, which is used as an overclad in the EOLE experiments [64] [65]. The overclad is used in the experimental facility to recreate the moderation ratio at lower temperature than the one in a functioning PWR. Even in this simplified configuration, we tried to incorporate elements of the experiments that could be studied by hybrid methods. In this traverse problem, we choose to use the fuel cell from the EPICURE UH1.2 configuration [65].

The geometry presented in Figure 31 and Figure 32 was used to run a reference stochastic calculation using LAST and a deterministic calculation using IDT. The stochastic code does not allow for true 2D calculation. We had to create a thin 3D geometry with reflections on the z-axis. In the IDT calculation, submeshes were created to ensure the spatial convergence of our problem. A preliminary analysis of the problem was performed comparing the eigenvalues for both IDT and LAST calculation determined the number of submeshes. Since we chose to run a constant MOSC calculation, we created twenty subspaces along each axis, which can be seen in Figure 33.

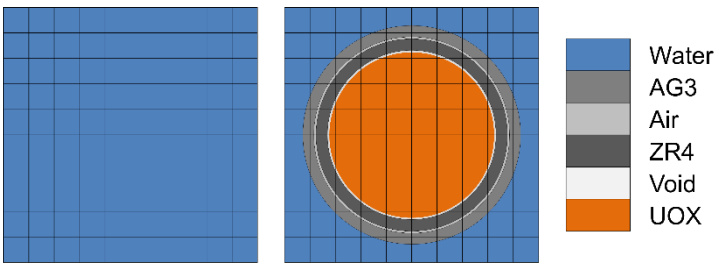


Figure 33: Cell separation into a 20x20 submesh – Water cell (left) – Fuel cell (right)

The geometry used to run our coupled calculation was a bit different. We still used our twenty-cell traverse, but instead of filling the fuel cells with the appropriate media, each ring and the moderator were filled with a voided media. This is illustrated below in Figure 34. We decided to use the same geometry as the complete traverse, to facilitate geometric correspondences.

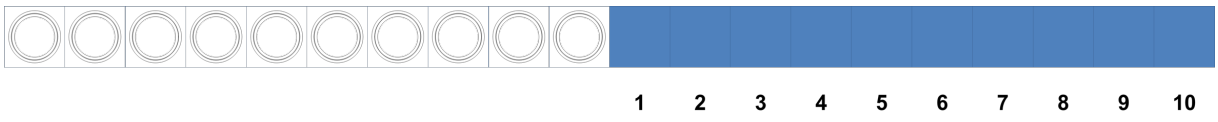


Figure 34: Core-reflector traverse geometry with cell numbering with voided fuel cells used in the coupled calculation

Originally, we had created a geometry simply made up of the water traverse as seen below in Figure 35. This geometry was unsuited for specifying the appropriate boundary conditions, because we could not differentiate the boundary condition entering the first cell along the y-axis with the two boundaries on the x-axis. This is because the whole traverse is surrounded by the same medium. To overcome this issue a small volume needs to be added before the first water cell.

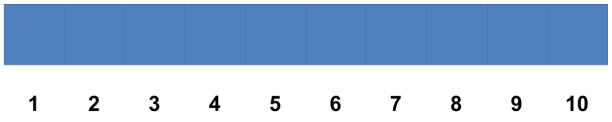


Figure 35: Second Core-reflector traverse geometry with cell numbering cells which was not used

A secondary issue with this geometry is that the origin of the problem is no longer the middle of the interface between the fuel and water subdomains. The origin is now between cells 5 and 6 in Figure 35. While this is not a problem, it is important to bear in mind for both the spatial correspondences between the deterministic and stochastic codes and the development of an interface between two codes.

An important difference with the whole traverse is that the water traverse is not surrounded by reflective boundary conditions. Three of the external surfaces have a reflective boundary condition, while the interface surface has a void boundary condition. The interface surface already has the imposed boundary source obtained from a converged calculation from IDT. If we impose a reflective boundary condition, we bias the problem by not allowing particles slowed down in the water cells to leave this area.

4.2.2 Calculation options

Not only was the geometry used in the IDT and LAST calculations a bit different due to the requirements of each method, but the parameters of each code were as well. For instance, we mentioned above the need to refine each spatial cell to ensure the spatial convergence of the problem, this is the number of subdivisions on each edge in Table 22. Table 22 presents the different methods and options used to treat the phase space using IDT.

Spatial method	Constant MOSC
Number of subdivisions on each edge	10
Number of energy groups	281
Number of directions	144 (~S_{16})
Angular method	Chebyshev-Legendre

Table 22: IDT calculation options

LAST does not discretize the phase space, so there is no specific spatial treatment. However, to reduce probabilistic uncertainties, we need to ensure that a sufficient number of particles were simulated and batches were run. These options are synthesized in Table 23. Each MPI process runs a batch, when it completes said batch, the process moves on the next. Each MPI process will run approximately 667 batches.

Number of batches	10000
Number of MPI processes	15
Number of neutrons per batch and per MPI process	2000
Number of neutron to discard	100
Initial source	Ponctual (0, 0, 0), Isotropic, Watt spectre
Simulation type	Critical

Table 23: LAST calculation options

The 1-way coupled calculation can be seen as a combination of the IDT and LAST calculation options. We only present the stochastic options used in Table 24, because they differ from the stochastic options in Table 23, while the initial deterministic calculation is the whole problem IDT calculation.

Number of batches	10
Number of MPI processes	15
Number of neutrons per batch and per MPI process	20000
Number of neutrons to discard	0
Initial source	IDT output file
Simulation type	Propagation with fixed source

Table 24: LAST calculation options for the 1-way coupled calculation

It can be noted in Table 24 that the number of batches to be run is much smaller, while the number of particles per batch is larger. The reason for this is to strive for the same magnitude of statistical uncertainties. The total number of neutrons simulated in both calculations lead to 10^{-3} of relative uncertainty. Contrary to the complete LAST simulation, we no longer need to discard the first few batches, since we are already using a converged source distribution. The analysis of the IDT distribution will be performed below in §4.2.3. To be representative of the IDT boundary flux distribution, we need to generate a sufficient number of particles per batch. All the neutrons simulated in the 1-way calculation are in the water section of the traverse, they all contribute to the flux in this part of the geometry. However, in the complete geometry LAST calculation, the neutrons simulated can contribute to the flux in the fuel region, the water region or both. This is why we can run fewer batches in the 1-way coupled calculation.

4.2.3 Comparing the sampling method to the boundary flux distribution

Before running the 1-way coupled calculation, we started by comparing the two complete calculations. This analysis is meant to serve as an envelope parameter for the results of the 1-way coupled calculation. In Figure 36, we take a look at the normalized flux in each cell of the traverse. The flux distribution is separated into two groups, a fast group covering energies from 0.625 eV to 20 MeV, and a thermal group covering energies below 0.625 eV. The fast (or thermal) flux is divided to the sum of the fast and thermal fluxes in the water cells. The results are normalized to the flux obtained in the water cells to be coherent with 1-way coupled calculation. In this figure, we can see good agreement between the two calculations.

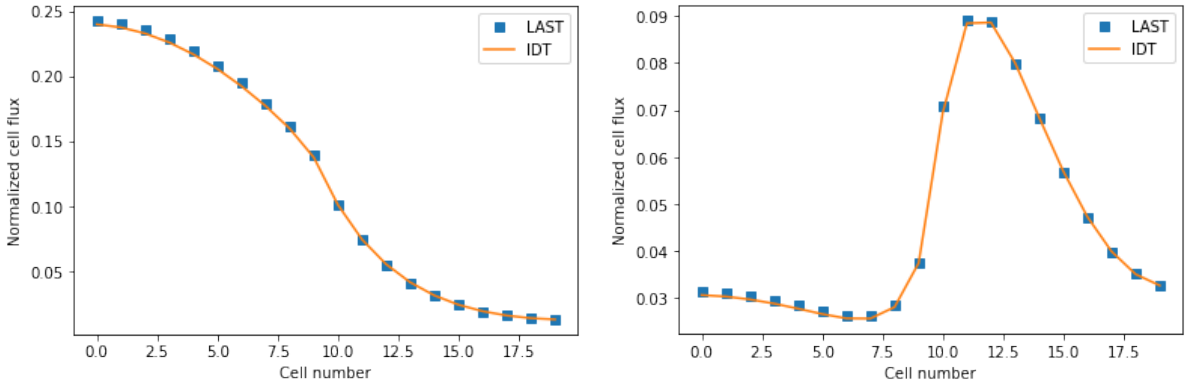


Figure 36: Comparison of the complete IDT calculation (orange) to the complete LAST calculation (blue) – Fast group comparison (left) – Thermal group comparison (right)

In Figure 37, we plot the relative differences between IDT and LAST. In Figure 36, it seemed that the two curves overlapped completely so we need the results of Figure 37 to quantify the differences between the two calculations. The error bars shown are the uncertainty at 3σ obtained by the stochastic calculation only, since we have no associated uncertainty to the flux obtained by IDT.

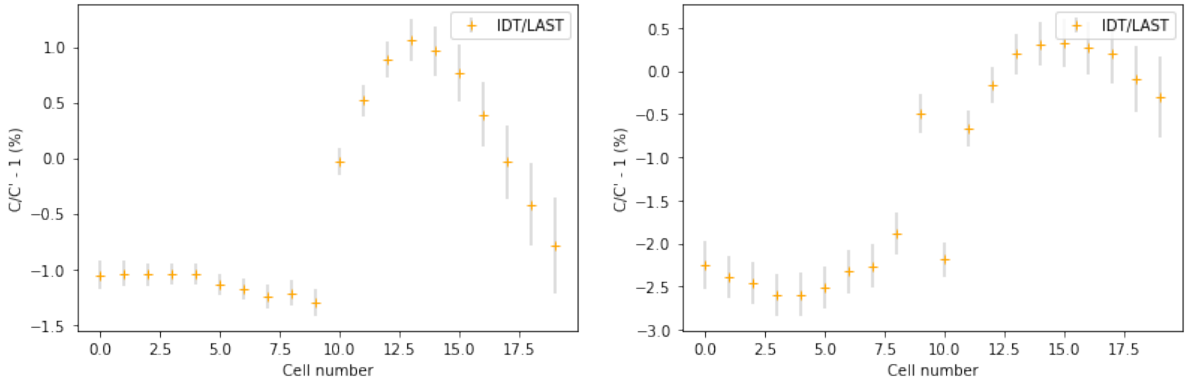


Figure 37: C/C' comparing complete IDT calculation to complete LAST calculation – Fast group comparison (left) – Thermal group comparison (right)

We compared calculation times for both of these calculations in Table 25.

Calculation	Time spent in IDT	Time spent in LAST	Total calculation time
IDT only	3 min		3 min
LAST only		2h 7 min	2h 7 min

Table 25: Calculation time spent in each code

Having a good idea of how our calculation is expected to behave, we then looked at the data generated by the sampling algorithm.

Our goal is to have a sampled distribution that reproduces the IDT boundary flux distribution. To be able to compare the two distributions, while sampling our boundary source with LAST, we counted the surface, group, and direction numbers before converting this data into a point in the phase-space. Using this data, we were able to compare the normalized boundary flux on each surface with the normalized counts of particles on each surface as shown in Figure 38.

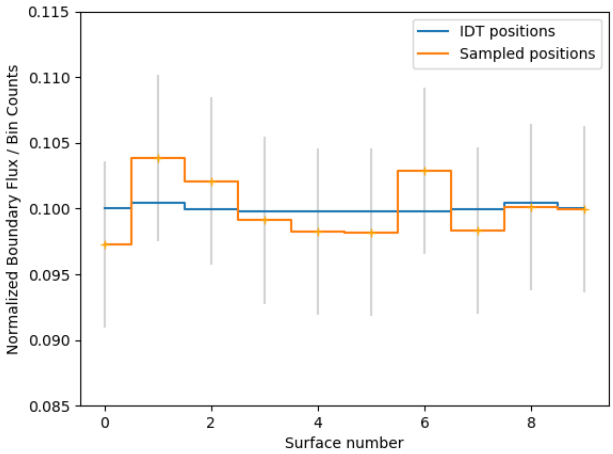


Figure 38: Comparison of IDT spatial distribution (blue) with the sampled spatial distribution (orange)

The spatial distribution showed good agreement, and the energetic distribution is shown in Figure 39. It can be noted that some statistical fluctuations disturb the energetic distribution slightly, but the overall shape is conserved after sampling.

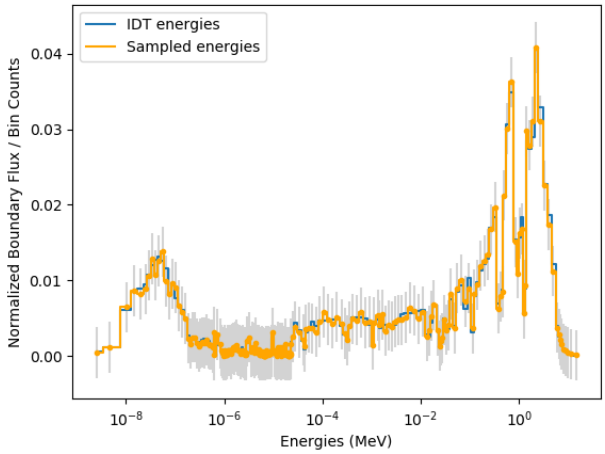


Figure 39: Comparison of the IDT energetic distribution (blue) with the sampled energetic distribution (orange)

The same method is used to compare the angular distribution of the sampled source to IDT's. The comparison of the direction numbers is shown on the left side of Figure 40. Some statistical fluctuation can also be observed here. The axis shows the increasing arbitrary index given to the most directions with the largest flux (blue).

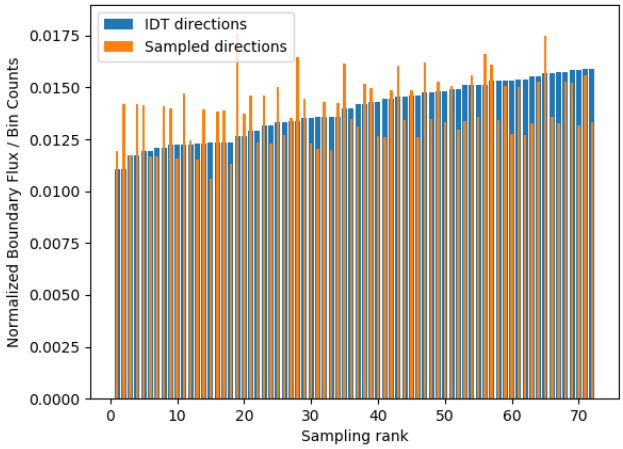


Figure 40: Comparison of the IDT angular distribution (blue) with the sampled angular distribution (orange)

In the last three figures, we can see that there is good agreement between the two distributions. This sampling method seems to be able to recreate the boundary flux distribution.

4.2.4 Results

4.2.4.1 Initial results

Next we compared the normalized flux in each cell of the water subdomain in Figure 41. Cell numbering is illustrated in Figure 34. The results are surprising because the crosses in green (resulting from the coupled calculation) do not follow the expected shape.

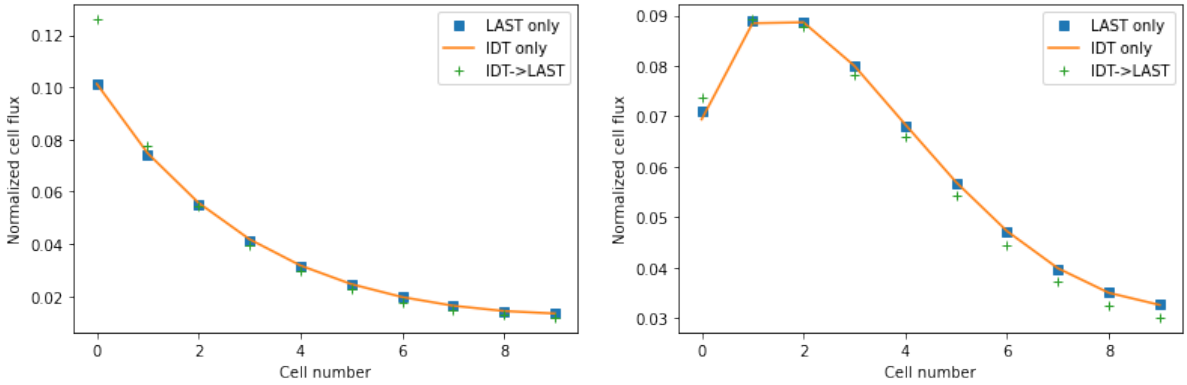


Figure 41: Normalized flux in each water cell – Fast flux (left) – Thermal flux (right)

The fast range seems to be poorly predicted with an important overestimation of the flux for the first few cells on the left of Figure 41. The prediction seems slightly better for the thermal range. In both cases, there seems to be an overestimation of the flux in the first few cells. To get a closer look at the behavior of the coupled calculation compared to the complete IDT calculation, we use the same method described for Figure 37. Figure 42 presents the way the 1-way coupled calculation and complete IDT calculation differ from the complete LAST calculation.

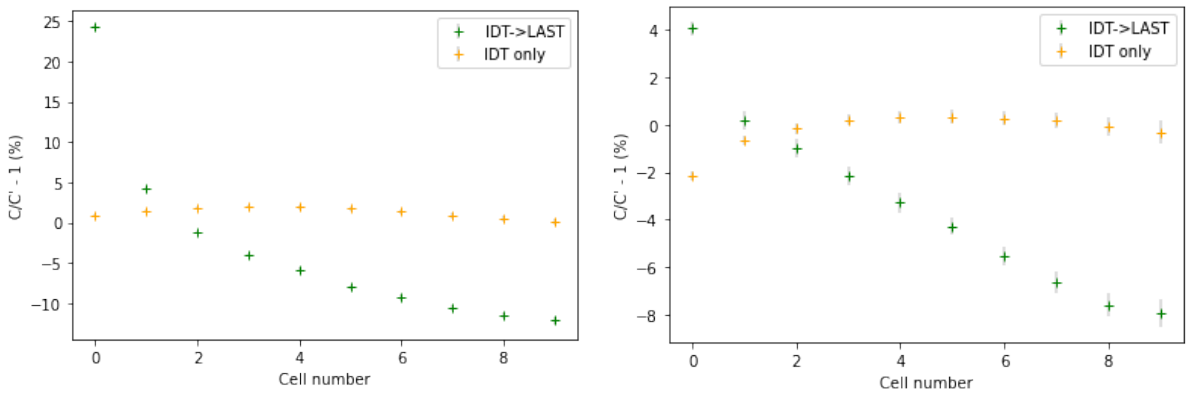


Figure 42: C/C' comparing complete IDT calculation (yellow) and coupled IDT-LAST calculation (green) to complete LAST calculation – Fast comparison (left) – Thermal comparison (right)

Figure 42 confirms the observations made above. The C/C' variation for the 1-way coupled calculation does not follow the IDT only variation and far exceeds the 3σ uncertainty. In particular, the 20% error obtained for the first cell of the fast flux is much too large. It was expected that the 1-way coupled method would lead to equivalent or better flux. These results must indicate a problem with our methodology.

4.2.4.2 Problem analysis

After comparing the result of the sampling method with the initial deterministic calculation, we compared data crossing the interface between the two subdomains in the complete deterministic and stochastic calculation. We started by comparing the angular distribution, by summing the boundary fluxes in energy and space. After this, we separated the polar coordinates into intervals, following IDT directions. We then summed up all the particles in each interval to obtain the final count. We also only represented the directions obtained in a single polar plan. From Table 5, in a stochastic calculation fluxes are estimated using $\frac{\omega}{\vec{\Omega}.\vec{n}}$, while currents are estimated using ω. On the left side of Figure 43, we only summed up weights in each direction, which means that we are comparing a flux and a current. To be able to compare stochastic data to this boundary flux, we started by converting our weight distribution into a flux distribution by dividing the particle weight by $\vec{\Omega}.\vec{n}$. In this case, this product can be simplified since we are comparing data going through a single interface, the product now becomes $\vec{\Omega}.\vec{n} = \Omega_x$. After this, we separated the spatial coordinates into intervals, to fit in with the deterministic subspaces. We then summed all the particles in each interval to obtain the final count. In Figure 43, we represent the two normalized distributions. These two distributions now show good agreement and both seem almost isotropic.

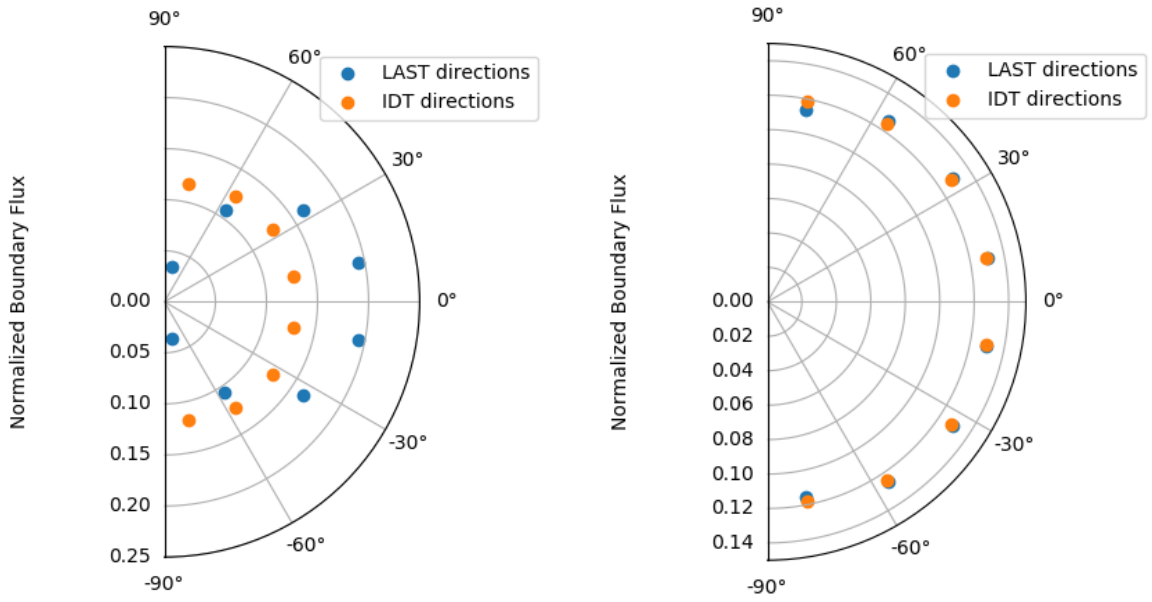


Figure 43: Polar plot comparing the normalized boundary flux from IDT (orange) and the number of particles in each direction of solid angle from LAST (blue)

We use the same procedure to convert the weight distribution into a spatial distribution. The subspaces are shown above in Figure 33. The results are condensed in Figure 44.

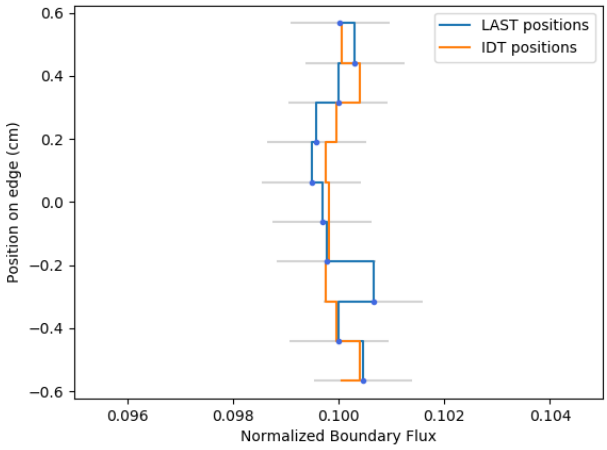


Figure 44: Spatial plot comparing the normalized boundary flux from IDT (orange) and the number of particles in each sub-edge (blue)

This time instead of separating the weight distribution into directions, we converted the energies into groups, using the procedure explained in §4.1.2. The results are shown in Figure 45.

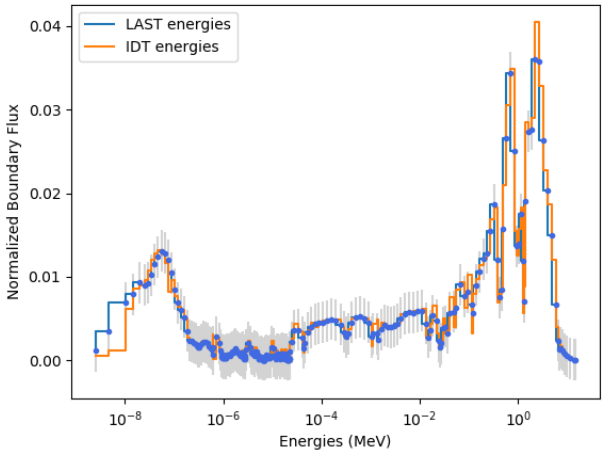


Figure 45: Energy plot comparing the normalized boundary flux from IDT (orange) and the number of particles in each group from LAST (blue)

In the last three figures, we can see that there is good agreement between the two calculations. The three distributions show that the similar fluxes go through the interface in the two codes. The stochastic flux is higher in the thermal region, so we expect that we may have slightly underestimated this using IDT. Now that we have seen how similar the data transiting between the subdomains is, we can see that the problem comes from the sampling algorithm used in LAST. By considering our boundary flux distribution as a distribution of weights, we were biasing our calculation. To correct this error, we need to add an extra step that converts a boundary flux distribution into a true distribution of weights as shown in Algorithm 18. Once the boundary flux file is obtained from the IDT calculation, it needs to be converted into a point source to be run in the stochastic calculation. To do this, we start by transforming our boundary fluxes into surface sources. In our method, this means that we are going to use our boundary flux distribution on each surface as the discrete distribution to sample surface sources.

Sampling algorithm

- Loop over parallel threads
  - Read boundary flux output file from IDT
  - Create distribution of weights to sample from
    - Convert boundary fluxes into weights by multiplying each value by $\vec{\Omega}.\vec{n}$
  - Sample N particles
    - Use C++ discrete distribution to sample the weights distribution and get index
    - Use index to find the corresponding subsurface on each edge and convert into a position (x, y, z)
    - Use index to find the corresponding energy group and sample uniformly within group to determine energy E
    - Use index to find the corresponding direction and associated $\{\mu, \eta, \xi\}$

Algorithm 18: Corrected algorithm implemented in LAST to sample IDT boundary fluxes

In Algorithm 16, we mentioned converting boundary fluxes into weights. The reason we need to do this is that, in a stochastic calculation, fluxes are estimated using $\frac{\omega}{\vec{\Omega}.\vec{n}}$, as seen in Table 5. To get a distribution of weights, we need to multiply each component of the boundary flux distribution with the associated $\vec{\Omega}.\vec{n}$. After implementing this new correction, a new analysis of the hybrid scheme is performed.

4.2.4.3 Final results

We first start by looking at the normalized flux in each cell of the water subdomain of the traverse in Figure 46. Cell numbering is shown in Figure 34. The crosses in green seem to follow the curves obtained with the single method calculation scheme for both the fast and thermal ranges.

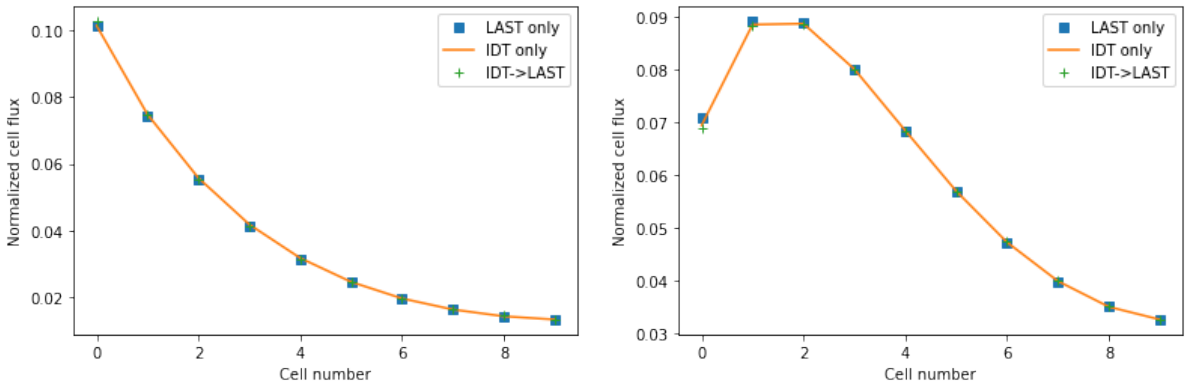


Figure 46: Normalized flux in each water cell – Fast flux (left) – Thermal flux (right)

To get a closer look at the difference between the 1-way coupled calculation and the complete stochastic calculation, the relative difference to the stochastic simulation is shown below in Figure 47. This allows us to see how the 1-way coupled scheme differs from the complete calculation, and how its behavior compares to the complete deterministic calculation.

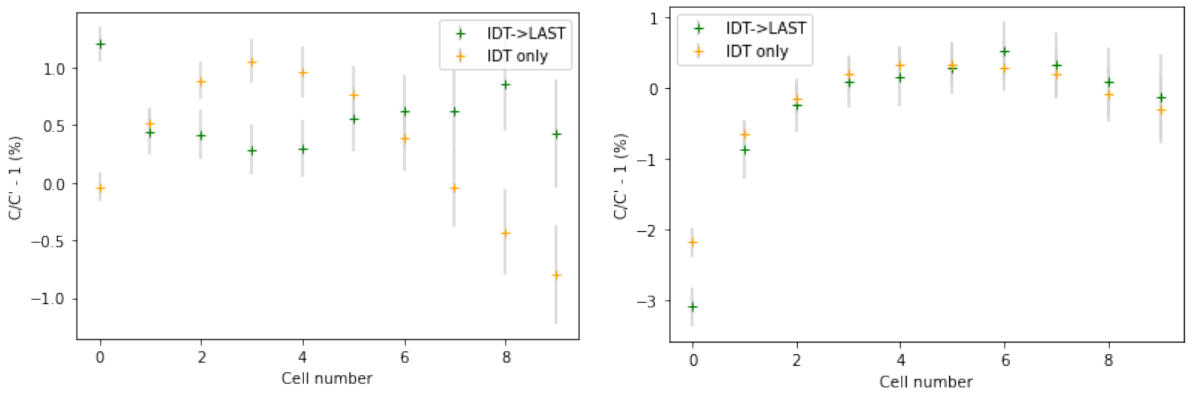


Figure 47: C/C' comparing complete IDT calculation (yellow) and coupled IDT-LAST calculation (green) to complete LAST calculation – Fast comparison (left) – Thermal comparison (right)

From Figure 47 it seems that the fast flux is improved by the 1-way coupled scheme. The 1-way coupled scheme is closer to the fast-flux obtained by LAST, than the one calculated by IDT. However, it seems that the thermal flux is close to the one obtained by IDT. On the right side of Figure 47, the two curves are very similar. Similar results from the IDT calculation and the coupled calculation were expected. We chose to look at a fuel-reflector traverse that was well modeled by the deterministic calculation already, which is why gains were expected in the 1-way coupled calculation.

4.2.5 Parametric study

The results obtained in the previous section came from a fully converged and finely refined deterministic calculation. However, in a 2-way coupled calculation scheme, the deterministic calculation will not perform as many power iterations, or as fine as studied above. This will have an impact on the data used by the stochastic calculation, which will in turn affect the partially coupled calculation. We assume that the refinement and convergence of the deterministic problem will probably be case-dependent in a 2-way coupled scheme. The reference parameters are those used in the previous IDT calculation (Table 22).

4.2.5.1 Number of groups

Table 26 presents the different methods and number of meshes used to treat the phase space using IDT. The original calculation was run in IDT using 281 energy groups; in this section, we are going to reduce the number of energy groups to see how this might affect the calculation.

Spatial method	Constant MOC
Number of subdivisions on each edge	10
Number of energy groups	51 and 23
Number of directions	144
Angular method	Chebyshev-Legendre

Table 26: IDT calculation options simplified in energy

An important part of the calculation time (1 min 23 s) is spent preparing the DDM simulation and initializing subdomains. This part of the calculation is barely affected by the change in the number of groups. The different calculation times for the new 1-way coupled calculations are presented in Table 27.

Decreasing the number of groups, decreased both the size of the distribution to sample from and the size of the file to be read. The creation of the surface source distribution from IDT's boundary fluxes was significantly quicker for these new distribution. For 281 groups it took about 51 minutes, 10 minutes for 51 groups and 5 minutes for 23 groups. The 1-way coupled calculation time was then much shorter compared to the 281 group calculation. The calculation times obtained for each are presented in Table 27.

Calculation	Time spent in IDT	Time spent in LAST	Total calculation time
IDT only	3 min		3 min
LAST only		2h 7 min	2h 7 min
IDT->LAST 281 groups	3 min	55 min 15 s	58 min 15 s
IDT->LAST 51 groups	2 min 42 s	14 min 30 s	17 min 12 s
IDT->LAST 23 groups	2 min 16 s	9 min 28 s	11 min 44 s

Table 27: Calculation time spent in each code

Despite the lower number of groups used in the deterministic calculation, there is still good agreement with the normalized flux obtained in LAST. To better see the differences between all our calculations, we decided to only present the (C-C')/C' results in Figure 48 and Figure 49. The IDT curve is in yellow, the coupled IDT-LAST 281 group calculation in green, the 51-group calculation in red and the 23-group calculation in purple.

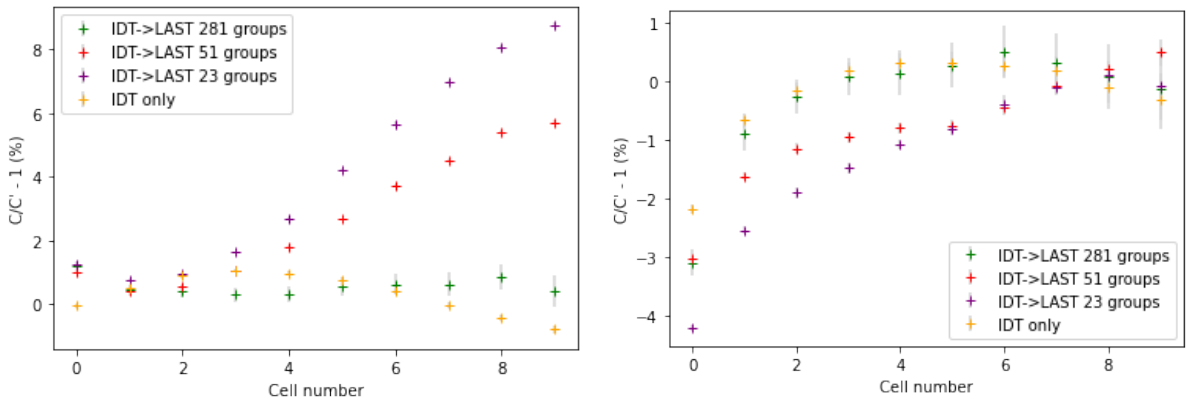


Figure 48: C/C' comparing complete IDT calculation (yellow) and coupled IDT-LAST calculation (green, red, and purple) to complete LAST calculation – Fast comparison (left) – Thermal comparison (right)

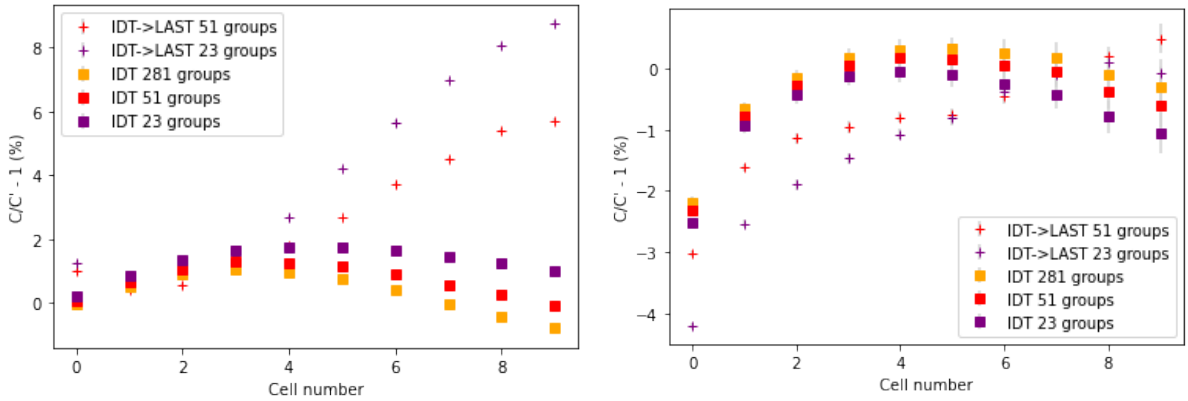


Figure 49: C/C' comparing complete IDT calculation (yellow, red, and purple) and coupled IDT-LAST calculation (red, and purple) to complete LAST calculation – Fast comparison (left) – Thermal comparison (right)

As expected, decreasing the number of groups in the deterministic calculation leads to a poorer flux calculation in the coupled calculation. The deterministic calculation performed much better than the 1-way coupled calculation with fewer energy groups. It seems more beneficial with a coarser energetic calculation to use the deterministic method. In §4.1.2, we explained that the energy used in the stochastic calculation is obtained by sampling uniformly all the energies within the selected group. From Figure 48, the thermal flux seems much less affected by the number of groups in the initial calculation. This might be explained by the width in lethargy for thermal groups (<0.625 eV) which is still close to 0.1-0.3 as was the case in the 281-group mesh. The discrepancies obtained in the fast region might be due to uniform sampling method, which samples in wider groups. This might lead to higher energies being sampled, which could explain why the flux is overestimated in the last few cells of the traverse.

The results seem to indicate the need to run energetically fine IDT calculations when using the coupled method. Another solution could be to change the sampling method and adapt the sampling method with the energy range. This could be done using weighting functions such as those proposed by NJOY [66] [67], samples energies in the fast range using a Watt spectrum, a $^1/_E$ distribution for the epithermal range, and a Maxwell-Boltzmann distribution for the thermal range. The three distributions are illustrated below for a thermal reactor in Figure 50.

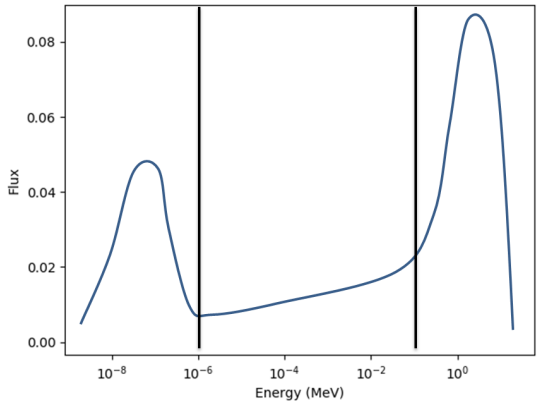


Figure 50: Energy ranges and neutron spectrum

4.2.5.2 Number of spatial meshes

Now that we have seen the impact of changing the energy grids has on the coupled calculation, let us take a look at the number of surfaces on each edge. The original calculation was performed with 10 surfaces on each edge. Table 28 presents the different methods and number of meshes used to treat the phase space using IDT in this new coupled calculation.

Spatial method	Constant MOC
Number of subdivision on each edge	5 and 3
Number of energy groups	281
Number of directions	144
Angular method	Chebyshev-Legendre

Table 28: IDT calculation options simplified in space

Figure 51 compares the complete IDT calculation and the three coupled calculations to the complete LAST calculation. IDT is in yellow, the 10-subdivision calculation in green, the 5-subdivision calculation in red, and the 3-subdivision calculation in purple.

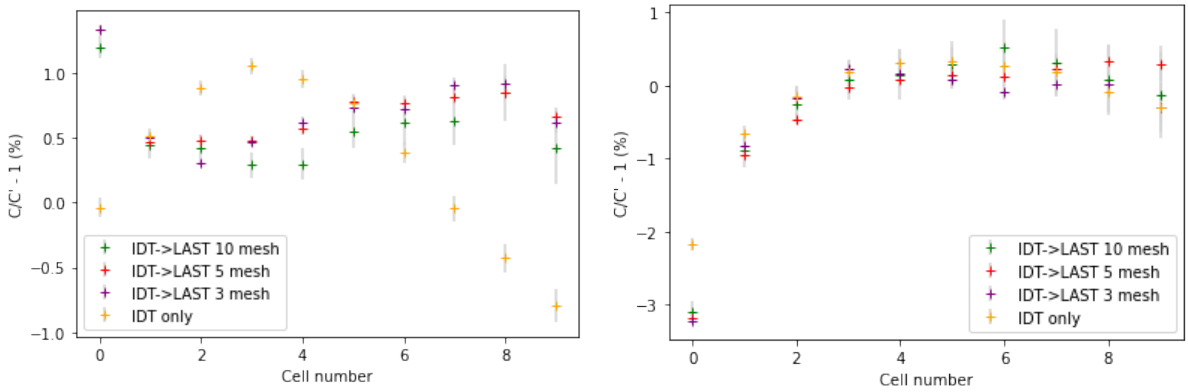


Figure 51: C/C' comparing complete IDT calculation (yellow) and coupled IDT-LAST calculation (green) to complete LAST calculation – Fast comparison (left) – Thermal comparison (right)

From Figure 51, the number of subdivisions on the interface seems to have very little impact on the coupled calculation scheme. This is not very surprising given that the spatial flux distribution is relatively flat according to Figure 44. This is a consequence of the problem that we chose to study, presented in Figure 34, and the reflective boundary condition surrounding the pattern. We are also dealing with an

interface of only 1.26 cm. All of this combined lead to an even distribution of the flux, which explains the small impact the number of subdivisions have on the calculation.

Once again, most of the difference in calculation time obtained in Table 29 comes from the decreased preparation time in LAST. The preparation time could be decreased if an interface was used, which pass information in a more time-efficient way.

Calculation	Time spent in IDT	Time spent in LAST	Total calculation time
IDT only	3 min		3 min
LAST only		2h 7 min	2h 7 min
IDT->LAST 10 subdivisions	3 min	55 min 15 s	58 min 15 s
IDT->LAST 5 subdivisions	2 min 34 s	20 min 34 s	23 min 8 s
IDT->LAST 3 subdivisions	2 min 01 s	12 min 53 s	14 min 54 s

Table 29: Calculation time spent in each code

While the number of subdivisions had very little impact on this problem, we expect that it would not be the case on larger problems with spatial gradients.

4.2.5.3 Angular directions

The next step in this parametric study was to observe the impact of the number of directions on the coupled simulation. The original coupled calculation was run using 144 directions. Table presents the different methods and number of meshes used to treat the phase space using IDT.

Spatial method	Constant MOC
Number of subdivisions on each edge	10
Number of energy groups	51
Number of directions	60 (~S_{10}) and 40 (~S_8)
Angular method	Chebyshev-Legendre

Table 30: IDT calculation options simplified in directions

Figure 52 IDT and the coupled calculations to the complete LAST calculation. IDT is in yellow, the 144-direction calculation in green, the 60-direction calculation in red, and the 40-direction calculation in purple.

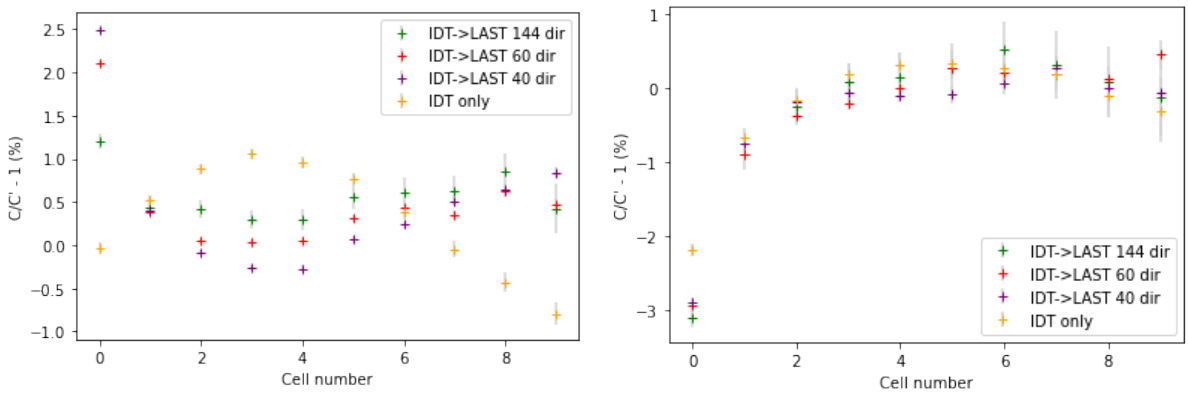


Figure 52: C/C' comparing complete IDT calculation (yellow) and coupled IDT-LAST calculation (green) to complete LAST calculation – Fast comparison (left) – Thermal comparison (right)

There seems to be very little impact to the flux calculation by decreasing the number of directions. Some impact is observed in the fast flux region with up to 1% difference between the original coupled calculation and the 40-direction calculation. The impact is much more limited in the thermal region, as shown in Figure 52. There is no sampling of the angular distribution, the direction used is the one obtained in the deterministic calculation, and the small differences observed cannot be attributed to the sampling method.

Decreasing the number of directions in the initial IDT calculation had an important impact on the sampling procedure. It originally took about 51 minutes to read and sample 144-direction boundary flux distribution, but only 27 minutes to sample the 60-direction distribution, and 15 minutes to sample the 40-direction distribution. The LAST calculation time for the 1-way coupled calculation was always around 4 to 6 minutes. This had decreased the overall calculation time of the 1-way coupled procedure as shown in Table 31.

Calculation	Time spent in IDT	Time spent in LAST	Total calculation time
IDT only	3 min		3 min
LAST only		2h 7 min	2h 7 min
IDT->LAST 144 directions	3 min	55 min 15 s	58 min 15 s
IDT->LAST 60 directions	2 min 37 s	31 min 45 s	34 min 22 s
IDT->LAST 40 directions	2 min 06 s	19 min 26 s	21 min 32 s

Table 31: Calculation time spent in each code

The angular source distribution created from the boundary fluxes shown on the left of Figure 43 revealed that directions within a -60° to 60° intervals are more likely to be sampled. As long as there are directions within this interval, most sampled sources will be associated to these directions. The limited changes due to the angular distribution used in the deterministic calculation might be attributed to the traverse geometry and the shape of the angular distribution.

4.2.5.4 Deterministic calculation convergence

In this section, we use the same options as presented in Table 22, but instead of letting the calculation run its course (13 power iterations), we stopped it after [1, 4, 8] outer iterations. Figure 53 compares IDT and the coupled calculations to the complete LAST calculation.

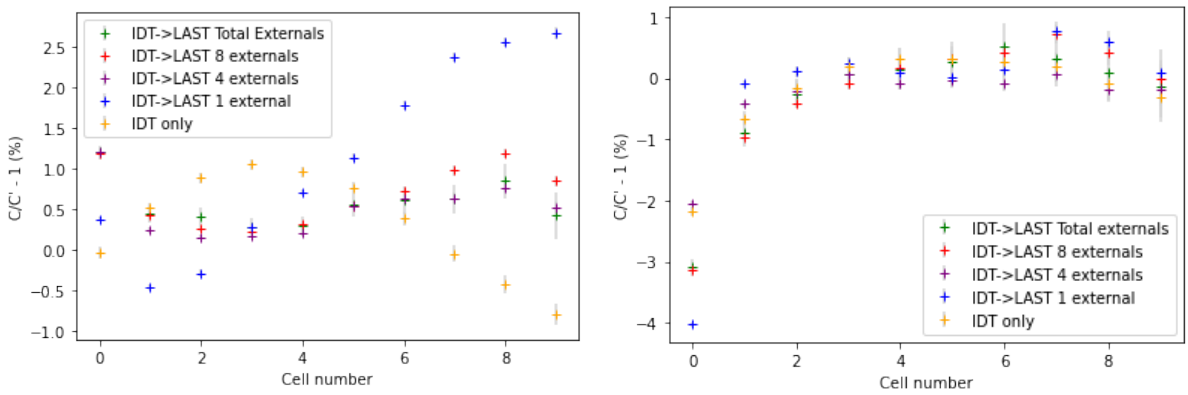


Figure 53: C/C' comparing complete IDT calculation (yellow) and coupled IDT-LAST calculation (green) to complete LAST calculation – Fast comparison (left) – Thermal comparison (right)

Figure 53 shows that performing four power iterations, eight power iterations or the complete IDT calculation leads to very similar results (less than 0.5 % difference). However, only using the first power iteration negatively influences the calculation. It leads to an overestimation of the fast flux in the last few cells by 2.5 %, compared to the 0.5% for the other coupled calculations. It seems that for the traverse problem studied, the newer results show good agreement with the complete LAST calculation, despite using deterministic data that has not fully converged.

It was expected that the first power iteration would lead to poorer results. The results of the first power iteration are obtained only after considering each subdomain as an infinite medium. The boundary conditions has not yet been updated using neighboring subdomains. This explains why the shape is quite different for the fast flux in Figure 53.

Performing fewer externals will shorten the IDT calculation, let us see how that impacts the overall calculation time of the 1-way coupled calculation in Table 32. Decreasing the number of iterations performed by the IDT calculation had little impact on the calculation time of the 1-way method. Most of the calculation time is attributed to the preparation of the surface sources, which is unaffected by the number of external iterations. This procedure depends on the size of the boundary source distribution.

Calculation	Time spent in IDT	Time spent in LAST	Total calculation time
IDT only	3 min		3 min
LAST only		2h 7 min	2h 7 min
IDT->LAST total externals	3 min	55 min 15 s	58 min 15 s
IDT->LAST 8 externals	2 min 15 s	56 min	58 min 15 s
IDT->LAST 4 externals	1 min 53 s	55 min 52 s	57 min 45 s
IDT->LAST 1 external	1 min 16 s	55 min 34 s	56 min 50 s

Table 32: Calculation time spent in each code

The four previous sections have shown that partially coupled calculation using simplifying and shortening the deterministic calculation lead to similar results to those obtained using fine and fully converged data. However, some work can still be done to improve the energetic sampling of the selected group. The good results obtained in the last few paragraphs are, in part, due to the simple geometry studied, which is why spatial subdivision had little impact on the coupled calculation.

The sampling method is able to recreate the deterministic distribution and lead to improved results in the flux calculation over the water subdomain. For this single interface problem, with a non-multiplying medium, the 1-way coupled scheme leads to improved results in the flux calculation with very few batches. The goal is to use this method for more complex geometries with multiple interfaces between

the two codes, which will require studying additional test cases to see if similar results are obtained. Toward a 2-way coupling scheme

4.2.6 Conclusions on the 1-way coupling results

The sampling method is able to recreate the deterministic distribution and lead to improved results in the flux calculation over the water subdomain. For an IDT calculation using 281 groups, 144 directions and 10 subdivisions per cell, the source preparation time was almost three times longer in the coupled calculation compared to the complete LAST calculation. The 1-way coupled calculation simulated more neutrons per batch for fewer batches compared to the complete LAST calculation. While the source preparation was longer, the overall calculation time was shorter. The relative difference between the fluxes of the 1-way coupled calculation and LAST was below 3 %. The relative difference between the 1-way calculation and LAST, was always in the same range or smaller as the one obtained between IDT and LAST. For this single interface problem, with a non-multiplying medium, the 1-way coupled scheme leads to improved results in the flux calculation with very few batches.

Performing a 281-group, 144-direction, and 10-subdivision calculation on a larger problem would lead to much longer deterministic calculations. The stochastic calculation is only performed on a region of interest, to lead to more precise results without having to run a complete stochastic calculation. The goal is to be able to run a coarser and/or less converged deterministic calculation to initialize the stochastic problem. This leads us to perform a “parametric” study of the flux calculation in §4.2.5. Good results were obtained despite decreasing either the number of directions or the number of subdivisions. Decreasing the number of external iterations or groups did increase the relative difference to the complete LAST calculation. One key learning from this study is the role of the energetic refinement of the IDT calculation. Refinement directly improves the 1-way coupled calculation.

The importance of energetic refinement also applies to the previously studied multiple-energy-grid scheme (1). As a result, future work should focus on improving the energetic sampling. This might be done by choosing energy grids that better match the flux spectrum. Another possibility might be adapting the energetic sampling method to the energy range. This method would not require knowing the flux spectrum beforehand which would facilitate its use. Nevertheless, this would require implementing additional tests to determine the distribution from which to sample, thus increasing the computational cost.

4.3 CONCLUSION: TOWARD A 2-WAY COUPLED SCHEME

4.3.1 Main limitations faced

As described in §4.2.4.2, we encountered significant errors in the flux calculation. After tedious analysis we discovered the root caused to be in the interpretation of the weight in LAST which should have been $\frac{\omega=}{\vec{\Omega}.\vec{n}}$, Coherent results were obtained afterwards (§4.2).

One of the challenges linked to our choice of methods came from the lack of common interfaces between the two codes, IDT and LAST. This meant that geometries needed to be defined separately for the two codes, vigilance when creating the initial geometries to solve with each method is paramount. In addition, technical challenges came from the state of development of the codes used.

Other difficulties came from the need to implement and test different functionalities in both codes. For instance, flux estimators were not implemented in LAST. This also required a verification procedure using the already implemented flux estimators in TRIPOLI-4®. In IDT, we needed to implement a number of post-treatment functions to be able to print data. The challenge in developing these functions is that they needed to be compatible with the DDM scheme and parallel method used. While this step is necessary to make our tools compatible, the choice of methods here was at times a contributing factor to slowing down progress.

4.3.2 Remaining work

The development of this partially coupled method can serve multiple purposes. Future developments can continue to complete work done to date and extend this method into an industrial code. In addition, our work can serve as a stepping-stone towards a fully coupled method.

4.3.2.1 Partially coupled hybrid method

Let us start by looking at the work required to complete this method as a partially coupled method. The parametric study conducted on the traverse problem gave us a better understanding of how the deterministic calculation could be simplified before running the stochastic problem. However, the low sensitivity to the angular and spatial distribution are likely specific to the traverse problem. The next step would be to continue the sensitivity study on the second problem. This might help identify which parameters are likely to be case-dependent.

The partially coupled calculation fast flux showed important variations with the deterministic energetic grid used. As mentioned in §4.2.5.1, adapting the energetic sampling method with the energy might be beneficial to partially coupled calculations. Sampling uniformly within each group yielded good results for a fine energy grid but it could be interesting to use a more representative spectrum such as the weighting spectrum proposed by NJOY [66] [67]: a Watt spectrum for the fast range, a Maxwell-Boltzmann distribution in the thermal range and a $1/E$ distribution in the epithermal one. However, the challenge might be finding the appropriate distribution within each group. Before implementing the three distributions, two energy bounds need to be set, the upper thermal boundary $E_{thermal}$ (0.625 eV), and the lower fast boundary E_{fast} (100 keV). An energetic sampling algorithm would need to replace the uniform sampling procedure. Let us consider the selected group $g = [E_{g-1}, E_g]$, with $E_g > E_{g-1}$.

Sampling an energy in the selected energy group

Check the energy range the group belongs to

If $E_g < E_{thermal}$

> Use the partial thermal distribution to sample the energy

If $E_{g-1} > E_{fast}$

> Use the partial fast distribution to sample the energy

If $E_g > E_{thermal}$ and $E_{g-1} > E_{fast}$

> Sample using a linear distribution over $\left[\frac{1}{E_{g-1}}, \frac{1}{E_g}\right]$.

Algorithm 19: Checks to perform to change the sampling distribution with energy

This algorithm still needs some work, particularly when dealing with energy grids that overlap both energy ranges. The effect of such groups would need to be investigated further when developing this method.

At the beginning of each batch, the sampled source is imposed on the stochastic problem's interface and then propagated. However, from one batch to the next, it does not create a new source which would be the sum of the sampled source and all the new source points generated by fission. While the partially coupled method was able to provide microscopic reactions, fluxes, and currents, it cannot yet calculate an eigenvalue. Having an eigenvalue in the partially coupled method would also be an interesting point of comparison with the complete IDT and LAST calculations. To obtain this information, a fixed-source criticality calculation mode would need to be implemented.

In §4.3.1, we mentioned that one of the challenges of working with two independent codes is that the problem definition in each code can be quite different. In this case, IDT defines its geometry natively while LAST uses ROOT geometries. This leads to having to manually update the spatial correspondence scheme for each problem studied. This process is not sustainable, and a future partially coupled hybrid scheme would need an interface that automates this process. The method was currently imagined with IDT controlling everything. This is not the best solution given the important differences between IDT and LAST. In the long run, developing an external interface which controls both codes would be the best solution as it would not only help with the problem definition in both codes but would also facilitate the data exchanges between them. This interface could also offer the possibility of being developed for different deterministic solvers making it possible to treat a wider range of geometries.

A more complete parametric study of the impact of the convergence and refinement of boundary fluxes on the stochastic scheme is also an important step in the development of a fully coupled hybrid method. Let us consider a scenario where a coarse energy grid leads to increased errors in all configurations. This would mean that the energetic description of the deterministic problem would also have an effect on the fully coupled calculation which is dependent on the results from the partially coupled hybrid method. The developments mentioned above would be interesting for a future fully coupled hybrid method.

4.3.2.2 Towards the fully coupled method

To be able to develop the fully coupled hybrid method, it would be pertinent to start by continuing the development of the partially coupled method. The fully coupled hybrid method process is illustrated below in Figure 54.

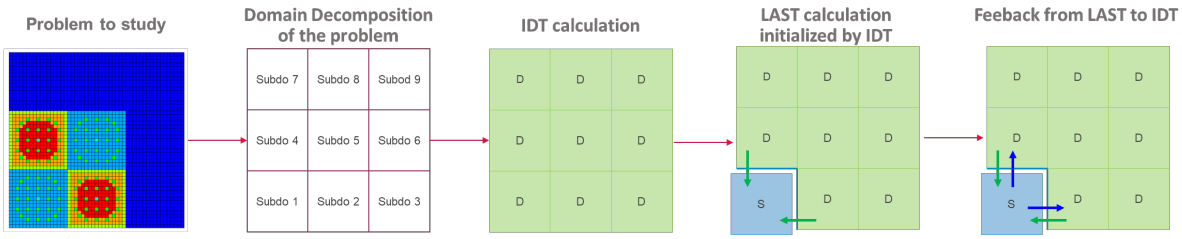


Figure 54: Illustration of the main steps in the fully coupled process

The fully coupled calculation was imagined with the deterministic calculation performing the initialization of the whole problem. This means that the hybrid subdomain would first be treated by IDT and then by LAST. This choice was made by imagining the DDM implemented in IDT controlling the hybrid method. Having IDT control the hybrid process, means that the whole problem is described to IDT to set up the scheme. Having IDT initialize each subdomain in the hybrid calculation would then be interesting to have data accounting for neighboring subdomains. The sources created by stochastic code during its first iteration would not be based on a user-defined source but would instead be created using deterministic fluxes that have already been updated with neighboring subdomains. The deterministic fluxes would account for the real environment, as opposed to the user-defined stochastic source. This would speed up the stochastic calculation. However, this process requires verifying the partially coupled deterministic-stochastic scheme and identifying sources of biases.

Once the 1-way deterministic to stochastic calculation is optimized and fully automated, the next step would be to work on feedback from the stochastic code to the deterministic solver. Before moving straight to automation of this feedback, an important step would involve developing the necessary procedures in IDT to receive this information. This step may seem like the opposite 1-way hybrid scheme, but is just an important element to prepare the fully coupled hybrid method. It would give us an idea of how the DDM procedure needs to be adapted to wait for the stochastic calculation. In this step it seems that it would be best to treat smaller problems and work with fewer subdomains. An option for this might be to study two neighboring subassemblies, each treated with its own method. This would be easier to implement since there would be fewer interfaces. During this first step in the development of the fully coupled hybrid scheme, we will need to deal with the following problems and steps:

- Before starting to develop the 1-way stochastic-deterministic method, some preparation will be necessary. IDT uses angular boundary fluxes and a fission source to proceed. The appropriate boundary flux estimators will need to be implemented in LAST [68]. The challenge here is that the angular distribution needs to match the one used in IDT. IDT uses the discrete ordinates method, which means that by implementing the same partitioning of the unit sphere, we could collect neutrons in the solid angle associated to each direction and pass this information to IDT. The boundary flux estimators can be implemented in LAST or can be dealt with by an interface linking the two codes.
- To ensure that the deterministic calculation really waits for the stochastic calculation, a first step would be to wait until the stochastic calculation converges. IDT would be able to use fully converged boundary fluxes from LAST. This can be done either using files (much more costly) or in memory (using an interface for example). The reasoning here is that using converged data

to initialize the calculation should help move IDT in the right direction, and speed up the calculation while reaching the expected results. However, the overall calculation process in this step would still be slower than desired since IDT would have to wait for a fully converged LAST calculation between each power iterations.

- After the stochastic-deterministic data exchange has been tested and approved using converged data, a sensitivity study will need to be conducted using data that has not been fully converged. Different cases can be imagined by reducing either the number of batches, the number of neutrons per batch or both.
- When passing data from LAST to IDT some errors may arise due to the different normalization processes used in both codes. LAST normalizes all its outputs to one neutron, which is not the case in IDT, where fluxes can be normalized to power or fission rates for instance. More investigation into how the data sent to IDT needs to be normalized is necessary.
- In the previous points, the stochastic nature of LAST's data was not discussed. The average value of the estimator will be sent to IDT at each power iteration. To see how much the statistical uncertainty of the stochastic calculation may affect the overall convergence of the stochastic-deterministic method, it would be interesting to use different values of the boundary fluxes. This could be done at first by sampling a value of the boundary flux within its uncertainty range.

Having completed two opposite 1-way hybrid methods and their performance analysis, we will be able to inform future research on the viability of a 2-way hybrid method. However, before really focusing on the development of the 2-way hybrid scheme, it will be necessary to modify the deterministic-stochastic scheme to run the two codes in parallel so that IDT DDM's deals with calling the LAST calculation and waits for its results. Once this step is complete, it will be time to add the second 1-way calculation. Once this is completed, the next step will be to run both deterministic and stochastic calculations at the same time. These steps will be tricky because problems will need to be separated into categories:

- Implementation: these problems will be due to errors coming from the implementation of the boundary exchange. Spatial correspondences are such examples. The errors may also be due to modifications of the DDM scheme.
- Calculation time: DDM make it possible to split the global calculation into smaller problems, which can be run using parallel processes. This parallel calculation decreases significantly the calculation time when the total calculation time for each parallel thread is equivalent. This might not be the case when using the fully coupled method. In some problems, the stochastic calculation may be significantly longer than all the other deterministic iterations. This would cause the overall method to be significantly slower. In such cases, it will be necessary to study more closely the cost/precision balance.
- Data: these errors may come from poor convergence of either the deterministic or stochastic calculation, or from the statistical fluctuations in the stochastic calculation. To correct these errors, we should start by making sure that the stochastic data completely covers the data exchanges, which should have been checked in the previous step. At first, the average value of the flux will be used to implement the stochastic-deterministic interface, but it will be necessary to study the effect of uncertainties on the convergence of the overall calculation.

5. CONCLUSIONS

5.1 GENERAL CONCLUSIONS

With the goal of improving neutronic numerical method, this works tackles challenge of coupling deterministic - stochastic calculations. While, existing publications on hybrid methods are promising, hybrid deterministic-stochastic methods that use DDM to solve problems more efficiently than either method remains to be explored. The goal of this work was to move towards a fully coupled hybrid method and examine its feasibility for problems such as ICARE.

To reach this goal, different hybrid methods were examined and developed. To get a wider view of the variety offered by hybrid methods, an analysis was performed. This brought to light three main categories of methods: variance reduction, partially coupled and fully coupled. Partially coupled methods can be easier to implement since the two methods only interact one way, and one calculation scheme is completely unaffected and still lead to interesting gains in precision and calculation time. However, fully coupled methods have the potential to treat a much wider range of applications and may lead to important advances in reactor modeling.

One of the main challenges faced during this Ph.D. thesis has been the lack of a common interface between the two methods to couple. Given the independent nature of the two methods, it was decided to first develop different hybrid methods before working on the fully coupled deterministic-stochastic method. The multiple-energy-grid scheme was created to gain more information on the DDM architecture, which is a central topic in the fully coupled hybrid method. Modifying the DDM scheme to allow for different energy grids in each subdomain helps to understand the challenges of unequal boundary exchanges, and seeing how the quality of data influences the overall calculation scheme. Overall, the multiple-energy-grid results were bounded by the corresponding single-energy-grid results, which means that the method developed led to more precise absorption rates.

The next step towards fully coupled methods is to develop a partially coupled deterministic-stochastic hybrid scheme. This was an essential step to ensure that the communication from deterministic to stochastic behaves as expected. In this method, the converged boundary fluxes from the surrounding subdomains were fixed at the stochastic problem's boundaries. An IDT calculation was performed over the whole problem and converged boundary fluxes were used to create a source for the stochastic calculation.

The partially coupled method was implemented and tested for a simplified 2D problem. For this single interface problem, with a non-multiplying medium, the 1-way coupled scheme leads to improved results in the flux calculation with very few batches. The partially coupled calculation was then tested using data from a partially converged and coarser deterministic calculation. The 1-way coupled method led to similar results to those obtained using fine and fully converged data. The partially coupled calculation is not only a stepping stone towards the fully coupled method, it can also be used to obtain more precision on problems with little feedback between the problem and the region of interest.

In this work, two hybrid methods were developed and studied. Energetic refinement was confirmed to play a key role in both methods and might also for future developments. But energy refinement also comes at a cost for deterministic calculation. In our view, future work towards fully coupled hybrid methods should include a step to deal specifically with energetic refinement.

5.2 PERSPECTIVES

During this three-year Ph.D., we have explored various paths, encountered several issues. Some were resolved, whereas others may become the subject of future work towards a fully coupled hybrid method. In this section, we have selected some of the most promising perspectives for the hybrid methods presented and towards a fully coupled hybrid method.

- Energetic deterministic-deterministic hybrid.
    - A first step would be to continue updating the multiple-energy-grid scheme to go beyond the fine/coarse mesh separation. Being able to refine around specific resonances might help the reconstruction methods. We observed that the reconstruction method

performed better when the fuel's energy mesh was finer than the reflectors. Having group refinement in both energy meshes might be a solution to this problem.
    - To make this method more user-friendly, the next step would be to allow energy grids to have different group boundaries. However, the flux reconstruction and condensation for unequal group sizes become more complex and might not lead to better results compared to grids with common boundaries.
    - Another difficulty for the user with this method might be choosing appropriate energy grids. To overcome this, a solution might be to move towards an adaptive method, starting from a coarse grid and refining it as needed. This is obviously a much more complicated step to perform compared to the previous ones. This method would need to create estimators to test the precision at each iteration to determine if the subdomain's grid needs to be refined. This might at first be done using precalculated cross sections with different energy grids for each subdomain.
- Partially coupled deterministic-stochastic method.
    - A more complete parametric study of the effect of the coarseness and convergence of the deterministic calculation would serve to provide more information on the viability of a fully coupled deterministic-stochastic method.
    - This method was only developed on propagation cases, to be able to give a more complete picture of critical problems, where multiplying media are studied using the stochastic code. Currently, fission sources generated during a batch are not stored; the only sources used are those sampled from the deterministic output. A fixed criticality mode would also be interesting to get an indication of the k_{eff} of the stochastic region of interest. This would also be important for the development of a fully coupled hybrid method, which requires feedback from the stochastic calculation.
    - This method could also be made more efficient by changing the method of passing information. Currently data is exchanged using files, this is obviously not suitable, and future work could focus on building an interface between the two codes and more efficient exchanges. This same interface could also be used for the creation of appropriate geometries for both codes to limit sources of error for the user.
- Fully coupled hybrid method.
    - Assuming the deterministic to stochastic boundary exchanges has been completed and verified as explained above, the next step would be to focus on the stochastic to deterministic communication. This step can also be separated into much smaller steps. To start, an angular surface flux estimator needs to be implemented to be able to initialize an IDT calculation. Once LAST is capable of sending the necessary data to IDT, the next step would be to start with a converged calculation. After ensuring the convergence of the stochastic to deterministic scheme using converged data, it would be important to understand how the stochastic nature of LAST might impact the convergence of the deterministic calculation. This would mean studying the impact of sending information earlier. An alternative to sending information from a less converged calculation could be to add fluctuations manually by sampling data within the uncertainty range. A difficult aspect of this is creating or implementing a stochastic estimator, which will send the necessary data to the deterministic calculation. Part of the challenge is using an angular discretization compatible with the chosen deterministic method. IDT requires implementing a discrete ordinates discretization.
    - An interface between the two codes could also be beneficial here. While the DDM offers interesting possibilities in terms of problem description, when working with two independent codes, adding an interface makes the method much more user-friendly and limits potential errors from the user in problem definition and translation between the two codes.

In the development of a future hybrid method, a first key result derived from our work is the importance of finding a balance between the fixed source from the determinist calculation and the sources created during the stochastic calculations is paramount to model critical problems in hybrid methods. Accurate modeling of this balance will yield to a better neutron production estimation. A second take way is the convergence duration of stochastic methods over the determinist method. Energy refinement was

shown to significantly impact hybrid calculations. However, the resulting precision comes at a cost of slower calculation. The precision / cost ratio remains a challenging balance to be found. Lastly, having a common data model compatible with both stochastic (e.g., LAST) and determinist (e.g., IDT) is paramount to ensure greater compatibility of the two codes coupled in a fully hybrid method.

REFERENCES

[1] Gouvernement Français, «Code de l'environnement,» 1 March 2020. [En ligne]. Available: https://www.legifrance.gouv.fr/affichCode.do?cidTexte=LEGITEXT000006074220. [Accès le 16 March 2020].

[2] Gouvernement Français, «Arrêté du 7 février 2012 fixant les règles générales relatives aux installations nucléaires de base,» 24 September 2018. [En ligne]. Available: https://www.legifrance.gouv.fr/affichTexte.do?cidTexte=JORFTEXT000025338573. [Accès le 16 March 2020].

[3] ASN, «Guide 28 : Qualification des outils de calcul scientifique utilisés dans la démonstration de sûreté nucléaire,» 26 June 2017. [En ligne]. Available: https://www.asn.fr/l-asn-reglemente/guides-de-l-asn/guide-de-l-asn-n-28-qualification-des-outils-de-calcul-scientifique-utilises-dans-la-demonstration-de-surete-nucleaire. [Accès le 16 March 2020].

[4] Direction de l'énergie nucléaire, Neutronics, Paris: CEA Saclay and "Le Moniteur", 2015.

[5] D. Bernard et A. Santamarina, «Qualification of gadolinium burnable poison: Interpretation of MELUSINE/GEDEON-II spent fuel analysis,» *Annals of Nuclear Energy,* vol. 87, pp. 21-33, 2016.

[6] P. Mas, «Description of Melusine,» Boyd, A.W. International Atomic Energy Agency (IAEA), Vienna, Austria, 1970.

[7] P. Reuss, Neutron Physics, EDP Sciences - Collection Génie Atomique, 2008.

[8] J. J. Duderstadt et W. R. Martin, Transport Theory, Wiley and Sons: New York NY USA, 1979.

[9] G. I. Bell et S. Glasstone, Nuclear Reactor Theory, Van Nostrand Reinhold Company, 1970.

[10] J. R. Lamarsh et A. J. Baratta, Introduction to Nuclear Engineering, vol. 3, Prentice Hall, 2001.

[11] W. M. Stacey, Nuclear Reactor Physics, vol. Second Edition, Wiley-VCH, 2007.

[12] R. Sanchez et A. Chetaine, «A Synthetic Acceleration for a Two-Dimensional Characteristics Method in Unstructured Meshes,» *Nuclear Science and Engineering,* vol. 136, pp. 122-139, 13 May 2000.

[13] R. Lenain, «Amélioration des méthodes de calcul de coeurs de réacteurs nucléaires dans APOLLO3 : décomposition de domaine en théorie du transport pour des géométries 2D et 3D avec une accélération non linéaire par la diffusion,» 2015.

[14] R. Lenain, E. Masiello, F. Damian et R. Sanchez, «Domain decomposition method for 2D and 3D transport calculations using hybrid MPI/OPENMP parallelism,» chez *Proceedings of M&C2015*, Nashville TN United States, 2015.

[15] R. Lenain, E. Masiello, R. Sanchez et F. Damian, «A Parallel Full Core Transport Calculation Based on Domain Decomposition Method,» chez *Proceedings of Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo 2013 (SNA+ MC 2013)*, Paris, France, 2013.

[16] E. Masiello et I. Zmijarevic, «IDT Theory Manual,» Gif-sur-Yvette, 2020.

[17] D. Schneider et e. al, «APOLLO3: CEA/DEN Deterministic multi-purpose code for reactor physics analysis,» chez *Proceedings of The Physics of Fuel Cycles and Advanced Nuclear Systems - Global Developments (PHYSOR-2016)*, Sun Valley, United States, 2016.

[18] H. Golfier, R. Lenain, C. Calvin, J. J. Lautard, A. M. Baudron, P. Fougeras, P. Magat, E. Martinolli et Y. Dutheillet, «APOLLO3: A common project of CEA, AREVA and EDF for the development of new deterministic multi-purpose code for core physics analysis,» chez *Proceedings of the International Conference on Mathematics and Computational Methods (M&C2009)*, New York NY USA, 2009.

[19] D. Sciannandrone et S. Santandrea, «Tracking strategies in 3D axial geometries for a MOC solver,» chez *Proceedings of the Joint International Conference on Supercomuting in Nuclear Applications and Monte Carlo (SNA + MC 2013)*, Paris France, 2013.

[20] A. M. Baudron et J.-J. Lautard, «MINOS: A simplified Pn solver for core calculation,» *Nuclear Science and Engineering,* vol. 155, pp. 250-263, 10 April 2007.

[21] N. Odry, «Méthode de décomposition de domaines avec parallélisme hybride et accélération non linéaire pour la résolution de l'équation du transport Sn en géométrie non-structurée,» Aix-Marseille Université, Aix-en-Provence, 2016.

[22] J.-J. Lautard, J.-Y. Moller et O. Mulla, «MINARET or the quest toward the use of time-dependent neutron solver for nuclear core calculation on a regular basis,» chez *Proceedings of the Joint International Conference on Supercomputing in Nuclear Applications and Monte Carlo (SNA + MC 2013)*, Paris, France, 2013.

[23] I. Zmijarevic, «Multidimensional discrete ordinates nodal and characteristics methods for the Apollo2 Code,» chez *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C1999)*, Madrid, Spain, 1999.

[24] E. Masiello et I. Zmijarevic, «Short Characteristics Method for Two Dimensional Heterogeneous Cartesian Cells,» chez *Proceedings of the Physics of Fuel Cycles and Advanced Nuclear Systems - Global Developments (PHYSOR-2006)*, Vancouver BC Canada, 2006.

[25] E. Masiello, R. Lenain et J.-M. Do, «Domain Decomposition and CMFD Acceleration Technique Applied to Neutron Discrete-Ordinates Transport Equation in XYZ Geometries,» chez *Proceedings of the International Conference on Mathematics & Computational Methods Applied to Nuclear Science and Engineering (M&C2011)*, Rio de Janeiro, Brazil, 2011.

[26] E. Girardi, «Couplage de méthodes et décomposition de domaine pour la résolution de l'équation du transport de neutrons,» Université d'Evry Val d'Essonne, 2004.

[27] B. Faure, «Development of Neutronic Calculation Schemes for Heterogeneous Sodium-Cooled Nuclear Core in the APOLLO3 Code,» Aix-Marseille Université, Aix-en-Provence, 2019.

[28] J.-F. Vidal, «Contribution à l'amélioration des méthodes déterministes et schémas de calcul neutronique des réacteurs nucléaire,» Université de Grenoble, Grenoble France, 2018.

[29] E. Masiello et I. Zmijarevic, «Short Characteristics Method for Two Dimensional Heterogeneous Cartesian Cells,» chez *Proceedings of the Physics of Fuel Cycles and Advanced Nuclear Systems - Global Developments (PHYSOR-2006)*, Vancouver BC Canada, 2006.

[30] International Website Domain Decomposition Methods, «Conferences and Proceedings - Domain Decomposition Methods,» [En ligne]. Available: http://www.ddm.org/conferences.html. [Accès le 20 April 2020].

[31] H. A. Schwarz, «Ueber einige Abbildungsaufgaben,» *Journal für die riene und angewandte Mathematik,* vol. 70, pp. 105-120, 1869.

[32] P. Lions, «On the Schwarz Method,» chez *Proceedings of the first International Conference on Domain Decomposition Methods*, Paris, 1988.

[33] C. M. Diop, «Simulation of the neutron transport using Monte Carlo method,» International School in Nuclear Engineering, Cadarache, 2019.

[34] E. Brun, F. Damian, E. Dumonteil, F. X. Hugot, C. Jouanne, Y. K. Lee, A. Malvagi, A. Mazzolo, O. Petit, J. C. Trama, T. Visonneau et A. Zoia, «TRIPOLI-4®, CEA, EDF and AREVA reference Monte Carlo code,» *Annals of Nuclear Energy,* pp. 151-160, 22 August 2015.

[35] C. J. Werner, J. S. Bull, C. J. Solomon, F. B. Brown, G. W. McKinney, M. E. Rising, D. A. Dixon, R. L. Martz, H. G. Hughes, L. J. Cox, A. J. Zukaitis, J. C. Armstrong, R. A. Forster et L. Casswell, «MCNP Version 6.2 Release Notes,» LANL, Los Alamos, NM (United States), 2018.

[36] P. K. Romano, N. E. Horelik, B. R. Herman, A. G. Nelson, B. Forget et K. Smith, «OpenMC: A state-of-the-art Monte Carlo code for research and development,» *Annals of Nuclear Energy,* pp. 90-97, 14 August 2015.

[37] J. Leppänen, M. Pusa, T. Viitanen, V. Valtavirta et T. Kaltiaisenaho, «The Serpent Monte Carlo code: Status, development and applications in 2013,» *Annals of Nuclear Energy,* pp. 142-150, 12 August 2015.

[38] E. Vandermeersch, «Développement de calculs à énergie continue pour un retour intégral complet sur les paramètres de modèle nucléaire,» Université Grenoble Alpes, 2021.

[39] Y. Chen, «Coupled Monte Carlo-Discrete Ordinates Computational Scheme for Three-Dimensional Shielding Calculations of Large and Complex Nuclear Facilities,» Universität Karlsruhe (TH), Karlsruhe, Germany, 2005.

[40] T. L. Becker, A. B. Wollaber et E. W. Larsen, «A Hybrid Monte Carlo-Deterministic Method for Global Particle Transport Calculations,» *Nuclear Science and Engineering,* vol. 155, pp. 155-167, 2007.

[41] E. Guadagni, C. Le Loirec, Y. Pénéliau, J. M. Létang, D. Mancusi et C. Diop, «A new hybrid next-event estimator for photon-based Monte Carlo dose rate calculations,» *The European Physical Journal Plus,* vol. 136, n° %11135, 2021.

[42] J. C. Wagner, D. E. Peplow, S. W. Mosher et T. M. Evans, «Review of Hybrid (Deterministic/Monte Carlo) Radiation Transport Methods, Codes, and Applications at Oak Ridge National Laboratory,» *Progress in NUCLEAR SCIENCE and TECHNOLOGY,* vol. 2, pp. 808-814, 2011.

[43] Y. Wang, S. Schunert, M. DeHart, R. Martineau et W. Zheng, «Hybrid Pn-Sn with Lagrange multiplier and upwinding for the multiscale transport capability in Rattlesnake,» *Progress in Nuclear Energy,* vol. 101, n° %1C, pp. 381-393, 2017.

[44] E. Girardi et J.-M. Ruggieri, «Mixed First- and Second-Order Transport Method using Domain Decomposition Techniques for Reactor Core Calculations,» chez *Proceedings of Supercomputing for Nuclear Applications*, Paris France, 2003.

[45] E. Girardi, J.-M. Ruggieri, P. Sireta et G. Ritter, «A New Method for the Treatment of Local Strong Heterogeneities and its Application to the Phebus Experimental Facility,» chez *Proceedings of the Physics of Fuel Cycles and Advanced Nuclear Systems : Global Developments (PHYSOR-2004)*, Lagrange Park IL USA, 2004.

[46] H. Lee, S. Choi et D. Lee, «A Hybrid Monte Carlo/Method-of-Characteristics Method for Efficient Neutron Transport Analysis,» *Nuclear Science and Engineering,* vol. 180, pp. 69-85, 2015.

[47] R. S. Baker, «A fully coupled Monte Carlo/discrete ordinates solution to the neutron transport equation,» Graduate College of the University of Arizona, 1990.

[48] F. Rahnema et D. Zhang, «Continuous energy coarse mesh transport (COMET) method,» *Annals of Nuclear Energy,* vol. 115, pp. 601-610, May 2018.

[49] M. A. Kowalski et E. Shwageraus, «Study of the interaction of regions with Multi-Group and Continuous Energy representation in Monte Carlo calculations,» chez *International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering (M&C2019)*, Portland, OR, USA, 2019.

[50] M. A. Kowalski et E. Shwageraus, «A hybrid continuous energy and multi-group Monte Carlo method,» *Annals of Nuclear Energy,* vol. 140, 17 December 2019.

[51] D. Bernard, O. Fabbris et R. Garder, «Validation of JEFF-3.1.1 Thermal and Epithermal Neutron-Induced Capture Cross Sections Trough MELUSINE Experiment Analysis,» *Nuclear Science and Engineering,* vol. 179, n° %13, pp. 302-312, 2015.

[52] A. Santamarina, D. Bernard, P. Blaise, P. Leconte, R. Le Tellier, C. Vaglio-Gaudard et J. F. Vidal, «APOLLO2: A validated code package for PWR neutronics calculations,» chez *Proceedings of Advances in Nuclear Fuel Management IV (ANFM 2009)*, Hilton Head Island, SC, United States, 2009.

[53] M. Livolant et F. Jeanpierre, «Autoprotection des résonances dans les réacteurs nucléaires,» Rapport CEA-R-4533-1974, 1974.

[54] M. A. Smith, E. E. Lewis et B.-C. Na, «Benchmark on Deterministic Transport Calculations Without Spatial Homogenisation - A 2-D/3-D MOX Fuel Assembly Benchmark,» OECD/NEA, 2003.

[55] S. Cathalau, J. Lefebvre et J. P. West, «Proposal for a Second Stage of the Benchmark on Power Distributions within Assemblies,» OECD/NEA, 1996.

[56] G. Marleau, A. Hébert et R. Roy, «A User's Guide for DRAGON,» Ecole Polytechnique de Montréal, Montréal QC, Canada, 1997.

[57] J.-F. Vidal, R. Le Tellier, P. Blaise, G. Guillot, N. Huot, O. Litaize, A. Santamarina, N. Thiollay et C. Vaglio-Gaudard, «Analysis of the FLUOLE experiment for the APOLLO2 validation of PWR core reflectors,» chez *Proceedings of the International Conference on the Physics of Reactors (PHYSOR2008)*, Interlaken, Switzerland, 2008.

[58] F. Desplats, P. Archier, J.-F. Vidal, J.-M. Palau, R. Lenain et E. Masiello, «Innovative Multiple Energy Grid Deterministic Method to Treat Core-Reflector Interfaces,» chez *Proceedings of the International Conference on Mathematics and Computational Methods Applied to Nuclear Science and Engineering of 2021 (M&C2021)*, Rayleigh, NC, USA, 2021.

[59] F. Desplats, P. Archier, J.-F. Vidal, J.-M. Palau, R. Lenain et E. Masiello, «Investigation of a Multiple Energy Grid Determinstic Method - Application to core-reflector problems,» *Annals of Nuclear Energy,* vol. 170, 2022.

[60] cplusplus.com, «<random> - C++ reference,» [En ligne]. Available: https://m.cplusplus.com/reference/random/. [Accès le 9 April 2021].

[61] ROOT Team 2021, «Geometry - ROOT,» ROOT Team 2021, [En ligne]. Available: https://root.cern/manual/geometry/. [Accès le 16 11 2021].

[62] V. Pascal, R. Selabi et J. Tommasi, «PHENIX: Interpretation of ECRIX-H transmuatation experiment with TRIPOLI4D®,» chez *Proceedings of the International Conference on Mathematics & Computational Methods Applied to Nuclear Science & Engineering (M&C2017)*, Jeju, Korea, 2017.

[63] P. J. Coelho, «Thermopedia,» 2 August 2012. [En ligne]. Available: https://www.thermopedia.com/content/9171/. [Accès le 14 May 2022].

[64] J. -M. Palau, E. Masiello, J. -F. Vidal, P. Archier et B. Faure, «Recent progress in the V&V of the French APOLLO3® code: 3D full core analysis of the UH1.2 integral experiment using IDT method of characteristics,» chez *Proceedings of the International Conference (PHYSOR-2018)*, Cancun, Mexico, 2018.

[65] P. Blaise et A. Colomba, «Interpretation of 3D void measurements with TRIPOLI4.6/JEFF3.1.1 Monte Carlo code,» chez *Proceedings of ICAPP-12*, Chicago IL, USA, 2012.

[66] R. E. MacFarlane, D. W. Muir, R. M. Boicourt, A. C. Kahler et J. L. Conlin, «The NJOY Nuclear Data Processing System Version 2016,» Los Alamos National Laboratory, 2019.

[67] A. D. Caldeira, «On the Maxwellian + 1/E + fission-spectrum weighting function of NJOY 99.90 systems' GROUPR module,» *Annals of Nuclear Energy,* vol. 32, n° %16, pp. 606-611, 2005.

[68] J. Bericic et L. Snoj, «On the calculation of angular neutron flux in MCNP,» *Annals of Nuclear Energy,* vol. 100, n° %12, pp. 128-149, 2017.

[69] D. Bernard, O. Fabbris et R. Gardet, «Validation of JEFF-3.1.1 Thermal and Epithermal Neutron-Induced Capture Cross Sections Through MELUSINE Experiment Analysis,» *Nuclear Science and Engineering,* vol. 179, pp. 302-3012, 2015.

[70] Nuclear Power, «Nuclear Power Neutron Flux Spectra,» [En ligne]. Available: https://www.nuclear-power.com/nuclear-power/reactor-physics/nuclear-engineering-fundamentals/neutron-nuclear-reactions/neutron-flux-spectra/. [Accès le 2022 06 20].

APPENDICES

APPENDIX 1: 51-GROUP ENERGY MESH

Group number	Group number in 281-SHEM	Upper boundary in energy (eV)	Lower boundary in energy (eV)	Lethargy width
1	1	1,96403E+07	6,70320E+06	1,07500E+00
2	9	6,70320E+06	4,96585E+06	3,00001E-01
3	11	4,96585E+06	2,23130E+06	8,00000E-01
4	15	2,23130E+06	1,33694E+06	5,12201E-01
5	19	1,33694E+06	8,60007E+05	4,41198E-01
6	24	8,60007E+05	4,94002E+05	5,54401E-01
7	27	4,94002E+05	1,95008E+05	9,29499E-01
8	34	1,95008E+05	6,73795E+04	1,06270E+00
9	41	6,73795E+04	4,99159E+04	3,00001E-01
10	43	4,99159E+04	2,49991E+04	6,91500E-01
11	50	2,49991E+04	9,11882E+03	1,00850E+00
12	57	9,11882E+03	1,91045E+03	1,56300E+00
13	66	1,91045E+03	9,07502E+02	7,44398E-01
14	71	9,07502E+02	4,10796E+02	7,92599E-01
15	75	4,10796E+02	1,32701E+02	1,13000E+00
16	82	1,32701E+02	7,50456E+01	5,70003E-01
17	85	7,50456E+01	4,57914E+01	4,94000E-01
18	88	4,57914E+01	2,76077E+01	5,06002E-01
19	92	2,76077E+01	1,44703E+01	6,45996E-01
20	127	1,44703E+01	1,19795E+01	1,88901E-01
21	136	1,19795E+01	8,13028E+00	3,87601E-01
22	151	8,13028E+00	7,13988E+00	1,29899E-01
23	157	7,13988E+00	6,28016E+00	1,28301E-01
24	178	6,28016E+00	5,41025E+00	1,49100E-01
25	187	5,41025E+00	4,93324E+00	9,22993E-02
26	192	4,93324E+00	4,76785E+00	3,41005E-02
27	193	4,76785E+00	4,00001E+00	1,75599E-01
28	197	4,00001E+00	2,46994E+00	4,82103E-01

29	211	2,46994E+00	1,90008E+00	2,62298E-01
30	218	1,90008E+00	1,44397E+00	2,74500E-01
31	223	1,44397E+00	1,25094E+00	1,43501E-01
32	228	1,25094E+00	1,16999E+00	6,69001E-02
33	230	1,16999E+00	1,14797E+00	1,90000E-02
34	231	1,14797E+00	1,10395E+00	3,91005E-02
35	234	1,10395E+00	1,00904E+00	8,98953E-02
36	239	1,00904E+00	9,63961E-01	4,57038E-02
37	242	9,63961E-01	9,19979E-01	4,67000E-02
38	244	9,19979E-01	8,80026E-01	4,43994E-02
39	245	8,80026E-01	6,25000E-01	3,42200E-01
40	248	6,25000E-01	5,20012E-01	1,83900E-01
41	251	5,20012E-01	3,52994E-01	3,87401E-01
42	255	3,52994E-01	2,79989E-01	2,31701E-01
43	258	2,79989E-01	1,90005E-01	3,87700E-01
44	262	1,90005E-01	1,61895E-01	1,60102E-01
45	263	1,61895E-01	1,04298E-01	4,39696E-01
46	266	1,04298E-01	7,64970E-02	3,10001E-01
47	268	7,64970E-02	5,54982E-02	3,20901E-01
48	270	5,54982E-02	4,03000E-02	3,19999E-01
49	272	4,03000E-02	2,92989E-02	3,18801E-01
50	274	2,92989E-02	1,04505E-02	1,03090E+00
51	278	1,04505E-02	1,10000E-04	4,55392E+00

Table 33: Upper and lower boundaries for each energy group for the 51-group mesh

APPENDIX 2: 23-GROUP ENERGY MESH

Group number	Group number in 281-SHEM	Upper boundary in energy (eV)	Lower boundary in energy (eV)	Lethargy width
1	1	1,96403E+07	4,96585E+06	1,37500E+00
2	11	4,96585E+06	2,23130E+06	8,00000E-01
3	15	2,23130E+06	1,33694E+06	5,12201E-01
4	19	1,33694E+06	4,94002E+05	9,95599E-01
5	27	4,94002E+05	1,95008E+05	9,29499E-01
6	34	1,95008E+05	6,73795E+04	1,06270E+00
7	41	6,73795E+04	2,49991E+04	9,91501E-01
8	50	2,49991E+04	9,11882E+03	1,00850E+00
9	57	9,11882E+03	1,91045E+03	1,56300E+00
10	66	1,91045E+03	4,10796E+02	1,53700E+00
11	75	4,10796E+02	7,50456E+01	1,70000E+00
12	85	7,50456E+01	4,00001E+00	2,93180E+00
13	197	4,00001E+00	1,25094E+00	1,16240E+00
14	228	1,25094E+00	1,14797E+00	8,59001E-02
15	231	1,14797E+00	1,10395E+00	3,91005E-02
16	234	1,10395E+00	1,00904E+00	8,98953E-02
17	239	1,00904E+00	9,63961E-01	4,57038E-02
18	242	9,63961E-01	8,80026E-01	9,10994E-02
19	245	8,80026E-01	6,25000E-01	3,42200E-01
20	248	6,25000E-01	3,52994E-01	5,71301E-01
21	255	3,52994E-01	1,90005E-01	6,19401E-01
22	262	1,90005E-01	7,64970E-02	9,09799E-01
23	268	7,64970E-02	1,10000E-04	6,54453E+00
24	178	6,28016E+00	5,41025E+00	1,49100E-01

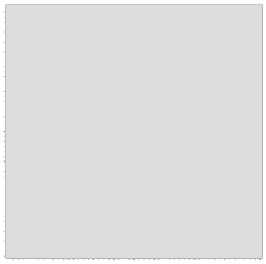
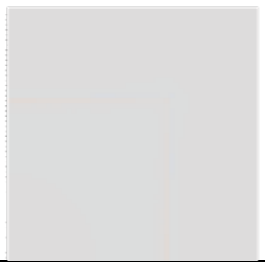
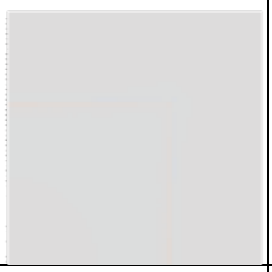
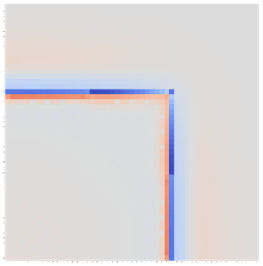
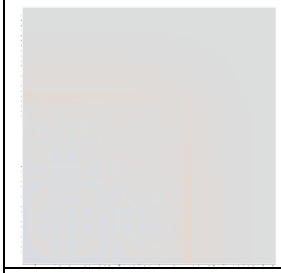
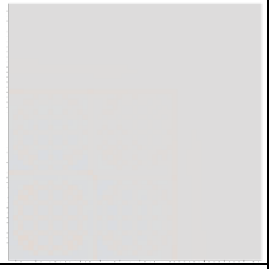
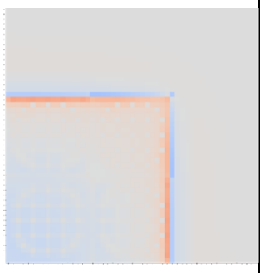
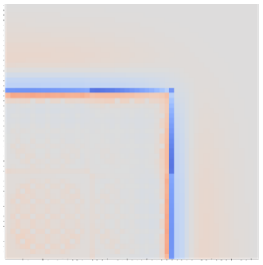
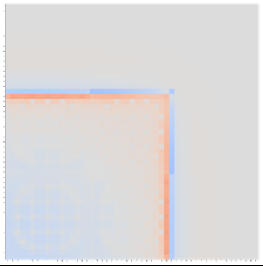
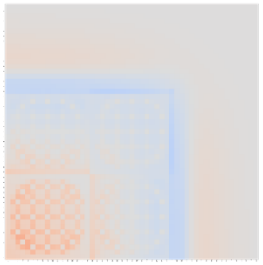
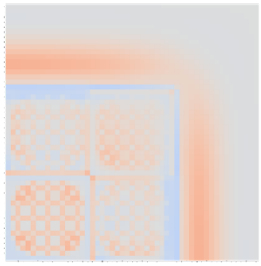
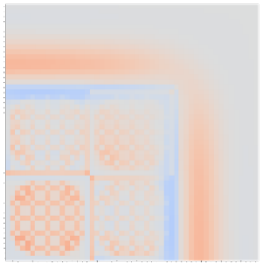
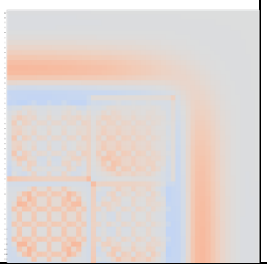
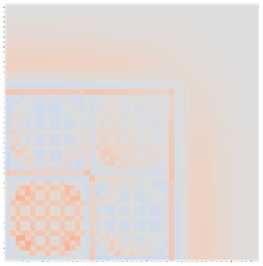
Table 34: Upper and lower boundaries for each energy group for the 23-group mesh

APPENDIX 3: 7-GROUP ENERGY MESH

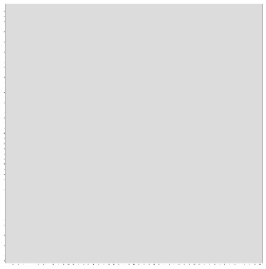
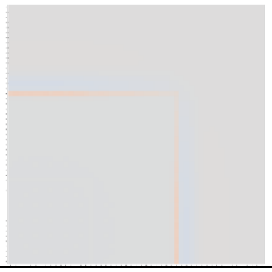
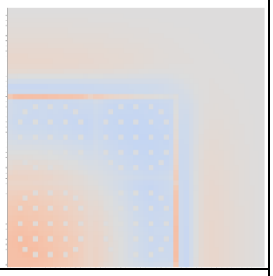
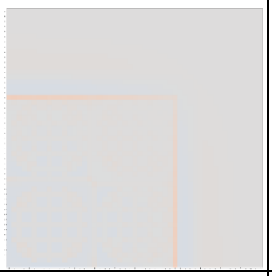
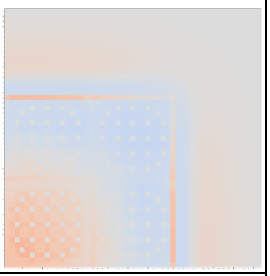
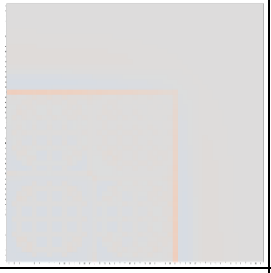
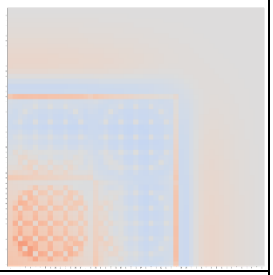
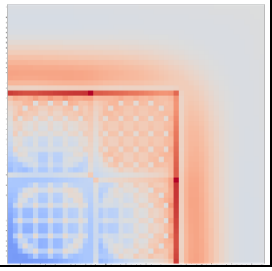
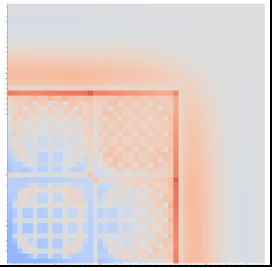
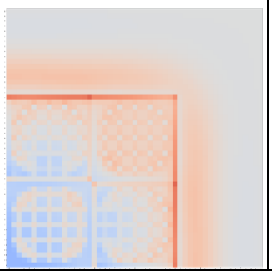
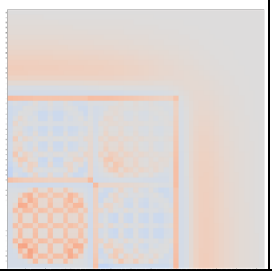
Group number	Group number in 281-SHEM	Upper boundary in energy (eV)	Lower boundary in energy (eV)	Lethargy width
1	1	1,96403E+07	1,33694E+06	2,68720E+00
2	19	1,33694E+06	4,10796E+02	8,08780E+00
3	75	4,10796E+02	7,50456E+01	1,70000E+00
4	85	7,50456E+01	4,00001E+00	2,93180E+00
5	197	4,00001E+00	6,25000E-01	1,85630E+00
6	248	6,25000E-01	1,90005E-01	1,19070E+00
7	262	1,90005E-01	1,10000E-04	7,45433E+00

Table 35: Upper and lower boundaries for each energy group for the 7-group mesh

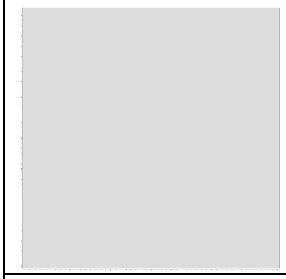
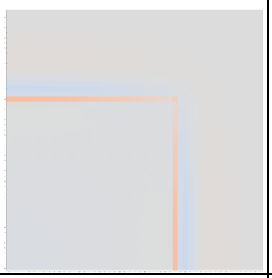
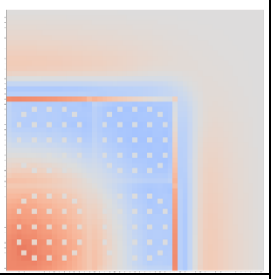
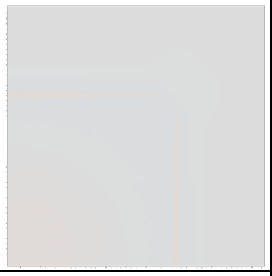
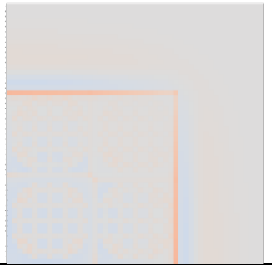
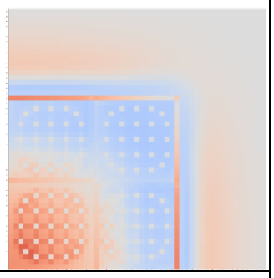
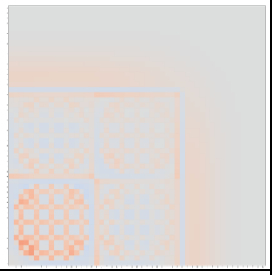
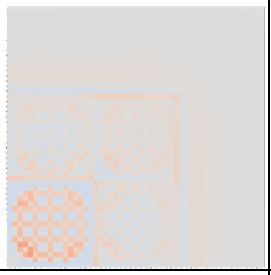
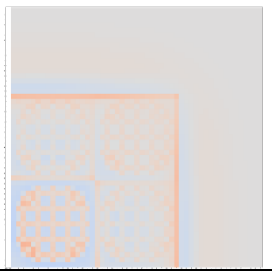
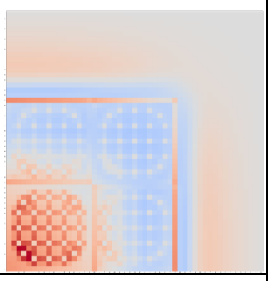
APPENDIX 4: CELL-BY-CELL NORMALIZED ABSORPTION RATE DIFFERENCE (PCM) USING THE BRM

F 281 – R 281	F 281 – R 51	F 281 – R 23	F 281 – R 7	
				
F 51 – R 281	F 51 – R 51	F 51 – R 23	F 51 – R 7	
				
F 23 – R 281	F 23 – R 51	F 23 – R 23	F 23 – R 7	
				
F 7 – R 281	F 7 – R 51	F 7 – R 23	F 7 – R 7	
				

APPENDIX 5: CELL-BY-CELL NORMALIZED ABSORPTION RATE DIFFERENCE (PCM) USING THE CRM

F 281 – R 281	F 281 – R 51	F 281 – R 23	F 281 – R 7	
				
F 51 – R 281	F 51 – R 51	F 51 – R 23	F 51 – R 7	
				
F 23 – R 281	F 23 – R 51	F 23 – R 23	F 23 – R 7	
				
F 7 – R 281	F 7 – R 51	F 7 – R 23	F 7 – R 7	
				

APPENDIX 6: CELL-BY-CELL NORMALIZED ABSORPTION RATE DIFFERENCE (PCM) USING THE FRM

F 281 – R 281	F 281 – R 51	F 281 – R 23	F 281 – R 7	
				
F 51 – R 281	F 51 – R 51	F 51 – R 23	F 51 – R 7	
				
F 23 – R 281	F 23 – R 51	F 23 – R 23	F 23 – R 7	
				
F 7 – R 281	F 7 – R 51	F 7 – R 23	F 7 – R 7