

Titre: Traçage de lignes d'intégration cycliques en 3D pour la résolution
Title: de l'équation de transport des neutrons

Auteur: Mickael Gunka
Author:

Date: 2023

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Gunka, M. (2023). Traçage de lignes d'intégration cycliques en 3D pour la
Citation: résolution de l'équation de transport des neutrons [Mémoire de maîtrise,
Polytechnique Montréal]. PolyPublie. <https://publications.polymtl.ca/53458/>

 **Document en libre accès dans PolyPublie**
Open Access document in PolyPublie

URL de PolyPublie: <https://publications.polymtl.ca/53458/>
PolyPublie URL:

**Directeurs de
recherche:** Alain Hébert, & Guy Marleau
Advisors:

Programme: Génie énergétique
Program:

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

**Traçage de lignes d'intégration cycliques en 3D pour la résolution de l'équation
de transport des neutrons**

MICKAEL GUNKA

Département de génie physique

Mémoire présenté en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
Génie énergétique

Mai 2023

POLYTECHNIQUE MONTRÉAL

affiliée à l'Université de Montréal

Ce mémoire intitulé :

**Traçage de lignes d'intégration cycliques en 3D pour la résolution de l'équation
de transport des neutrons**

présenté par **Mickael GUNKA**

en vue de l'obtention du diplôme de *Maîtrise ès sciences appliquées*
a été dûment accepté par le jury d'examen constitué de :

Cédric BÉGUIN, président

Alain HÉBERT, membre et directeur de recherche

Guy MARLEAU, membre et codirecteur de recherche

Benoît ARSENAULT, membre externe

REMERCIEMENTS

Je tiens tout d’abord à remercier grandement M. Marleau Guy mon co-directeur de recherche et professeur au sein du génie nucléaire de Polytechnique Montréal, pour m’avoir encadré durant ces 2 ans. Il a toujours été disponible et bienveillant à mon égard. J’ai pu grâce à lui développer mes compétences et en acquérir de nouvelles qui me serviront dans mon futur professionnel. Je le remercie aussi de m’avoir donné l’opportunité de contribuer au développement du logiciel DRAGON.

Je remercie également mon directeur de recherche M. Hébert Alain professeur au sein du génie nucléaire de Polytechnique Montréal afin de m’avoir donné cette opportunité au sein du laboratoire de génie nucléaire.

Je remercie également l’école Centrale Marseille qui m’a permis de réaliser mon double diplôme au sein de Polytechnique Montréal et ainsi offert l’opportunité d’étudier à l’étranger pendant 2 ans à Polytechnique Montréal que je remercie aussi.

Je tiens aussi à remercier tous les étudiants avec lesquels j’ai pu travailler au sein du laboratoire de génie nucléaire. Merci à Yvan, Gabrielle, Samy et Tex mes collègues pour avoir été là ainsi que Atyab et Ahmed, tous deux doctorants au laboratoire de génie nucléaire pour leurs conseils avisés et leurs soutiens dans mon travail.

Je remercie aussi énormément les personnes que j’ai découvertes au Canada et qui m’ont soutenu sur ces années et en particulier Médéric, Émile, Léo, Alban et Antonin.

Je remercie enfin mes amis en France qui ont toujours été là pour me soutenir et en particulier Yaël et Gwendoline.

RÉSUMÉ

L'utilisation des centrales nucléaires nécessite de normes de sûretés drastiques. Afin de respecter ces normes, les ingénieurs du nucléaire doivent avoir à leurs dispositions des outils performants afin de comprendre et de simuler l'intérieur des cœurs de réacteurs. Afin d'effectuer ces tâches, des codes de calculs ont été mis au point. Notamment au Canada où le code DRAGON a été développé à Polytechnique Montréal. DRAGON regroupe différents modèles et algorithmes en un seul code et il met l'accent sur la facilité d'implémentation de nouvelles méthodes en ayant une structure en module.

Ces codes de calculs permettent d'étudier la population neutronique et son évolution au travers d'une équation : l'équation de transport des neutrons. C'est une équation primordiale en neutronique, mais aussi dans de nombreuses disciplines liées au nucléaire. Elle décrit de manière statistique le comportement d'une grande population de neutrons dans la matière.

Afin de résoudre cette équation numériquement, de nombreuses méthodes existent. Afin de les mettre au point, il est nécessaire d'avoir un outil permettant de suivre le trajet des neutrons. C'est le rôle des lignes d'intégration. Les lignes d'intégration simulent le trajet des neutrons dans un cœur de réacteur nucléaire. Le traçage de ces lignes s'appelle le tracking. Il existe plusieurs méthodes afin de les tracer. Les méthodes retenant notre attention sont : le tracé uniforme ou isotrope et le tracé cyclique ou spéculaire. Le tracé uniforme est le plus commun et le plus utilisé, mais il nécessite une approximation des conditions frontières, contrairement au tracé cyclique où les conditions frontières sont appliquées directement lors du tracé, cela permet d'avoir des conditions frontières exactes. De plus son nom provient du fait que les lignes reviendront à leur point de départ en formant des cycles sur la géométrie.

L'évolution actuelle des centrales nucléaires voit le développement de nouveaux réacteurs avec des cœurs de plus en plus complexes. Les représentations en deux dimensions deviennent lacunaires et nécessitent des approximations. C'est pour cela que le développement d'une méthode en trois dimensions est nécessaire.

Ce type de tracé en 3D existe déjà dans des logiciels de calcul d'assemblage comme WIMS-11 ou DRAGON. Cependant, il est encore très limité. En effet, dans WIMS-11 il ne permet qu'une analyse lacunaire sur des géométries simples et DRAGON ne permet que des tracés cycliques sur des géométries en 2D même si DRAGON permet des tracés 3D avec une méthode uniforme. Ainsi ma recherche porte sur le développement et l'implémentation d'une méthode cyclique en 3D dans le logiciel DRAGON pour résoudre l'équation de transport des neutrons.

Une méthode de tracé des lignes d'intégrations cycliques en 3D sera développée et implémentée dans DRAGON. De plus, les différences entre tracé cyclique et uniforme seront évaluées afin de connaître l'impact de l'approximation des conditions frontières lors d'un traçage uniforme.

La méthode de tracé cyclique développée en 3D a été inspirée de la méthode 2D déjà programmée dans DRAGON. Elle consiste en l'application de la méthode 2D à deux reprises pour former des lignes d'intégration. En 3D nous devons utiliser les angles azimutales φ et polaires θ et tracer dans les quatre octants supérieurs de la géométrie, les autres octants sont tracés automatiquement par symétries. Nous avons adapté la condition d'arrêt afin qu'elle puisse tracer suffisamment de cycles en 3D et aussi dans les deux directions perpendiculaires à la trajectoire du neutron pour être uniforme sur la géométrie.

Pour implémenter notre nouvelle méthode, nous avons pris la décision de créer de nouveaux programmes à la place de modifier les programmes actuels. Ces nouveaux programmes permettront une plus grande flexibilité. Cela nous permettra aussi de mettre à jour plus aisément les programmes dans l'avenir. Nous avons créé des programmes pour les différentes actions nécessaires au tracé des lignes d'intégrations. De manières globales, nous avons une sous-routine permettant le calcul de toutes les informations nécessaires au tracé cyclique NXTICT (NXT Information cyclic tracking). Et une deuxième sous-routine afin d'exploiter ces résultats : NXTTC3.

Afin de vérifier l'implémentation des sous-routines ainsi que de la validité de notre méthode, nous avons simulé différents cas afin de voir différentes situations ainsi que leurs résultats. Nous avons dans un premier temps effectué une simulation simple avec un cube de dimensions 1 puis nous voulions voir l'effet des lignes d'intégration cyclique dans une géométrie avec plusieurs régions comme la cellule de réacteur REP. Ces calculs ont été effectués afin de vérifier les quadratures surfaciques et volumiques et leur évolution lors d'une augmentation de densité ou/et d'angles.

Pour la quadrature des volumes, les résultats obtenus sont satisfaisants et permettent de conclure sur la validité de notre tracking. Dans une cellule de réacteur REP avec différentes régions nos erreurs sont du même ordre de grandeur qu'avec un tracking uniforme, mais pour un temps de calcul beaucoup plus long. Cependant il faut noter que le calcul des surfaces présente une erreur trop grande pour être exploitable.

Cependant nous n'avons pas réussi à exploiter les lignes d'intégration dans le but de résoudre l'équation de transport des neutrons.

Les résultats sur le tracking de la cellule REP permettent d'affirmer que notre génération de

lignes d'intégration fonctionne. Ce qui valide notre méthode.

ABSTRACT

The use of nuclear reactors is bound by safety rules. To follow these standards, the nuclear engineers must have performant tools to understand and to simulate the inside of the reactor core. In order to carry out these tasks, lattice code have been developed. Especially in Canada, where the code DRAGON has been developed at Polytechnique Montréal. DRAGON is composed of different model and algorithms, and has a modular structure to facilitate the implementation of new methods.

This code gives the opportunity to study the neutronic population and its evolution with an equation : the neutron transport equation. It is important in neutronics but also in a lot of nuclear fields. This equation describes the behavior of a large population of neutron in a statistical way.

We have several numerical methods to solve this equation, but to create them, we need a tool to follow the neutrons. It is the function of the integration line, they simulate the neutron path in the core. The generation of them it called the tracking. We have different means to generate them, but we will focus on the uniform tracking and the cyclic tracking. The uniform tracking is more simple and frequently use, but it makes an approximation to the boundaries conditions. Contrary, to the cyclic one where the boundaries condition are applied during the tracking. Moreover, its name come from the fact that each line is part of a cycle in the geometry.

Due to the evolution of nuclear reactors, the nuclear cores are becoming more and more complex. The 2-dimensions representation lacks of precision for more advanced reactor core, so we need 3-dimensional methods to improve our calculations.

Those 3D methods have already been implemented in lattice code such as WIMS-11 or DRAGON. However, they are limited. Indeed, WIMS-11 can process simple geometry and DRAGON is able to analyze 3D geometrize with standard tracking. So, the goal of my research is to develop and to implement a 3D cyclic method in DRAGON to solve the neutron transport equation. Moreover, we will assess the difference between the cyclic tracking and the uniform tracking, to understand the impact of the approximation of the boundaries conditions in the uniform tracking.

TABLE DES MATIÈRES

REMERCIEMENTS	iii
RÉSUMÉ	iv
ABSTRACT	vii
TABLE DES MATIÈRES	viii
LISTE DES TABLEAUX	xi
LISTE DES FIGURES	xii
LISTE DES SIGLES ET ABRÉVIATIONS	xiv
CHAPITRE 1 INTRODUCTION	1
1.1 Mise en Contexte	1
1.2 Objectifs de recherche	2
1.3 Plan du mémoire	5
CHAPITRE 2 NEUTRONIQUE : BASE THÉORIQUE	6
2.1 Introduction	6
2.2 Concepts neutroniques fondamentaux	6
2.2.1 Neutron	6
2.2.2 Les angles solides	8
2.2.3 Les distributions	9
2.2.4 Le flux angulaire	10
2.2.5 Les groupes d'énergie	12
2.2.6 Le parcours optique	12
2.3 Équation de Boltzmann	13
2.3.1 Équation de transport des neutrons	14
2.3.2 Les conditions frontières	18
2.4 Méthodes de résolutions déterministes	18
2.4.1 Méthode MOC : méthode des caractéristiques	19
2.4.2 Méthode des probabilités de collision	20
2.4.3 Méthode S_n	21

2.4.4	Méthode P_n	23
2.5	Méthode de résolution stochastique	24
2.5.1	Méthode de Monte-Carlo	24
2.6	Conclusion	25
CHAPITRE 3	LES LIGNES D'INTÉGRATION	26
3.1	Introduction	26
3.2	Définitions	26
3.2.1	Le tracking uniforme ou isotrope	26
3.2.2	Le tracking cyclique ou spéculaire	27
3.3	Calcul avec les lignes d'intégration	28
3.3.1	Outil pour le calcul numérique	28
3.3.2	Équivalence des grandeurs	31
3.4	Résolution de l'équation de transport	33
3.4.1	Méthode MOC : méthode des caractéristiques	33
3.4.2	Conditions frontières	35
CHAPITRE 4	IMPLÉMENTATIONS DU TRACÉ DES LIGNES CYCLIQUE EN 3D	37
4.1	Présentation	37
4.2	Autre méthode de tracé	37
4.2.1	MRT Modular Ray Tracing	37
4.2.2	3DGT The 3D global meshing method	37
4.3	Tracking en 2D	38
4.3.1	Choix des angles	38
4.3.2	Les conditions frontières	41
4.3.3	Les directions de la quadrature	44
4.3.4	Poids angulaire et spatial	45
4.3.5	Nombre de points	50
4.3.6	Condition d'arrêt	51
4.3.7	Positionnement du point de départ du cycle	53
4.3.8	Algorithme	54
4.3.9	Conclusion méthode 2D	55
4.4	Tracking cyclique en 3D	55
4.4.1	Choix des angles	57
4.4.2	Les conditions frontières	59
4.4.3	Les directions de la quadrature	59
4.4.4	Poids spatial et angulaire	61

4.4.5	Nombre de points	64
4.4.6	Condition d'arrêt	64
4.4.7	Points de départ du cycle	65
4.4.8	Incrémentation en 3D	66
4.4.9	Traitement des coins	67
4.4.10	Motif	68
4.4.11	Rotation en 3D	70
4.4.12	Numérotation des angles	70
4.4.13	Algorithme	73
4.4.14	Conclusion méthode 3D	74
CHAPITRE 5	VALIDATION ET VÉRIFICATION DU PROGRAMME	75
5.1	Le Code DRAGON	75
5.1.1	Les modules de Tracking	75
5.1.2	Le module NXT :	76
5.1.3	Implantation dans le module NXT : de DRAGON	77
5.2	Vérification des programmes	80
5.2.1	Vérification analogique	81
5.2.2	Vérification numérique	85
CHAPITRE 6	CONCLUSION	90
6.1	Synthèse des travaux	90
6.2	Limitations de la solution proposée	91
6.3	Améliorations futures	92
RÉFÉRENCES		93

LISTE DES TABLEAUX

Tableau 4.1 :	Les entiers m_x et m_y pour $N_{ANGLE} = 8$ en 2D.	41
Tableau 4.2 :	Numérotation des octants.	70
Tableau 4.3 :	Indexation des angles pour une quadrature d'ordre 2.	71
Tableau 5.1 :	Les sous-routines principales afin de générer les lignes d'intégration uniformes.	78
Tableau 5.2 :	Les sous-routines principales afin de générer les lignes d'intégration cycliques en 2D.	78
Tableau 5.3 :	Les différentes sous-routines afin de générer les lignes d'intégration cyclique en 3D.	79
Tableau 5.4 :	Récapitulatif des erreurs de quadrature surfaciques et volumiques pour différents angles et différentes densités avec un tracé uniforme pour référence.	86
Tableau 5.5 :	Récapitulatif des erreurs quadratiques pour les quadrature surfa- ciques et volumiques pour une cellule de réacteur REP.	88

LISTE DES FIGURES

Figure 1.1 :	Assemblage de réacteur nucléaire REP.	3
Figure 1.2 :	Mécanisme de contrôle de réactivité d'un réacteur CANDU.	4
Figure 2.1 :	Représentation du volume de contrôle.	7
Figure 2.2 :	Représentation de l'angle solide.	8
Figure 2.3 :	Représentation d'une quantité de neutrons de direction Ω , de vitesse V_n durant un temps dt	11
Figure 2.4 :	Représentation des différentes méthodes de résolutions numériques de l'équation de transport des neutrons.	17
Figure 2.5 :	Représentation d'une discrétisation spatial en une dimension.	22
Figure 3.1 :	Représentation des lignes d'intégration uniformes sur une cellule carrée.	27
Figure 3.2 :	Représentation d'une procédure de tracking avec 8 angles sur une cellules carrée.	28
Figure 3.3 :	Représentation des lignes d'intégration dans une cellule comportant différentes régions.	30
Figure 3.4 :	Mise en évidence des conditions frontières de réflexion (droite) et de translation (gauche).	36
Figure 4.1 :	Représentation de la cellule de travail et notations des surfaces en 2D.	38
Figure 4.2 :	Représentation de 8 angles dans une cellule.	39
Figure 4.3 :	Première ligne d'un cycle avec condition frontière de translation sur une figure dépliée et non dépliée.	42
Figure 4.4 :	Première ligne d'un cycle avec condition frontière de réflexion sur une figure dépliée et non dépliée avec $m_x = 2$ et $m_y = 3$	43
Figure 4.5 :	Représentation des 4 directions et des 4 quadrants en 2D.	45
Figure 4.6 :	Représentation de l'écartement des lignes d'intégration.	47
Figure 4.7 :	Représentation d'une quadrature pour une direction et des conditions frontières de translation.	49
Figure 4.8 :	Nombre de point pairs, impairs sur des lignes d'intégration à très faible densité.	51
Figure 4.9 :	Représentation d'un nombre de points supérieure à la limite admissible par la géométrie	52
Figure 4.10 :	Différence entre le nombre de points effectif et possible.	53

Figure 4.11 :	Représentation des lignes limites, de la première ligne ainsi que du point de départ.	54
Figure 4.12 :	Algorithme en pseudo code pour les lignes d'intégration cycliques en 2D.	55
Figure 4.13 :	Représentation du plan de base en 2D et du plan U sur la géométrie 3D.	56
Figure 4.14 :	Représentation de la cellule de travail et notation des surfaces en 3D.	57
Figure 4.15 :	Représentation des angles polaires pour un angle azimutal.	58
Figure 4.16 :	Représentation des 8 octants et des 8 directions.	60
Figure 4.17 :	Représentation des 4 directions de départ des 4 cycles pour des conditions frontières de translation totale.	61
Figure 4.18 :	Représentation d'une quadrature du volume en 3D, pour des conditions frontières de translation totale.	63
Figure 4.19 :	Représentation des lignes limites en rouge pour la 3D et en bleu pour la 2D.	65
Figure 4.20 :	Alignement des points de départ selon un angle de vue quelconque et selon le plan perpendiculaire à la direction $\vec{t}$	67
Figure 4.21 :	Représentation d'un motif en 2D pour un angle azimutal fixe.	69
Figure 4.22 :	Motif valide (droite) et invalide (gauche) en 3D.	69
Figure 4.23 :	Algorithme en pseudo-code pour les lignes d'intégration cycliques en 3D.	73
Figure 5.1 :	Structure des données d'entrée descnxt.	76
Figure 5.2 :	Structure descnxt pour les lignes d'intégration uniformes (TISO).	78
Figure 5.3 :	Structure descnxt pour les lignes d'intégration cycliques en 2D (TSPC 2D).	79
Figure 5.4 :	Structure descnxt pour les lignes d'intégration cycliques en 3D (TSPC 3D).	80
Figure 5.5 :	Représentation des surfaces et des régions pour une cellule de réacteur REP.	81
Figure 5.6 :	Représentation des quadratures sur les surfaces.	82
Figure 5.7 :	Projection des surfaces en 2D.	83
Figure 5.8 :	Segment des lignes d'intégration dans les différentes régions.	84
Figure 5.9 :	Projection des surfaces en 2D.	85

LISTE DES SIGLES ET ABRÉVIATIONS

CANDU	Canada Deuterium Uranium
EPR	Réacteur Pressurisé Européen
MOC	Méthode des Caractéristiques
REP	Réacteur à Eau sous Pression

CHAPITRE 1 INTRODUCTION

1.1 Mise en Contexte

L'énergie sous toutes ses formes accompagne l'histoire de l'humanité. La première révolution énergétique, celle du charbon a permis une évolution et des progrès titanesques dans la société. Aujourd'hui, issue des énergies fossiles, renouvelables ou nucléaires ; l'électricité est devenue peu à peu le sang de notre économie et a une place primordiale dans nos vies personnelles [1]. Notre manière de consommer implique un besoin toujours plus grand d'électricité, cependant les enjeux environnementaux et la question du réchauffement climatique dû aux gaz à effet de serre demandent une production d'énergie bas-carbone. Cette exigence donne une place d'importance à l'électricité dans l'avenir. Cependant, cette électricité doit être disponible à tout moment, ces propriétés appartiennent uniquement à l'énergie hydraulique et à l'énergie nucléaire. L'hydraulique demande des caractéristiques géographiques particulières, les sites permettant son développement se raréfient et les nouveaux lieux de productions demandent souvent de noyer de grandes zones habitables ou exploitables [2]. De plus le nucléaire propose un coût de production compétitif face aux énergies renouvelables et profite d'une production pilotable avec un faible impact environnemental [3].

Grâce à ses caractéristiques le nucléaire apparaît alors comme la solution pour une transition énergétique et une abolition de l'utilisation des combustibles fossiles. Cependant, le nucléaire est une énergie ne gagnant pas le cœur du public. Les déchets nucléaires restent une question en suspens, la peur des catastrophes nucléaires comme Fukushima ou Tchernobyl reste gravée dans l'inconscient collectif. La bombe nucléaire qui ne possède comme points communs avec l'énergie que le nom est toujours une question de société. Ces sujets sont utilisés pour créer une peur des énergies nucléaires, basée non pas sur des avis scientifiques, mais sur la peur provenant d'une incompréhension d'une technologie complexe. Les événements géopolitiques actuels comme la guerre entre la Russie et l'Ukraine ont permis de relancer le nucléaire à un niveau mondial, dû à la peur d'une dépendance énergétique à des pays comme la Russie. La France relance la construction de plusieurs EPR (European Pressurised Reactor) sur son territoire et de plus en plus de pays choisissent le nucléaire.

L'industrie nucléaire est une industrie de précision où les erreurs ne sont pas admises. La sûreté des centrales nucléaires prend comme référence, pour la partie cœur de réacteur et l'étude du comportement des neutrons, des codes de calculs dits d'assemblage. Il existe de nombreux codes de calculs d'assemblage, nous allons utiliser le code DRAGON développé à Polytechnique Montréal. DRAGON est un code de calcul OpenSource développé en partie

par des étudiants. C'est un code de calcul neutronique qui résout l'équation de transport des neutrons dans un assemblage de réacteur nucléaire.

L'équation de transport des neutrons dérivant de l'équation de Boltzmann réalise un bilan sur une population de neutron dans un espace défini comme une cellule de réacteur nucléaire. C'est l'équation la plus importante en physique neutronique. Elle permet de déterminer la position des neutrons ainsi que leur évolution au cours du temps et ainsi expliciter les différents phénomènes physiques se passant au sein du cœur. Pour utiliser cet outil, nous aurons besoin de la trajectoire des neutrons qui nous sera donnée par notre objet d'étude : les lignes d'intégrations. Le tracé cyclique que nous étudions ici en trois dimensions nous permet l'utilisation directe des conditions frontières appliquées sur le réacteur. Cela nous permettra une amélioration dans la précision des calculs.

1.2 Objectifs de recherche

Les objectifs de la recherche sont d'analyser dans un premier temps la possibilité de créer et d'implémenter dans DRAGON une méthode de tracé des lignes d'intégrations cycliques en 3D. Une fois la méthode de tracé définie, nous devons élaborer un algorithme et valider notre tracé sur des géométries simples. À la suite de la validation nous intégrerons notre méthode de tracé des lignes d'intégration cyclique dans DRAGON afin qu'elles fonctionnent avec les différentes méthodes de résolution de l'équation de transport. Une fois l'intégration dans le logiciel DRAGON effectué, nous pourrons utiliser notre méthode sur un cube d'arrêt 1 centimètre, une cellule 3D de réacteur REP ou un mécanisme de contrôle de réactivité entouré de deux cellules CANDU.

Pour comprendre les différences entre les réacteurs REP et CANDU il est nécessaire de comparer cellules ou assemblages de chacun de ces réacteurs.

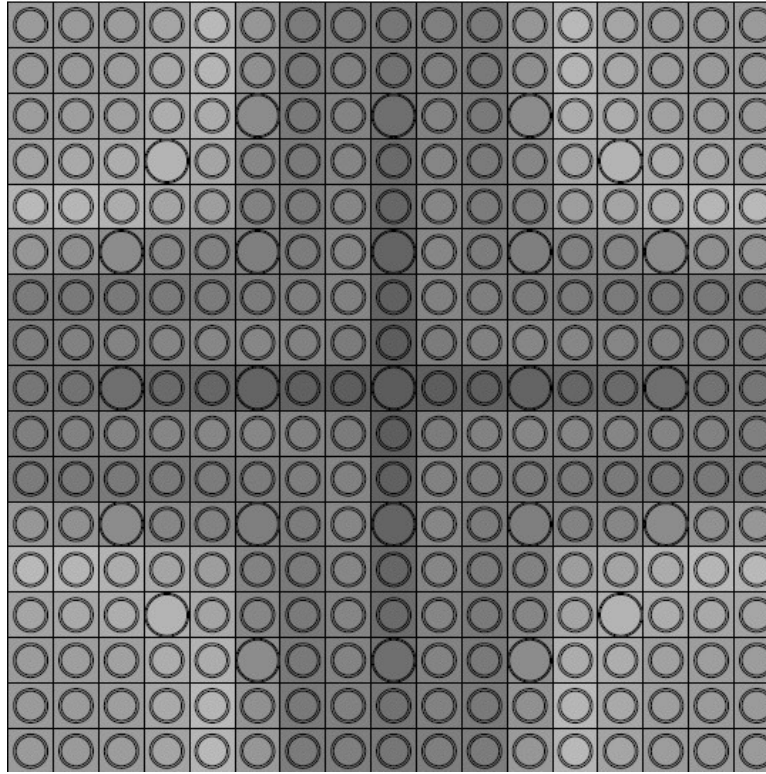


FIGURE 1.1 Assemblage de réacteur nucléaire REP.

La représentation d'un assemblage de crayons de combustible de réacteur à eau sous pression voir figure (1.1) est une représentation effectuée par DRAGON, les caractéristiques géométriques proviennent de références de réacteur nucléaire issu de [4], cette représentation nous permet de voir que le coeur d'un réacteur REP peut être décrit par une représentation en 2D.

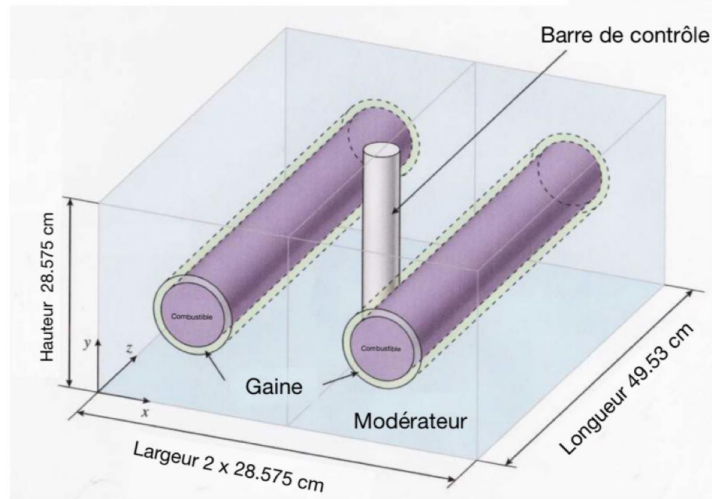


FIGURE 1.2 Mécanisme de contrôle de réactivité d'un réacteur CANDU.

La figure (1.2) représente schématiquement un mécanisme de réactivité CANDU localisé entre deux grappes de combustible. Cette représentation met en évidence que dans un coeur de réacteur CANDU les barres de contrôle sont placées de façon orthogonale aux grappes de combustible. Cette disposition rend les représentations 2D d'un tel réacteur incomplètes. Les représentations 3D permettent une meilleure compréhension du système. C'est pour cela que nous effectuerons notre étude en trois dimensions, car les conditions aux frontières isotropes pour les tubes de combustible et les mécanismes de réactivité sont inadéquates pour tenir compte des flux directionnels provenant de l'intérieur de ces régions.

Notre étude a pour objectif de résoudre l'équation de transport des neutrons dans les réacteurs REP et CANDU. La méthode appliquée sera une méthode de tracé cyclique des lignes d'intégration. Le tracé cyclique ou spéculaire permet de prendre en compte les conditions frontières de manière exactes contrairement au tracé uniforme. Ce tracé nous permettra d'évaluer l'impact des conditions frontières approximatives du tracé uniforme sur par exemple la valeur en réactivité des mécanismes de contrôle d'un CANDU ce que ne fait que grossièrement la méthode uniforme actuellement codée dans DRAGON, car elle fait une approximation des conditions frontières. Le but sera de voir l'efficacité de la méthode cyclique sur des géométries où les conditions frontières jouent un rôle important.

1.3 Plan du mémoire

Pour présenter notre sujet, nous effectuerons dans un premier temps une mise en contexte dans le domaine neutronique. Ainsi nous définirons les différents outils dont nous aurons besoin afin de résoudre l'équation de transport des neutrons, et aussi les différentes méthodes employées pour sa résolution. Dans un second temps nous ferons la présentation des lignes d'intégration, leur utilisation dans les méthodes de résolution de l'équation de Boltzmann sera mise en évidence et les équivalences numériques seront détaillées afin de permettre une compréhension plus globale. Nous expliquerons par la suite notre méthode de tracé des lignes d'intégration cyclique en 3D, pour cela nous détaillerons le tracé en 2D dans un premier temps. Enfin nous expliciterons l'implémentation de notre méthode dans le logiciel DRAGON et nous effectuerons une qualification où les résultats sont comparés aux surfaces et volumes calculés de façon analytique.

CHAPITRE 2 NEUTRONIQUE : BASE THÉORIQUE

2.1 Introduction

Dans le but de comprendre notre étude, nous avons besoin de mettre en contexte le concept des lignes d'intégration et leur importance dans la résolution de l'équation de transport des neutrons. Nous allons nous intéresser à l'étude de la physique du neutron : la neutronique. Les notions fondamentales pour notre étude seront introduites dans cette partie, ainsi que la présentation de l'équation de transport des neutrons et sa signification physique. Enfin, différentes méthodes de résolution seront détaillées.

2.2 Concepts neutroniques fondamentaux

Pour résoudre l'équation de transport des neutrons nous allons dans un premier temps reprendre certaines bases physiques et mathématiques essentielles en neutronique.

2.2.1 Neutron

Le neutron est une des particules constituant les noyaux des atomes et donc la matière. Il est de charge électrique nulle, donc nous n'aurons pas d'effet électromagnétique en neutronique. C'est une particule essentielle en nucléaire étant l'un des produits de fission les plus importants, permettant d'entretenir la réaction, elle est aussi utilisée afin de démarrer les réactions nucléaires. Elle est notée par le symbole : n ou encore n^0 . Son importance est primordiale en physique nucléaire et le contrôle de la population neutronique permet d'assurer la sûreté nucléaire. Nous l'étudierons au travers de l'équation de transport, mettant en évidence l'évolution d'une population de neutrons dans un réacteur nucléaire ou alors une cellule de réacteur nucléaire.

Nous étudierons les neutrons avec une approche de mécanique statistique : la propriété de chaque particule sera identifiée par 7 coordonnées.

1. Trois positions dans l'espace $\vec{r} = x\vec{i} + y\vec{j} + z\vec{k}$. Les vecteurs $\vec{i}, \vec{j}$ et $\vec{k}$ forment une base orthonormée directe dans l'espace
2. Trois vitesses, elles sont obtenues en dérivant le vecteur position par le temps. $\vec{V} = \frac{d\vec{r}}{dt}$. Nous définissons par la même occasion la direction : $\vec{\Omega} = \frac{\vec{V}}{V_n}$ avec V_n la norme du vecteur vitesse.
3. Le temps t , utilisé dans les régimes transitoires.

L'intérêt dans notre étude est de suivre l'évolution d'une population de neutrons dans un volume défini. Nous utiliserons la densité de population afin de connaître la quantité de neutrons. Nous la représentons par une distribution : $n(\vec{r}, V_n, \vec{\Omega}, t)$ de ce fait nous aurons la quantité suivante décrivant le nombre de particules par unité de volume à un temps t :

$$n(\vec{r}, V_n, \vec{\Omega}, t) d^3r dV_n d^2\Omega \quad (2.1)$$

- d^3r Le volume élémentaire autour du point $\vec{r}$.
- dV_n l'élément de vitesse élémentaire autour de $\vec{V}_n$.
- $d^2\Omega$ étant l'élément d'angle solide autour de $\vec{\Omega}$.

Nous cherchons à définir un volume de contrôle C . Pour le définir, utilisons une distance $\vec{r}$ dans un repère orthonormé direct. Sur la figure (2.1) nous pouvons voir le volume de contrôle C défini ainsi que la direction des neutrons : $\vec{\Omega}$

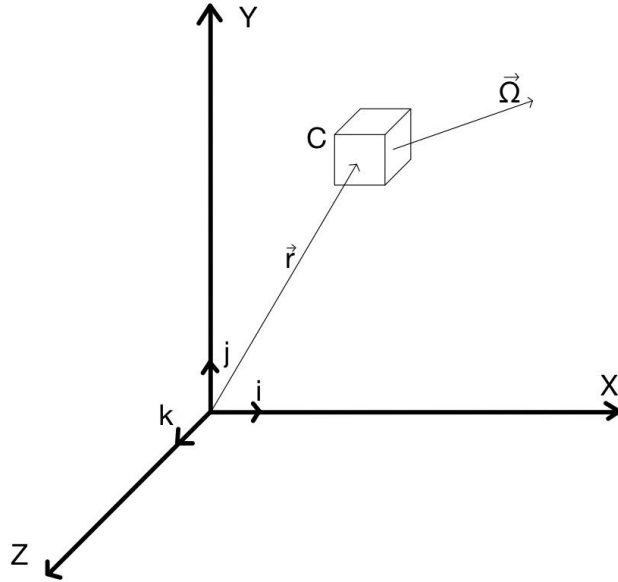


FIGURE 2.1 Représentation du volume de contrôle.

En utilisant le volume C nous pouvons intégrer la formule (2.1) afin d'obtenir le nombre de particules dans ce volume :

$$N_{\text{particule}}(t) = \int_{V_n} \int_{\Omega} \int_C n(\vec{r}, V_n, \vec{\Omega}, t) d^3r dV_n d^2\Omega \quad (2.2)$$

On remarque que nous avons une fonction du temps, mais une distribution par rapport aux autres paramètres.

2.2.2 Les angles solides

L'angle solide nous permettra de donner la trajectoire du neutron dans la cellule. Dans un espace en trois dimensions, nous définissons un angle azimutal φ et un angle polaire θ . Dans l'espace en considérant une surface β sur la sphère de rayon unitaire en plaçant l'observateur au centre de ce système, l'angle sous lequel on voit la surface β est par définition l'angle solide [5]. Si nous déplaçons nos angles de valeur infinitésimale : $d\varphi$ et $d\theta$ nous obtenons un élément d'angle solide : $d^2\Omega$ que nous pouvons aussi écrire :

$$d^2\Omega = \sin(\theta)d\theta d\varphi \quad (2.3)$$

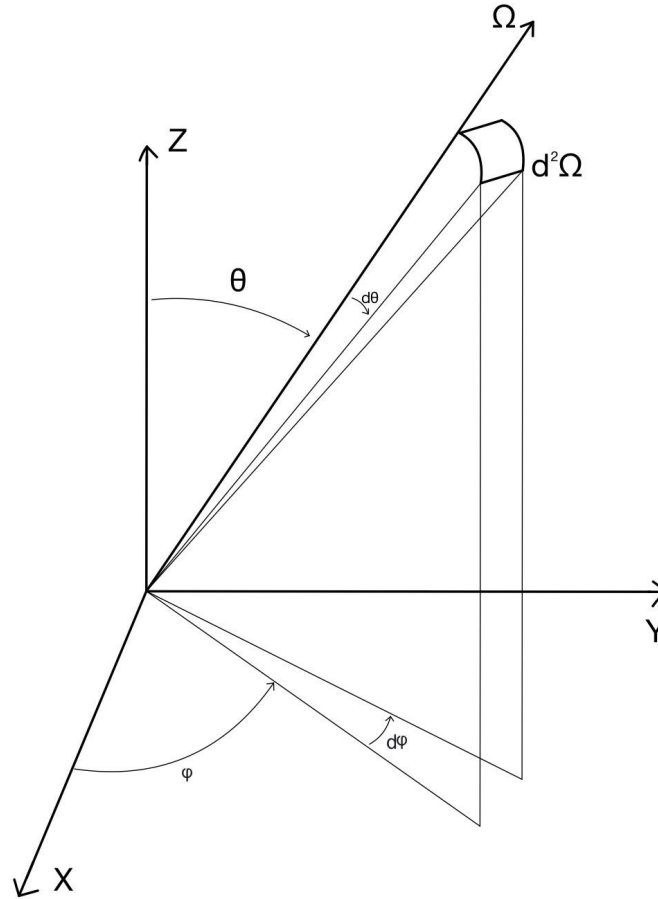


FIGURE 2.2 Représentation de l'angle solide.

L'angle solide nous permettra d'exprimer toutes les valeurs physiques sur une sphère ou partie de sphère. On note que la direction sous laquelle on voit tout l'espace vaut 4π stéradians, soit la surface de la sphère de rayon unitaire.

2.2.3 Les distributions

Nous allons ici faire un point mathématique afin d'expliquer les distributions et leurs utilisations dans les différentes démonstrations suivantes.

Il est important de noter qu'en physique, l'outil des fonctions peut être assez limité lors de son utilisation. En effet, les fonctions posent des problèmes à un niveau théorique en physique. Une fonction associe en un point une valeur, ce qui est en physique parfois impossible. Certes nous pouvons associer en un point une valeur physique et cela nous permettra de résoudre les équations, cependant il n'est pas toujours physiquement juste de le faire. On peut prendre comme exemple la température, qui pour résoudre des équations comme l'équation de chaleur peut être donnée en chaque point d'un domaine. Cependant la température n'est pas une variable que l'on peut associer à point étant donné qu'elle provient de l'agitation des particules moyennée. Il existe une autre raison venant des limites des appareils de mesures : les mesures effectuées par un appareil ne sont pas parfaites et ne peuvent décrire un point spatial exact, mais donnent plutôt une moyenne spatiale de la valeur physique. Comme nous pouvons le voir dans [6] qui donne une définition mathématique très précise des distributions, qui est la suivante :

"Une distribution sur l'ouvert Ω est une forme linéaire T , telle que pour toutes suites d'éléments φ_n avec $n \in N$ qui converge vers 0 alors $T(\varphi_n) \rightarrow_{n \rightarrow \infty} 0$ ".

C'est une définition très abstraite et générale. En pratique on utilisera surtout les différentes propriétés des distributions afin de les manipuler dans nos équations. Nous allons reprendre les propriétés des distributions énoncées dans [7] afin de les utiliser par la suite. Soit une distribution de x : $D(x)$

- $D(x) \geq 0$
- Le produit d'une distribution et d'une fonction est une distribution
- Le produit de deux distributions n'a pas de sens
- Soit $D_2(x, y)$ une distribution de x et y on peut obtenir une distribution de x en intégrant y sur son support S_y

$$D(x) = \int_{S_y} dy D_2(x, y) \quad (2.4)$$

- Soit une distribution $D_a(x)$ sur un support S_x . On peut obtenir $D_b(y)$ avec $y = f(x)$ où f est une fonction strictement croissante ou décroissante :

$$D_b(y) = D_a(x) \left| \frac{dx}{dy} \right| \quad (2.5)$$

l'expression prendra un signe négatif si $\frac{dy}{dx} < 0$

Nous utiliserons à de nombreuses reprises les distributions dans nos calculs afin de démontrer les différentes formes possibles de l'équation de transport des neutrons.

2.2.4 Le flux angulaire

Nous avons défini le neutron ainsi que la densité de population neutronique. Cependant, lors de l'étude des réacteurs nucléaires nous utiliserons aussi comme variables le flux angulaire. La raison la plus importante pour son utilisation et sa simplicité de manipulation dans les équations.

$$\phi(\vec{r}, V_n, \vec{\Omega}, t) = V_n n(\vec{r}, V_n, \vec{\Omega}, t) \quad (2.6)$$

Ce choix est justifié par la simplification des équations qu'il induit. N'ayant pas de signification physique exacte, le flux angulaire est une définition mathématique. Nous utiliserons une formule modifiée en utilisant la formule de réduction (2.5) sur la variable $\vec{\Omega}$ et nous obtiendrons le flux intégré.

$$\phi(\vec{r}, V_n, t) = \int_{4\pi} d^2\Omega \phi(\vec{r}, V_n, \vec{\Omega}, t) \quad (2.7)$$

Ici nous pouvons établir une signification physique. Le flux intégré sur l'angle solide représente le nombre de neutrons de vitesse V_n croisant une surface de 1 cm^2 par seconde. Il peut être écrit en fonction de l'énergie des particules ou bien de la léthargie u .

$$E = \frac{1}{2} m V_n^2 \quad u = \ln \frac{E_0}{E} \quad (2.8)$$

Avec E_0 l'énergie maximale des particules et m la masse de la particule.

De plus, nous allons définir aussi le courant de particules. Soit le nombre de particules de vitesse V_n et de direction $\vec{\Omega}$ passant par une surface d^2S durant un temps dt . Le courant de particule est égal au nombre de particules présentes dans un cylindre de hauteur égale à la distance parcourue durant dt à la vitesse V_n c'est-à-dire $V_n dt$. La représentation 2.3 montre un cylindre de hauteur $V_n * dt$. $\vec{N}$ représente la normale à la surface d^2S

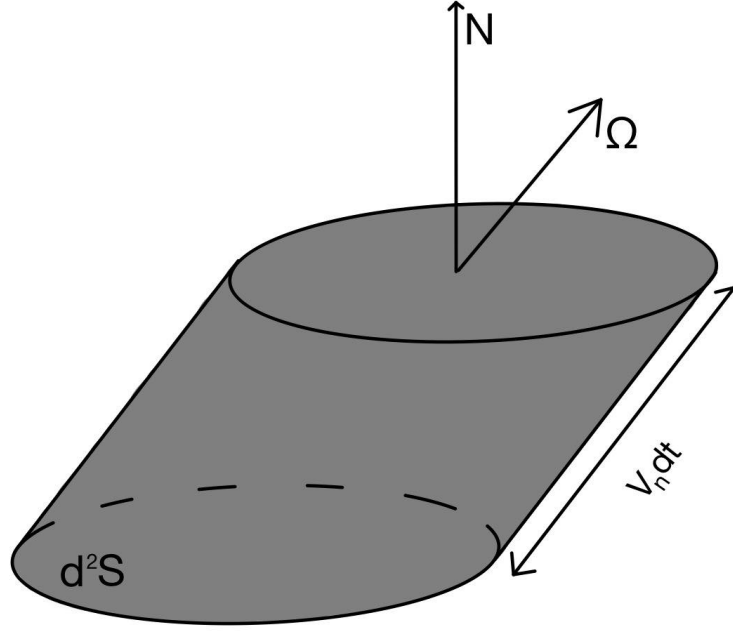


FIGURE 2.3 Représentation d'une quantité de neutrons de direction Ω , de vitesse V_n durant un temps dt .

La quantité de neutrons peut être exprimée par la formule suivante :

$$d^3n = n(\vec{r}, V_n, \vec{\Omega}, t)(d^2S(V_n dt)(\vec{\Omega} \cdot \vec{N})) \quad (2.9)$$

La quantité décrite par (2.9) est représenté par le cylindre sur la figure (2.3).

Nous l'exprimons en termes de flux de particule :

$$d^3n = \vec{\Omega} \cdot \vec{N} \phi(\vec{r}, V_n, \vec{\Omega}, t) d^2S dt \quad (2.10)$$

Et ainsi nous pouvons définir le courant angulaire $J(r, V_n, \Omega, t)$:

$$\begin{aligned} \vec{J}(\vec{r}, V_n, \vec{\Omega}, t) \cdot \vec{N} &= \frac{d^3n}{d^2S dt} \\ \vec{J}(\vec{r}, V_n, \vec{\Omega}, t) &= \vec{\Omega} \phi(\vec{r}, V_n, \vec{\Omega}, t) \end{aligned} \quad (2.11)$$

Le courant angulaire intégré est sortant si nous sommes dans la direction N et entrant sinon.

$$\begin{aligned}
\vec{J}^+(\vec{r}, V_n, t) &= \int_{\Omega \cdot N > 0} d^2\Omega (\vec{\Omega} \cdot \vec{N}) \phi(\vec{r}, V_n, \vec{\Omega}, t) \\
\vec{J}^-(\vec{r}, V_n, t) &= - \int_{\Omega \cdot N < 0} d^2\Omega (\vec{\Omega} \cdot \vec{N}) \phi(\vec{r}, V_n, \vec{\Omega}, t)
\end{aligned} \tag{2.12}$$

Nous verrons dans la suite comment exploiter ces expressions afin de créer des expressions numériques.

2.2.5 Les groupes d'énergie

L'énergie d'un neutron fait référence à l'énergie cinétique de ce dernier. Le neutron peut être décrit par son énergie cinétique ou alors par le module de sa vitesse, ces deux grandeurs sont reliées par la formule suivante : $E = \frac{1}{2}m|\vec{V}|^2$. Les différents groupes d'énergie représentent alors aussi différents groupes de neutrons dont le module de leur vitesse est identique.

Afin de résoudre l'équation de transport des neutrons, une méthode de multigroupe d'énergie est utilisée. Le but de cette méthode est de discrétiser le spectre d'énergie en G groupe à l'aide d'un maillage énergétique. Dans chaque groupe d'énergie, les neutrons ont la même vitesse, et les grandeurs dépendantes de l'énergie seront moyennées sur la valeur énergétique du groupe. Les différentes équations sont établies pour une énergie spécifique appartenant au groupe d'énergie. Cette méthode s'appelle aussi la condensation en énergie [7]. On utilise la condensation à plusieurs reprises dans le code notamment pour le calcul de section efficaces définies sur la librairie des sections efficaces, dans ce cas $G \in [50, 400]$. Nous aurons aussi une condensation lors du calcul du réacteur. Dans ce cas si nous aurons $G \in [2, 8]$. Nous pouvons avoir d'autres condensations au cours des calculs. Dans l'exemple de deux groupes, nous aurons les neutrons rapides et les neutrons thermiques [8]. Une condensation à un seul groupe d'énergie n'est pas applicable due aux variations importantes des vitesses de neutrons qui ne peuvent être représentées par un seul groupe.

2.2.6 Le parcours optique

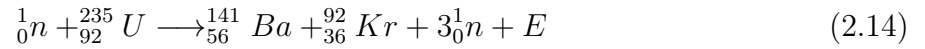
Le parcours optique est défini comme le nombre de collisions entre le neutron et la matière vue par le neutron capturé au point $\vec{r}$ et provenant du point $\vec{r} - s'\vec{\Omega}$. C'est pour cela que la section efficace macroscopique rentre en jeu [9]. Il est exprimé pour une vitesse $\vec{V}$ par :

$$\tau(s, V_n, \vec{\Omega}) = \int_0^s ds' \Sigma(\vec{r} - s'\vec{\Omega}, V_n, \vec{\Omega}) \tag{2.13}$$

2.3 Équation de Boltzmann

Dans un réacteur nucléaire, la connaissance de l'évolution des neutrons est primordiale, car ce sont les acteurs majoritaires des réactions de fission alimentant les réacteurs. Dans les équations de désintégration nucléaire, les neutrons sont produits et ils mènent à de nouvelles réactions. En soi, on peut admettre que le contrôle sur la population neutronique est un critère obligatoire dans un réacteur nucléaire pour la sécurité, mais aussi pour son fonctionnement.

On peut prendre en exemple l'uranium 235 (U^{235}) dans les réacteurs REP qui, bombardé par des neutrons, réagira en fissionnant et en libérant 2 noyaux plus légers et des neutrons (en moyenne cette réaction libérera 2,5 neutrons). On peut prendre comme exemple : l'équation (2.14) de l'uranium 235 donnant du baryum et du krypton.



On note que c'est un résultat possible de la désintégration de l'uranium, ce n'est pas la seule.

L'équation de Boltzmann est une équation permettant d'évaluer l'évolution d'une population dans un volume de contrôle. Elle a été adaptée pour l'étude nucléaire de l'équation originelle de Boltzmann écrite en 1879 pour des gaz [10]. Dans notre cas, nous allons étudier cette équation dans son application sur des particules neutres : les neutrons et leur évolution dans un réacteur nucléaire. D'un point de vue mathématique, c'est une équation intégral-différentielle dont la solution n'est pas triviale à obtenir. Nous verrons qu'il existe de nombreuses méthodes afin de la résoudre de manière exacte ou non.

Cette équation prendra en compte les différents changements dans la population neutronique dans le domaine afin de trouver son évolution. Pour mettre en évidence la variation des neutrons, nous allons citer différentes manières dont cette population peut fluctuer (cette liste est non exhaustive) :

- La définition de la densité neutronique (2.1) inclut la présence d'un volume de contrôle. Les neutrons se déplaçant dans le réacteur et peuvent entrer et sortir du volume de contrôle. Si nous prenons comme volume de contrôle le réacteur entier, nous n'aurons pas de neutrons entrants.
- En fonction des sections efficaces, les neutrons peuvent être capturés ou diffusés. Les neutrons capturés disparaissent et les neutrons diffusés changent de direction ainsi que d'énergie.
- Des neutrons instantanés peuvent être produits lors d'une réaction d'absorption menant à une réaction de fission nucléaire.
- Des neutrons retardés peuvent provenir de la désintégration de noyaux produits par

la fission.

- Les neutrons peuvent provenir de sources externes. On peut citer ici les canons à électrons permettant le démarrage de certains réacteurs nucléaires.

Cette liste a été obtenue grâce au livre [9] ainsi que [10].

Cette équation prend donc en compte ces différentes façons de générer ou d'éliminer les neutrons du volume de contrôle. Afin de pouvoir réaliser les calculs et résoudre l'équation, nous devons établir plusieurs hypothèses. En neutronique, les hypothèses suivantes nous permettent d'établir l'équation de Boltzmann ainsi que de la résoudre :

1. Le nombre de neutrons dans le réacteur est suffisamment important pour qu'une densité neutronique soit définie. Nous nous servons de la valeur de 10^8 neutrons par cm^3 afin de considérer cette condition réalisée.
2. Les interactions neutrons-neutrons sont négligeables face aux interactions neutron - matière. Le rapport entre la densité d'atome du milieu et celle des neutrons varie généralement entre 10^{10} et 10^{15} . C'est une condition importante simplifiant grandement l'équation de Boltzmann.
3. Les effets relativistes des neutrons sont négligeables, nous travaillons donc en physique classique et non en physique relativiste.
4. Les trajectoires des neutrons sont des lignes droites, les neutrons n'ont pas de charge électrique.
5. La désintégration des neutrons en protons est négligée. En effet, sa demi-vie est bien supérieure au temps mis entre sa production et sa disparition par absorption dans le réacteur.
6. La matière est isotrope dans l'espace. Elle possède les mêmes propriétés physiques dans toutes les directions. Les effets d'une collision neutron-matière ne dépendront donc pas de l'angle de collision du neutron.

Après avoir défini le cadre de notre équation, nous présentons dans la partie suivante une méthode afin d'obtenir l'équation de Boltzmann aussi nommée équation de transport.

2.3.1 Équation de transport des neutrons

L'équation de transport dans un milieu est une relation regroupant une grande diversité de phénomènes de la propagation de matières comme les photons, les neutrons ou encore des courants en mécanique des fluides.

L'équation de transport est utilisée en physique nucléaire dans le but de décrire une population de particules neutres (neutrons ou photons), dans un domaine fermé, en régime

transitoire ou en régime stationnaire. Elle est basée sur le comportement statistique d'une large population de particules. La quantité de particules par unité de volume est en variation permanente, y compris en régime stationnaire. La solution de l'équation de transport des neutrons est primordiale dans de nombreux domaines du nucléaire comme en physique des réacteurs ou encore en radioprotection.

Nous allons démontrer l'équation de transport des neutrons dans un volume de contrôle que nous allons définir et nommer C et son contour ∂C . Les particules dans le volume C , se déplacent à une vitesse V_n avec un intervalle de vitesse dV_n et dans une direction $\vec{\Omega}$ avec un intervalle $d^2\Omega$. La quantité de particules dans le volume C évoluera au cours de la période Δt . En utilisant l'équation (2.1) nous pouvons en intégrant sur le volume, obtenir le nombre de particules au temps t initial.

$$\int_C n(\vec{r}, V_n, \vec{\Omega}, t) d^3r dV_n d^2\Omega \quad (2.15)$$

et final, $t + \Delta t$

$$\int_C n(\vec{r}, V_n, \vec{\Omega}, t + \Delta t) d^3r dV_n d^2\Omega \quad (2.16)$$

Nous réalisons la différence des deux expressions précédentes 2.15 et 2.16 afin d'obtenir le taux de variation de particules sur un temps Δt dans le volume C . Nous nommons cette variation : ΔA

$$\Delta A = \int_C (n(\vec{r}, V_n, \vec{\Omega}, t + \Delta t) - n(\vec{r}, V_n, \vec{\Omega}, t)) d^3r dV_n d^2\Omega \quad (2.17)$$

ΔA représente la variation du nombre de neutrons dans le volume C sur un temps Δt . Nous allons maintenant étudier les différentes quantités modifiant le nombre total de neutrons et les exprimer :

- Le nombre de neutrons sortant du volume de contrôle.
- Le nombre de neutrons entrant dans le volume de contrôle.
- Le nombre de neutrons subissant des collision.
- Le nombre de neutrons créée.

Le nombre de neutrons sortant du volume de contrôle durant Δt est obtenu en réalisant l'intégrale sur le contour du courant de particule (2.11).

$$\Delta B = \left[\int_{\partial C} (\vec{\Omega} \cdot \vec{n}) \Phi(\vec{r}, V_n, \vec{\Omega}, t) d^2r dV_n d^2\Omega \right] \Delta t \quad (2.18)$$

$\vec{n}$ est un vecteur unitaire normal à ∂C . On remarque que l'intégrale sur le contour peut être transformée en intégrale sur le volume en utilisant le théorème de la divergence sur (2.18). Nous obtenons alors :

$$\Delta B = \left[\int_C \nabla \cdot \vec{\Omega} \Phi(\vec{r}, V_n, \vec{\Omega}, t) d^3 r dV_n d^2 \Omega \right] \Delta t \quad (2.19)$$

Nous allons maintenant voir le nombre de particules rentrant en collision :

$$\Delta C = \left[\int_C \Sigma(\vec{r}, V_n) \Phi(\vec{r}, V_n, \vec{\Omega}, t) d^3 r dV_n d^2 \Omega \right] \Delta t \quad (2.20)$$

Pour rendre la notation plus compréhensible, nous considérons dans l'équation (2.20) la section efficace macroscopique Σ est indépendante du temps.

Et enfin en contribution positive nous avons le nombre de particules créées :

$$\Delta D = \left[\int_C Q(\vec{r}, V_n, \vec{\Omega}, t) d^3 r dV_n d^2 \Omega \right] \Delta t \quad (2.21)$$

où $Q(\vec{r}, V_n, \vec{\Omega}, t)$ est la densité de la source.

Nous pouvons donc écrire avec les équations : (2.17), (2.19), (2.20) et (2.21), en prenant en compte les contributions positives et négatives, nous obtenons :

$$\Delta A = -\Delta B - \Delta C + \Delta D \quad (2.22)$$

En simplifiant toutes les intégrales sur le volume C et en prenant la limite pour $\Delta t \rightarrow 0$ nous obtenons la forme différentielle de l'équation de transport :

$$\frac{1}{V_n} \frac{\partial}{\partial t} \Phi(\vec{r}, V_n, \vec{\Omega}, t) + \nabla \cdot \vec{\Omega} \Phi(\vec{r}, V_n, \vec{\Omega}, t) + \Sigma(\vec{r}, V_n) \Phi(\vec{r}, V_n, \vec{\Omega}, t) = Q(\vec{r}, V_n, \vec{\Omega}, t) \quad (2.23)$$

En utilisant l'identité : $\nabla \cdot \vec{\Omega} f(\vec{r}) = \vec{\Omega} \cdot \nabla f(\vec{r})$ pour écrire l'équation (2.24) sous la forme la plus fréquente de l'équation linéaire de Boltzmann.

$$\frac{1}{V_n} \frac{\partial}{\partial t} \Phi(\vec{r}, V_n, \vec{\Omega}, t) + \vec{\Omega} \cdot \nabla \Phi(\vec{r}, V_n, \vec{\Omega}, t) + \Sigma(\vec{r}, V_n) \Phi(\vec{r}, V_n, \vec{\Omega}, t) = Q(\vec{r}, V_n, \vec{\Omega}, t) \quad (2.24)$$

En nous plaçant en régime permanent nous obtenons l'équation suivante :

$$\vec{\Omega} \cdot \nabla \Phi(\vec{r}, V_n, \vec{\Omega}) + \Sigma(\vec{r}, V_n) \Phi(\vec{r}, V_n, \vec{\Omega}) = Q(\vec{r}, V_n, \vec{\Omega}) \quad (2.25)$$

Ces différentes équations seront utilisées dans les méthodes de résolution soit directement soit sous une autre forme que nous préciserons.

La démonstration précédente a été tirée du livre Applied Reactor Physic d'Alain Hébert [7].

Nous allons décrire différentes méthodes de résolution de l'équation de transport des neutrons. Des méthodes déterministes ainsi qu'une méthode stochastique seront présentées. Les méthodes stochastiques simulent de manière approximative l'équation exacte, alors que les méthodes déterministes résolvent un problème approximatif de manière exacte [11]. Nous pouvons présenter les différentes méthodes grâce à la figure (2.4) :

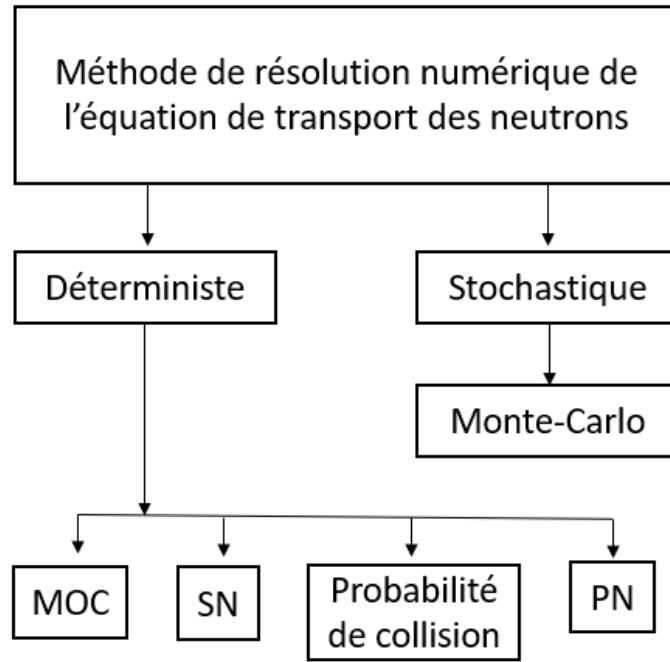


FIGURE 2.4 Représentation des différentes méthodes de résolutions numériques de l'équation de transport des neutrons.

2.3.2 Les conditions frontières

L'équation (2.25) est résolue pour des conditions frontières précises. Elles définissent la géométrie et permettent d'évaluer le flux aux limites de la géométrie. Nous pouvons discerner différentes conditions frontières. Les conditions frontières de vide, de réflexion, de translation et de flux entrant imposé. Nous considérons le flux unidimensionnel en x à un temps t pour une vitesse V_n et en utilisant le cosinus directeur μ sur une géométrie de longueur L_x selon la coordonnée x .

- La condition frontière de vide. Soit condition frontière appliquée sur la surface X d'une géométrie.

$$\Phi(x, \mu) = 0 \text{ if } x > L_x \quad (2.26)$$

- La condition frontière de réflexion. Cette condition applique une symétrie sur le flux.

$$\Phi(x, \mu) = \Phi(x, -\mu) \quad (2.27)$$

- Le flux entrant imposé. Pour un flux imposé en $x = 0$ nous aurons :

$$\Phi(x = 0, \mu) = \alpha \quad (2.28)$$

Avec α un réel décrivant un flux.

- La condition frontière de translation. Cette condition permet de lier deux faces opposées.

$$\Phi(x, \mu) = \Phi(x - L_x, \mu) \quad (2.29)$$

Si cette condition frontière est choisie alors elle est appliquée sur les deux faces, par exemple X^+ et X^-

Ils existent aussi de nombreuses autres conditions que nous ne détaillerons pas ici.

2.4 Méthodes de résolutions déterministes

Il existe de nombreuses méthodes de résolutions déterministes pour l'équation de transport des neutrons. Nous allons voir ici les 4 principales méthodes. Ce sont les méthodes les plus importantes d'un point de vue historique, mais aussi d'un point de vue utilisation aujourd'hui.

- La méthode MOC : méthode des caractéristiques.
- La méthode S_n : méthode aux ordonnées discrètes.
- La méthode des probabilités de collision.
- La méthode P_n : méthode dite des harmoniques sphériques

Nous commencerons par la méthode la plus utilisée à savoir la méthode MOC et nous finirons par la méthode P_n . La majorité des informations sur les différentes méthodes proviennent de [7]. On note que les méthodes utilisent différentes formes de l'équation de transport des neutrons (2.24) que nous détaillerons.

2.4.1 Méthode MOC : méthode des caractéristiques

La méthode des caractéristiques ou méthode MOC, est une méthode déterministe permettant de résoudre l'équation de transport des neutrons sous sa forme caractéristique que nous détaillerons par la suite. C'est l'une des méthodes les plus utilisées dans l'industrie car elle permet une précision suffisante pour un temps de calcul acceptable en 2D [10]. Un avantage certain de la méthode MOC est son adaptabilité et flexibilité sur des géométries complexes [12]. Ceci rend cette méthode très utile pour les nouvelles générations de réacteurs nucléaires émergentes nécessitant une méthode de résolution flexible [13]. La première application de la méthode MOC revient à Askew en 1972 suivi de sa première implémentation en production au module CACTUS de WIMS-E [7]. La flexibilité de cette méthode provient du fait que le maillage adopté n'est pas structuré et permet tout type de géométrie. Nous pouvons noter que la méthode MOC a surtout été développée et utilisée sur des géométries 2D, en effet en 3D les calculs deviennent longs et fastidieux. Bien qu'étant connue depuis de longues années, cette méthode a retrouvé un essor dans les années 90 grâce à l'amélioration de la vitesse de calcul des ordinateurs. De plus, la demande en précision et flexibilité due aux nouveaux projets nucléaires ont incité les chercheurs à trouver des techniques d'accélération permettant d'accélérer cette méthode.

La méthode des caractéristiques nous permet de résoudre des équations aux dérivées partielles. Pour cela nous allons nous placer sur une ligne aussi nommée caractéristique où notre équation se réduit à une équation aux dérivées totales. En résolvant cette seconde équation, nous pouvons résoudre la première. Dans notre contexte cette équation aux dérivées totales est nommée : la forme caractéristique de l'équation de transport des neutrons. Nous résolvons l'équation caractéristique en suivant chaque ligne droite modélisant le parcours d'un neutron dans le réacteur. Ces solutions sont ensuite combinées de façon itérative pour obtenir le flux directionnel sur l'ensemble du domaine.

Nous allons réaliser la démonstration afin d'obtenir l'équation caractéristique de transports des neutrons. Lors du parcours du neutron nous allons supposer qu'à chaque instant la particule est à une distance s de son point de référence R sur sa caractéristique. La position de la particule est donc définie à chaque instant.

Nous allons étudier le neutron se déplaçant sur sa caractéristique avec une direction $\vec{\Omega}$ à une

vitesse V_n . Nous allons définir à un temps t son point de référence $\vec{r}$. Le neutron se déplace d'une distance s sur la caractéristique. Donc à un temps $t + \frac{s}{V_n}$ le neutron sera à une position $\vec{r} + s\vec{\Omega}$. De plus nous allons adopter une transformation mathématique afin de réécrire l'opérateur $\vec{\Omega} \cdot \nabla \Phi$. Pour exprimer l'opérateur nabla ainsi que la dérivée temporelle partielle, nous utilisons la règle de la dérivée en chaîne ou Théorème de dérivation des fonctions composées :

$$\frac{d}{ds} = \frac{dx}{ds} \frac{\partial}{\partial x} + \frac{dy}{ds} \frac{\partial}{\partial y} + \frac{dz}{ds} \frac{\partial}{\partial z} + \frac{dt}{ds} \frac{\partial}{\partial t} \quad (2.30)$$

L'opérateur de base comporte un produit scalaire que nous appliquons entre le vecteur direction et le vecteur unitaire afin d'obtenir l'équation suivante (2.31)

$$\frac{d}{ds} = \vec{\Omega} \cdot \nabla \phi + \frac{1}{V_n} \frac{\partial}{\partial t} \quad (2.31)$$

Notre point de référence est placé avant le neutron en modifiant l'équation : (2.24) nous obtenons la forme caractéristique arrière.

$$\frac{d}{ds} \phi(\vec{r} + s\vec{\Omega}, V_n, \vec{\Omega}, t + \frac{s}{V_n}) + \Sigma(\vec{r} + s\vec{\Omega}, V_n) \phi(\vec{r} + s\vec{\Omega}, V_n, \vec{\Omega}, t + \frac{s}{V_n}) = Q(\vec{r} + s\vec{\Omega}, V_n, \vec{\Omega}, t + \frac{s}{V_n}) \quad (2.32)$$

Par la suite nous verrons comment l'écriture de la forme (2.24) nous permettra de résoudre l'équation de transport grâce aux lignes d'intégration.

2.4.2 Méthode des probabilités de collision

La méthode des probabilités de collision ou méthode CP est une méthode de résolution de l'équation de Boltzmann sous sa forme intégrale avec une discrétisation spatiale. L'utilisation de cette méthode pour un problème avec I régions mène à la résolution d'un système linéaire impliquant une matrice de dimensions $I * I$ par groupe d'énergie. C'est pour cela que pour cette méthode il est mieux d'éviter que le problème comporte trop de régions. Cette matrice contient les liens entre les flux scalaires par région ce qui peut être interprété comme étant des probabilités de collision [10]. Les probabilités de collision peuvent être définies sur un nombre de régions fini simulant à l'infini le maillage originel. Le principe de cette méthode est d'obtenir le flux à partir des probabilités de collision. C'est-à-dire en partant de l'équation de transport dans sa forme intégrale sur un nombre de domaines fini, on peut obtenir la relation (2.33) :

$$\phi_{i,j} = \sum_i Q_j p_{ij} \quad (2.33)$$

Où nous aurons les probabilités de collisions définies par l'équation (2.34).

$$p_{ij} = \frac{P_{ij}}{\Sigma_j} = \frac{1}{4\pi V_i} \int_{V_i}^{\infty} d^3r \int_{V_j} \frac{e^{-\tau(s)}}{s^2} d^3r \quad (2.34)$$

Avec $p_{ij,g}$ la probabilités de collisions réduites et $P_{ij,g}$ la probabilité pour un neutron provenant de n'importe quelle région V_i du maillage de subir une collision dans n'importe quelle région V_j . Avec $Q_{j,g}$ la source des neutrons dans V_j . l'indice g indique le groupe d'énergie. Les détails de cette démonstration ne seront pas montrés ici, mais se trouvent dans [9].

Cette méthode est composée de trois étapes :

1. Une étude du parcours des neutrons est effectuée, avec un nombre suffisant de régions indépendantes.
2. L'obtention de la matrice de probabilité de collision pour chaque groupe d'énergie.
3. Le calcul du flux intégral grâce à la formule (2.33)

La méthode CP à deux limitations principales. Tout d'abord comme nous l'avons vu, les matrices produites par cette méthode peuvent devenir très grandes. Cela peut engendrer des problèmes pour la vitesse de calcul. De plus, cette méthode est limitée à des sources isotopiques. Ceci implique soit de ne travailler que dans des cas de sources isotropiques ou alors accepter de faire cette hypothèse en plus.

2.4.3 Méthode S_n

La méthode S_n ou méthode des caractéristiques courtes où l'on va suivre les neutrons cellule par cellule. C'est une méthode de résolution pour l'équation de Boltzmann sous sa forme différentielle. Le n dans S_n représente le nombre d'angles utilisé. Cette méthode provient d'une méthode en astrophysique et a été utilisée pour la première fois par Carlson en 1955 dans un contexte de physique des réacteurs [14]. L'idée est de réaliser une discrétisation explicite et de l'évaluer à différents angles. Dans cette méthode, nous verrons l'émergence d'une quadrature angulaire et d'un cosinus directeur. De façon générale, elle utilise un schéma aux différences finies de type diamant. Comme nous le retrouvons dans l'écrit de Carlson afin d'utiliser la méthode S_n , en 1D on définit le cosinus directeur comme étant $\mu = \cos(\theta)$ avec θ l'angle du flux neutronique et nous aurons n intervalle (μ_{j-1}, μ_j) avec $j = 1, \dots, n$ ainsi que $\mu_0 = -1$ et $\mu_n = 1$ [15]. Cette définition nous permet d'obtenir le flux neutronique à un angle choisi, c'est l'objectif d'une partie des méthodes afin de résoudre l'équation de transport des neutrons.

Pour illustrer la méthode, nous appliquerons l'équation de transport des neutrons (2.25) dans

son état stationnaire sur une ligne, donc en seule dimension. L'équation deviendra :

$$\mu \frac{d\phi(x, \mu)}{dx} + \Sigma(x, \mu)\phi(x, \mu) = Q(x, \mu) \quad (2.35)$$

En plus de la discrétisation angulaire, nous effectuons une discrétisation sur la ligne, comme illustré dans la figure (2.5)

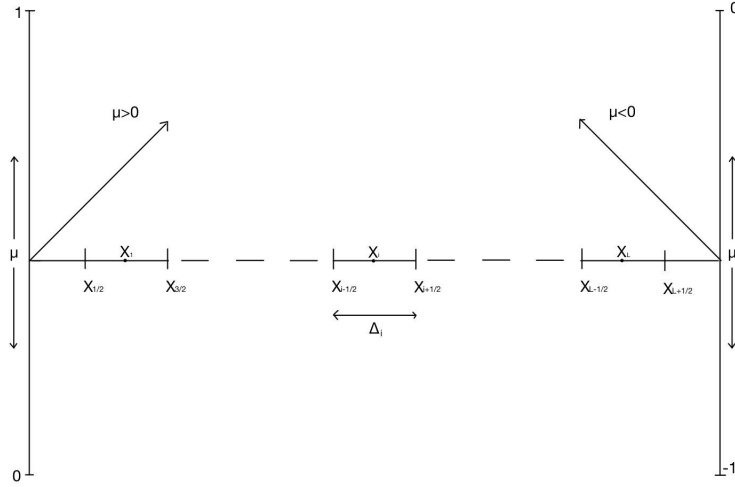


FIGURE 2.5 Représentation d'une discrétisation spatial en une dimension.

Cette illustration nous servira de base afin de comprendre les discrétisations en 2D puis en 3D.

En discrétisant l'espace, nous pouvons exprimer les différentes grandeurs grâce notamment au polynôme de Legendre, et la source $Q(x, \mu)$, nous ne la détaillerons pas ici, mais tous les détails des calculs sont issus de [14].

Afin de résoudre l'équation (2.35) nous avons besoin d'une forme numérique pour les grandeurs angulaires. Nous définissons N directions angulaires ayant N cosinus directeurs μ_n et leur poids correspondant ω_n . Soit une fonction $f(\mu)$ dépendant du cosinus directeur de l'angle nous pouvons écrire :

$$\int_{-1}^1 f(\mu) d\mu = \sum_{n=1}^N f(\mu_n) \omega_n \quad (2.36)$$

Avec :

$$\sum_{n=1}^N \omega_n = 2 \quad (2.37)$$

La somme des poids (2.37) est égale à deux dans ce cas, car nous utilisons les cosinus directeurs positifs et négatifs donc nous avons deux fois plus d'angles $\theta \in [0, \pi]$.

En appliquant ces relations à l'équation (2.35) nous aurons :

$$\mu_n(\phi_{i+\frac{1}{2},n} - \phi_{i-\frac{1}{2},n}) + \Delta_i \Sigma_{t,i} \phi_{i,n} = \Delta_i Q_{i,n} \quad (2.38)$$

En connaissant les valeurs aux extrémités ; c'est-à-dire les flux entrants dans la géométrie, on peut résoudre l'équation (2.38) et ainsi obtenir le flux sur toute la géométrie.

Cette méthode peut être établie pour un nombre d'angles N et pour un nombre de noeuds L sur le maillage spatial.

2.4.4 Méthode P_n

La méthode P_n ou méthode dite des harmoniques sphériques, est une méthode où on développe le flux angulaire en harmoniques sphériques ainsi que le terme source. C'est la méthode la plus ancienne pour résoudre les équations de transport. [9] Le flux angulaire est représenté comme un développement limité en harmoniques sphériques réelles. Les harmoniques sphériques sont un ensemble de fonctions permettant la représentation d'une fonction sur une sphère. Les harmoniques sphériques peuvent être définies comme les valeurs propres de la partie angulaire d'un laplacien en 3D. Nous utilisons la méthode des harmoniques sphériques afin de résoudre l'équation de transport des neutrons [16].

Si nous réécrivons l'équation de transport stationnaire à une vitesse

$$\vec{\Omega} \cdot \vec{\nabla} \phi(\vec{r}, \vec{\Omega}) + \Sigma(\vec{r}) \phi(\vec{r}, \vec{\Omega}) = (\vec{r}, \vec{\Omega}) \quad (2.39)$$

Nous pouvons écrire pour le terme de flux ϕ et le terme source Q

$$\begin{aligned} \phi(\vec{r}, \vec{\Omega}) &= \sum_{m=0}^{\infty} \sum_{l=-m}^m \phi_m^l(\vec{r}) Y_m^l(\vec{\Omega}) \\ Q(\vec{r}, \vec{\Omega}) &= \sum_{m=0}^{\infty} \sum_{l=-m}^m Q_m^l(\vec{r}) Y_m^l(\vec{\Omega}) \end{aligned} \quad (2.40)$$

Nous avons dans l'expression (2.40) $Y_m^l(\vec{\Omega})$ l'expansion angulaire en harmonique sphérique.

Pour l'utilisation en trois dimensions, le nombre d'harmonique devient important, car il croît en n^2 avec n le nombre d'angles [10]. C'est donc une méthode compliquée à implémenter en trois dimensions due aux équations et au temps de calcul. Cependant, il existe des simplifi-

cations de cette méthode comme la méthode SP_n produite par E.M.Gelbard [17].

2.5 Méthode de résolution stochastique

En opposition aux méthodes de résolution déterministes, il existe les méthodes de résolution stochastiques. Ces méthodes se basent sur générateurs de nombres aléatoires afin de simuler le comportement des neutrons dans le réacteur. Ce sont les méthodes les plus précises afin de résoudre l'équation de transport des neutrons et elles nous permettent ainsi de valider nos méthodes déterministes. Nous allons présenter la méthode de Monte-Carlo, qui est la seule méthode que nous étudierons. On note cependant que ce n'est pas la seule méthode non déterministe qui existe.

2.5.1 Méthode de Monte-Carlo

Dans les méthodes stochastiques, nous allons effectuer une simulation directe de la population neutronique. On dit de la méthode Monte-Carlo qu'elle est exacte si la simulation est suffisamment précise et que le nombre de particules est convenable. La méthode de Monte-Carlo permet de trouver des résultats dans des configurations complexes de manière relativement simple. Le but est de calculer pour chaque particule un comportement moyen en simulant grâce à l'aléatoire tous les événements physiques susceptibles de se dérouler et en respectant toutes les valeurs propres aux neutrons comme les sections efficaces ainsi que les différentes lois physiques régissant leur déplacement. On dit stochastique, car on se base sur la simulation de nombre aléatoire sur l'intervalle $[0, 1]$. C'est la méthode la plus précise pour simuler le transport des neutrons dans la matière, mais c'est aussi la plus chère. En effet, les méthodes de Monte-Carlo sont les méthodes avec le temps de calcul le plus long parmi toutes les méthodes citées précédemment. Même si de nombreux moyens existent pour accélérer ces méthodes, leur temps élevé d'exécution reste problématique [10].

En pratique dans la méthode de Monte-Carlo, nous prendrons une particule positionnée en $\vec{r}$ se déplaçant à une vitesse V sur une trajectoire décrite par $\vec{r} + Vt\vec{\Omega}$ avec t représentant le temps et $t \geq 0$ jusqu'à ce qu'un des événements suivants se produise.

- La particule rencontre les limites de la géométrie et les conditions frontières sont appliquées.
- Lorsqu'un neutron s'approche d'un atome et change de vitesse on peut noter un temps aléatoire T_s lorsque cet événement se produit on peut définir une probabilité. De plus, on obtiendra une nouvelle vitesse V' obtenue de manière aléatoire aussi.
- Un événement de fission se produit. Lorsqu'un neutron rencontre un noyau d'atome.

Nous écrivons le temps T_f le moment choisi aléatoirement de l'évènement de fission. On peut alors définir une probabilité pour l'évènement. Lors de cet évènement de fission une quantité non nulle de neutrons est produite par la réaction et éjectée à une vitesse $\vec{v}$ elle aussi obtenue de manière aléatoire.

Les nombres aléatoires pour les méthodes stochastiques sont trouvés à l'aide de distributions [18].

2.6 Conclusion

Le but de cette présentation est de comprendre comment résoudre de manière mathématique les équations de transport des neutrons. Les méthodes que nous avons proposées sont les méthodes essentielles à une compréhension globale du problème. On note qu'il existe de nombreuses autres méthodes afin de résoudre nos équations, certaines méthodes étant une évolution d'une méthode citée précédemment ou encore des associations entre les différentes méthodes.

Nous allons voir dans la prochaine partie comment en pratique nous pouvons résoudre les équations posées précédemment à l'aide d'un outil : les lignes d'intégration.

CHAPITRE 3 LES LIGNES D'INTÉGRATION

3.1 Introduction

Nous allons définir un outil essentiel pour la résolution de l'équation de transport des neutrons : les lignes d'intégration. C'est aussi un point essentiel de ma maîtrise. Nous verrons en particulier le tracé des lignes d'intégration cycliques, l'importance des différents tracés et leurs influences dans la résolution de l'équation de transport des neutrons.

3.2 Définitions

La ligne d'intégration ou ligne caractéristique dans la méthode MOC représente d'un point de vue physique le parcours des neutrons dans un réacteur nucléaire. D'un point de vue mathématique, c'est un outil pour résoudre numériquement l'équation de transport des neutrons en fonction de la méthode de résolution choisie. Nous verrons l'utilisation des lignes d'intégration dans les différentes équations ainsi que leur génération sur la géométrie à savoir le tracking. Globalement, les lignes d'intégration sont utilisées pour choisir les trajectoires des neutrons (direction et point de départ) requises pour définir les lignes caractéristiques de la méthode MOC (2.32).

Le tracking représente la méthode utilisée afin de tracer nos lignes d'intégration. Il existe principalement deux méthodes de tracé : la méthode uniforme et la méthode cyclique.

3.2.1 Le tracking uniforme ou isotrope

C'est le tracé le plus simple. Nous commençons sur le bord d'une cellule et nous avançons par pas dans la direction orthogonale à la trajectoire choisie afin de parcourir la géométrie. Les lignes sont tracées de manières indépendantes entre elles. Les conditions frontières seront appliquées par la suite de manière approximative.

Nous pouvons voir sur la figure (3.1) les lignes d'intégration tracées selon une procédure finie. À noter que cette image est issue du livre de M.Hébert : [7]. Nous pouvons voir un pas constant entre les différentes lignes ainsi que plusieurs directions. Il est aussi possible de trouver un pas non constant avec cette méthode. Cela permet de mettre l'accent sur des régions plus petites, et ainsi avoir une répartition adaptée des lignes par régions. Nous notons aussi que la représentation (3.1) à un tracé des lignes particulier. En effet nous pouvons discerner 3 direction selon différents octants. Cependant, c'est un tracé qui n'est pas reproductible avec

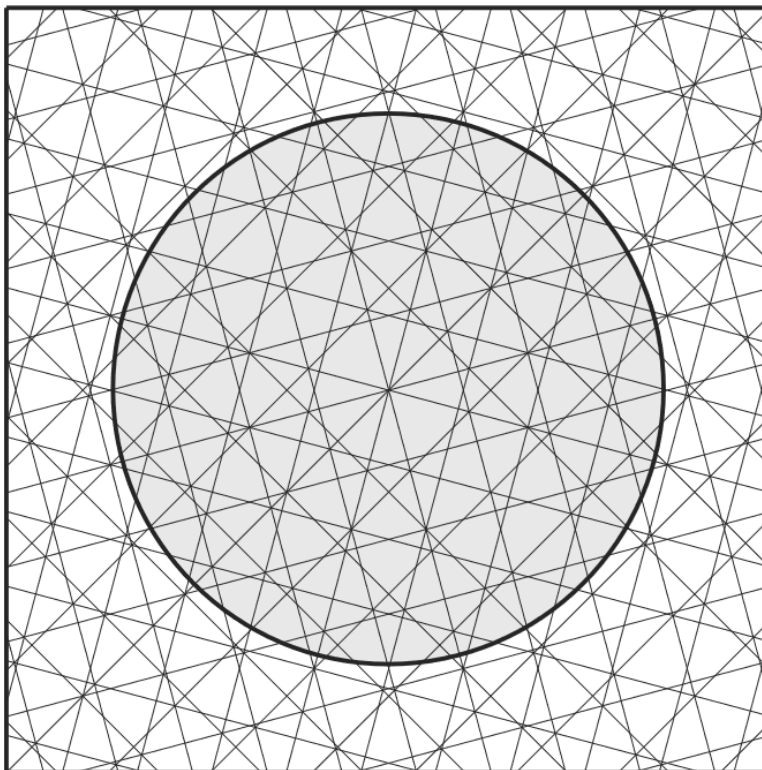


FIGURE 3.1 Représentation des lignes d'intégration uniformes sur une cellule carrée.

un procédé cyclique, car la fin des lignes ne correspond pas au début d'une autre ligne, peu importe les conditions frontières (réflexion ou translation). Nous ne pouvons pas distinguer de cycles.

3.2.2 Le tracking cyclique ou spéculaire

Le tracé cyclique ou spéculaire. Dans ce tracé plus complexe, nous allons tracer une ligne et à chaque intersection avec la surface externe de la géométrie nous allons appliquer les conditions frontières. Chaque ligne possédera finalement plusieurs segments et potentiellement plusieurs directions. De plus, comme son nom l'indique le point de départ est identique au point final. Dans ce tracé nous procéderons cycle par cycle.

Il est important de noter que si nous choisissons d'imposer les mêmes conditions de tracé aux deux méthodes c'est à dire les mêmes angles et le même espacement de lignes, alors le motif décrit par les lignes d'intégration dans une procédure finie et une procédure cyclique est identique. De plus, il est important de noter que dans un tracking cyclique les lignes passant par un coin seront évitées.

Sur la figure (3.2) nous pouvons voir les différents cycles tracés. Nous pouvons de plus re-

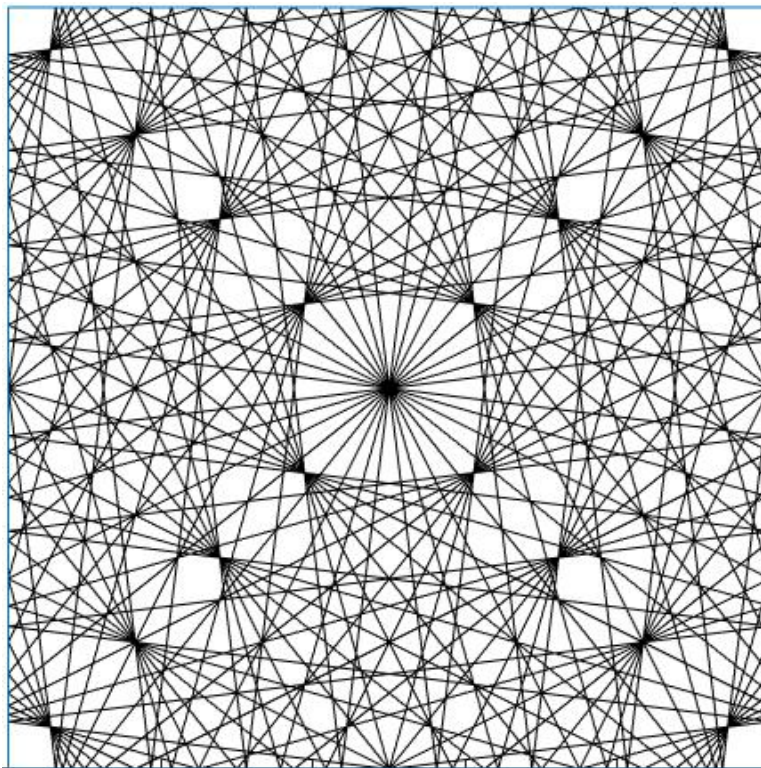


FIGURE 3.2 Représentation d'une procédure de tracking avec 8 angles sur une cellules carrée.

marquer : des zones blanches. C'est-à-dire des zones possédant une densité de lignes globale plus faible, on notera que par angle la densité de lignes est uniforme.

Contrairement aux lignes uniformes, le pas est ici constant par direction afin de conserver des lignes d'intégration cycliques.

3.3 Calcul avec les lignes d'intégration

3.3.1 Outil pour le calcul numérique

Comme nous l'avons vu précédemment, les lignes d'intégration nous permettent de résoudre des équations différentielles numériquement. Il est impossible de demander directement à un compilateur comme Fortran de résoudre une équation différentielle. En effet, nous devons lui donner une forme équivalente qu'il pourra interpréter.

Nous allons voir ici les différents outils qui nous permettent grâce aux lignes d'intégration de calculer les différentes quantités nécessaires à la résolution de notre équation de transport. Pour ce faire, nous allons effectuer le balayage d'une cellule en 2D afin de connaître les données importantes que nous devons conserver pour réaliser nos intégrations.

Nous allons à présent utiliser une cellule comportant des sous-régions. Nous utilisons pour notre démonstration des sous-régions rectangulaires ou carrées. Cependant, les sous-régions dans les réacteurs nucléaires peuvent avoir des formes variées.

Lorsque nous utilisons un procédé de tracking sur une cellule, nous balayons toute la cellule grâce aux lignes d'intégration. Il est important de retenir les points d'intersection entre les surfaces des zones extérieures et les lignes. De plus, il est aussi important de retenir les points d'intersection entre les lignes et les surfaces appartenant aux sous-régions. En général, les cellules de réacteurs seront présentées par région circulaire entourée de plusieurs régions annulaires, représentant respectivement le combustible puis la gaine [4].

Afin d'illustrer notre explication, nous utiliserons des lignes d'intégration uniformes, par souci de clarté dans nos illustrations. La procédure de tracking permet de tracer l'ensemble des lignes d'intégration $\vec{T}$ que nous nommerons Δ . Nous définissons un domaine subdivisé en région, chaque région est délimitée par un contour. Sur la figure (3.3), chaque ligne est définie par un point de départ et une direction nommée $\vec{\Omega}$, les lignes sont incrémentées avec un pas régulier selon $\vec{\Omega}_\perp$. Et les lignes sont découpées en segments selon les différentes régions.

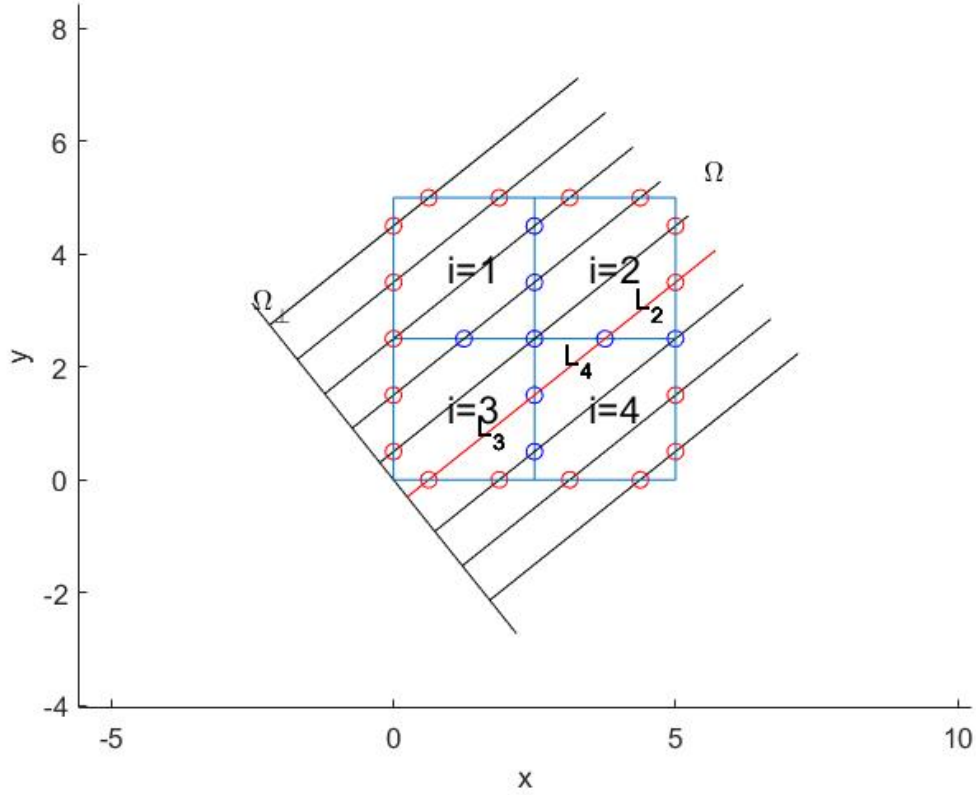


FIGURE 3.3 Représentation des lignes d'intégration dans une cellule comportant différentes régions.

Notre objectif est l'intégration numérique d'une équation différentielle partielle. Chaque région traversée est numérotée i_k , k est l'indice de la région traversée. Pour chaque ligne, il faut connaître la longueur du segment dans la région : L_i .

Il faut réaliser un lien entre les segments des lignes d'intégration et l'intégration sur le volume afin de résoudre les équations. Pour cela, il faut effectuer un changement de variable d'un volume V_i à un ensemble de lignes d'intégration Δ . Afin d'illustrer ce changement de variable, nous utilisons une fonction quelconque f définie sur un volume V_i et nous cherchons son intégrale sur le volume F_i :

$$\begin{aligned}
 F_i &= \int_{V_i} d^3r \int_{4\pi} f(\vec{r}, \Omega) d^2\Omega \\
 F_i &= \frac{1}{4\pi} \int_{4\pi} d^2\Omega \int_{\Pi_\Omega} d^2p ds \int_{-\infty}^{\infty} \xi_{V_i}(s, \vec{T}) f(\vec{T}) \\
 F_i &\approx \frac{1}{4\pi} \int_{4\pi} d^2\Omega \int_{\Pi_\Omega} d^2p \sum_{k=1}^K \delta_{i,k} f(i) L_k
 \end{aligned} \tag{3.1}$$

K représente le nombre total de segments d'une ligne. ξ_{V_i} représente une fonction caractéristique prenant en compte uniquement les segments de ligne appartenant à la bonne région. La dernière forme comporte le delta de Kronecker, il possède le même but que la fonction ξ ; c'est une première approximation qui nous permettra d'écrire une forme numérique des grandeurs nécessaires à la résolution de l'équation de transport. Cette équation sera utilisée à de nombreuses reprises dans la partie suivante.

La formule (3.1) nous permettra de calculer toutes les quantités nécessaires à résoudre les différentes équations, elle est extraite d'une thèse : [19], réalisée à Polytechnique Montréal par Mudimbi Musongela en 2019, nous utiliserons certaines parties de cette dernière dans la suite.

3.3.2 Équivalence des grandeurs

Nous allons chercher à effectuer l'équivalence des différentes grandeurs. Ce qui est nécessaire pour résoudre nos équations. Dans un premier temps, nous allons appliquer la formule (3.1) afin d'utiliser la notion de lignes d'intégration. Dans un second temps, nous appliquerons une quadrature angulaire et une densité de trajectoire afin d'exprimer nos expressions à l'aide de sommes pour les résoudre numériquement.

Afin de réaliser les équivalences, il nous faut définir les différents points d'intersection entre les lignes, les surfaces du domaine et les sous-régions ainsi que le flux angulaire.

$$\begin{aligned}\vec{r}_k &= \vec{r}_{k-1} + L_k \vec{\Omega} \\ \Phi_k(\vec{T}) &= \Phi(\vec{r}_k, \Omega)\end{aligned}\tag{3.2}$$

Avec $k \in [1, K]$ où K est le nombre total de segments.

Nous pouvons de cette manière calculer un flux moyen par segment.

$$\bar{\phi}_k(\vec{T}) = \frac{1}{L_k(\vec{T})} \int_0^{L_k(\vec{T})} \phi(\vec{r}_{k-1} + s\vec{\Omega}, \vec{\Omega}) ds\tag{3.3}$$

Grâce à l'équation (3.3) et l'application du changement de variable, que nous avons explicité (3.1) sur l'équation (2.7) nous obtenons :

$$V_i \phi_i \approx \sum_{k=1}^K \Delta_{i,k} L_k(\vec{T}) \bar{\phi}_k(\vec{T})\tag{3.4}$$

Et le volume numérique devient :

$$V_i \approx \sum_{k=1}^K \Delta_{i,k} L_k(\vec{T}) \quad (3.5)$$

Les formules (3.4) et (3.5) sont des formules pseudonumériques et que pour les résoudre avec un code de calcul nous devons établir des formules numériques. L'intégrale sur les directions deviendra une discrétisation angulaire avec un poids : ω_Ω . Nous aurons de même pour la densité surfacique avec son poids : $\omega_\perp$.

$$\int_{4\pi} d^2\Omega \approx \sum_m^{N_\Omega} \omega_{\Omega_m} \vec{\Omega}_m = 4\pi \quad (3.6)$$

$$\int_\Delta d^2p \approx \sum_p^{N_\perp} \omega_{\perp_p} = 1 \quad (3.7)$$

Avec ces nouvelles notations, nous pouvons exprimer toutes les grandeurs nécessaires sous une forme numérique.

Le flux scalaire moyen par région avec les volumes :

$$V_i \phi_i = \sum_p^{N_\perp} \sum_m^{N_\Omega} \sum_k^K \omega_{\perp_p} \omega_{\Omega_m} \delta_{i,k} L_k \bar{\phi}_{k,p,m} \quad (3.8)$$

$$V_i = \frac{1}{4\pi} \sum_p^{N_\perp} \sum_m^{N_\Omega} \sum_k^K \omega_{\perp_p} \omega_{\Omega_m} \delta_{i,k} L_k \quad (3.9)$$

Nous allons aussi définir directement les courants de surface ainsi que leur surface en utilisant le procédé précédent et la formule (2.12).

Pour le courant entrant à une surface α et sa surface :

$$\begin{aligned} S_\alpha \phi_\alpha^- &= -4 \sum_p^{N_\perp} \sum_m^{N_\Omega} \omega_{\perp_p} \omega_{\Omega_m} \xi_\alpha(\vec{r}_0, \vec{T}) \phi_{0,p,m} \\ S_\alpha &= \frac{1}{\pi} \sum_p^{N_\perp} \sum_m^{N_\Omega} \omega_{\perp_p} \omega_{\Omega_m} \xi_\alpha(\vec{r}_0, \vec{T}) \end{aligned} \quad (3.10)$$

Et le courant sortant à une surface β et sa surface :

$$\begin{aligned} S_\beta \phi_\beta^+ &= 4 \sum_p^{N_\perp} \sum_m^{N_\Omega} \omega_{\perp_p} \omega_{\Omega_m} \xi_\alpha(\vec{r}_K, \vec{T}) \phi_{K,p,m} \\ S_\beta &= \frac{1}{\pi} \sum_p^{N_\perp} \sum_m^{N_\Omega} \omega_{\perp_p} \omega_{\Omega_m} \xi_\alpha(\vec{r}_K, \vec{T}) \end{aligned} \quad (3.11)$$

Pour les probabilités de collisions p_{ij} .

Nous allons utiliser le parcours optique définie dans l'équation (2.13) sous une forme numérique.

$$\tau_{i,j}^{m,n} = \sum_{l=1}^n \Sigma_k L^{m,l} \quad (3.12)$$

En utilisant l'équation (2.34) nous pouvons écrire une formule numérique d'après [19] et [20].

$$p_{ij} = \frac{1}{\Sigma_j \Sigma_i V_i} \sum_p^{N_\perp} \sum_m^{N_\Omega} \omega_{\perp p} \omega_{\Omega_m} \sum_{n \in j} \left(K_{i,j}(\tau_{i,j}^{m,n}) - K_{i,j}(\tau_{i,j}^{m,n-1}) - K_{i,j}(\tau_{i,j}^{m+1,n}) + K_{i,j}(\tau_{i,j}^{m+1,n-1}) \right) \quad (3.13)$$

Nous avons utilisé la fonction de Bickley-Naylor [7] définie dans l'équation (3.14).

$$K_{i,n}(x) = \int_0^1 \exp\left(\frac{-x}{\sqrt{1-u^2}}\right) \sqrt{1-u^{2n-2}} du \quad (3.14)$$

3.4 Résolution de l'équation de transport

Nous présenterons ici la résolution de l'équation de transport grâce aux lignes d'intégration. Cette partie nous permet de mettre en évidence l'importance des lignes d'intégration et leur utilisation mathématique dans la résolution de l'équation de transport. Nous avons choisi d'étudier leur implication uniquement au travers de la méthode MOC. Nous avons également choisi de mettre en évidence la résolution selon cette méthode, car la méthode MOC consiste déjà à résoudre l'équation de transport des neutrons selon un segment traversant la géométrie.

3.4.1 Méthode MOC : méthode des caractéristiques

Afin de présenter cette méthode de résolution, nous avons utilisé [19] comme référence.

Nous allons mettre en évidence l'utilisation des lignes d'intégration pour résoudre : la méthode des caractéristiques. Sachant que chaque ligne possède un ensemble unique de points et de directions. Nous écrivons l'équation de transport sur une ligne caractéristique : $\vec{T}_{pm}$ définie par un point de départ p et une direction $\vec{\Omega}$. En utilisant l'équation de transport sous sa forme MOC (2.32), nous la réécrivons sur un segment de ligne T_k :

$$\frac{d}{ds} \phi_{pm}(\vec{T}_k(s)) + \Sigma_{pm}(\vec{T}_k(s)) \phi_{pm}(\vec{T}_k(s)) = Q_{pm}(\vec{T}_k(s)) \quad (3.15)$$

Afin de résoudre l'équation, nous devons appliquer certaines approximations afin de simplifier le problème.

1. Les sections efficaces seront considérées comme constantes sur un segment de ligne appartenant à une région i :

$$\Sigma_{pm}(\vec{T}_k(s)) = \Sigma_{ik} \quad (3.16)$$

2. Nous appliquons le même raisonnement sur les sources, qui seront constantes par région :

$$Q_{pm}(\vec{T}_k(s)) = Q_{ik} \quad (3.17)$$

En utilisant les différentes hypothèses, nous pouvons écrire l'équation (3.15) sous la forme :

$$\frac{d}{ds}\phi_{pm}(\vec{T}_k(s)) + \Sigma_{ik}\phi_{pm}(\vec{T}_k(s)) = Q_{ik} \quad (3.18)$$

En utilisant la formule (3.3) nous pouvons écrire :

$$\bar{\phi}_{pm}^k = \frac{1}{L_K} \int_{s_{k-1}}^{s_k} ds \phi_{pm}(\vec{T}_k(s)) \quad (3.19)$$

Afin de résoudre notre calcul, nous utiliserons un outil de calcul à savoir e^{τ_k} qui représente la fonction exponentielle appliquée au chemin optique, d'après l'équation (2.13) et les symboles de ce chapitre nous pouvons écrire :

$$\tau_k = \int_{s_{k-1}}^{s_k} ds \Sigma_{ik} = \Sigma_{ik} L_{ik} \quad (3.20)$$

Nous obtenons en multipliant l'équation (3.18) par le parcours optique. De plus, nous utilisons l'identité suivante : $\frac{d(uv)}{ds} = u \frac{dv}{ds} + v \frac{du}{ds}$ sur la partie gauche de notre équation. Nous obtenons :

$$\left(\frac{d}{ds} \phi_{pm}(\vec{T}_k(s)) + \Sigma_{ik} \phi_{pm}(\vec{T}_k(s)) \right) e^{\tau_k} = \frac{d}{ds} (\phi_{pm}(\vec{T}_k(s)) e^{\tau_k}) \quad (3.21)$$

Nous utilisons l'identité précédente (3.21) sur notre équation (3.18) afin d'obtenir une équation plus simple que nous intégrons sur le segment :

$$\begin{aligned}
\frac{d}{ds}(\phi_{pm}(\vec{T}_k(s))e^{\tau_k}) &= Q_{ik}e^{\tau_k} \\
\int_{s_{k-1}}^{s_k} d(\phi_{pm}(\vec{T}_k(s))e^{\tau_k}) &= \int_{s_{k-1}}^{s_k} Q_{ik}e^{\tau_k} \\
\phi_{pm}^k e^{\tau_k} - \phi_{pm}^{k-1} &= \frac{Q_{ik}}{\Sigma_{ik}}(e^{\tau_k} - 1)
\end{aligned} \tag{3.22}$$

En organisant notre équation (3.22), nous obtenons un schéma d'incrémentation :

$$\phi_{pm}^k = \phi_{pm}^{k-1}e^{-\tau_k} + \frac{Q_{ik}}{\Sigma_{ik}}(1 - e^{-\tau_k}) \tag{3.23}$$

Le schéma d'incrémentation (3.23) nous permet d'obtenir de manière itérative le flux sortant. Pour obtenir le flux moyenné, nous devons réaliser l'intégrale de l'équation (3.15) sur un segment s , compris entre $s_{k-1} < s < s_k$:

$$\int_{s_{k-1}}^{s_k} d\phi_{pm}(\vec{T}_k(s)) + \int_{s_{k-1}}^{s_k} ds \Sigma_{ki} \phi_{pm}(T(s)) = \int_{s_{k-1}}^{s_k} ds Q_{ki} \tag{3.24}$$

$$\phi_{pm}^k - \phi_{pm}^{k-1} + \Sigma_{ki} \bar{\phi}_{pm}^k(T(s)) = L_k Q_{ki} \tag{3.25}$$

En remplaçant l'équation (3.25) dans le schéma itératif (3.23) nous obtenons le flux moyen pour chaque segment, mais de manière itérative :

$$\bar{\phi}_{pm}^k = \phi_{pm}^{k-1} \left(\frac{1 - e^{-\tau_k}}{L_{kik}} \right) + \frac{Q_{ik}}{\Sigma_{ik}} \left(1 - \frac{1 - e^{-\tau_k}}{L_{kik}} \right) \tag{3.26}$$

Les équations itératives (3.26) et (3.23) sont les moyens d'obtenir selon la méthode MOC les flux itératifs sortants et moyens. En fixant les conditions initiales ; c'est-à-dire les sources de départ, nous pouvons balayer le domaine dans toutes les directions. Ainsi, nous procédons de manière itérative en obtenant les différents flux. Cela nous permet de mettre à jour les sources et nous recommençons le processus avec ces nouvelles valeurs de sources afin d'avoir la convergence de notre flux.

3.4.2 Conditions frontières

Nous travaillons dans un système fermé représentant une cellule nucléaire ou une partie d'assemblage, il est important de définir des conditions frontières afin d'avoir une continuité du flux. Ces conditions permettent de définir le comportement des neutrons sur l'interface

de notre géométrie, qui sera défini sur un domaine $\mathcal{D}$ et sa frontière $\partial\mathcal{D}$. Nous allons nous intéresser aux conditions frontières de réflexion et de translation.

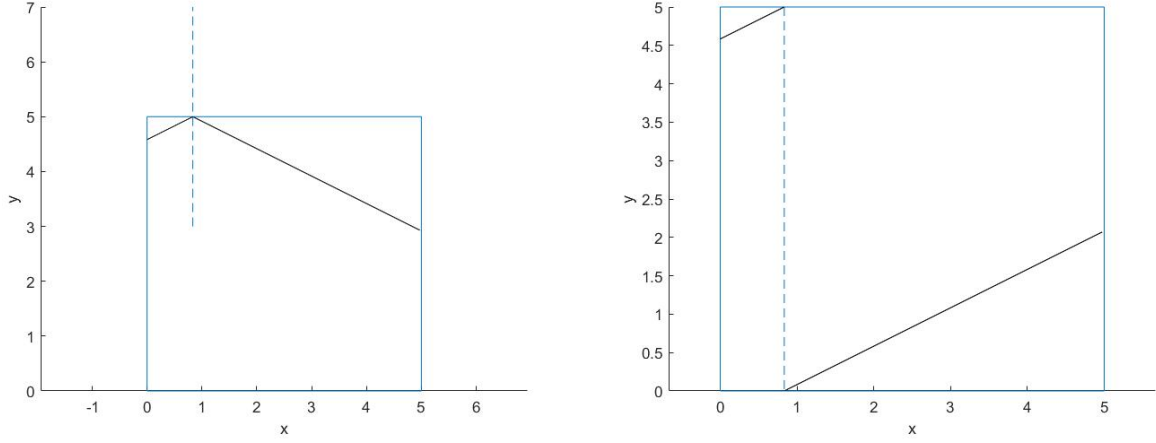


FIGURE 3.4 Mise en évidence des conditions frontières de réflexion (droite) et de translation (gauche).

Réflexion : La condition frontière de réflexion a le même effet qu'un miroir : aux mêmes points elle applique l'angle symétrique et conserve le flux.

$$\phi(\vec{r}, \vec{\Omega}) = \phi(\vec{r}, \vec{\Omega}') \quad (3.27)$$

Translation : la condition frontière de translation va décaler le point d'entrée de la longueur de la géométrie et conserve le même flux.

$$\phi(\vec{r}, \vec{\Omega}) = \phi(\vec{r} + \vec{L}, \vec{\Omega}) \quad (3.28)$$

Il existe de nombreuses façons d'appliquer les conditions frontières. Sur un tracé cyclique, les conditions frontières sont dites exactes, comparées à un tracé uniforme. On peut noter que les considérer comme exactes est un abus de langage, car elles sont fonction de la quadrature angulaire, c'est donc une approximation. Cependant, pour un tracé cyclique les conditions frontières sont plus proches de la réalité que sur un tracé uniforme. Il existe de nombreuses autres conditions frontières que celles énoncées précédemment, et aussi différentes méthodes de tracking afin de les appliquer [21].

CHAPITRE 4 IMPLÉMENTATIONS DU TRACÉ DES LIGNES CYCLIQUE EN 3D

4.1 Présentation

Dans ce chapitre, nous allons présenter la méthode que nous avons utilisée afin de tracer des lignes d'intégration cycliques en 3D. Afin de simplifier notre travail, nous allons tout d'abord décrire la méthode pour tracer les lignes d'intégration en 2D, qui est déjà programmée dans DRAGON. La méthode 3D consiste en une succession de deux méthodes 2D, il est donc important de commencer par les décrire.

4.2 Autre méthode de tracé

D'autres méthodes existent dans la littérature pour effectuer le tracé des lignes d'intégration. L'utilisation de ces méthodes est assez répandue dans la résolution de l'équation de transport des neutrons, surtout lors de l'application de la méthode MOC. Il existe aussi de nombreuses façons de faire varier la méthode que nous appliquons, comme des modifications pour les point de départ. Le solveur CACTUS est en un exemple avec sa méthode cyclique en 2D [22].

4.2.1 MRT Modular Ray Tracing

La méthode MRT ou méthode de tracé modulaire repose sur le principe que toute géométrie peut être décomposée en série de pavés rectangulaires de mêmes dimensions. Cela permet de réaliser un unique tracking sur une seule des sous-géométries et par la suite de l'appliquer sur toute la géométrie. Cette méthode est décrite plus en détail dans les références [12] et [23]. La méthode MRT se résume à deux étapes :

1. Découpage de la géométrie.
2. Tracé des lignes sur une sous géométrie.

Pour le découpage de la géométrie, une quadrature spatiale globale sera utilisée. Pour le tracé des lignes, les différentes références appliquent des méthodes différentes.

4.2.2 3DGT The 3D global meshing method

La méthode 3DGT utilise une procédure proche de la méthode que nous avons utilisée. Cependant, elle trace dans un premier temps un tracking complet en 2D. En utilisant les

informations de tracking 2D, le tracking en 3D est calculé, et les différentes informations telles que les angles et les écartements. Les angles et les écartements sont corrigés à chaque boucle sur les angles, ces corrections ont pour but d'augmenter la précision afin d'obtenir les angles souhaités. Cette méthode est démontrée plus en détail dans la référence [23].

4.3 Tracking en 2D

Pour cette démonstration, nous allons utiliser une cellule carrée (méthode très facile à adopter pour une cellule rectangulaire) comme suit :

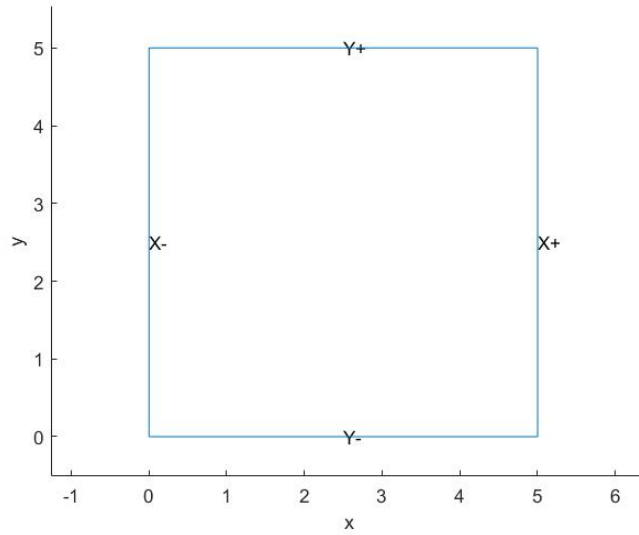


FIGURE 4.1 Représentation de la cellule de travail et notations des surfaces en 2D.

Cette figure nous sera utile pour la suite du travail. Elle nous sert de point de départ afin d'explicitier notre méthode. C'est aussi la représentation la plus simple d'un réacteur que nous pouvons avoir comme expliqué dans [4]. En effet, les cellules de certains réacteurs peuvent être schématisées par un carré avec à l'intérieur un crayon de combustible circulaire. En choisissant ici cette représentation, c'est la façon la plus claire de vérifier dans un premier temps la validité de notre procédé.

4.3.1 Choix des angles

Afin de choisir nos angles, nous appliquerons une quadrature angulaire. Pour cela, nous allons commencer par définir notre direction principale : $\hat{u}_\alpha$. Elle représente physiquement la

direction de déplacement de notre neutron dans le réacteur.

$$\hat{u}_\alpha = (u_{x,\alpha}, u_{y,\alpha}) = (\cos(\varphi_\alpha), \sin(\varphi_\alpha)) \quad (4.1)$$

Avec φ_α l'angle entre la direction $\hat{u}_\alpha$ et l'axe X .

Nous devons aussi sélectionner un nombre d'angles avec lequel nous voulons travailler. Tous les nombres d'angles ne sont pas permis par DRAGON pour les méthodes cycliques. D'après [24] pour des lignes d'intégration cycliques en 2D il est recommandé de choisir une valeur égale à $P + 1$, où P est un nombre premier. Ce qui nous donne comme choix possibles 8, 12, 14, 18, 20, 24, ou 30 pour le nombre d'angles. Dans DRAGON, le nombre d'angles a une borne supérieure, on ne peut pas dépasser la limite de 30 angles différents.

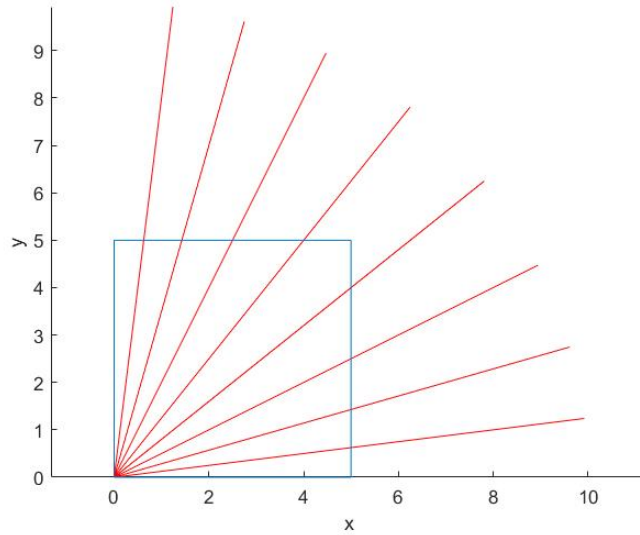


FIGURE 4.2 Représentation de 8 angles dans une cellule.

Nous allons tracer la trajectoire dans le plan de la cellule. Pour ce faire, nous allons déplier la cellule à l'infini en respectant les conditions frontières que nous détaillerons par la suite. Notre neutron commence en un point $\vec{r}_s$ et termine dans une cellule équivalente à la première en un point $\vec{r}_e$. Ils sont distants de $\vec{r}_r$. On peut écrire :

$$\vec{r}_e = \vec{r}_s + \vec{r}_r \quad (4.2)$$

La distance $\vec{r}_r$ est définie comme suit :

$$\vec{r}_r = m_x X_p \vec{x} + m_y Y_p \vec{y} \quad (4.3)$$

Avec X_p et Y_p respectivement les dimensions de la cellule selon $\vec{x}$ et $\vec{y}$, m_x et m_y sont des entiers définissant la quadrature angulaire. On note aussi que la norme du vecteur $\vec{r}_r$ représentera la distance totale parcourue par le neutron dans la cellule avant de revenir à son point de départ.

Pour le choix angulaire, nous utiliserons la définition suivante pour l'angle azimutal :

$$\varphi_\alpha = \varphi_{m_x, m_y} = \arctan\left(\frac{m_y Y_p}{m_x X_p}\right) \quad (4.4)$$

Les entiers m_x et m_y sont premiers entre eux. En pratique, nous calculerons le vecteur directeur ainsi que sa normale :

$$\vec{u}_\alpha = \vec{u}_{m_x, m_y} = \left(\frac{m_x * X_p}{d_{m_x, m_y}}, \frac{m_y * Y_p}{d_{m_x, m_y}}\right) = (\cos(\varphi_{m_x, m_y}), \sin(\varphi_{m_x, m_y})) \quad (4.5)$$

$$\vec{s}_\alpha = \vec{s}_{m_x, m_y} = \left(-\frac{m_y * Y_p}{d_{m_x, m_y}}, \frac{m_x * X_p}{d_{m_x, m_y}}\right) = (-\sin(\varphi_{m_x, m_y}), \cos(\varphi_{m_x, m_y})) \quad (4.6)$$

Avec $d_{m_x, m_y} = \sqrt{(m_x X_p)^2 + (m_y Y_p)^2}$ la diagonale sur la géométrie dépliée.

Nous allons maintenant voir comment choisir les entiers m_x et m_y en 2D dimensions. Comme énoncé précédemment les angles dépendent de la quadrature angulaire que nous choisissons ainsi que du nombre d'angles que nous voulons. Pour un nombre d'angles N_{ANGLE} , les valeurs de m_x et m_y sont trouvées par les formules suivantes avec $i \in [1, N_{ANGLE}]$:

$$\begin{aligned} m_x &= i \\ m_y &= N_{ANGLE} - i + 1 \end{aligned} \quad (4.7)$$

Plusieurs propriétés en découlent, notamment le fait que $m_x + m_y = N_{ANGLE} + 1$ que nous utiliserons par la suite.

Si nous prenons comme exemple une valeur de $N_{ANGLE} = 8$, nous aurons les valeurs données au tableau (4.1) pour m_x et m_y .

Nous avons défini précédemment que les entiers m_x et m_y devaient être premiers entre eux. Cependant, dans certains cas du tableau (4.1) comme $m_x = 3$ et $m_y = 6$, ils ne le sont pas. Dans ces cas, l'angle sera tracé, mais en utilisant l'équation (4.8).

$$m'_x = \frac{m_x}{\text{PGCD}(m_x, m_y)} \quad m'_y = \frac{m_y}{\text{PGCD}(m_x, m_y)} \quad (4.8)$$

où $\text{PGCD}(m_x, m_y)$ représente le plus grand commun diviseur des deux entiers m_x et m_y .

TABLEAU 4.1 Les entiers m_x et m_y pour $N_{ANGLE} = 8$ en 2D.

m_x	m_y
1	8
2	7
3	6
4	5
5	4
6	3
7	2
8	1

De cette façon, nous pouvons voir que les entiers m_x et m_y sont inclus dans l'intervalle $[1, N_{ANGLE}]$, et ainsi nos angles sont toujours compris dans l'intervalle ouvert $]0, \pi/2[$ en radian.

D'après le document [24], nous utiliserons dans notre cas la quadrature à poids égal numéro 2 dans DRAGON soit EQW2, où les angles 0 et $\pi/2$ ne sont pas considérés. En effet, la prise en compte des angles 0 et $\pi/2$ lors d'un tracking cyclique peut s'avérer complexe dû aux tracés des lignes suivant les directions des axes x et y .

4.3.2 Les conditions frontières

En général, le tracé des lignes d'intégration ne les prend pas en compte. Il est possible d'en tenir compte après le tracé en faisant des approximations sur les lignes d'intégration finies. Seul le tracé cyclique prend en compte ces conditions lors du tracé, cela permet une plus grande précision dans le calcul. Tout d'abord, nous rencontrerons 2 conditions frontières possibles c'est-à-dire : La condition frontière de réflexion et celle de translation. Nous aurons 4 faces où ces conditions peuvent intervenir : X^- , X^+ , Y^- et Y^+ explicitées dans la figure (4.1). Il existe aussi la condition frontière dite de vide, elle définit un vide sur un côté de notre géométrie. Dans DRAGON, la condition de vide correspond à une réflexion avec albedo (coefficient de réflexion) nul. À noter que lorsqu'une condition de translation est imposée sur une face elle est aussi imposée sur sa face réciproque. Donc, si nous avons une condition frontière de translation en X^- nous l'aurons aussi en X^+ , nous parlerons simplement de condition frontière en X . En fonction des conditions frontières pour la distance, $\vec{r}_r$ nous ne prendrons pas les mêmes valeurs pour effectuer le déploiement de la cellule. Pour les

conditions de réflexion et de translation, nous aurons respectivement :

$$\begin{aligned} X_{\text{pR}} &= 2\Delta x & Y_{\text{pR}} &= 2\Delta y \\ X_{\text{pT}} &= \Delta x & Y_{\text{pT}} &= \Delta y \end{aligned} \quad (4.9)$$

Nous allons déplier la cellule en prenant en compte les conditions frontières de translation et de réflexion. Nous allons montrer un déploiement de la cellule avec un premier angle selon les entiers m_x et m_y . Pour notre exemple (4.3), nous allons prendre les entiers suivants $m_x = 4$ et $m_y = 5$.

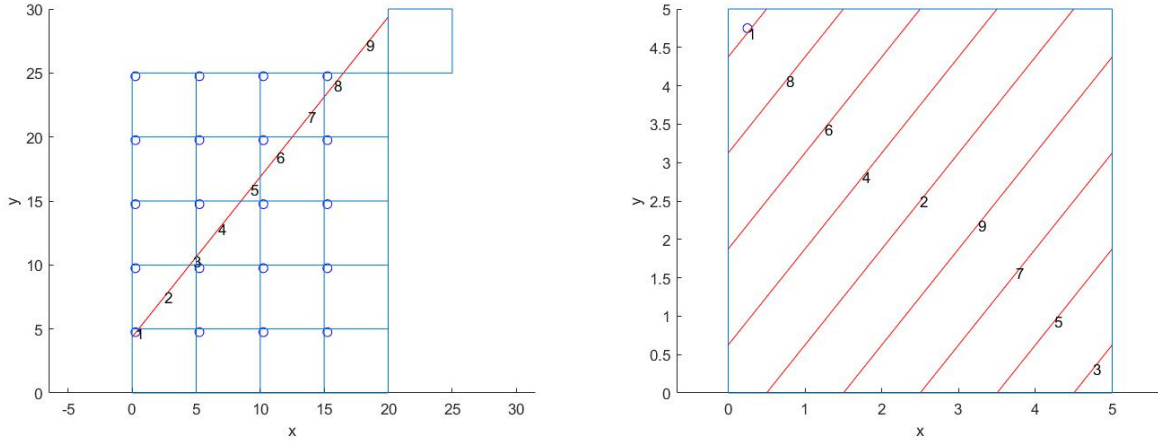


FIGURE 4.3 Première ligne d'un cycle avec condition frontière de translation sur une figure dépliée et non dépliée.

Pour illustrer la réflexion, nous utiliserons $m_x = 2$ et $m_y = 3$, car comme nous l'avons vu dans l'équation (4.9) nous doublons l'espace en réflexion. Ainsi par souci d'illustration sur la figure (4.8) nous avons choisi des valeurs d'entiers plus faibles.

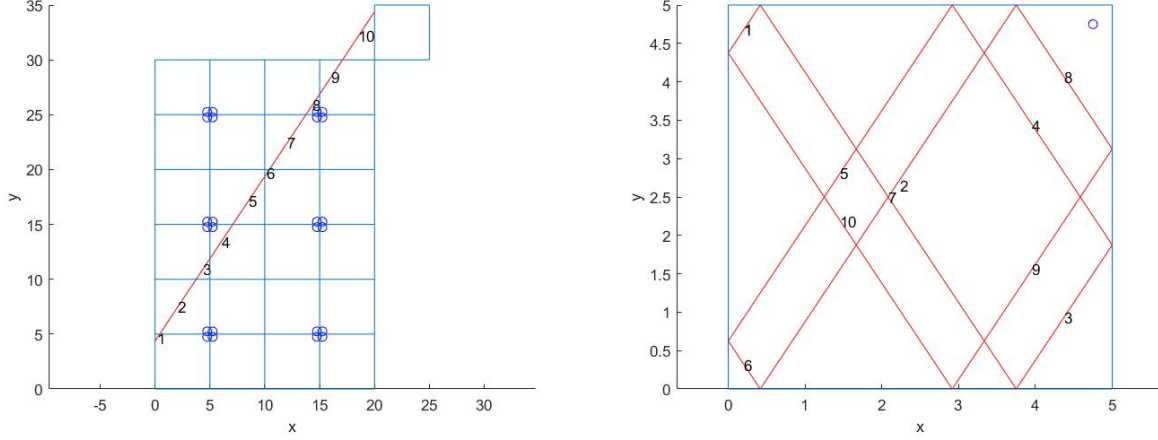


FIGURE 4.4 Première ligne d'un cycle avec condition frontière de réflexion sur une figure dépliée et non dépliée avec $m_x = 2$ et $m_y = 3$.

En fonction des conditions frontières, nous pouvons voir que la position des cellules les unes par rapport aux autres est différente. Nous pouvons le voir grâce au repère placé dans les coins des cellules sur les figures (4.3) et (4.4).

Nous définissons le point D de position $P_D = (X, Y)$ et de direction principale $\vec{u}_D = (\vec{u}_D(1), \vec{u}_D(2))$. C'est sur ce point de base que nous appliquerons les différentes conditions frontières. Ci-dessous, l'ensemble des évolutions possibles pour les valeurs du point et de sa direction après application des conditions frontière. Nous nommerons ces points après application D' avec sa position $P_{D'}$ et sa direction $\vec{u}_{D'}$.

Translation X^-

$$P_{D'} = (X + L_x, Y)$$

$$\vec{u}_{D'} = \vec{u}_D = (\vec{u}_D(1), \vec{u}_D(2))$$

Translation X^+

$$P_{D'} = (X - L_x, Y)$$

$$\vec{u}_{D'} = \vec{u}_D = (\vec{u}_D(1), \vec{u}_D(2))$$

(4.10)

Réflexion X

$$P_{D'} = P_D = (X, Y)$$

$$\vec{u}_{D'} = (-\vec{u}_D(1), \vec{u}_D(2))$$

(4.11)

$$\begin{array}{ll}
\text{Translation } Y^- & \text{Translation } Y^+ \\
P_{D'} = (X, Y + L_y) & P_{D'} = (X, Y - L_y) \\
\vec{u}_{D'} = \vec{u}_D & \vec{u}_{D'} = \vec{u}_D
\end{array} \tag{4.12}$$

$$\begin{array}{l}
\text{Réflexion } Y \\
P_{D'} = P_D \\
\vec{u}_{D'} = (\vec{u}_D(1), -\vec{u}_D(2))
\end{array} \tag{4.13}$$

Avec L_x et L_y respectivement les dimensions de la cellule en X et Y. On précise que pour appliquer ces conditions sur le point D , celui-ci doit appartenir à une face, il y aura dans ce cas un point D avant application des conditions frontières et un point D' après application des conditions frontières. Le point D et D' peuvent avoir les mêmes coordonnées, mais une direction différente, lors des quadratures il faudra le prendre en compte. Cette particularité interviendra lors des quadratures surfaciques où nous devrons appliquer un poids de $\frac{1}{2}$ à chacune des surfaces (2 fois la même surface pour la réflexion et deux surfaces différentes pour la translation).

4.3.3 Les directions de la quadrature

Pour que le tracking soit efficace, nous devons suivre nos lignes dans toutes les directions. De plus, nous notons que DRAGON est programmé de sorte que lorsqu'une direction est choisie sa direction opposée est aussi tracée, c'est-à-dire que les 2 sens d'une direction sont considérés. Ceci nous évitera d'analyser plus de directions que nécessaire. En effet, on voit que sur la figure (4.4) nous avons toutes les directions alors que sur la figure (4.3) nous avons qu'une des directions qui est suivie. Nous devons par conséquent tracer nos lignes dans la direction complémentaire afin d'avoir une quadrature complète.

Dans un cas de translation en X et en Y, seule la direction (+,+) sera tracée. Afin d'avoir toutes les directions, nous devons tracer une seconde direction soit (+,-) ou alors (-,+). Nous tracerons alors quadrant par quadrant. Cette notion est très importante afin d'avoir un tracé complet et elle prendra plus d'importance en trois dimensions où il sera possible avec des conditions frontières de translation en X,Y et Z, de devoir tracer 4 directions différentes afin de pouvoir décrire le motif complet.

Dans DRAGON nous allons notifié ces conditions avec un vecteur : IPER. Ce vecteur possède

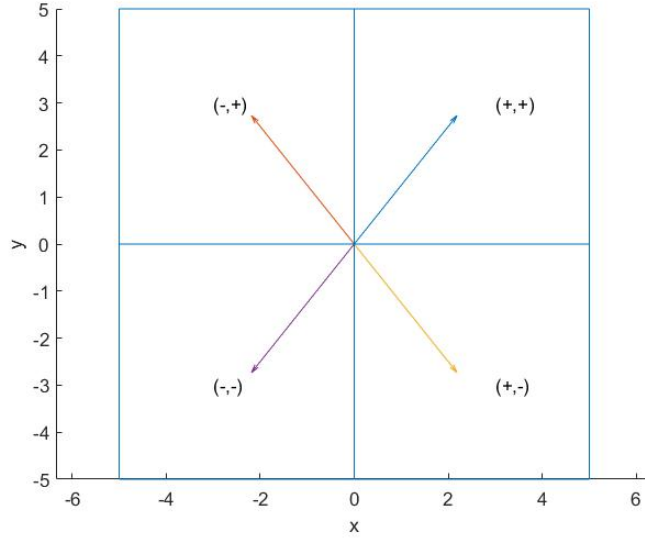


FIGURE 4.5 Représentation des 4 directions et des 4 quadrants en 2D.

2 coordonnées en 2D. $IPER(1)$ correspond aux faces X et $IPER(2)$ correspond aux faces Y .

$$\begin{aligned} IPER(i) &= 1 && \text{Translation sur les faces } i. \\ IPER(i) &= 2 && \text{Réflexion sur les faces } i. \end{aligned} \quad (4.14)$$

De plus, nous devons connaître le nombre de quadrant que nous devons tracer, cette valeur correspond à $NSCAN$ selon la formule (4.15)

$$NSCAN = 2^{(\max(0, 3 - (IPER(1) + IPER(2))))} \quad (4.15)$$

4.3.4 Poids angulaire et spatial

L'objectif de notre travail est de résoudre l'équation de transport des neutrons. Nous avons vu dans la partie (3) plusieurs méthodes pour le faire. Afin d'utiliser ces méthodes, nous devons réaliser des intégrales sur les différentes variables de notre équation. Afin de pouvoir intégrer, nous allons avoir besoin de poids. Nous aurons besoin de ces poids sur l'espace et sur les angles. Nous allons donc voir la quadrature spatiale ainsi que la quadrature angulaire.

Pour le calcul du poids angulaire, nous rappelons que le nombre d'angles est limité comme nous l'avons vu auparavant. Nous définissons le poids angulaire pour $i \in [2, N_{angle} - 1]$ comme suit :

$$\begin{aligned}
\omega_1 &= \frac{\varphi(2) + \varphi(1)}{\pi} \\
\omega_N &= 1 - \frac{\varphi(N) + \varphi(N-1)}{\pi} \\
\omega_i &= \frac{\varphi(i+1) - \varphi(i-1)}{\pi}
\end{aligned} \tag{4.16}$$

Afin de vérifier que les poids sont corrects, il est important que tous les poids angulaires soient positifs et doivent vérifier :

$$\sum_i^N \omega_i = 1 \tag{4.17}$$

Pour l'espacement des lignes, le poids que nous définirons sera l'écartement entre chaque ligne. Cet écartement nous permettra aussi de calculer la surface de notre cellule, ce calcul étant utilisé plus tard pour l'évaluation du volume 2D et la vérification de la cohérence entre les poids et les lignes d'intégration.

$$\begin{aligned}
N_{T_{2D}} &= \max(1, \text{round}(\frac{Dens}{e_{mx,my}} + 0.5)) \\
\Delta_{2D_{mx,my}} &= (e_{mx,my} N_{T_{2D}})^{-1}
\end{aligned} \tag{4.18}$$

L'entier $N_{T_{2D}}$ correspond au nombre de fois que nous aurons l'écartement $e_{mx,my}$ dans la densité fournie par l'utilisateur. Où : $e_{mx,my} = \sqrt{\left(\frac{m_y}{X_p}\right)^2 + \left(\frac{m_x}{Y_p}\right)^2}$ représente la densité minimale de ligne (en ligne/cm), qui permet un espacement uniforme des lignes d'intégration cycliques selon la ligne de direction $\vec{s}$. La fonction round donne le nombre de fois que $e_{mx,my}$ est contenu dans $Dens + \frac{1}{2}e_{mx,my}$, soit un entier N tel que $\text{round}(\frac{Dens}{e_{mx,my}} + 0.5) = N$ avec $Dens$ la densité choisie par l'utilisateur. Comme nous le voyons dans la formule (4.18), la densité choisie par l'utilisateur est remplacée par $N e_{mx,my}$ afin que le calcul ne devienne jamais incohérent, peu importe le choix de la densité de l'utilisateur.

Il est important de noter que la densité minimale de ligne $e_{mx,my}$ est la plus petite densité de ligne possible afin d'obtenir un tracé cyclique cohérent, nous aurons d'après l'équation (4.18), le plus grand écartement admissible comme étant égal à $\Delta_{2D_{mx,my}} = 1/e_{mx,my}$. Les autres écartements possibles sont une division du plus grand écartement par un entier. L'objectif est d'obtenir un pas régulier ayant comme densité de ligne un multiple entier de $e_{mx,my}$.

Nous allons démontrer la formule (4.18). Pour ce faire nous allons utiliser la figure (4.6). Nous allons dans un premier cas le faire pour une $Dens$ très faible choisie par l'utilisateur, tel que $\Delta_{2D_{mx,my}} = 1/e_{mx,my}$. Il est important que les lignes cycliques soient uniformément espacées selon une direction perpendiculaire $\vec{u}$ afin d'assurer que l'intégrale selon cette direction soit

à poids constant.

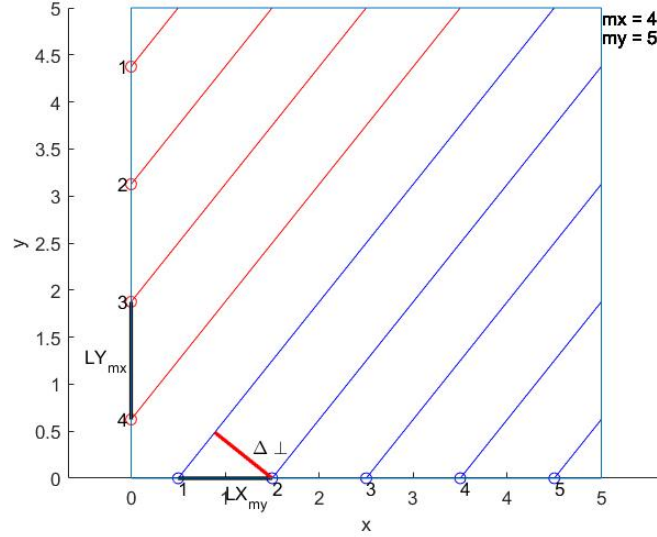


FIGURE 4.6 Représentation de l'écartement des lignes d'intégration.

Nous cherchons donc à évaluer la valeur $e_{mx,my}$ qui représente la densité minimale de ligne selon la direction orthogonale à la direction des lignes d'intégration. Nous allons pour commencer regarder les points d'intersection sur les faces X et Y .

Le nombre d'intersections entre les lignes d'intégration et une face X est égale à m_y , nous remarquons la même chose sur une face Y avec m_x points d'intersection. De plus, afin de conserver la propriété cyclique de nos lignes chaque point d'intersection sur l'axe X ou Y doit être espacé de la même distance (cette distance dépend de m_x et m_y donc elle n'est pas égale entre les deux axes). Le premier et le dernier point sur l'axe ne sont pas espacés de la même distance par rapport à la limite de la géométrie, cela est dû au fait que nous déplaçons la cellule comme sur la figure (4.3). Nous aurons notre écartement en additionnant la distance à la limite de la géométrie du premier et du dernier point.

Nous définissons ces distances que nous retrouvons sur la figure (4.6) :

$$\begin{aligned} LY_{mx} &= \frac{Y_p}{m_x} \\ LX_{my} &= \frac{X_p}{m_y} \end{aligned} \quad (4.19)$$

Nous avons les valeurs : LX_{my}^{-1} et LY_{mx}^{-1} qui représentent respectivement la densité de point d'intersection selon l'axe X et Y . L'espacement des lignes $\Delta_{\perp}$ sur la ligne orthogonale à $\vec{u}$

est donné par :

$$\Delta_{2D_{mx,my}} = \Delta_{\perp} = \frac{LY_{mx}}{\cos(\varphi(mx, my))} = \frac{LX_{my}}{\sin(\varphi(mx, my))} \quad (4.20)$$

Nous avons les valeurs de $\cos(\varphi(mx, my))$ et $\sin(\varphi(mx, my))$ avec la formule (4.5) que nous pouvons écrire de la manière suivante :

$$\begin{aligned} \cos(\varphi(mx, my)) &= \frac{m_x * X_p}{\sqrt{(m_x X_p)^2 + (m_y Y_p)^2}} = \frac{m_x}{Y_p \sqrt{\left(\frac{m_y}{X_p}\right)^2 + \left(\frac{m_x}{Y_p}\right)^2}} \\ \sin(\varphi(mx, my)) &= \frac{m_y * Y_p}{\sqrt{(m_x X_p)^2 + (m_y Y_p)^2}} = \frac{m_y}{X_p \sqrt{\left(\frac{m_y}{X_p}\right)^2 + \left(\frac{m_x}{Y_p}\right)^2}} \end{aligned} \quad (4.21)$$

D'après les formules (4.19),(4.21) et en utilisant : $e_{mx,my} = \sqrt{\left(\frac{m_y}{X_p}\right)^2 + \left(\frac{m_x}{Y_p}\right)^2}$, Nous obtenons :

$$\begin{aligned} \cos(\varphi(mx, my)) &= \frac{1}{LY_{mx} e_{mx,my}} \\ \sin(\varphi(mx, my)) &= \frac{1}{LX_{my} e_{mx,my}} \end{aligned} \quad (4.22)$$

En utilisant l'équation (4.22) dans l'équation (4.20), nous obtenons bien :

$$\begin{aligned} \Delta_{\perp} &= \frac{1}{\sqrt{\left(\frac{m_y}{X_p}\right)^2 + \left(\frac{m_x}{Y_p}\right)^2}} = \frac{1}{e_{mx,my}} \\ \Delta_{\perp} &= \Delta_{2D_{mx,my}} = \frac{1}{e_{mx,my}} \end{aligned} \quad (4.23)$$

Pour une *Dens* choisie par l'utilisateur très faible nous avons bien ce que nous voulions démontrer : $\Delta_{2D_{mx,my}} = 1/e_{mx,my}$. Afin de démontrer la formule (4.18) dans son cas général, c'est-à-dire pour n'importe quelle densité *DENS* choisie par l'utilisateur, nous allons multiplier la densité de ligne minimale par un entier afin de conserver les propriétés des lignes cycliques. Cet entier dépendra du nombre de fois que $e_{mx,my}$ est contenu dans $Dens + \frac{1}{2}e_{mx,my}$. Nous obtenons alors l'équation (4.18).

Dans l'évaluation du poids spatial nous allons, en deux dimensions, prendre l'écartement entre 2 lignes, car il est constant en fonction de l'angle et nous allons donc avoir l'égalité suivante :

$$\omega s_{\alpha} = \omega s_{mx,my} = \Delta_{2D_{mx,my}} \quad (4.24)$$

De plus, afin de vérifier notre précision du calcul des poids, nous utilisons un calcul de vérification dont nous connaissons le résultat. Nous allons évaluer la surface de la cellule à partir des lignes d'intégration en reprenant la formule (3.9) vue dans la partie (3) :

$$\sum_{i=1}^N \omega_i \left(\sum_{j=1}^{LINMAX} L_{j,i} * \omega_{s_{\alpha},i} \right) = V_{2D} \quad (4.25)$$

La variable $L_{j,i}$ représente la longueur du segment j pour l'angle i . Ces segments sont numérotés de $[1, LINMAX]$ où $LINMAX$ est le nombre maximal de segments admissibles pour les angles. $LINMAX$ ne dépend pas d'un angle, mais du nombre total d'angles N , et on peut en admettre un maximal pour tous les angles. Afin de calculer $LINMAX$ nous devons trouver le nombre maximal de segments possibles, quel que soit le cycle.

Ce calcul nous permet de trouver pour chaque angle l'aire de notre cellule que nous pondérons par les poids angulaires. Nous pouvons ainsi calculer l'aire de la cellule pondérée par le poids spatial et les poids angulaires.

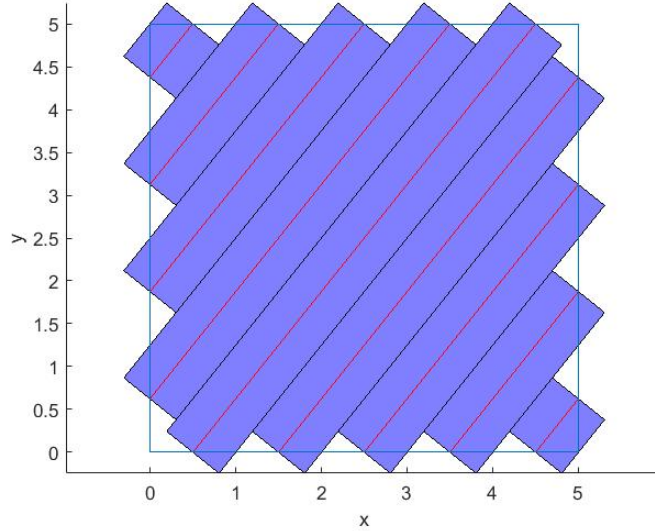


FIGURE 4.7 Représentation d'une quadrature pour une direction et des conditions frontières de translation.

Comme nous l'avons expliqué précédemment, il est nécessaire d'avoir 4 directions pour avoir une quadrature juste. Ces directions se répercutent dans le calcul de l'aire. En effet, le calcul de l'aire de notre cellule s'effectue en sommant l'aire dans les 4 directions. Ainsi, nous trouverons 4 fois l'aire de cellule, il faut donc être vigilant dans nos calculs.

Pour évaluer la précision de notre méthode nous comparerons la valeur obtenue par ce calcul avec celle de la surface théorique calculée avec les données du problème.

De plus, nous pouvons évaluer le nombre de segments par cycles. Nous aurons besoin de ce terme afin de dimensionner les variables en FORTRAN. Il est présent dans la formule (4.25) sous le nom de $LINMAX$. Le nombre de segments se trouve en fonction du nombre de fois où nous déplions notre géométrie. Donc en fonction de m_x et m_y comme nous l'avons vu dans l'équation (4.7). Nous avons noté que :

$$\begin{aligned} m_x + m_y &= N_A + 1 \\ N_{STRAN} &= N_A + 1 \\ N_{SREFL} &= 2 * (N_A + 1) \end{aligned} \tag{4.26}$$

Avec N_A le nombre d'angles. Et comme nous pouvons le voir il y a une différence entre des conditions frontières de translation et de réflexion, cela est dû au fait en réflexion nous prenons un facteur 2 pour les distances. Le nombre de segments est constant en 2D pour tous les cycles, nous verrons par la suite qu'il varie en 3D selon les cycles.

4.3.5 Nombre de points

Nous allons établir un nombre de points représentant le nombre de cycles maximum que nous pouvons faire. Il est aisément calculé à partir de la densité minimale de ligne.

$$NP_{2D} = \text{floor}(D * e_{mx,my} + 1) \tag{4.27}$$

Avec $D = \sqrt{(X_p)^2 + (Y_p)^2}$ soit la diagonale du cube et floor : l'arrondi inférieur. L'équation (4.27) correspond au nombre de points maximum admissible par la géométrie. Nous devons nous assurer que le nombre de points pour le cycle est pair. Une séparation impaire de la cellule pourrait donner lieu à un phénomène de coins. C'est-à-dire qu'une droite passe dans un coin de notre géométrie, c'est une situation que nous voulons éviter, car l'application des conditions frontières est alors impossible.

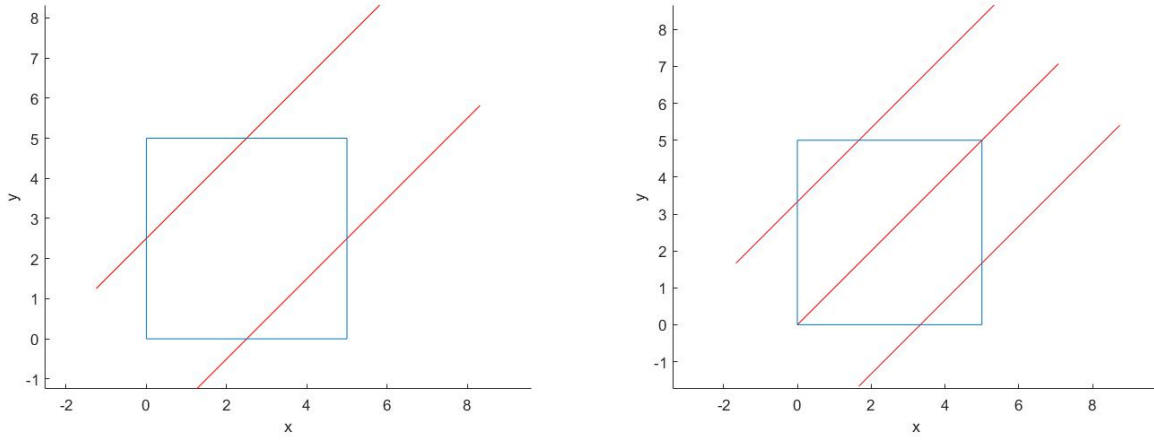


FIGURE 4.8 Nombre de point pairs, impairs sur des lignes d'intégration à très faible densité.

Cependant, avec la formule (4.27) nous surestimons le nombre de points. Il se peut dans certaines situations que des points sortent de la géométrie comme nous le voyons dans la figure (4.9).

Pour remédier à ce phénomène, nous vérifierons à chaque itération que chaque ligne coupe la géométrie en deux points distincts. En pratique en 2D nous n'atteindrons pas de ligne sortant de la géométrie, car la méthode que nous avons limite le nombre de lignes que nous traçons.

4.3.6 Condition d'arrêt

Dans le tracé uniforme, nous traçons toutes les lignes imposées par le nombre de points, alors que dans le tracé cyclique si nous faisons cela certains cycles seront tracés plusieurs fois. Chaque ligne doit être tracée uniquement une fois sinon le poids accordé à certaines lignes sera trop important, et les calculs seront faux. Nous devons arrêter notre tracé lorsque toutes les lignes sont tracées. Dans DRAGON la méthode actuellement implémentée en 2D utilise le nombre de segments en calculant exactement le nombre de segments nécessaires afin d'avoir l'ensemble des cycles. La condition d'arrêt en 2D est donc le nombre de segment tracé. Cependant nous devons aussi prendre en compte la densité de ligne d'intégrations donnée par l'utilisateur. Le nombre de segment par cycle est constant en 2D comme nous l'avons vu dans l'équation (4.26) afin de prendre en compte la densité de ligne nous utiliserons le nombre entier $N_{T_{2D}}$ défini dans l'équation (4.18).

Chaque cycle tracera $N_A + 1$ segment, et nous aurons besoin de $N_{T_{2D}}$ nombre de cycle. Donc

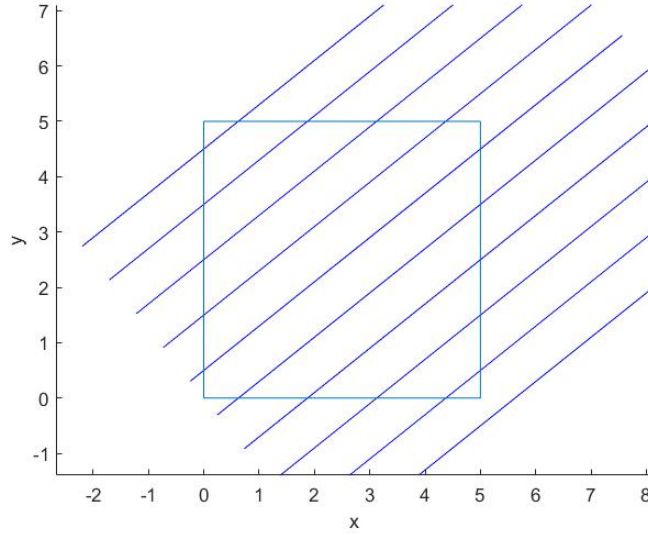


FIGURE 4.9 Représentation d'un nombre de points supérieure à la limite admissible par la géométrie

en comptant le nombre de segments N_S notre condition d'arrêt sera décrite par la formule (4.28).

$$N_S \geq N_{CF} * N_{T_{2D}}(N_A + 1) \quad (4.28)$$

N_{CF} est un facteur permettant de prendre en compte le facteur multiplicatif sur la géométrie pour des conditions frontières de réflexions. $N_{CF} = 2$ si les conditions frontières sur toutes les faces sont des conditions de réflexions et $N_{CF} = 1$ sinon.

Nous pouvons simplifier l'équation (4.28) en utilisant le nombre de cycles N_C au lieu du nombre de segments N_S .

$$N_C = N_{CF} * N_{T_{2D}} \quad (4.29)$$

N_C correspond au nombre de cycles que nous devons tracer afin d'avoir une bonne quadrature ainsi qu'aucune ligne en double. Nous parlons de conditions d'arrêt, car nous arrêterons notre tracé lorsque nous atteindrons N_C cycles.

La figure (4.10) mais en évidence 2 cycles (un en rouge et un en bleu), ainsi que les deux points permettant de tracer ces cycles : les points 1 et 2 de couleur cyan. Les autres points en magenta sont intersectés par les cycles en rouge et en bleu et il est inutile de les tracer à nouveau. Dans ce cas, la formule (4.27) donnera 10 (les points représentés en magenta et en cyan) et la formule (4.29) donnera 2 (les points représentés en cyan).

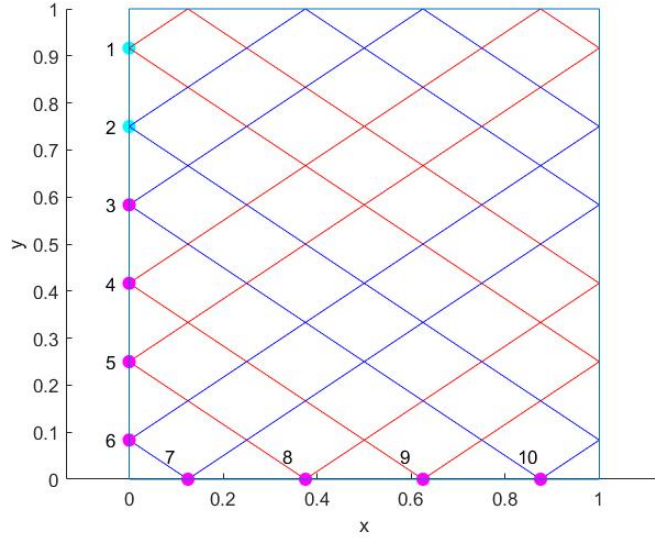


FIGURE 4.10 Différence entre le nombre de points effectif et possible.

4.3.7 Positionnement du point de départ du cycle

La position du premier point du cycle en 2D est primordiale. En effet, l'ensemble des cycles tracés pour un angle dépend de sa position et de son incrémentation. Nous allons voir la méthode afin de trouver ce point.

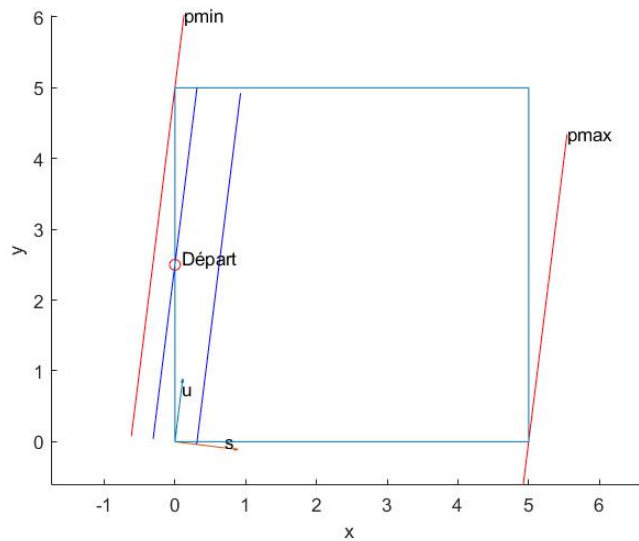
Pour rappel le vecteur directeur et sa normale sont $\vec{u}$ et $\vec{s}$. Nous cherchons les positions maximales et minimales où une ligne de direction $\vec{u}$ est dans la géométrie. On utilise pour cela les coins de la géométrie : $\vec{c}_i (i = 1, 4)$, que nous projetons sur le vecteur $\vec{s}$ et nous prenons les valeurs extrêmes. Les lignes extrêmes sont ainsi données par l'équation : (4.30).

$$\begin{aligned} p_{\min} &= \min(\vec{s} \cdot \vec{c}_i) \\ p_{\max} &= \max(\vec{s} \cdot \vec{c}_i) \end{aligned} \quad (4.30)$$

Nous avons choisi de commencer à partir de $p_{\min}$ et d'incrémenter avec le pas, qui est l'écartement entre les lignes : l'équation (4.18). Cependant, dans la formule (4.31) nous ne commençons pas à $p_{\min}$ mais à un demi-pas après $p_{\min}$ afin d'éviter les coins.

Ainsi nous avons trouvé la ligne p_m qui représente la ligne de départ du cycle m avec $m = [1, N_p]$

$$p_m = \left(p_{\min} + \left(m - \frac{1}{2} \right) * \Delta_{2D} \right) * \vec{s} \quad (4.31)$$



Algorithme Calculer des lignes cycliques en 2D

Obtenir les valeurs de la géométrie

Do $l \in \text{NANGL}$ {Boucle sur les angles azimutaux}

Obtenir m_x, m_y

Calculer $D_{m_x, m_y}, \vec{s}, \vec{u}$

Do $n \in \text{NP}$ {Nombre de points}

Do $\text{ISCAN} \in \text{NSCAN}$ {Nombre de quadrants}

Appliquer le changement de quadrant

Calculer $p_{\min}, p_{\max}$

Calculer Le point de départ du cycle

Calculer Le cycle avec tous les segments

Obtenir Incrémenter le nombre de cycles N_c

IF $N_c \geq \text{ISCAN} * N_{T2D}$

EXIT

END IF

Incrémenter le point de départ

End Do

End Do

End Do

FIGURE 4.12 Algorithme en pseudo code pour les lignes d'intégration cycliques en 2D.

4.3.9 Conclusion méthode 2D

L'objectif principal de la méthode 2D est d'expliquer la base de la méthode 3D, en effet la méthode 3D repose sur 2 tracés de type 2D. Afin de présenter cette méthode en 3D dans la partie suivante, nous utiliserons en grande partie la méthode 2D.

4.4 Tracking cyclique en 3D

La partie précédente nous a présenté la méthode utilisée en 2D afin de calculer les lignes d'intégration. C'est le point de départ de la méthode 3D, nous l'utiliserons et expliquerons les différentes subtilités afin de créer le tracking cyclique en 3D. On note que la partie 3D

consiste à effectuer deux fois la méthode 2D sur deux plans différents. En récupérant les premières informations du tracking en 2D, nous créons un nouveau plan où nous appliquons de nouveau la méthode.

La figure (4.13) nous permet d'illustrer le plan U, nous pouvons voir aussi la première ligne du tracking 2D, qui permet de tracer le plan U. En pratique, nous traçons la première ligne du cycle en 2D et nous cherchons les points d'intersection entre cette ligne et les limites de la géométrie. Pour cela nous projetons ces points sur l'axe vertical, nous pouvons ainsi tracer le plan U.

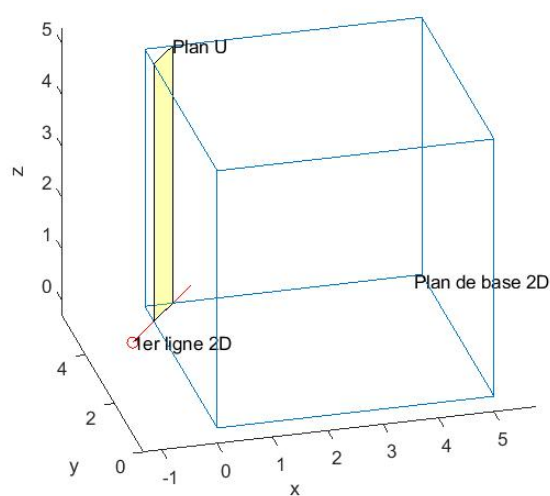


FIGURE 4.13 Représentation du plan de base en 2D et du plan U sur la géométrie 3D.

Nous allons aussi définir notre cellule de travail en 3D. Comme pour la figure (4.1) nous allons nommer nos différentes surfaces sur la figure (4.14).

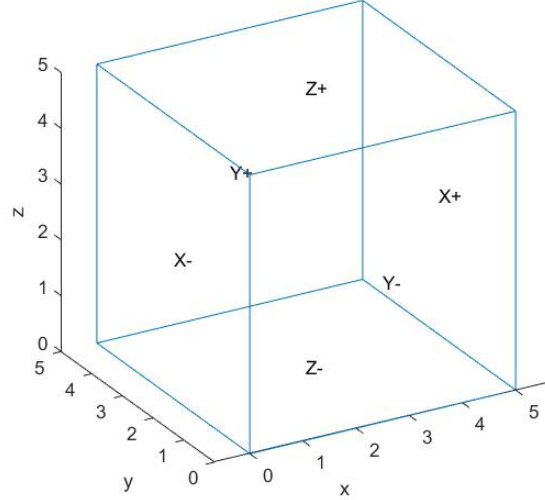


FIGURE 4.14 Représentation de la cellule de travail et notation des surfaces en 3D.

4.4.1 Choix des angles

Notre premier angle, c'est-à-dire l'angle azimutal φ sera identique à celui décrit dans la partie (4.3.1). Nous devons cependant introduire un deuxième angle afin de se positionner en trois dimensions. Cet angle est appelé angle polaire θ . Nous allons donc avoir une modification des directions :

$$\begin{aligned}\vec{u}_{\varphi,\theta} &= (\cos(\varphi) \sin(\theta), \sin(\varphi) \sin(\theta), \cos(\theta)) \\ \vec{s}_{\varphi,\theta} &= (-\vec{u}_{\varphi,\theta}(2), \vec{u}_{\varphi,\theta}(1), 0) \\ \vec{t}_{\varphi,\theta} &= \vec{u}_{\varphi,\theta} \wedge \vec{s}_{\varphi,\theta}\end{aligned}\tag{4.32}$$

Les directions $\vec{s}_{\varphi,\theta}$ et $\vec{t}_{\varphi,\theta}$ sont les directions normales à la direction principale $\vec{u}_{\varphi,\theta}$.

La direction principale calculée à l'aide des deux angles $\vec{u}_{\varphi,\theta}$ représente toujours la direction des neutrons. Cependant il est important de noter que pour un angle azimutal nous aurons N_{angle} polaires. le total d'angles sera alors : $N_{angle\ total} = N_{angle}^2$.

Pour le choix de l'angle azimutal φ , il est identique à celui en 2D. Pour l'angle polaire : θ , nous allons procéder d'une manière similaire à la méthode utilisée en 2D. Nous introduisons deux entiers : m_z et m_{xy} . Ces entiers ont les mêmes propriétés que les entiers m_x et m_y . Ainsi, le tableau (4.1) est toujours applicable. Cependant, nous aurons pour couple de valeurs m_z

et m_{xy} un ensemble de valeur de m_x et m_y que nous retrouvons dans le tableau (4.1) Nous allons mettre en évidence les calculs des directions en 3D (4.32) en reprenant aussi les valeurs 2D (4.5) et (4.6).

Afin de simplifier les notations, nous utiliserons la variable M pour les notations avec $M = [m_x, m_y, m_z, m_{xy}]$. Elle ne sera utilisée que lorsqu'une variable dépend des 4 entiers $[m_x, m_y, m_z, m_{xy}]$.

$$\begin{aligned}\vec{u}_{\varphi,\theta} &= \vec{u}_{m_x,m_y,m_z,m_{xy}} = \vec{u}_M = \left(\vec{u}_{m_x,m_y}(1) * \frac{m_{xy}d_{m_x,m_y}}{d_M}, \vec{u}_{m_x,m_y}(2) * \frac{m_{xy}d_{m_x,m_y}}{d_{m_z,m_{xy}}}, \frac{m_z Z_p}{d_M} \right) \\ \vec{s}_{\varphi,\theta} &= \vec{s}_M = \left(\vec{s}_{m_x,m_y}(1), \vec{s}_{m_x,m_y}(2), 0 \right) \\ \vec{t}_{\varphi,\theta} &= \vec{t}_M = \vec{u}_{m_x,m_y,m_z,m_{xy}} \wedge \vec{s}_{m_x,m_y,m_z,m_{xy}}\end{aligned}\tag{4.33}$$

Avec :

$$d_{m_x,m_y} = \sqrt{(m_x X_p)^2 + (m_y Y_p)^2}\tag{4.34}$$

$$d_M = \sqrt{(m_z Z_p)^2 + (m_{xy} d_{m_x,m_y})^2}\tag{4.35}$$

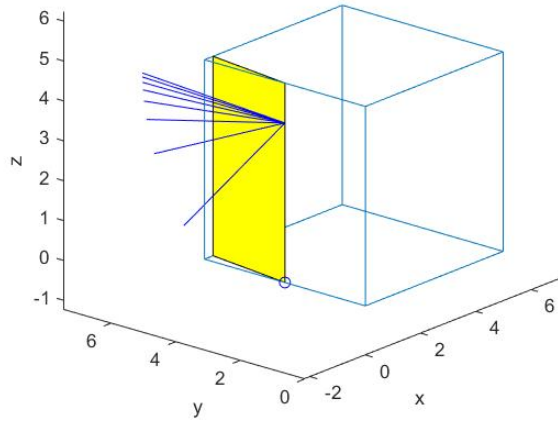


FIGURE 4.15 Représentation des angles polaires pour un angle azimutal.

Nous pouvons voir sur la figure (4.15) l'apparition d'un plan en jaune. Ce plan est la justification qui nous permet d'associer notre méthode de tracé en 3D à l'utilisation d'une méthode de tracé 2D à deux reprises.

Nous notons aussi que la répartition des angles entre $[0; \frac{\pi}{2}]$ est différente par rapport à l'angle azimutal que nous pouvons voir sur la figure (4.2).

4.4.2 Les conditions frontières

Pour les conditions frontières nous pouvons appliquer les mêmes changements pour les faces X et Y, que pour la face Z. En prenant comme point de départ $P_D = (X, Y, Z)$ et comme direction $\vec{u}_D = (\vec{u}(1), \vec{u}(2), \vec{u}(3))$

$$\begin{array}{ll}
 \text{Translation } Z^- & \text{Translation } Z^+ \\
 P_{D'} = (X, Y, Z + L_z) & P_{D'} = (X, Y, Z - L_z) \\
 \vec{u}_{D'} = \vec{u}_D & \vec{u}_{D'} = \vec{u}_D
 \end{array} \tag{4.36}$$

$$\begin{array}{l}
 \text{Réflexion } Z \\
 P_D = P_{D'} \\
 \vec{u}_{D'} = (\vec{u}(1), \vec{u}(2), -\vec{u}(3))
 \end{array} \tag{4.37}$$

Les changements sur les faces X et Y sont identiques à celles décrites dans la partie 2D mais en prenant un point et une direction en trois dimensions.

4.4.3 Les directions de la quadrature

En trois dimensions nous allons travailler sur un total de 8 octants et nous aurons à tracer 8 directions. On peut calculer le nombre de directions en utilisant la formule $N_{direction} = 2^{dimension}$. On rappelle que DRAGON parcourt le cycle dans les deux sens et que l'on ne doit pas tracer toutes les directions. Seulement 4 sont nécessaires afin d'obtenir une quadrature complète. En pratique nous tracerons uniquement les directions dans les octants supérieurs.

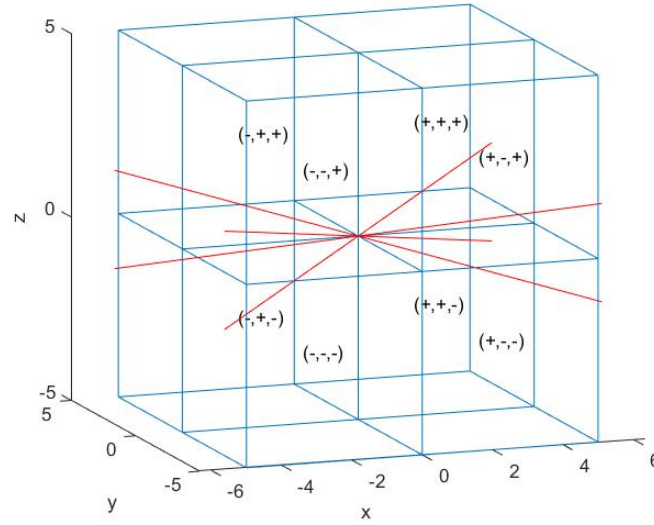


FIGURE 4.16 Représentation des 8 octants et des 8 directions.

Il est important de noter que le nombre de directions que nous devons implémenter varie en fonction des conditions frontières. En effet les conditions frontières de réflexion permettent de faire évoluer la direction contrairement aux conditions frontières de translation. En pratique, on tracera des directions supplémentaires lorsque le nombre de conditions frontières de translation est supérieur à 1.

Nous allons illustrer dans la figure (4.17), les différentes directions utilisées grâce à un exemple avec des conditions frontières de translation sur toutes les faces. Dans ce cas, nous devons tracer 4 directions afin d'avoir une quadrature complète. Ainsi, dès que nous tracerons un cycle, nous devons tracer 3 autres cycles avec des points de départ différents et des directions différentes ayant pour point le plan U de départ.

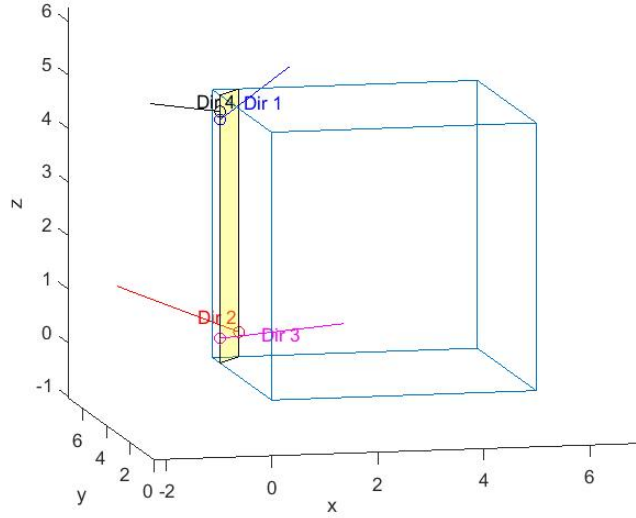


FIGURE 4.17 Représentation des 4 directions de départ des 4 cycles pour des conditions frontières de translation totale.

Les 4 directions tracées sur la figure (4.17) sont bien différentes comme nous le voulions. Elles représentent les directions des octants supérieurs si on les parcourt dans un sens et inférieurs dans l'autre sens. Nous aurons bien nos 8 directions à la fin du tracking.

De plus de la même manière qu'en 2D nous utiliserons IPER ainsi que NSCAN afin de connaître les conditions frontières sur les surfaces et le nombre d'octant à tracer. IPER suit les mêmes règles que l'équation (4.14), mais sur 3 dimensions. Nous reprenons l'équation (4.15) en ajoutant toutes les dimensions de IPER.

$$NSCAN = 2^{(5-(IPER(1)+IPER(2)+IPER(3)))} \quad (4.38)$$

4.4.4 Poids spatial et angulaire

Pour le poids angulaire, nous utiliserons la formule (4.39) permettant d'obtenir le volume en coordonnées sphériques, nous pouvons remarquer le terme $\sin(\theta)d\theta$ pour calculer le volume infinitésimal dV .

Nous nous plaçons dans un espace avec un volume A décrit par les coordonnées sphériques classiques.

$$\begin{aligned} V &= \int_A dV \\ V &= \int_0^R \int_{-\pi/2}^{\pi/2} \int_0^{2\pi} r^2 dr \cdot \sin(\theta) d\theta \cdot d\varphi \end{aligned} \quad (4.39)$$

Nous allons utiliser pour la seconde quadrature angulaire le cosinus directeur de l'angle : $\cos(\theta_\alpha) = \mu_\alpha$. Pour ce faire nous allons appliquer un changement de variable dans l'équation (4.39). Nous aurons :

$$\begin{aligned}\mu &= \cos(\theta) \\ d\mu &= \sin(\theta)d\theta\end{aligned}\tag{4.40}$$

avec $\theta \in [0, 2\pi]$ et $\mu \in [0, 1]$.

Pour un entier j appartenant à l'intervalle $[2; N - 1]$ et i appartenant à l'intervalle $[1; N]$ représentant le numéro de l'angle azimutal avec N le nombre angle azimutal et nous aurons N angle polaire par angle azimutal.

$$\begin{aligned}\omega_{3D \ i,1} &= \frac{1}{2}(\mu_{i,2} + \mu_{i,1}) \\ \omega_{3D \ i,N} &= 1 - \frac{1}{2}(\mu_{i,N} + \mu_{i,N-1}) \\ \omega_{3D \ i,j} &= \frac{1}{2}(\mu_{i,j+1} - \mu_{i,j-1})\end{aligned}\tag{4.41}$$

On rappelle que la formule (4.17) est aussi valable sur les poids angulaires polaires :

$$\sum_j^{N_\alpha} \omega_{3D \ i,j} = 1\tag{4.42}$$

Pour le poids spatial en trois dimensions, nous allons utiliser la même méthode que pour la formule (4.18). La formule que nous avons est identique à celle en 2D, mais avec les paramètres du plan U :

$$\begin{aligned}N_{T_{3D}} &= \max(1, \text{round}(\frac{Dens}{e_M} + 0.5)) \\ \Delta_{3DM} &= (e_M N_{T_{3D}})^{-1} \\ \omega_{SM} &= \Delta_{2D_{m_x, m_y}} * \Delta_{3DM}\end{aligned}\tag{4.43}$$

L'entier $N_{T_{3D}}$ correspond au nombre de fois que nous aurons l'écartement dans e_M dans la densité fournie par l'utilisateur.

Nous aurons par analogie avec l'équation 2D (4.18) $e_M = \sqrt{\left(\frac{m_z}{\sqrt{(m_x X_p)^2 + (m_y Y_p)^2}}\right)^2 + \left(\frac{m_{xy}}{Z_p}\right)^2}$. Δ_{3DM} est toujours un écartement en centimètre donc nous aurons un poids aux dimensions d'une aire. En multipliant par la longueur de la ligne d'intégration nous obtiendrons un volume.

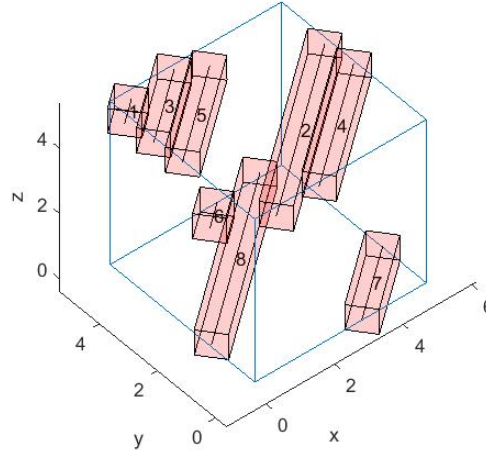


FIGURE 4.18 Représentation d'une quadrature du volume en 3D, pour des conditions frontières de translation totale.

Nous voyons sur la figure (4.18) la représentation des pavés droits permettant le calcul du volume de la cellule. Cette représentation est incomplète et permet une visualisation pas-à-pas de la construction de la quadrature volumique.

D'une manière analogue à ce que nous avons fait pour les lignes en 2D, nous vérifions notre calcul avec le calcul du volume 3D.

$$V_{exp} = \sum_{i=1}^N \omega_{2D_i} DP_{2D_i} \left(\sum_{j=1}^N \omega_{3D_{i,j}} DP_{3D_{i,j}} \left(\sum_{l=1}^{LINMAX_{i,j}} L_{l,j,i} \right) \right) \quad (4.44)$$

Avec $L_{l,j,i}$ la longueur du segment l , pour l'angle polaire j et l'angle azimutal i .

Remarque : Il est important de savoir que la variable $L_{l,j,i}$ est décomposable en différentes sommes. En effet, on peut prendre en compte les différents cycles de chaque angle ainsi que les directions d'intégration que nous avons en fonction de nos conditions frontières. Nous pourrions l'écrire comme suit pour un angle polaire et un angle azimutal fixés :

$$\sum_{c=1}^{N_{cycle}} \sum_{IDIR=1}^{NSCAN} \sum_{l=1}^{N_{segment}} L_{l,c,IDIR,j,i} \quad (4.45)$$

Le nombre de segments se trouve en fonction du nombre de fois où nous déplaçons notre géométrie, il est constant, peu importe la densité choisie. D'après le tableau (4.1) nous pouvons

lier les valeurs m_x, m_y et m_z, m_{xy} .

$$\begin{aligned} m_x &= i & m_y &= N_A - i + 1 \\ m_z &= j & m_{xy} &= N_A - j + 1 \end{aligned} \quad (4.46)$$

Avec i et j des entiers appartenant à $[1, N_A]$ et N_A le nombre d'angles choisis pour la quadrature.

$$\begin{aligned} N_{seg} &= (m_x + m_y) * m_{mxy} + m_z \\ N_{seg} &= (i + (N + 1 - i)) * (N + 1 - j) + j \\ N_{seg} &= (N + 1)^2 - j * N \\ N_{seg_{\max}} &= (N + 1)^2 - 1 * N \end{aligned} \quad (4.47)$$

On peut donc obtenir un nombre de segments maximal par cycle. Ce nombre sera utilisé afin de définir nos variables en FORTRAN.

4.4.5 Nombre de points

Nous pouvons calculer le nombre de points en 3D de manière analogue à ce que nous avons déjà fait en 2D avec la formule (4.27).

$$NP_{3D} = \text{floor}(D_{3D}e_M + 1) \quad (4.48)$$

Avec D_{3D} la diagonale de la géométrie de base soit $D_{3D} = \sqrt{X^2 + Y^2 + Z^2}$.

4.4.6 Condition d'arrêt

Comme en 2D, nous ne pouvons pas tracer tous les cycles possibles sinon nous aurons plusieurs fois le même et des quadratures fausses. Pour cela, nous devons trouver un nombre de cycles maximal à tracer en 3D. Il devra évoluer en fonction de la densité demandée par l'utilisateur. Il est important de noter que peu importe la densité des lignes, la taille des cycles est toujours la même. Donc si la densité augmente nous devons augmenter le nombre de cycles. Afin de lier le nombre de cycles à tracer et la densité de ligne nous avons décidé d'utiliser $N_{T_{2D}}$ et $N_{T_{3D}}$ que l'on trouve dans les équations (4.18) et (4.43). Nous allons tracer en tout $N_{CF} * N_{T_{2D}} * N_{T_{3D}}$ points. Cela nous permettra de tracer un nombre suffisant de points afin d'avoir des quadratures satisfaisantes. Nous verrons la répartition des cycles selon les directions dans une partie suivante.

On peut remarquer que cela réduira aussi le nombre de points par rapport à l'équation (4.48). Cette équation calcule le nombre de points maximal admissible. Afin de réaliser des quadratures justes, nous n'avons pas besoin de tous les points. De plus, l'utilisation d'un grand nombre de points peut mener à des lignes tracées en double.

4.4.7 Points de départ du cycle

Afin de trouver le premier point du cycle, nous allons appliquer de nouveau la méthode décrite dans la partie 2D. Cependant, cette fois, nous allons l'appliquer sur le cube. Nous utiliserons les nouvelles directions en 3D : $\vec{u}_{\varphi,\theta}$, $\vec{s}_{\varphi,\theta}$ et $\vec{t}_{\varphi,\theta}$.

Nous utiliserons les 8 coins du cube soit $\vec{c}_k$ avec $k \in [1, 8]$.

$$\begin{aligned} p_{min3D} &= \min(\vec{t}_{\varphi,\theta} \cdot \vec{c}_k) \\ p_{max3D} &= \max(\vec{t}_{\varphi,\theta} \cdot \vec{c}_k) \end{aligned} \tag{4.49}$$

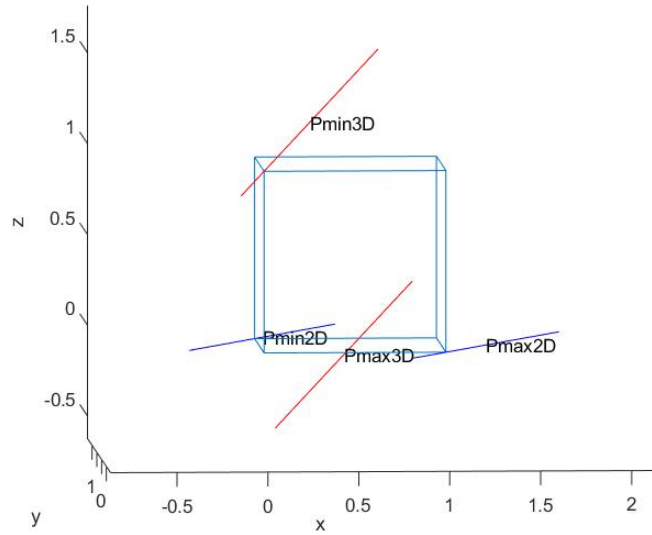


FIGURE 4.19 Représentation des lignes limites en rouge pour la 3D et en bleu pour la 2D.

Nous calculons $P3D_{min}$ le point tel qu'avec la direction $\vec{u}$ il passe par un des coins du plan U :

$$P3D_{min} = p_{\min 3D} \vec{t} \quad (4.50)$$

On utilisera uniquement la troisième coordonnée du calcul (4.50) les deux premières coordonnées sont identiques à la 2D.

4.4.8 Incrémentation en 3D

L'incrément en 3D est différente de l'incrément en 2D. En effet, nous devons incrémenter selon 2 directions différentes : $\vec{s}_{\varphi,\theta}$ et $\vec{t}_{\varphi,\theta}$, toutes deux normales à la direction initiale de notre neutron $\vec{u}_{\varphi,\theta}$.

Nous utiliserons l'incrément suivante, permettant d'incrémenter selon les vecteurs $\vec{s}_{\varphi,\theta}$ et $\vec{t}_{\varphi,\theta}$.

$$\begin{aligned} p_{m,n} &= p_m + \left(p_{\min 3D} + \left(n - \frac{1}{2} \right) \Delta_{3D} \right) \vec{t} \\ p_m &= \left(p_{\min 2D} + \left(m - \frac{1}{2} \right) * \Delta_{2D} \right) * \vec{s} \end{aligned} \quad (4.51)$$

La formule pour p_m est identique à la formule 2D (4.31). Nous partons de la base de notre formule 2D et nous utilisons $p_{\min 3D}$ afin de nous positionner en 3D.

Les entiers m et n dépendront du nombre de cycles à tracer. Plus précisément $m \in [1 : N_{CF} * N_{T_{2D}}]$ et $n \in [1 : N_{T_{3D}}]$.

Comme nous pouvons le voir sur la figure(4.20) pour une densité relativement faible et un angle quelconque, nous avons tracé les points de départ des différents cycles. Ces points sont tracés pour des conditions frontières de réflexions sur toutes les faces. Avec $N_{T_{2D}} = 3$ et $N_{T_{3D}} = 1$.

De plus, nous trouvons que les points sont alignés dans un plan perpendiculaire à la direction $\vec{t}$.

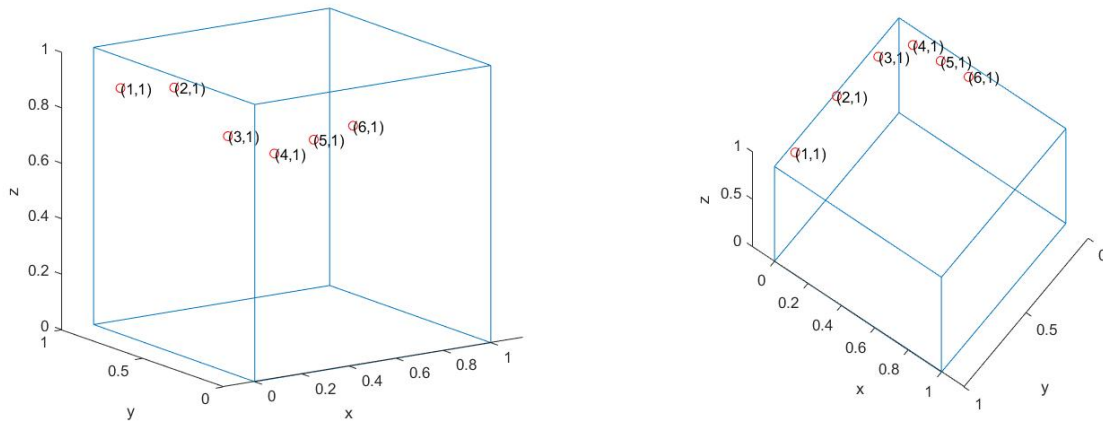


FIGURE 4.20 Alignement des points de départ selon un angle de vue quelconque et selon le plan perpendiculaire à la direction $\vec{t}$.

4.4.9 Traitement des coins

Comme nous l'avons déjà dit, nous utilisons différentes approches afin d'éviter les coins, cependant ce n'est pas toujours possible. Ici utiliser le terme de coins est un abus de langage, nous devrions plutôt utiliser la notion d'intersection entre 2 surfaces. Ce qui regroupe les coins et les arêtes des géométries 3D. Durant ma recherche la question des lignes aboutissant à un coin est revenue à plusieurs reprises et il existe plusieurs méthodes afin de la traiter. Le problème de ces lignes en tracé cyclique est l'application des conditions frontières. Elle devient complexe, car un choix entre les deux surfaces doit être fait. On peut alors résoudre le problème de différentes façons :

1. Dupliquer la ligne et diviser son poids par 2. En faisant cela, on considère avoir 2 lignes arrivant de part et d'autre du coin ce qui permet d'appliquer les conditions frontières. Cette solution a aussi ses limites, en précision notamment, en effet si la ligne provient d'un cycle cela peut créer une incohérence sur le cycle et potentiellement empêcher la ligne de revenir à son point de départ.
2. Supprimer la ligne arrivant dans un coin. En faisant cela on élude complètement le problème en ne traitant pas la ligne, cependant cette solution peut être valable uniquement à très haute densité. En effet, à basse densité cela peut créer une zone de vide dans la géométrie.
3. Effectuer un déplacement de la ligne. En choisissant un pas arbitraire, par exemple 1% du pas normal on effectue un déplacement de notre ligne.

Dans DRAGON, nous avons choisi la deuxième option, si nous rencontrons une ligne passant

dans un coin nous ne la traitons pas. Cependant, il est important de noter que nous prenons beaucoup de précautions afin d'éviter ce cas de figure. Dans DRAGON, nous avons décidé d'effectuer un déplacement de ligne si nous avons un point de départ d'un cycle dans un coin, et si un point durant le cycle aboutit à un coin nous supprimons la ligne.

4.4.10 Motif

Le motif créé par les lignes d'intégration sur la géométrie n'est pas aléatoire. Ce motif est important, car il nous permet d'invalider un tracé si le motif est incohérent. En effet, le motif effectué en 3D lorsqu'il est projeté sur le plan $X - Y$ doit être identique au motif des lignes en 2D. C'est une vérification visuelle qui peut infirmer un tracé, mais ne peut pas confirmer qu'un tracé est juste.

Nous allons montrer que pour le même angle azimutal, on peut avoir une erreur sur le cycle est ainsi créer des problèmes de cohérence si le motif n'est pas le même.

Nous pouvons voir dans la figure (4.22) une densité de lignes d'intégration irrégulière. Ce motif est donc invalide pour un tracé cyclique. En effet, pour avoir une quadrature cohérente nous devons utiliser une densité de lignes constante.

Nous avons choisi un angle azimutal fixe et les cycles en 3D sont montrés uniquement pour un seul angle polaire. Nous avons une parfaite similitude sur la figure de droite de la figure (4.22) et (4.21), ainsi la projection des lignes 3D sur un plan 2D donne le même motif que les lignes 2D. Sur le modèle invalide, nous observons des distances non constantes sur les différentes lignes transversales à la direction du neutron, c'est un signe que l'écartement a été mal choisi. Cela peut mener à des quadratures spatiales incomplètes et à des zones sans ligne d'intégration.

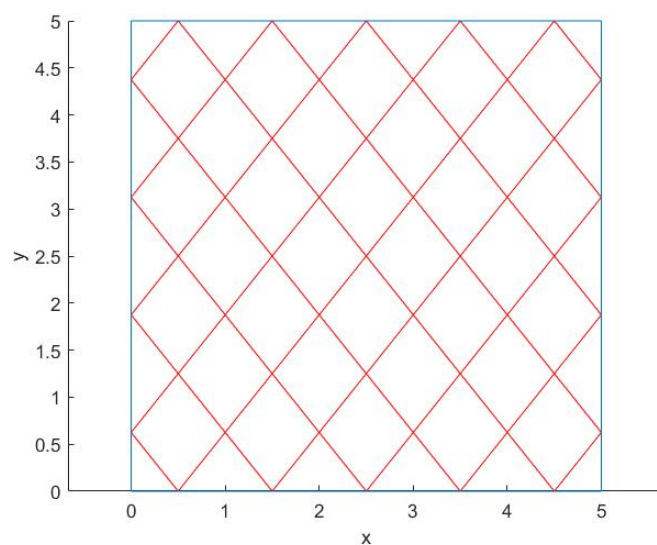


FIGURE 4.21 Représentation d'un motif en 2D pour un angle azimutal fixe.

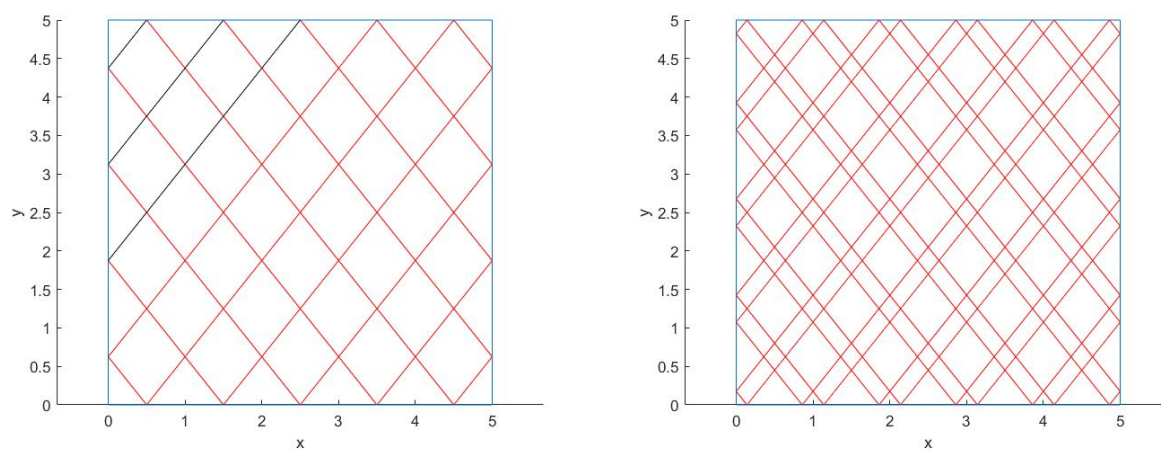


FIGURE 4.22 Motif valide (droite) et invalide (gauche) en 3D.

4.4.11 Rotation en 3D

Il existe différentes façons d'augmenter la précision de notre quadrature volumique. Elles sont toutes basées sur le même principe : augmenter le nombre de lignes. Pour augmenter le nombre de lignes, nous pouvons augmenter la densité de lignes d'intégration, c'est-à-dire demander en tant qu'utilisateur une densité plus grande afin de diminuer l'écartement comme nous l'avons vu : (4.18).

Une autre façon d'augmenter la précision de la quadrature volumique est de réaliser une rotation de notre géométrie. Sachant que les quadratures angulaires des angles azimutaux et polaires sont différentes nous n'avons pas de symétrie sur les lignes selon les différents axes. Cela est surtout marqué entre les axes X et Y et l'axe Z . En modifiant l'axe vertical par l'un des autres axes, nous aurons de nouvelles lignes d'intégration. En réalisant les trois rotations : XYZ , YZX et ZXY le critère de symétrie sera rétabli et en augmentant grandement notre nombre de lignes, nous affinerons notre précision.

4.4.12 Numérotation des angles

Nous allons préciser la numérotation des angles et des octants utilisés dans DRAGON, leurs positions sont détaillées dans la figure (4.16). Le besoin d'une numérotation des angles vient de la structure du code de DRAGON. Pour réaliser les quadratures relatives à chaque direction, nous devons associer aux angles un numéro d'index.

Nous allons respecter la numérotation suivante pour les octants :

TABLEAU 4.2 Numérotation des octants.

(+,+,+)	1
(-,+,+)	2
(-,-,+)	3
(+,-,+)	4
(+,+,-)	5
(-,+,-)	6
(-,-,-)	7
(+,-,-)	8

Pour effectuer cette numérotation, nous commençons par l'octant supérieur principal à savoir $(+, +, +)$ et nous parcourons les octants supérieurs dans le sens trigonométrique positif puis nous faisons de même pour les octants inférieurs. De plus, nous allons prendre en compte dans la numérotation les trois rotations.

Soit une quadrature de N_A angles. Nous aurons N_A^2 direction par quadrant pour une rotation. Au total nous aurons la formule (4.52) donnant le nombre total de directions en fonction de l'ordre de quadrature.

$$N_T = N_A^2 * 8 * 3 \quad (4.52)$$

Afin d'associer un angle à son numéro d'angle nous allons étudier la direction $\vec{U}$. Chaque direction peut être associée à un octant selon la numérotation vue précédemment. Pour cette association nous utilisons le signe de chaque coordonnée et nous utilisons la numérotation des octants.

Pour un angle quelconque, nous obtiendrons son numéro d'index avec la formule suivante :

$$N_I = N_\alpha + (N_R - 1)N_A^2 + 3(N_O - 1)N_A^2 \quad (4.53)$$

- N_I Le numéro de l'index de l'angle. $N_I \in [1 : N_T]$
- N_α Le numéro de l'angle parmi les angles de la quadrature. $N_\alpha \in [1 : N_A^2]$.
- N_R Numéro de la rotation $N_R \in [1, 3]$, nous associerons chaque numéro de rotation de la manière suivante : $1 \rightarrow XYZ$, $2 \rightarrow YZX$ et $3 \rightarrow ZYX$.
- N_A L'ordre de quadrature des angles.
- N_O Le numéro de l'octant comme décrit dans le tableau (4.2).

Afin d'illustrer cette méthode d'indexation nous allons utiliser une exemple. Pour une quadrature d'ordre 2. Nous aurons $N_A = 2$, $N_A^2 = 4$ et $N_T = 96$. Nous aurons le tableau (4.3) montrant la répartition des angles parmi les rotations et les octants.

TABLEAU 4.3 Indexation des angles pour une quadrature d'ordre 2.

OCTANT	XYZ				YZX				ZXY			
1	1	2	3	4	5	6	7	8	9	10	11	12
2	13	14	15	16	17	18	19	20	21	22	23	24
3	25	26	27	28	29	30	31	32	33	34	35	36
4	37	38	39	40	41	42	43	44	45	46	47	48
5	49	50	51	52	53	54	55	56	57	58	59	60
6	61	62	63	64	65	66	67	68	69	70	71	72
7	73	74	75	76	77	78	79	80	81	82	83	84
8	85	86	87	88	89	90	91	92	93	94	95	96

En pratique, nous allons dans un premier temps tracer pour une rotation les premiers angles. Ici, les angles numérotés de 1 à 4. Pour chacun de ces angles, nous obtenons leurs colonnes relatives grâce à l'application des conditions frontières. Puis nous passons à une autre rota-

tion.

Cela montre aussi le nombre d'angles en fonction de la quadrature. Pour la quadrature la plus grande admissible : 16, nous aurons en tout 6144 angles. Le nombre d'angles total est proportionnel au carré du nombre d'angles. Une augmentation d'angle augmentera la précision, mais aussi grandement le temps de calcul.

4.4.13 Algorithme

Ici nous allons vous présenter l'algorithme afin de tracer les lignes cycliques en 3D. C'est un algorithme simplifié en pseudo-code.

Algorithme Calcul des lignes cycliques en 3D

Obtenir les valeurs de la géométrie

Do $I \in \text{NANGL}$ {Boucle sur les angles azimutaux}

Obtenir m_x, m_y

Calculer $D_{m_x, m_y}, \vec{s}, \vec{u}$

Do $J \in \text{NANGL}$ {Boucle sur les angles Polaires}

Do $\text{ISCAN} \in \text{NSCAN}$ {Nombre d'octants}

Appliquer le changement de quadrant

Obtenir m_z, m_{xy}

Calculer $D_M, \vec{u}_{3D}, \vec{s}_{3D}, \vec{t}_{3D}$

Calculer $P_{\text{MIN}2D}, P_{\text{MAX}2D}, P_{\text{MIN}3D}, P_{\text{MAX}3D}$

Calculer Le point de départ du cycle

Calculer Le cycle avec tous les segments

Incrémenter le nombre de cycles N_c

IF $N_c \geq \text{ISCAN} * N_{T2D} * N_{T3D}$

EXIT

END IF

Incrémenter le point de départ

End Do

End Do

End Do

FIGURE 4.23 Algorithme en pseudo-code pour les lignes d'intégration cycliques en 3D.

4.4.14 Conclusion méthode 3D

Nous avons mis en avant notre méthode complète en 3D. D'autres méthodes existent aussi en 3D où des corrections angulaires sont apportées. Nous pouvons notamment citer : [12]. Il existe différentes méthodes afin d'obtenir le tracé des lignes d'intégration cycliques en 3D. Nous avons opté pour la méthode la plus directe. Pour résumer brièvement notre méthode en 3D, nous partons d'une méthode 2D appliquée à notre géométrie pour créer un nouveau plan dans la dimension verticale de notre géométrie. À ce plan, nous appliquons de nouveau notre méthode 2D. Cette méthode nous permet plus de simplicité et d'aisance dans le code, même si nous avons dû prendre en compte de nombreux cas particuliers et apporter de nombreux correctifs afin générer les bonnes lignes. Cette méthode apporte pour nous les meilleurs résultats en termes de complexité de mise au point et d'implémentation par rapport aux autres méthodes. De plus, c'est la méthode respectant le plus la logique globale de DRAGON afin de limiter un maximum nos modifications du logiciel.

CHAPITRE 5 VALIDATION ET VÉRIFICATION DU PROGRAMME

5.1 Le Code DRAGON

Le code DRAGON est composé de sous-routines développées en FORTRAN. Elles sont regroupées sous forme de modules, directement appelés par le programme principal. Chaque module permet d'accomplir une tâche précise au sein du calcul de réseau. L'interaction avec le logiciel DRAGON est réalisée à l'aide d'un langage à part entière : CLE-2000. Le but de ce langage est sa simplicité afin de faciliter sa lecture par tous langages de programmation [26].

Nous allons nous intéresser uniquement aux modules de tracking.

5.1.1 Les modules de Tracking

Les modules de tracking permettent à DRAGON d'effectuer l'analyse de l'espace défini par le module géométrique (GEO :). Il permet aussi de calculer les différentes grandeurs nécessaires aux méthodes de résolution de l'équation de transport. Il effectuera la numérotation des régions, le calcul des volumes, des surfaces ainsi que des lignes d'intégration. Ces opérations sont effectuées en fonction de l'opérateur choisi. Le module de tracking est subdivisé en différents opérateurs, chacun ayant une fonction précise :

- **SYBILT** : Ce module traite des géométries en 1D et la méthode à courant d'interface en 2D.
- **EXCELT** : Ce module permet d'effectuer le tracking sur une cellule complète pour les probabilités de collisions ou la méthode des caractéristiques avec un tracking isotrope ou spéculaire.
- **NXT** : Ce module est une extension du module EXCELT : avec plus de possibilités. C'est aussi le module de travail où nous implémenterons la méthode cyclique en 3D.
- **MCCGT** : Ce module est une implémentation d'une méthode des caractéristiques dite ouverte en 2D et 3D.
- **SNT** : Ce module est une implémentation d'une méthode S_n en 1D, 2D et 3D.
- **BIVACT** : Ce module est une implémentation de la théorie de la diffusion ou d'une méthode SP_n en 2D.
- **TRIVAT** : Ce module est une extension de **BIVACT** : en 3D.

Les informations de sortie de ces modules seront sous la forme d'une "tracking data structure". Ces informations sont spécifiées dans le document [24].

Nous allons nous concentrer sur le module de tracking NXT :

5.1.2 Le module NXT :

Le module NXT : est le module de tracking où la majorité des nouvelles implémentations sont effectuées. Il signifie : New Excell Tracking. Comme les autres modules, il permet l'analyse d'une géométrie ainsi que la génération des lignes d'intégration. Il a la particularité de pouvoir être appelé par n'importe quel autre module, cela lui permet une grande flexibilité. Il comporte une structure parallélisée afin de diminuer le temps de son exécution. Enfin, sa capacité à incorporer directement de nouvelles géométries ou des niveaux de géométrie le rend unique parmi les opérateurs de tracking. Cela simplifie les processus de maintenance et de mise à jour [27].

Le module NXT : contiens de nombreuses options différentes afin de faire varier les calculs et la génération des lignes d'intégration. Le tableau (5.1) tiré du guide utilisateur de DRAGON [24] présente les différentes options :

```
[ ANIS nanis ]
[ { ONEG | ALLG | XCLL } ]
[ [ QUAB iquab ] [ { SAPO | HEBE } ] ]
[ { PISO | PSPC [ CUT pcut ] } ]
[ { SYMM isymm | NOSY ]
[ { GAUS | CACA | CACB | LCMD | OPP1 | OGAU } [ nmu ] ]
[ TISO [ { EQW | GAUS | PNTN | SMS | LSN | QRN } ] nanl dens [ CORN pcorn ]
  | TSPC [ { EQW | MEDI | EQW2 } ] nanl dens ]
[ { NOTR | MC } ]
[ NBSLIN nbslin ]
[ LONG ]
[ PRIZ denspr ]
;
```

FIGURE 5.1 Structure des données d'entrée descnxt.

Les options les plus intéressantes pour notre étude sont :

- **TISO** permet un tracking isotrope.
- **TSPC** permet un tracking spéculaire.
- La quadrature angulaire permettant de définir les caractéristiques des angles que nous allons utiliser. Dans notre cas nous choisirons uniquement l’option **EQW2** afin de ne pas traiter les angles extrêmes 0 et $\pi/2$.
- **nangl**. Ce choix permet de choisir le nombre d’angles. Dans le cas 3D nous pouvons choisir **nangl**=2, 4, 8, 10, 12, 14 ou 16.
- **dens**. Ce choix permet la sélection de la densité de ligne par centimètre en 2D ou par cm^{-2} en 3D.
- **PISO**. Cette option permet d’effectuer le calcul des probabilités de collision avec des lignes d’intégration uniformes ou cycliques. Si **TISO** est sélectionnée, cette option l’est aussi par défaut
- **PSCP**. Cette option permet d’effectuer le calcul des probabilités de collision avec des lignes d’intégration spéculaires. Si **TSPC** est sélectionnée, cette option l’est aussi par défaut.

5.1.3 Implantation dans le module NXT : de DRAGON

Nous allons présenter la stratégie que nous avons suivie afin d’implémenter nos modifications dans le module **NXT** :. Comme nous l’avons déjà dit le module **NXT** : permet l’analyse des géométries ainsi que la génération des lignes d’intégration, cependant il ne pouvait pas effectuer la génération des lignes en 3D avec l’option spéculaire, ce que nous avons implémenté.

Pour implémenter cette option en plus dans **NXT** : nous avons pris la décision d’effectuer une divergence de cas. Si l’option **TSPC** est choisie en 3D, ce sont de nouvelles sous-routines qui s’exécuteront. Cette décision a été motivée par le fait que l’intrication des différentes sous-routines dans le module est complexe et provient de nombreuse évolution du code par différentes personnes. Une divergence de cas en créant de nouvelles sous-routines permettra donc une lecture globale plus simple ainsi que de permettre une maintenance et des mises à jour futur plus accessible. Les sous-routines permettant de tracer les lignes d’intégration uniformes et cycliques 2D suivent la même logique. Un programme principal **NXTTCG**, une sous-routine afin de calculer les poids spatiaux, une autre pour calculer les quadratures angulaires et les poids relatifs. Enfin, une sous-routine générant le tracking.

Comme nous pouvons le voir pour le tracking uniforme sur la figure suivante : (5.2) la chaîne d’appel des sous-routines principales avec la description de ces programmes dans le tableau

suivant (5.1). Et de même pour le tracking spéculaire en 2D avec la figure (5.3) et le tableau (5.2).

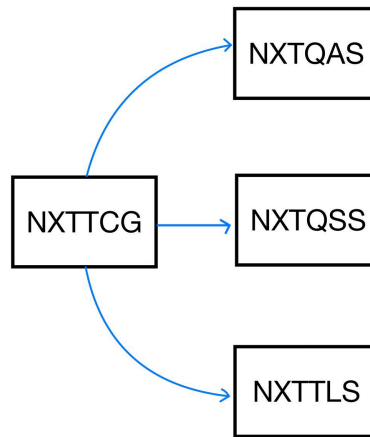


FIGURE 5.2 Structure descnxt pour les lignes d'intégration uniformes (**TISO**).

TABLEAU 5.1 Les sous-routines principales afin de générer les lignes d'intégration uniformes.

NXTQAS	Définis la quadrature angulaire pour les options données pour un tracking uniforme.
NXTQSS	Définis la quadrature spatiale pour les options données pour un tracking uniforme.
NXTTLS	Génère le tracking uniforme pour une géométrie.

TABLEAU 5.2 Les sous-routines principales afin de générer les lignes d'intégration cycliques en 2D.

NXTQAC	Définis la quadrature angulaire pour les options données pour un tracking cyclique.
NXTQSC	Définis la quadrature spatiale pour les options données pour un tracking cyclique.
NXTTLC	Génère le tracking cyclique pour une géométrie.

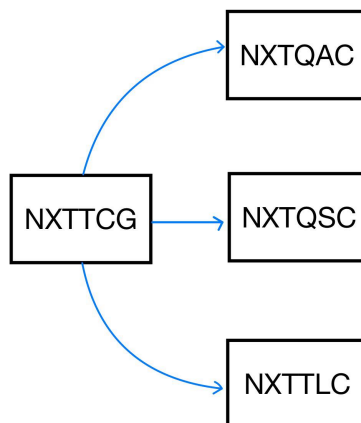


FIGURE 5.3 Structure descnxt pour les lignes d'intégration cycliques en 2D (**TSPC 2D**).

Pour la structure des sous-routines afin d'effectuer le tracking cyclique en 3D, nous avons préféré créer une sous-routine générant toutes les informations nécessaires au tracking et une sous-routine faisant le tracking. Nous mettrons en évidence la chaîne d'appel de sous-routine que nous avons mis au point pour calculer les lignes d'intégrations cycliques dans la figure (5.4) et la description des programmes dans le tableau (5.3).

TABLEAU 5.3 Les différentes sous-routines afin de générer les lignes d'intégration cyclique en 3D.

NXTTCG	Effectue le tracking sur une cellule ayant des sous-régions.
NXTTC3	Génère le tracking cyclique pour des géométries 3D.
NXTICT	Calcul toutes les informations nécessaires pour effectuer le tracking cyclique
NXTCFD	Trouve les changements de directions ou de points lors de l'application des conditions frontières de réflexion ou de translation.
NXTFNP	Trouve les points d'intersection entre une ligne et la géométrie en 3D.
NXTCLC	Trace le cycle sur la géométrie 3D.

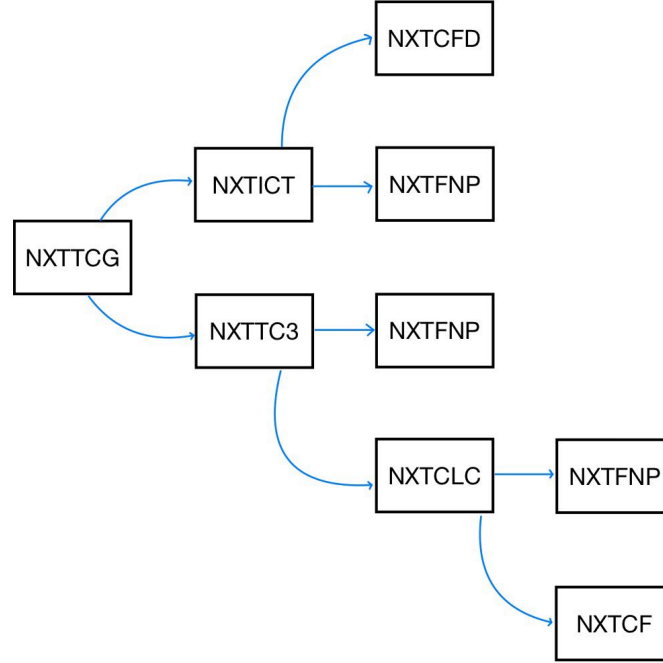


FIGURE 5.4 Structure descnxt pour les lignes d'intégration cycliques en 3D (**TSPC 3D**).

On peut voir que les sous-routines **NXTQAC** et **NXTQSC** (respectivement **NXTQAS** et **NXTQSS** en tracking uniforme) ont toutes deux été remplacées par une unique sous-routine **NXTICT** (NXT : Information Cyclic Tracking) dans notre méthode. Et la sous-routine **NXTTLC** (ou **NXTTLS** en tracking uniforme) a été remplacée par **NXTTC3**. Pour les méthodes **TISO** et **TSPC 2D** nous n'avons pas mis en évidence toutes les sous-routines appelées afin de ne pas alourdir les figures. On note que les anciennes routines n'ont pas été supprimées du code, mais nous avons introduit une divergence de cas afin d'accéder à nos sous-routines pour les options **TSPC** et **3D** dans la routine **NXTTCG**.

5.2 Vérification des programmes

Pour vérifier que nous programmes fonctionne nous allons effectuer différentes vérification analogique et numériques. Pour cela nous allons utiliser le tracé des lignes d'intégration par notre implémentation dans le module NXT : afin de calculer le volume et les surfaces de notre géométrie grâce aux équations (3.8) et (3.10) vues précédemment.

5.2.1 Vérification analogique

Dans un premier temps, nous allons vérifier notre procédé avec une méthode analogique. C'est à dire, pour vérifier la validité des quadratures sur le domaine nous allons étudier la présence des lignes d'intégration ainsi que des points permettant d'effectuer les quadratures surfaciques et volumiques. Nous nous intéresserons à leurs positions relatives aux différentes surfaces et régions.



FIGURE 5.5 Représentation des surfaces et des régions pour une cellule de réacteur REP.

Nous pouvons voir sur la figure (5.5) la représentation des différentes surfaces et des différents volumes dans une cellule de réacteur REP. Nous avons le combustible au centre, la section annulaire représente la gaine du combustible et enfin le reste du volume représente le modérateur, nous voulons effectuer la quadrature de ces volumes. Il est important de pouvoir différencier les régions afin de définir les matériaux les composant. Il est important de noter qu'une vérification analogique ne nous permet pas de valider un tracking cependant elle permet d'en invalider un.

Les surfaces

Lors du tracé des lignes d'intégration, les lignes toucheront les surfaces et nous appliquerons les conditions frontières, nous conserverons en mémoire les points d'intersection et leur surface. L'objectif est d'effectuer une quadrature grâce à ces points afin d'obtenir la valeur des différentes surfaces. Chaque point doit appartenir à une surface précise.

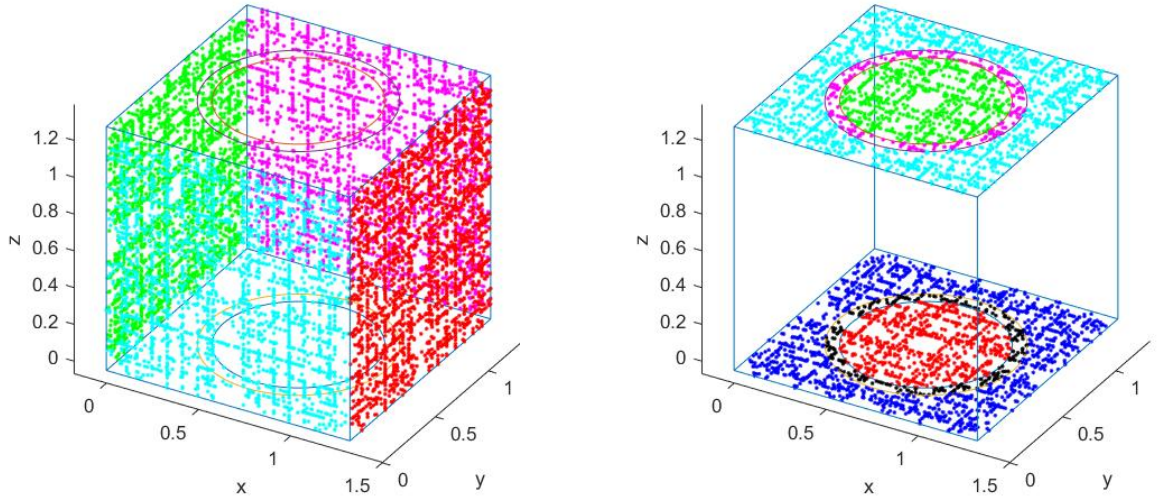


FIGURE 5.6 Représentation des quadratures sur les surfaces.

Nous pouvons voir sur la figure (5.6) la quadrature des surfaces. Chaque couleur définit une surface différente. Nous n'avons pas de points mal placés. De plus, nous pouvons voir un motif cohérent formé par les points. On ne voit pas d'anomalie sur les points. Les différentes surfaces du cube sont définies par les points et il en est de même pour les surfaces des faces Z^+ et Z^- constituées des surfaces du combustible, de la gaine et du modérateur. La surface de la gaine en Z^+ et Z^- représente la plus petite cellule ici et donc aura moins de points que les autres cellules elle présentera donc la plus grande erreur.

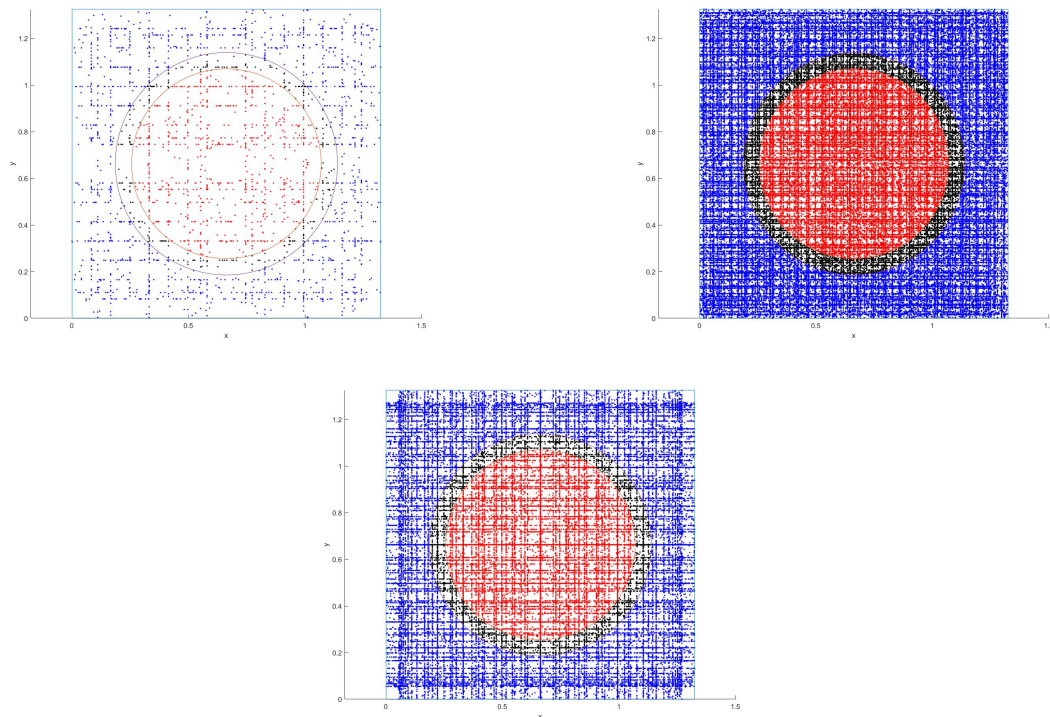


FIGURE 5.7 Projection en 2D des points d'intersection entre les lignes d'intégration et une surface Z .¹

La figure (5.7) montre l'évolution des points servant à la quadrature surfacique lors d'une augmentation de densité ou du nombre d'angle. L'augmentation de la densité de point reste cohérente, les points décrivent toujours les surfaces. De plus, chaque point est associé à la bonne surface. Cela valide dans un premier temps notre quadrature surfacique. On peut voir que l'augmentation du nombre d'angles donne plus de points par rapport à l'augmentation de la densité. En effet, lorsqu'on augmente le nombre d'angles, on augmente drastiquement le nombre de lignes, car le nombre d'angles augmente de manière quadratique.

Les figures (5.6) et (5.7) ont été obtenues grâce au module **TLM** : de DRAGON.

Les volumes

La quadrature des volumes est effectuée grâce aux lignes d'intégration. Il est donc important que chaque segment de lignes appartienne à la bonne région.

1. Pour une densité de 10 et de 1000 avec 4 angles (en haut) et une densité de 10 avec 12 angles (en bas).

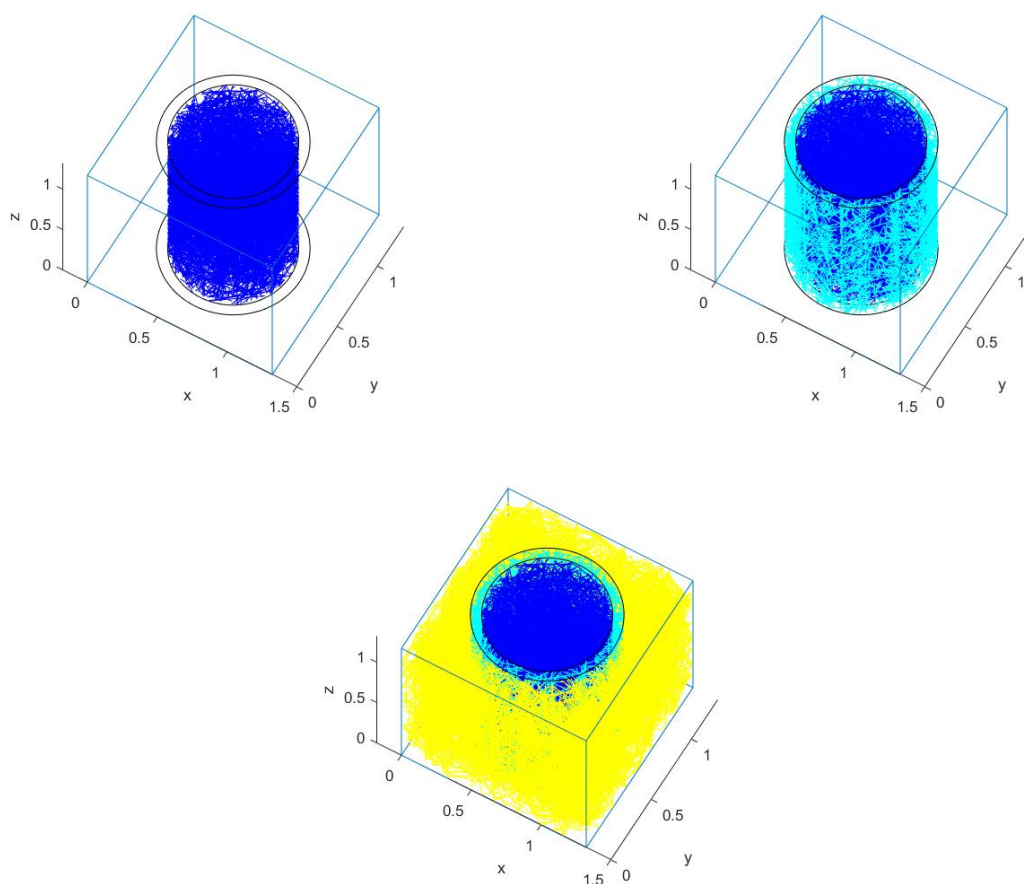


FIGURE 5.8 Segment des lignes d'intégration dans les différentes régions.

La figure (5.8) montre la quadrature de chaque volume par les lignes d'intégration pour la densité minimale. Chaque volume est décrit par une couleur différente. Pour une ligne d'intégration, ces segments peuvent passer dans différents volumes. Nous pouvons confirmer que chaque ligne est associée au volume d'après leurs couleurs et qu'il n'y a pas d'anomalie. Il est important de noter que même certains volumes vont présenter des erreurs plus grandes pour la quadrature, car si le volume est faible il y aura moins de ligne et de plus la densité de lignes sur toute la géométrie varie ce que nous pouvons notamment observer dans les coins de la géométrie.

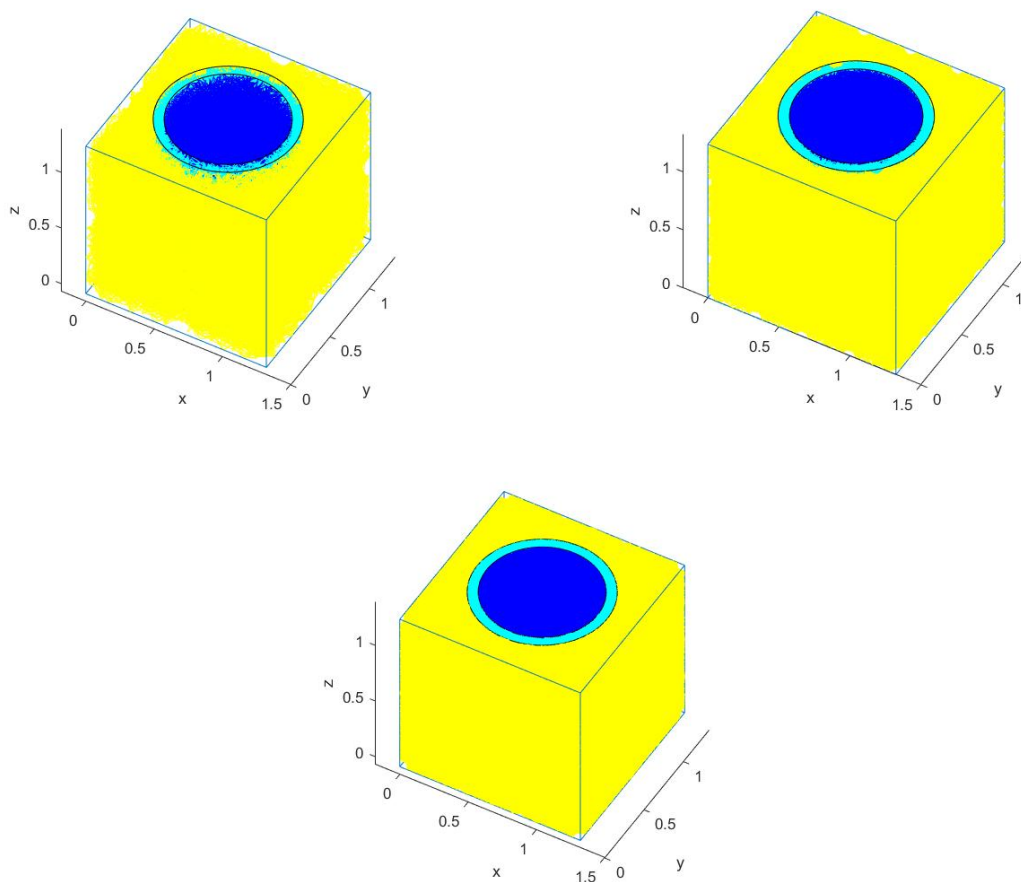


FIGURE 5.9 Lignes d'intégration dans les différentes régions.¹

Nous pouvons voir sur la figure (5.9) une augmentation cohérente de la densité de lignes dans les différentes régions. Pour l'augmentation de la densité de ligne ou bien l'augmentation du nombre d'angle nous avons une densité de lignes qui devient rapidement grande et nous ne pouvons pas voir de différence sur les figures. Cependant nous constatons que les résultats restent cohérent dans tous les circonstances.

5.2.2 Vérification numérique

La vérification numérique nous permet de confirmer la validité de nos algorithmes. Nous allons pour cela nous concentrer sur les valeurs des quadratures des surfaces, des volumes et aussi du temps de calcul. Nous effectuerons aussi les comparaisons avec la méthode uniforme. Il est important de noter qu'en 3D il est difficile de distinguer correctement les problèmes possibles avec les lignes d'intégration. En effet, en 3D il y a beaucoup plus de lignes par rapport à la 2D

1. Pour une densité de 10 et de 1000 avec 4 angles (en haut) et une densité de 10 avec 12 angles (en bas).

et se positionner selon les 3D peut être compliqué.

Les données sur les volumes et les surfaces représentent l'erreur relative par rapport à la valeur théorique en pourcentage.

Cube unité avec une région

Afin d'effectuer une vérification préliminaire nous utiliserons une géométrie simple : un cube de côté 1 centimètres. Le cube est une première simplification des cellules de réacteurs. Nous voulons mettre en évidence le calcul des lignes d'intégrations, analyser leurs générations, ainsi qu'effectuer une vérification de leur cohérence en fonction de l'évolution des paramètres. Nous nous intéresserons aux quadratures des surfaces et des volumes en fonction du nombre d'angles et de la densité de lignes. Nous analyserons enfin le temps de calcul . Pour comparer nos exécutions, nous utiliserons des tracés uniformes avec les mêmes caractéristiques.

TABLEAU 5.4 Récapitulatif des erreurs de quadrature surfaciques et volumiques pour différents angles et différentes densités avec un tracé uniforme pour référence.

	Spéculaire TSPC				Uniforme TISO		
	S(%)	V(%)	t(s)		S(%)	V(%)	t(s)
EQW4							
dens							
5	11.10394	0.00000	0.069		2.00000	0.09175	0.113
10	11.10394	0.00000	0.089		1.82879	0.05397	0.211
100	11.10394	0.00000	0.069		4.10085	0.01720	1.822
EQW12							
dens							
5	13.89265	0.00000	3.048		5.42169	0.96083	0.044
10	13.89265	0.00000	3.176		0.09709	0.36890	0.085
100	13.89265	0.00000	8.988		1.51988	0.01726	0.571
EQW16							
dens							
5	20.87132	0.00000	16.343		8.17308	1.27162	0.068
10	20.87132	0.00000	16.471		1.34228	0.28517	0.148
100	20.87132	0.00000	20.668		1.09589	0.07364	1.069
400	21.01321	0.20711	76.711		0.61300	0.02643	3.472

Le choix des différentes densités est justifiable afin d'avoir différentes expériences significatives. Une densité de 5 permet d'avoir une densité faible par rapport à la géométrie présente ici de 1 centimètre. Les densités suivantes permettent une évolution logique du nombre de lignes par centimètre. Il est important de noter que la densité de ligne évolue en fonction de $\sqrt{dens}$.

Pour les angles nous avons choisi 3 angles différents afin de voir l'implication de l'augmentation des angles dans la précision et le temps de calcul. Avec une quadrature à 16 angles représentant la plus grande quadrature que nous pouvons avoir.

Nous pouvons voir dans le tableau (5.4) l'évolution des quadratures surfaciques et volumiques en fonction des angles et de la densité comparée à un tracé uniforme.

Nous n'avons pas une évolution cohérente de notre précision lors des calculs. En effet, une augmentation de densité devrait donner une augmentation de la précision, cependant ce n'est pas ce que nous avons. Les quadratures des volumes semblent parfaites en simple précision. Cependant, les surfaces sont incohérentes, pour un nombre d'angle donné elle n'évolue pas. De plus, on peut faire que la précision diminue lors d'une grande densité, cela peut provenir de ligne supprimer dû à des coins.

Nous avons ici 6 surfaces différentes comme sur la figure (4.14). Et peut importe la surfaces nous avons la même quadrature, cela signifie que la rotation fonctionne.

La vitesse de calcul qui correspond au temps passé dans le module NXT: est cohérente. Le temps de calcul est bien proportionnel au nombre total de lignes. De plus, on remarque que lorsque le nombre d'angles augmente le temps de calcul augmente drastiquement comparé à une augmentation de densité, cela provient du fait que pour une augmentation d'angle on va tracer drastiquement plus de ligne que pour une augmentation de densité. Lorsqu'on multiplie par 2 la densité, le temps de calcul est lui aussi multiplié par 2 alors que pour un nombre d'angles 3 fois plus important le temps de calcul est multiplié par un facteur supérieur à 10, variation proportionnelle à N_A^2 comme attendu.

L'expérience effectuée pour le tableau (5.4) est une expérience où les conditions frontières sont des conditions de réflexions pures. Nous avons effectué l'expérience pour des conditions de translation pure et nous obtenons les mêmes résultats. Cela signifie que sur une géométrie simple (cube d'arête 1 centimètre) les tracés en réflexion et en translation sont équivalents pour les quadratures surfaciques et volumiques.

Maintenant nous pouvons comparer nos résultats au tracé uniforme. Le tracé uniforme est plus rapide et précis que le tracé spéculaire. La rapidité de tracé uniforme face au tracé spéculaire était quelque chose d'attendu, dû à une méthode de tracé plus complexe il est naturel que le tracé spéculaire prennent plus de temps. Il en est de même pour la précision sur les volumes et surfaces.

Cellule de réacteur REP.

Nous allons étudier dans cette partie une cellule de réacteur REP. Le but ici est d'évaluer la précision de notre programme sur de multiples régions et surfaces. Nous pouvons voir sur la figure (5.5) les 3 régions étant le combustible la gaine et le modérateur, nous aurons de plus 10 surfaces différentes. Il est important de noter que nous ne ferons pas la comparaison de chaque surface et de chaque région. Nous utiliserons la moyenne quadratique des erreurs relatives décrite dans l'équation (5.1) afin d'analyser les données. Jusque là nous n'avons pas eu besoin de l'utiliser, car nous n'avions qu'une seule région et que toutes les faces étaient identiques.

$$\bar{X}_2 = \sqrt{\sum_i^n \frac{x_i^2}{n}} \quad (5.1)$$

TABLEAU 5.5 Récapitulatif des erreurs quadratiques pour les quadrature surfaciques et volumiques pour une cellule de réacteur REP.

	Spéculaire TSPC				Uniforme TISO		
	S(%)	V(%)	t(s)		S(%)	V(%)	t(s)
EQW4							
dens							
5	30.77535	6.58202	0.076		10.88063	2.46665	0.017
10	22.91168	5.99055	0.242		6.89544	16.18464	0.027
100	12.47096	0.86291	1.098		10.54975	1.55968	0.162
EQW12							
dens							
5	31.09444	3.33955	3.332		12.47574	5.23149	0.092
10	29.18117	2.56702	3.722		6.93685	15.47690	0.147
100	15.30873	3.63753	15.955		0.46157	0.98436	1.024
1000	19.80984	0.76271	89.553		1.01647	0.49929	5.965
EQW16							
dens							
5	33.83359	3.89699	17.162		11.58618	4.06468	0.152
10	35.44459	4.22796	25.519		3.88406	16.30475	0.278
100	21.33949	2.65483	32.381		1.69089	1.05710	1.789
1000	20.43769	0.11282	223.325		0.50062	0.51918	10.580

Analyse des résultats.

Les résultats apparaissant dans les tableaux (5.4) et (5.5) mettent en évidence une quadrature de volumes cohérente pour les réacteurs REP et pour le cube d'arêtes 1 la quadrature volumique est exacte en simple précision. Les volumes pour le réacteur REP sont précis et du même ordre de grandeur que les résultats pour tracking uniforme. On remarque même une meilleure consistance des résultats, en effet, certains résultats du tracking uniforme présentent de fortes erreurs. De plus, à très haute densité de lignes et avec un grand nombre d'angles nous avons une meilleure précision sur le volume pour le tracking spéculaire. Cependant, nous avons des erreurs importantes sur la quadrature surfacique même si elle garde une évolution cohérente. Cette erreur provient du calcul utilisé pour la quadrature surfacique.

Il est important de noter que ces problèmes ne sont pas apparus sur les représentations graphiques des figures (5.7) et (5.9). Cela souligne un problème que nous avons en 3D et non en 2D : nous ne pouvons nous fier aux représentations graphiques. En effet, il est presque impossible d'utiliser avec efficacité les représentations graphiques. Le nombre de lignes devient rapidement très élevé et nous ne pouvons plus rien discerner.

Nous pouvons comparer le temps d'exécution. Il est sans équivoque très long pour le tracking spéculaire comparé au tracking uniforme.

Globalement, le tracking uniforme reste plus performant dans la quadrature volumique et surfacique. Parfois un tracking spéculaire sera plus précis, mais le temps de calcul est très important comparé à un tracking uniforme. Dans le cadre de quadrature volumique et surfacique, il faudra utiliser le tracking spéculaire afin de vérifier de résultat et créer une référence. Cependant, il est important de noter que le but d'utiliser des lignes cycliques se trouve dans la résolution de l'équation de transport des neutrons.

CHAPITRE 6 CONCLUSION

Ce projet de recherche a permis la résolution de l'équation de transport des neutrons en trois dimensions avec des lignes d'intégration cycliques dans DRAGON. Nous détaillerons les modifications que nous avons apportées au code DRAGON ainsi que les différentes simulations afin de les vérifier. Nous verrons aussi les limites de notre solution et nous ouvrirons sur les potentielles améliorations futures.

6.1 Synthèse des travaux

Cette recherche a été motivée par l'absence de méthode cyclique en 3D dans DRAGON. En effet jusqu'à ma recherche DRAGON pouvaient analyser des géométries en 3D uniquement grâce à des lignes d'intégration uniforme. La méthode cyclique permettant une meilleure prise en compte des conditions frontières donnerait une meilleure précision sur des géométries comme celle du réacteur CANDU.

Afin de réaliser la mise en place d'une méthode cyclique en 3D, nous avons dû :

- Créer une méthode afin de tracer les lignes d'intégration cycliques en 3D.
- Implémenter la nouvelle méthode dans DRAGON.
- Effectuer une vérification de la méthode avec des simulations.

Dans le but de créer une nouvelle méthode en 3D nous nous sommes inspiré de la méthode déjà existante en 2D, programmée dans DRAGON. Nous avons pour cela appliqué la méthode 2D à deux reprises pour former des lignes d'intégration en 3D. Nous avons dû prendre la gestion de deux angles, l'angle azimutal φ et l'angle polaire θ et leur quadrature respective. Ainsi qu'un nouveau poids en 3D Δ_{3D} qui permet de calculer, avec le poids 2D les volumes. Ainsi, nous pouvons réaliser les quadratures volumiques des différentes géométries. La 3D présente aussi plus de directions qu'en 2D. Nous devons prendre en considération 4 directions, celle dans les octants supérieurs. DRAGON permettra de calculer automatiquement les octants inférieurs par symétrie. De plus, nous devons toujours prendre en compte les différentes conditions frontières définissant notre géométrie. Nous avons rencontré des conditions frontières de translation et de réflexion applicables sur toutes les surfaces des géométries en 3D. La condition permettant d'arrêter le traçage des cycles est inspirée de la condition en 2D. Cependant, elle prend en compte le nombre de cycles et non le nombre segments, car il n'est pas constant en 3D. De plus, l'incrémentation selon les deux directions perpendiculaires à la direction des neutrons permettant aux lignes d'intégration de recouvrir l'ensemble de la géométrie uniformément en commençant l'incrémentation comme en 2D et ensuite en

rajoutant la nouvelle direction.

Pour implémenter notre nouvelle méthode, nous avons pris la décision de créer de nouveaux programmes à la place de modifier les programmes actuels. Ces nouveaux programmes permettront une plus grande flexibilité. Cela nous permettra aussi de mettre à jour plus aisément les programmes dans l'avenir. Nous avons créé des programmes pour les différentes actions nécessaires au tracé des lignes d'intégrations. De manières globales, nous avons une sous-routine permettant le calcul de toutes les informations nécessaires au tracé cyclique **NX-TICT** (NXT Information cyclic tracking). Et une deuxième sous-routine afin d'exploiter ces résultats : **NXTTC3**. Comme nous pouvons le voir dans la figure (5.4) nous avons aussi créé d'autres sous-routines effectuant des tâches plus simples, elles sont décrites dans le tableau (5.3).

Afin de vérifier l'implémentation des sous-routines ainsi que de la validité de notre méthode, nous avons simulé différents cas afin de voir différentes situations ainsi que leurs résultats. Nous avons dans un premier temps effectué une simulation simple avec un cube de dimensions 1 puis nous voulions voir l'effet des lignes d'intégration cyclique dans une géométrie avec plusieurs régions la cellule de réacteur REP comme nous pouvons voir dans la figure (5.5). Ces calculs ont été effectués afin de vérifier les quadratures surfaciques et volumiques et leur évolution lors d'une augmentation de densité ou/et d'angles.

Les résultats obtenus pour les quadratures des surfaces et des volumes sont concluants pour la cellule REP. Même si nous avons des incohérences sur le calcul des surfaces.

6.2 Limitations de la solution proposée

Comme nous l'avons vu dans la partie précédente, nous avons des résultats incohérents pour le calcul des surfaces. Nos résultats montrent une augmentation de la précision lors d'une augmentation du nombre de lignes, cependant l'erreur sur les surfaces est trop importante. L'origine de cette de cette provient très probablement d'une erreur de calcul pour la quadrature surfacique.

Pour l'instant, nous n'avons pas réussi à utiliser les lignes d'intégration cyclique afin de résoudre l'équation de transport des neutrons. Nous avons rencontré des difficultés afin d'utiliser les lignes dans ce contexte.

6.3 Améliorations futures

Les améliorations futures devront se concentrer dans un premier temps sur la quadrature surfacique afin d'améliorer sa précision.

L'utilisation des lignes d'intégrations cycliques pour des quadratures volumiques et surfaciques reste une utilisation limitée. En effet, le but de cette méthode de tracé est l'application des conditions frontières aux tracking, ce qui permet en théorie une meilleure précision pour le calcul des flux. Les calculs des quadratures surfaciques et volumiques sont des vérifications de la méthode, nous n'avons pas eu le temps nécessaire afin de tester la résolution de l'équation de transport des neutrons sur une géométrie où les conditions frontières jouent un rôle important, cela représente une des améliorations futures. Dans un premier temps, avec une méthode des probabilités de collision. Les calculs seront à effectuer avec les options (**TSCP/PISO**) donnant un tracking cyclique, mais les lignes cycliques seront considérées comme uniformes dans le calcul des flux. Et enfin l'option (**TSCP/PSPC**) pour un tracking cyclique et un calcul des flux avec des lignes cycliques. Dans un second temps, il faudra essayer les calculs avec une autre de résolutions comme la méthode MOC.

Enfin, une amélioration des algorithmes utilisés et créés est envisageable afin de réduire le temps de calcul de la méthode cyclique.

RÉFÉRENCES

- [1] F. Spengler, B. Accoyer et G. Maillet, *Une affaire d'état*. Hugo Document.
- [2] F. Lindsay et J. Parsons, "Why aren't we looking at more hydropower ?" 2021. [En ligne]. Disponible : <https://climate.mit.edu/ask-mit/why-arent-we-looking-more-hydropower>
- [3] "Economics of nuclear power," *World Nuclear Association*, 2022.
- [4] A. Yamamoto *et al.*, "Benchmark problem suite for reactor physics study of LWR next generation fuels," *Journal of Nuclear Science and Technology*, 2002.
- [5] D. Duverney, "Calcul différentiel et intégral pour la physique." [En ligne]. Disponible : <http://touteslesmaths.fr>
- [6] H. Boumaza, "Théorie des distributions," *Université Sorbonne Paris Nord*.
- [7] A. Hébert, *Applied reactor physics*. Polytechnique Montréal, 2016.
- [8] U. L. TENNIER, "Couplage thermohydraulique-neutronique (CATHENA3 DON-JON5) pour l'analyse de sûreté du SCWR canadien," 2021.
- [9] A. Hebert, "Acceleration of step and linear discontinuous schemes for the method of characteristics in DRAGON5," 2017.
- [10] CEA, "Neutronics," 2015, Commissariat à l'énergie atomique et aux énergies alternatives.
- [11] B. D. Rodriguez, M. T. Vilhena et B. E. Bodmann, "An overview of the boltzmann transport equation solution for neutrons, photons and electrons in cartesian geometry," vol. 53, n°. 8, p. 1119–1125, 2009.
- [12] Z. Liu *et al.*, "A new three-dimensional method of characteristics for the neutron transport calculation," 2010, *Annals of Nuclear Energy*.
- [13] M. Porhemmat, K. Hadad et M. Mahzoon, "Modular ray tracing in 2D whole core transport with MOC," vol. 99, p. 103–109.
- [14] M. Lahdour *et al.*, "The discrete ordinate method for angular flux calculations in slab geometry," vol. 5, n°. 8.
- [15] B. Carlson, "Solution of the transport equation by SN approximation," 1955, United States Atomic Energy Commission.
- [16] M. DeCross, M. Thum et J. Khim, "Spherical harmonics." [En ligne]. Disponible : <https://brilliant.org/wiki/spherical-harmonics/>

- [17] E.M.Gelbard, “Applications of the simplified spherical harmonics equations in spherical geometry,” *Bettis Atomic Power Laboratory*, 1960.
- [18] A. M. G. Cox *et al.*, “Monte-carlo methods for the neutron transport equation,” *SIAM/ASA Journal on Uncertainty Quantification*, p. 775–825, 2020.
- [19] M. Musongela, “Généralisation du modèle TIBÈRE pour la simulation des fuites B1 hétérogènes à la méthode des caractéristiques,” Thèse de doctorat, Polytechnique Montréal, 2019.
- [20] G. Marleau, “Dragon theory manual part 1 : Collision probability calculations,” 2001, Institut de génie nucléaire Département de génie physique École Polytechnique de Montréal.
- [21] R. Sanchez, L. Mao et S. Santandrea, “Treatment of boundary conditions in trajectory-based deterministic transport methods,” *Nuclear Science and Engineering*, vol. 140, n^o. 1, p. 23–50, 2002.
- [22] J. G. Hosking, B. A. Lindley et P. J. Smith, “Cactusot : A 3D method of characteristics solver in WIMS,” *International Conference on Mathematics Computational Methods Applied to Nuclear Science Engineering*, Jeju, Korea, 2017.
- [23] S. Shaner *et al.*, “Theoretical analysis of track generation in 3D method of characteristics,” 2015, Department of Nuclear Science Engineering Massachusetts Institute of Technology.
- [24] G. Marleau, A. Hebert et R. Roy, “A user guide for dragon version5,” 2015, Institut de génie nucléaire Département de génie mécanique École Polytechnique de Montréal.
- [25] G. Marleau, “Cyclic tracking for 3D geometries in DRAGON,” 2016, Technical report École Polytechnique de Montréal.
- [26] R. Roy, “The CLE-2000 tool-box,” 1999, École Polytechnique de Montréal.
- [27] G. Marleau, “New geometries processing in DRAGON : The NXT : Module,” 2005, Institut de génie nucléaire Département de génie physique École Polytechnique de Montréal.