

Titre: Développement dans DRAGON d'un module pour l'analyse du
Title: combustible compact des réacteurs prismatiques

Auteur: Marc-André Lajoie
Author:

Date: 2010

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Lajoie, M.-A. (2010). Développement dans DRAGON d'un module pour l'analyse
Citation: du combustible compact des réacteurs prismatiques [Mémoire de maîtrise, École
Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/249/>

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

URL de PolyPublie: <https://publications.polymtl.ca/249/>
PolyPublie URL:

**Directeurs de
recherche:** Guy Marleau
Advisors:

Programme: Génie énergétique
Program:

UNIVERSITÉ DE MONTRÉAL

DÉVELOPPEMENT DANS DRAGON D'UN MODULE POUR L'ANALYSE DU
COMBUSTIBLE COMPACT DES RÉACTEURS PRISMATIQUES

MARC-ANDRÉ LAJOIE
DÉPARTEMENT DE GÉNIE PHYSIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

MÉMOIRE PRÉSENTÉ EN VUE DE L'OBTENTION DU DIPLÔME DE
MAÎTRISE ÈS SCIENCES APPLIQUÉES
(GÉNIE ÉNERGÉTIQUE)
MARS 2010

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Ce mémoire intitulé :

DÉVELOPPEMENT DANS DRAGON D'UN MODULE POUR L'ANALYSE DU
COMBUSTIBLE COMPACT DES RÉACTEURS PRISMATIQUES

présenté par : M. LAJOIE Marc-André

en vue de l'obtention du diplôme de : Maîtrise ès sciences appliquées

a été dûment accepté par le jury constitué de :

M. TEYSSEDOU Alberto, Ph.D., président.

M. MARLEAU Guy, Ph.D., membre et directeur de recherche.

M. HÉBERT Alain, D.Ing., membre.

à ma grand-mère, Thérèse

Remerciements

J'aimerais tout d'abord remercier le professeur Guy Marleau, de m'avoir accepté comme étudiant et soutenu tout au long du projet par ses conseils et son aide précieuse sur la méthodologie à suivre. Sans son aide et son soutien, ce mémoire n'aurait jamais pu voir le jour.

Je tiens à remercier aussi Élisabeth Varin, Gilles Sabourin et Michel St-Denis et toute l'équipe d'ÉACL Montréal, ainsi que Marc-Antoine Petrilli, de Nucléonex Inc., qui, en m'offrant des stages passionnants tout au long de ce projet, m'ont ouvert les yeux sur le travail fait dans l'industrie nucléaire ici même au Québec.

Merci aussi à tous les membres anciens et actuels de l'IGN que j'ai pu croiser durant mon séjour, pour une ambiance de travail agréable qui m'a permis de réaliser ce travail avec beaucoup plaisir.

Je tiens à offrir un merci tout spécial à Nicolas Martin-Legendre, étudiant au doctorat à l'IGN qui, en plus de ses réflexions et de ses conseils toujours pertinents, m'a apporté une aide incommensurable concernant la validation de mes résultats.

Merci finalement à Émilie et aux membres de ma famille, ma mère Murielle, mes frères Daniel et Pascal-Julien, qui m'ont épaulé et encouragé à persévérer malgré les difficultés.

Résumé

Les réacteurs VHTR de génération IV présentent des caractéristiques qui sont difficilement analysables par les méthodes disponibles dans les codes de calcul déterministes traditionnels. Parmi ces particularités, mentionnons la disposition aléatoire d'éléments de combustible dans la géométrie, en plus d'incorporer pour ces mêmes éléments, des géométries sphériques.

Ce projet a pour but d'effectuer l'intégration dans le module `NXT`: du code de calcul de cellule DRAGON de fonctionnalités permettant l'analyse de géométries sphériques à plusieurs couches, générés à des positions aléatoires dans une cellule cartésienne contenant un maillage selon les trois axes. Une analyse géométrique complète des sphères nous a permis d'implanter le calcul des volumes analytiques, ainsi que le calcul des distances entre les points d'intersection des lignes d'intégration avec les éléments de géométrie sphériques. Une analyse exhaustive des combinaisons de maillages cartésiens et sphériques a aussi été effectuée, permettant une vérification des longueurs et des volumes calculés par les nouvelles routines.

De plus, une fonctionnalité permettant la génération aléatoire des positions des éléments de géométrie sphériques à l'intérieur d'une cellule a été incorporée, de manière à offrir une méthode alternative aux approches généralement utilisées pour traiter les problèmes présentant des caractéristiques de double hétérogénéité, pour éventuellement permettre des comparaisons avec ces mêmes modèles.

En dernier lieu, des tests de validation préliminaires ont aussi été effectués avec des *benchmarks* établis à l'aide du code Monte-Carlo Serpent et de deux méthodes de traitement de la double-hétérogénéité déjà implantées dans DRAGON. Les nouvelles capacités de `NXT`: ont été testées avec une discrétisation spatiale et angulaire de plus en plus raffinée, sur des géométries sphériques simples. Les résultats tendent à s'approcher des résultats de référence Monte-Carlo conséquemment au raffinement de la discrétisation spatiale, pour une différence finale en deçà de 0.5 mk.

Abstract

Fourth generation very high temperature reactors (VHTRs) present characteristics that tend to be harder than conventional geometries to analyze using conventional methods available in traditional deterministic neutronics computer codes. Worthy of interest in the scope of this document are the random character of fuel elements in a uniform matrix, and the spherical geometry of these same fuel elements.

The goal of this project was to integrate the capacity to analyze such properties in module `NXT`: of the transport lattice code DRAGON. A geometrical analysis of spherical volumes have allowed us to develop analytical volumes calculation, as well as tracking and intersection distances with spherical geometry elements. Routines enabling this exact geometrical analysis of multi-layered, randomly disposed spherical particles have been successfully incorporated and tested in this module. A thorough analysis of combinations of spherical and Cartesian meshes has also been performed, to ensure the validity of the lengths and volumes calculated by the new routines.

A new option has also been incorporated, allowing the program to randomly generate pin positions until the density specified by the user has been reached, offering an alternative to methods traditionally used to treat problems presenting characteristics of double-heterogeneity, and eventually permitting comparisons of results with these models.

Finally, preliminary validation runs have been carried out, comparing results obtained using the new module on a simple spherical geometry with a benchmark established using Monte-Carlo code Serpent and previously implemented double heterogeneity treatment in DRAGON. The new `NXT`: capabilities have been tested with an increasing angular and spatial discretization, leading to results approaching those generated by the Monte-Carlo method, for a difference inferior to 0.5 mk.

Table des matières

Dédicace	iii
Remerciements	iv
Résumé	v
Abstract	vi
Table des matières	vii
Liste des tableaux	ix
Liste des figures	x
Liste des annexes	xi
Chapitre 1 INTRODUCTION	1
1.1 Réacteurs actuels	1
1.2 Réacteurs GEN-IV	2
1.3 Problèmes rencontrés	3
1.4 Structure du document	4
Chapitre 2 RÉACTEURS À LIT DE BOULETS ET PRISMATIQUES	5
2.1 Réacteurs VHTR de quatrième génération	5
2.1.1 Designs	6
2.1.2 Propriétés	6
2.2 Structure du réacteur	9
2.2.1 Réacteurs à lit de boulets	9
2.2.2 Réacteurs prismatiques	10
2.2.3 Refroidissement	11
2.3 Besoins de modélisation	12
2.3.1 Solution proposée	13
Chapitre 3 ÉQUATIONS DE NEUTRONIQUE	15
3.1 Dérivation de l'équation de transport	15

3.1.1	Le terme source	17
3.1.2	Discrétisation multigroupe	18
3.2	Résolution de l'équation de transport	19
3.2.1	Probabilités de collision	21
3.2.2	Méthode des caractéristiques	24
3.2.3	Méthodes Monte-Carlo	27
Chapitre 4	ANALYSE GÉOMÉTRIQUE	28
4.1	Analyse des sphères	28
4.1.1	Pas d'intersection des plans à l'intérieur de la sphère	30
4.1.2	Intersection de deux des trois plans à l'intérieur de la sphère	31
4.1.3	Intersection des trois plans à l'intérieur de la sphère	34
4.2	Distribution spatiale des sphères	39
4.3	Lignes d'intégration	41
4.4	Implémentation	42
Chapitre 5	RÉSULTATS	43
5.1	Utilisation des géométries sphériques	43
5.2	Vérification des volumes analytiques	44
5.2.1	Cellule unitaire	44
5.2.2	Maillage cartésien dans la cellule	45
5.3	Vérification du <i>tracking</i>	47
5.3.1	Vérification des longueurs des lignes	47
5.3.2	Vérification de l'emplacement des lignes à l'aide de TLM :	49
5.3.3	Vérification des volumes numériques	49
5.4	Comparaison cellule à 8 pins	52
5.5	Génération aléatoire	54
5.6	Comparaison avec le <i>benchmark</i> HTGR	55
5.6.1	Comparaison avec les calculs Monte-Carlo	56
5.6.2	Comparaison avec les modèles de double-hétérogénéité	57
Chapitre 6	CONCLUSION	58
Références	60
Annexes	63

Liste des tableaux

TABLEAU 4.1	Volumes selon la position des plans - Aucune intersection dans la sphère	32
TABLEAU 4.2	Volumes selon la position des plans - Une seule intersection de deux plans dans la sphère	35
TABLEAU 4.3	Volumes selon la position des plans - Deux intersections de deux plans à l'intérieur de la sphère	36
TABLEAU 4.4	Volumes selon la position des plans - Trois intersections de deux plans à l'intérieur de la sphère	37
TABLEAU 5.1	Volumes calculés analytiquement par NXT :	45
TABLEAU 5.2	Maillage pour vérification des volumes d'intersection	45
TABLEAU 5.3	Volumes calculés analytiquement par NXT : (Cellule $3 \times 3 \times 3$)	46
TABLEAU 5.4	Propriétés des lignes de l'échantillon	47
TABLEAU 5.5	Échantillon de lignes d'intégration	48
TABLEAU 5.6	Volumes calculés numériquement par NXT :	51
TABLEAU 5.7	Test de cellule à 8 <i>pins</i>	54
TABLEAU 5.8	Description des pastilles TRISO utilisées pour le <i>benchmark</i>	56
TABLEAU 5.9	Influence de la discrétisation angulaire et spatiale vs. SERPENT	57
TABLEAU 5.10	Comparaison des constantes par groupe énergétique	57
Table C.1	Comparison between analytical and numerical volumes using an equal weight quadrature	97
Table C.2	Influence of the phase-space discretization for a full 3D spherical CP method	99
Table C.3	Group constant comparison	100

Liste des figures

FIGURE 2.1	Coupe d'une bille TRISO	7
FIGURE 2.2	Élément de combustible : Boulet	9
FIGURE 2.3	Élément de combustible : Bloc de graphite	10
FIGURE 2.4	Vue en coupe d'un cœur de réacteur prismatique	11
FIGURE 4.1	Analogie 2-D de la méthode de calcul des volumes	29
FIGURE 4.2	Intersection d'un plan avec la sphère	31
FIGURE 4.3	Volume à intégrer	33
FIGURE 4.4	Intersection de deux plans à l'intérieur de la sphère	35
FIGURE 4.5	Deux intersections de deux plans à l'intérieur de la sphère	36
FIGURE 4.6	Trois intersections de deux plans à l'intérieur de la sphère	37
FIGURE 4.7	Intersection des trois plans à l'intérieur de la sphère	38
FIGURE 4.8	Volumes découpés par l'intersection des trois plans à l'intérieur de la sphère	38
FIGURE 4.9	Intersection de deux sphères	40
FIGURE 5.1	Visualisation du <i>tracking</i> à l'aide du module TLM :	50
FIGURE 5.2	Intersection du <i>tracking</i> avec des plans à intervalles réguliers	50
FIGURE 5.3	Cellule de test 3-D avec <i>pins</i> cartésiennes et sphériques	53
FIGURE 5.4	Cellule de test 2-D avec <i>pins</i> cartésiennes et cylindriques	53
FIGURE 5.5	Pins générées aléatoirement pour des densités $d_{pin} = 0.002$ et $d_{pin} = 0.2$	55
Figure C.1	Intersection of a plane with a sphere	93
Figure C.2	Region formed by the intersection of two planes inside the sphere	93
Figure C.3	Volumes created by the intersection of three planes inside the sphere	95
Figure C.4	Tracking visualization with TLM: module	97
Figure C.5	Intersection of tracking lines with regularly spaced planes for one region	97

Liste des annexes

ANNEXE A	Code source	63
A.1	Routine NXTIRS	63
A.2	Routine NXTISS	69
A.3	Routine NXTVCS	72
A.4	Routine NXTLSP	76
A.5	Routine NXTPRD	84
ANNEXE B	Guide d'utilisation pour les géométries sphériques	86
ANNEXE C	Article : Application of 3D CP Theory to VHTR Spherical Geometries	88
C.1	ABSTRACT	88
C.2	INTRODUCTION	88
C.3	THEORY	89
C.3.1	The Collision Probability Method	89
C.3.2	Tracking	90
C.4	GEOMETRICAL ANALYSIS	91
C.4.1	Spherical 3-D Tracking	91
C.4.2	Volumes	92
C.5	VERIFICATION	96
C.5.1	Tracking	96
C.5.2	Volumes	96
C.6	INFINITE GRAIN LATTICE CALCULATION	98
C.6.1	Validation Against Monte-Carlo Calculations	98
C.6.2	Comparison With the Double-Heterogeneity Model	99
C.7	CONCLUSIONS	100

Chapitre 1

INTRODUCTION

L'électricité d'origine nucléaire a été, depuis son tout début, une source d'énergie controversée, notamment de la part de groupes écologistes et environnementalistes. Les principales critiques découlent des risques qui lui sont associés et de la problématique du combustible usé. Le travail effectué dans le cadre de ce mémoire est en lien avec l'une des alternatives viables aux réacteurs de la génération actuelle, les réacteurs prismatiques à très haute température, qui tentent d'apporter une solution à ces problèmes, tout en présentant des avantages supplémentaires notoires.

1.1 Réacteurs actuels

Les réacteurs présentement en opération peuvent habituellement être catégorisés en trois grandes classes, soient les réacteurs à eau bouillante (*Boiling Water Reactors* – BWR), les réacteurs à eau pressurisée (*Pressurised Water Reactors* – PWR), et les réacteurs pressurisée à eau lourde (*Pressurised Heavy Water Reactors* – PHWR). Tous ces types de réacteurs présentent les avantages de reposer sur une technologie déjà rodée et éprouvée, de pouvoir s'appuyer sur une expertise développée au cours des années, tout en permettant d'atteindre une rentabilité économique acceptable.

Cependant, ces technologies comportent toutes le désavantage de n'utiliser qu'une faible partie du combustible nucléaire. Avec des taux de combustion comparativement faibles, les réacteurs à eau lourde permettent à la fois de réduire les coûts du combustible, en utilisant de l'uranium naturel, en plus d'éliminer la production d'uranium appauvri. Les réacteurs à eau légère (PWR et BWR) utiliseront quant à eux de l'uranium légèrement enrichi, parfois mélangé avec du combustible usé retraité, atteindront des taux de combustion plus élevés. Peu importe le type de réacteur, le combustible usé contiendra encore une majorité de matériel qui n'aura subi aucune altération par son passage dans le réacteur.

Ce combustible usé mène à la production d'une quantité non négligeable de déchets radioactifs. L'intérêt d'obtenir des taux de combustion plus élevés réside en une utilisation plus

efficace des ressources en uranium, ainsi qu'une réduction du volume des déchets. Comme tous les réacteurs thermiques, les réacteurs de la génération actuelle sont limités par l'efficacité de la turbine, et, pour la plupart, d'un générateur de vapeur, ce qui a pour résultat que la plus grande partie de l'énergie produite par le réacteur n'est pas utilisée et est dissipée sous forme de rayonnement thermique.

1.2 Réacteurs GEN-IV

Une alternative se présente à plus ou moins long terme, qui permettrait de combler une partie des lacunes de ces réacteurs. Il s'agit des réacteurs de IV^e génération. L'idée est née d'un forum (GEN-IV, pour Génération 4) qui a retenu six principaux concepts pour des réacteurs d'avenir, incluant une fermeture du cycle du combustible, une meilleure utilisation du combustible nucléaire, et une plus grande résistance à la prolifération des déchets nucléaires.

Les six concepts se devaient de respecter les contraintes suivantes :

- Compétitivité économique, notamment par la réduction de l'investissement initial en capital ;
- Accroissement de la sécurité, tant dans l'opération du réacteur, en situation d'accident, et concernant les risques de prolifération ;
- Impact minimal sur l'environnement :
 - ◇ par une utilisation efficace du combustible ;
 - ◇ par une réduction appréciable des déchets à longue vie ;
- Capacité d'utilisation pour d'autres fins que la production d'électricité (ex : dessalement de l'eau de mer, production d'hydrogène, etc.).

On peut classer les six idées retenues par le forum en deux catégories, selon qu'ils utilisent des neutrons rapides ou thermiques. Les réacteurs à neutrons thermiques comprennent les réacteurs à très haute température, les réacteurs à eau surcritique et les réacteurs à sel fondu, alors que les réacteurs à neutrons rapides sont les réacteurs refroidis au plomb, au sodium ou au gaz.

Les designs envisagés pour ces nouveaux types de réacteurs sont parfois très proches de ceux des réacteurs existants, par exemple le réacteur à eau surcritique qui se rapproche beaucoup des réacteurs CANDU, alors que certains autres modèles comprennent des technologies complètement différentes. Dans le cadre de ce mémoire, les réacteurs à très haute température

(*Very High Temperature Reactors* – VHTRs) sont particulièrement dignes d'intérêt, en raison de particularités de conception qui seront explorées dans la section suivante, ainsi que plus en détails dans le premier chapitre.

1.3 Problèmes rencontrés

Les réacteurs à très haute température sont caractérisés par des éléments de conception différant grandement des réacteurs actuels. Notons entre autres le fait que le combustible n'est pas, comme dans les réacteurs actuels, réparti uniformément dans des régions homogènes, mais est plutôt constitué de petits grains (particules TRISO), disposés aléatoirement dans une matrice uniforme de graphite, elle-même formant une partie seulement du réacteur complet. Un autre niveau de difficulté vient s'ajouter lorsqu'on tient compte du fait que ces particules, sont non seulement sphériques, mais aussi composées de plusieurs couches imbriquées, formant sur l'ensemble du réacteur une géométrie très fortement hétérogène.

Les codes de calcul déterministes développés pour évaluer la population de neutrons dans un réacteur nucléaires supposent généralement un schéma de calcul à deux niveaux. On résout en premier lieu numériquement l'équation de transport de Boltzmann sur une section représentative (cellule) du réacteur complet, en supposant généralement une répétition de cette cellule à l'infini, pour ensuite créer un mélange homogène aux propriétés équivalentes. Ce dernier pourra par la suite être utilisé en remplaçant l'opérateur de transport par celui de diffusion (approximation de diffusion) pour résoudre le cœur complet du réacteur.

Les géométries très fortement hétérogènes des réacteurs VHTR viennent grandement compliquer la résolution de l'équation de transport, sans compter que les codes actuels nécessitent généralement une géométrie explicitement définie, ajoutant un niveau de difficulté supplémentaire à l'analyse de particules stochastiquement distribuées. Des méthodes alternatives ont donc été développées pour analyser ces géométries et générer des propriétés équivalentes, permettant de simplifier grandement la résolution de l'équation de transport. Dans ce contexte, la méthode proposée ici s'articule autour de deux points. Il s'agit dans un premier lieu de générer aléatoirement les positions des éléments de géométrie sphériques, afin de simuler la distribution stochastique obtenue lors de la fabrication, puis d'analyser à l'aide d'une méthode connue la géométrie globale ainsi obtenue.

1.4 Structure du document

Nous commencerons donc, dans le premier chapitre, par analyser plus en détails les réacteurs à très haute température, afin d'établir leur utilité et leur potentiel, et surtout les caractéristiques qui rendent l'analyse suivante pertinente. En plus de présenter les grandes lignes de ce type de réacteurs, nous exposerons donc aussi les besoins de modélisation particuliers, ainsi qu'un bref aperçu de certaines méthodes employées.

En second lieu, nous présenterons les bases des équations de neutronique, ainsi que quelques méthodes de résolution permettant de déterminer les quantités d'intérêt. En particulier, nous tenterons de mettre en valeur la procédure dite de *tracking*, qui est nécessaire pour la résolution du flux selon la méthode employée.

Puis, dans le troisième chapitre, nous exposerons l'analyse mathématique et géométrique nécessaire à l'implémentation de la méthode. Celle-ci se divise en deux sections : une première concernant un calcul de volumes analytiques, plus particulièrement l'intersection d'une sphère avec une région cartésienne, et une seconde qui, bien que mathématiquement simple, représente le cœur de l'analyse, impliquant les intersections d'une droite et d'une sphère.

Finalement, nous présenterons les résultats de la vérification effectuée pour s'assurer de la validité des données de sortie des calculs. Des tests préliminaires de validation sur un *benchmark* tri-dimensionnel sphérique simple ont aussi été effectués, dont nous exposerons les résultats.

Chapitre 2

RÉACTEURS À LIT DE BOULETS ET PRISMATIQUES

Afin de tenter d'orienter l'avenir de l'industrie nucléaire, le forum international de génération IV (*Generation IV International Forum* – GIF (Generation IV International Forum, 2002)) propose différentes avenues, différents modèles de réacteurs, qui pourront, si ils sont un jour réalisés, donner un nouveau visage à cette technologie. Ces réacteurs promettent, entres autres, une sécurité accrue, souvent passive, une résistance à la prolifération du matériel nucléaire, une meilleure utilisation de la ressource en matériel fissile et une réduction du volume des déchets.

Parmis les designs retenus par le GIF, on note, entres autres, les réacteurs rapides à sodium ou à plomb fondu, les réacteurs à sel fondu, ou encore les réacteurs à eau sur-critique. Ceux qui capteront particulièrement notre attention dans cette étude sont les réacteurs à très haute température (VHTR), notamment les réacteurs modulaires dits prismatiques (*Prismatic Modular Reactor* – PMR) ou à lit de boulets (*Pebble-bed modular reactor* – PBMR). Ces deux designs de réacteurs présentent entres autres, la particularité de contenir leur combustible dans de petites pastilles sphériques TRISO.

2.1 Réacteurs VHTR de quatrième génération

Les réacteurs à très haute température retiennent notre attention parce qu'ils offrent un design éprouvé en reposant sur la technologie des réacteurs à gaz, dont plusieurs ont été mis en fonction au cours des années. Comme leur nom l'indique, ce sont des réacteurs fonctionnant à des températures nominales beaucoup plus élevées ($\sim 1000^\circ\text{C}$) que les réacteurs présentement en opération, et présentent de ce fait des avantages non négligeables par rapport aux réacteurs actuels, que nous présenterons dans ce chapitre. Il est prévu que les premiers réacteurs commerciaux de ce type voient le jour aussi tôt qu'en 2015, une partie des concepts technologiques ayant déjà été développés pour les réacteurs à gaz, et certains prototypes ayant déjà été mis en fonction, notamment en Chine (Jing *et al.*, 2002) et au Japon (Shiozawa *et al.*,

2004).

2.1.1 Designs

Les deux designs sont suggérés pour les réacteurs à très haute température, soient le réacteur à lit de boulets, et le réacteur prismatique, utilisent le même élément de combustible de base, c'est-à-dire la pastille TRISO (*Tristructural-isotropic* – voir figure 2.1¹) (Nabielek *et al.*, 1990). Celles-ci sont constituées d'un cœur d'environ 100 μm de diamètre, composé d'oxyde ou de carbure d'uranium, suivi de couches successives de carbone poreux, de carbone pyrolytique, de carbure de silicium sous forme céramique, puis d'une autre couche de carbone pyrolytique. Ces éléments de combustible présentent de nombreux avantages que nous présenterons plus en détails à la section 2.1.2.

En plus d'avoir une résistance mécanique et une capacité de rétention des déchets très élevées, la très haute résistance thermique des pastilles TRISO ($>2000^\circ\text{C}$) permet d'obtenir des températures d'exploitation du réacteur très élevées. Le graphite utilisé comme modérateur et comme réflecteur n'accumulera pas de défauts de Winger à cause de la plus grande mobilité des molécules dans le réseau – il s'agit d'une sorte de recuit permanent, ce qui permet d'éviter des accidents comme le célèbre feu de Windscale (Duco, 2003).

Dans les réacteurs à lit de boulets (voir section 2.2.1), les éléments TRISO sont insérés dans une bille de la grosseur d'une balle de tennis (~ 60 mm de diamètre), qui à son tour sera insérée dans le réacteur, l'hélium agissant comme caloporteur circulant entre les billes, un réacteur pouvant contenir environ 700 000 boulets pour 1000 MW(th). Quant aux réacteurs prismatiques, les éléments sont incorporés dans des blocs de graphite qui seront rechargés à l'arrêt, et dans lesquels circule aussi de l'hélium pour évacuer la chaleur. Les détails en seront présentés à la section 2.2.2. Pour les deux types de réacteurs, un total de quelques milliards de particules TRISO se trouvent dans le cœur durant l'opération, ce qui en fait un système très difficile à analyser, d'où la pertinence de la méthode présentée ici.

2.1.2 Propriétés

Les deux types de réacteurs VHTR présentent des caractéristiques qui leur sont bien évidemment uniques, mais en partagent aussi certaines, la plupart étant dues aux particules TRISO. On compte parmi celles-ci une sécurité accrue, une résistance à la prolifération, un meilleur confinement des déchets, et une meilleure utilisation du combustible.

¹Image tirée de Wikimedia Commons - <http://en.wikipedia.org/wiki/File:TRISO.gif>

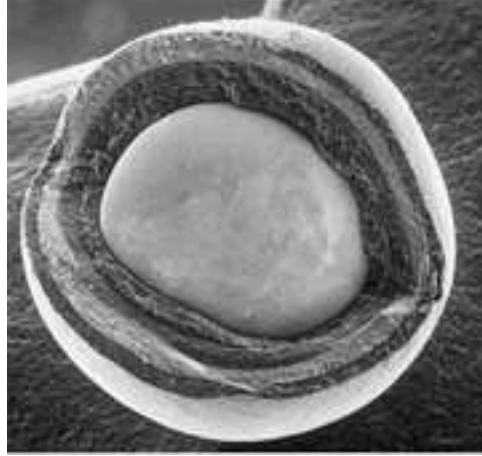


FIGURE 2.1 Coupe d'une bille TRISO

Dans les deux types de réacteurs, la température de sortie sensiblement plus élevée a pour objectif de permettre un rendement thermique nettement plus élevé que dans les réacteurs actuels. Cette particularité sera aussi exploitée en offrant la possibilité de produire de l'hydrogène à grande échelle sans l'utilisation d'hydrocarbures ou d'électrolyse, à l'aide du cycle soufre/iode, une série de réactions (globalement) endothermiques nécessitant des très hautes températures (Vitart *et al.*, 2006).

On pourra aussi choisir d'utiliser le thorium comme combustible pour ces réacteurs, afin de créer des cycles surgénérateurs, ou encore de brûler le plutonium accumulé dans les déchets des réacteurs actuellement en fonction (Talamo et Gudowski, 2005).

Sécurité

Une des principales exigences pour les réacteurs de IV^e génération étant une sécurité et une sûreté accrue, les réacteurs à très haute température sont dans une position stratégique, un de leurs principaux avantages étant une sécurité passive. Cet effet se fera notamment sentir lors de défaillances des systèmes entourant le réacteur, comme le circuit de refroidissement. Si, par exemple, une panne de celui-ci cause une hausse de la température du réacteur, une rétroaction négative en réactivité s'en suivra. Celle-ci est principalement due à un élargissement dû à l'effet Doppler des résonances de la section efficace dans l'²³⁸U. Suivra donc une augmentation de l'absorption de neutrons dans le réacteur, et, éventuellement, une stabilisation du niveau de puissance, qui sera évacuée par rayonnement ; on peut même concevoir le réacteur de façon à ce que les fuites thermiques surpassent la capacité du réacteur

dans un tel cas. Cet effet a d'ailleurs pu être observé expérimentalement (dans un réacteur à lit de boulets) (Gottaut et Krüger, 1990; Krüger *et al.*, 1988).

Non-Prolifération

Ce type de réacteur promet d'offrir une résistance à la prolifération accrue, notamment en raison des propriétés intrinsèques des pastilles TRISO, si on considère que le combustible utilisé sera de beaucoup plus difficile à récupérer. Il faut aussi voir qu'avec une meilleure utilisation du combustible (un burnup plus élevé), la quantité d'isotopes fissiles sera beaucoup plus faible : par exemple, une production accrue de ^{240}Pu par capture dans le ^{239}Pu réduira le ratio $^{239}\text{Pu} / ^{240}\text{Pu}$, menant à un plutonium de moins bonne qualité pour des fins militaires.

Confinement des déchets

Le confinement des éléments TRISO, en plus d'empêcher la récupération des déchets nucléaires à des fins militaires, agit aussi comme une barrière dans le sens inverse, en empêchant les produits de fission et autres éléments dangereux de s'échapper dans l'environnement. La couche de carbone poreux aura l'avantage de protéger la couche de SiC des produits de fissions pouvant être corrosifs (le carbone étant très stable chimiquement), cette dernière agissant comme principale barrière contre les fuites de déchets. Les couches de carbone pyrolytique, se comprimant sous irradiation, permettront d'obtenir la rigidité mécanique nécessaire pour maintenir la cohésion de l'élément de combustible.

Burnup

Ce confinement peut par contre se transformer en désavantage si on veut considérer un éventuel retraitement du combustible utilisé. Cependant, considérant les burnups recherchés dans ces types de réacteurs, on peut presque d'ores et déjà réduire la possibilité d'un éventuel retraitement, étant donné la beaucoup plus faible proportion d'isotopes fissiles ou fertiles qui resteront disponibles dans les particules usées. Les estimations théoriques prévoient un burnup atteignable grâce aux particules TRISO allant jusqu'à 200 GWj/t, ce qui est nettement supérieur à celui atteint avec les réacteurs actuels ($\sim 30\,000$ MWj/t). Un taux de combustion de 747 MWj/kg a même déjà été rapporté pour un combustible à base d'oxyde de plutonium (Nabielek *et al.*, 1984).

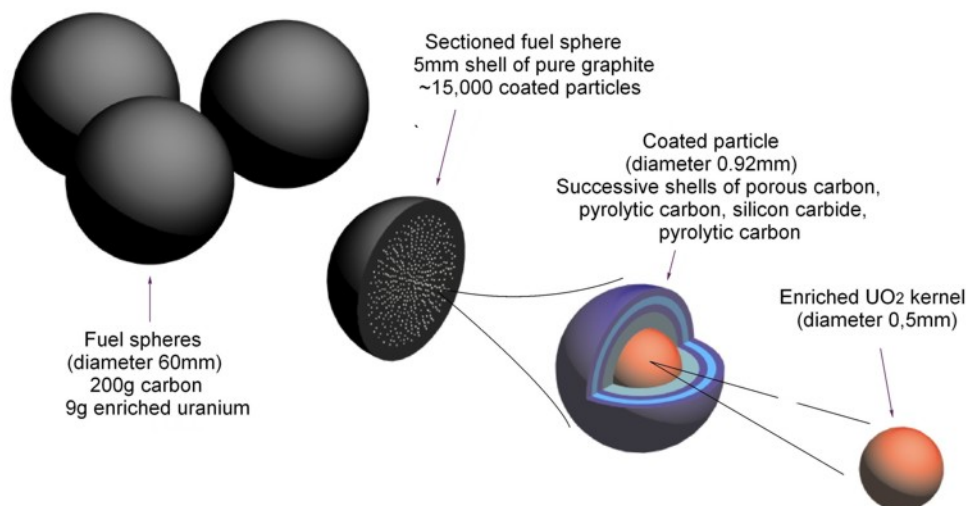


FIGURE 2.2 Élément de combustible : Boulet

2.2 Structure du réacteur

Les deux types de réacteurs présentant des caractéristiques de conception distinctes, nous présenterons les concepts séparément, mais nous attarderons davantage au concept prismatique, considérant que sa géométrie hexagonale rend plus intéressante son implantation dans des codes de calculs existants.

2.2.1 Réacteurs à lit de boulets

Les réacteurs à lits de boulets sont basés sur un concept de réacteur développé en Allemagne de l'ouest dès les années 1960 (Schulten, 1978). Les pastilles TRISO sont ici compactées dans une boule de graphite (voir Figure 2.2²), qui contiendra quelque milliers d'éléments, laquelle est à son tour est insérée dans un réacteur qui peut contenir quelques centaines de milliers de boulets similaires. Les barres de contrôle et d'arrêt n'auront pas d'emplacements spécifiques, mais s'inséreront plutôt dans le lit formé par les boulets en déplaçant ceux-ci.

De nouveaux boulets sont constamment ajoutés sur le dessus du réacteur, et retirés par le dessous, de manière à avoir un roulement constant et un rechargement en marche du réacteur. Cette manière de procéder permet aussi de trier les boulets à la sortie du réacteur

²Image tirée du Sud-Africain PBMR (Pty) Ltd – <http://www.pbmr.co.za/>



FIGURE 2.3 Élément de combustible : Bloc de graphite

à l'aide d'une mesure d'irradiation pour décider d'un passage subséquent dans le cœur. Cependant, le principal désavantage est d'ajouter un autre niveau de complexité à l'analyse du réacteur et à la gestion du combustible, en ne permettant pas de connaître explicitement la position des boulets dans le cœur.

2.2.2 Réacteurs prismatiques

Les réacteurs prismatiques sont largement basés sur la technologie de General Atomics, qui a développé le GT-MHR (*Gas Turbine Modular Helium Reactor*). Dans ces réacteurs, les billes TRISO présentées plus tôt seront formées en compacts des cylindres de petites dimensions ($\sim 5 \times 1,25$ cm de diamètre) qui seront installés dans des blocs de graphite de forme hexagonale. Ces blocs comportent aussi des ouvertures permettant d'insérer des barres de contrôle et d'arrêt, et d'assurer la circulation du fluide de refroidissement. Par la suite, ces structures hexagonales sont placées dans le cœur du réacteur pour un passage unique (voir figure 2.3³).

La figure 2.4⁴ présente une vue en coupe du cœur projeté d'un réacteur prismatique, qui reste pour le moment quasi-identique à celui d'un GT-MHR. On peut observer que les blocs prismatiques de combustible sont répartis en forme d'anneau dans le cœur, la partie centrale ne contenant pas de matériel fissile, et la partie extérieure servant presque exclusivement de réflecteur. Ce cœur annulaire offre une sécurité inhérente au réacteur. Le graphite au centre agit principalement comme réflecteur, mais aussi comme masse d'absorbant thermique, donnant une inertie thermique au réacteur durant les transitoires de puissance, ce qui aide au

³Image tirée de UARGA – http://www.uarga.org/nucleaire/reacteur_haute_temp.php

⁴Image tirée de l'article (Kim *et al.*, 2007)

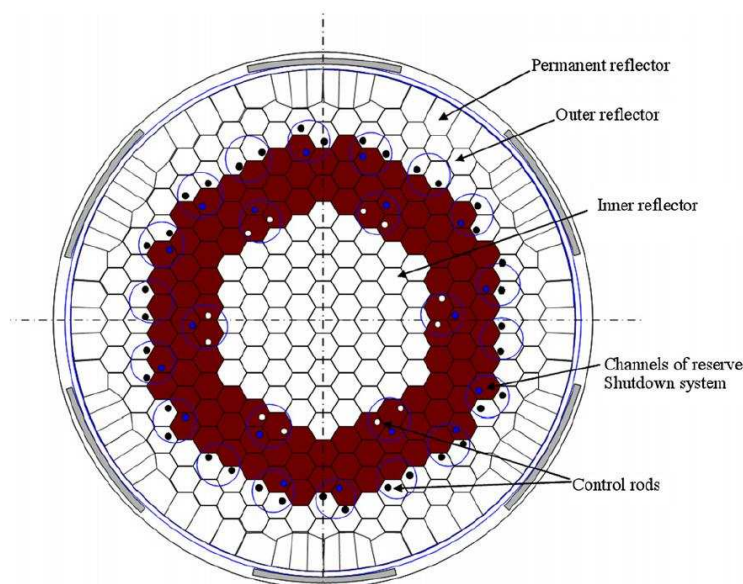


FIGURE 2.4 Vue en coupe d'un cœur de réacteur prismatique

contrôle et à l'opération, et stabilise le réacteur en cas d'accident.

2.2.3 Refroidissement

Le caloporteur gazeux comporte de nombreux avantages. On pourra notamment considérer le fait qu'en utilisant de l'hélium, qui a une section efficace totale d'interaction avec les neutrons très faible, on aura un effet presque nul en réactivité lors d'une vidange intempestive de ce dernier. Cette faible interaction avec le flux neutronique empêche aussi son activation, et par le fait même la contamination de son circuit, permettant l'utilisation directe dans la turbine, sans passer par des circuits d'échange de chaleur. De plus, le caloporteur étant déjà sous forme gazeuse, on n'a pas à se soucier des problèmes liés aux modifications des propriétés physiques associés aux changements de phase qui peuvent survenir dans un réacteur utilisant de l'eau (ébullition, assèchement, etc.) pouvant mener à des excursions de puissance. Le graphite possède aussi une chaleur massique (donc une inertie thermique) très importante, et une bonne stabilité mécanique à haute température, contribuant à la sûreté inhérente du réacteur.

Le modérateur qui servira à ralentir les neutrons rapides produits par la fission jusqu'à ce qu'ils puissent produire une nouvelle fission thermique est principalement le graphite composant le restant du cœur. C'est cependant ce même graphite qui sera aussi chauffé par le dégagement d'énergie provenant des réactions nucléaires, et qui devra être refroidi par

l'hélium qui ira par la suite dans le circuit thermique pour actionner la turbine à gaz et ainsi produire de l'électricité. Le graphite, de par sa résistance thermique élevée, pourra supporter des températures nécessaires à l'optimisation du cycle thermique de l'hélium, comme le cycle de Brayton qui permet de pousser les efficacités thermiques des turbines à gaz jusqu'à près de 50%.

2.3 Besoins de modélisation

L'analyse des réacteurs VHTR fait surgir plusieurs problèmes au niveau des schémas de calcul traditionnellement utilisés, notamment en raison d'une interaction plus forte entre le combustible et le réflecteur, un déplacement du spectre neutronique et une diffusion des neutrons vers le haut du spectre en raison notamment du graphite à haute température.

Le problème qui est d'intérêt pour ce travail est celui que ces réacteurs font surgir en raison de leur géométrie particulière. En premier lieu, la forte hétérogénéité à tous les niveaux du réacteur impliquera que le procédé d'homogénéisation des cellules représentera moins bien la géométrie initiale. L'étape d'autoprotection des résonances prendra donc ici une importance particulièrement cruciale si l'on veut obtenir des résultats précis.

Les codes de calcul déterministes ou stochastiques, ne peuvent non plus, pour la plupart, se représenter de manière adéquate la géométrie, à cause du fait que la position des éléments de combustible n'est pas connue précisément, mais aussi à cause du nombre important de ces pastilles typiquement contenues dans un seul assemblage. Ce problème, dit de la double-hétérogénéité, peut s'apparenter à ceux survenant entre autres lors de l'analyse associée à l'eau bouillante, de particules de poison consommable insérées dans des éléments de combustible, ou encore de transport γ dans du béton – donc de dispersion aléatoire de grains dans un milieu uniforme. On parle de deux niveaux d'hétérogénéité spatiale, soit un en lien avec l'hétérogénéité due aux différentes composantes du réacteur (modérateur, caloporteur, pastilles de combustible, barres de contrôle), et un autre dû à l'hétérogénéité intrinsèque du combustible, lui-même composé d'une matrice de graphite dans laquelle sont distribués aléatoirement des éléments sphériques à plusieurs couches.

Des méthodes à l'origine développées pour la modélisation des grains de poison consommable dans les assemblages de combustible des PWR ont été présentées notamment par Hébert (Hébert, 2008) et Sanchez et Pomraning (Sanchez et Pomraning, 1991), et implantées dans APOLLO (Sanchez, 1988) et DRAGON (Marleau *et al.*, 1992). Ces méthodes supposent

généralement qu'on peut remplacer le courant neutronique s'échappant de la sphère par une source uniformément distribuée dans la matrice. Une dérivation de l'équation de transport est ensuite effectuée en supposant une distribution aléatoire des éléments stochastiques dans la matrice les entourant, avec une utilisation de méthodes d'analyse statistique – notamment une généralisation de la loi de Poisson (*renewal statistics*) – pour gérer l'aspect aléatoire des positions des éléments granulaires.

Dernièrement, deux méthodes indirectes ont principalement été présentées pour contourner la double hétérogénéité qui surgit des configurations de tels réacteurs. La première est celle dite de la transformation physique à réactivité équivalente (*Reactivity-equivalent Physical Transform* – RPT), telle que présentée par Kim (Kim et Baek, 2005). La méthode consiste principalement à remplacer le mélange de modérateur au graphite et des pastilles TRISO par une pâte homogène dont les propriétés physiques macroscopiques seront équivalentes à celle du mélange non-homogénéisé.

Cependant, l'effet d'autoprotection des résonnances est généralement très prononcé dans les éléments de combustible fortement hétérogènes, comme les boulets contenant des particules TRISO, et de plus petite taille (l'autoprotection spatiale étant fonction du ratio volume-surface). Alors, afin de conserver une réactivité équivalente, le rayon du boulet homogénéisé sera forcément plus petit que l'original. Ce rayon r_{rpt} du boulet homogénéisé est calculé en utilisant des méthodes Monte-Carlo sur un seul boulet, dont les propriétés seront utilisées pour tous les éléments de combustible de l'assemblage, analysé par la suite à l'aide des techniques traditionnelles.

La seconde méthode, souvent utilisée conjointement avec la première (Noh *et al.*, 2008) est dite du combustible cylindrique équivalent (*Equivalent Cylindrical Fuel* – ECF) (Lee *et al.*, 2005). Plutôt que d'utiliser une bille sphérique aux propriétés équivalentes, elle consiste à utiliser un cylindre, qui pourra par la suite être analysé par des méthodes conventionnelles, conservant l'inventaire matériel et la longueur de corde moyenne des éléments de combustible. Cette méthode permet donc l'utilisation d'un code de cellule 2-D traditionnel pour la résolution du problème dont un des niveaux d'hétérogénéité a été levé.

2.3.1 Solution proposée

Le principal inconvénient de ces méthodes tient cependant du fait qu'il s'agit de méthodes indirectes. Ainsi, elles utilisent des approximations supplémentaires afin de permettre le traitement de ces géométries complexes en évaluant des propriétés physiques équivalentes, sans

pour autant permettre l'analyse du combustible TRISO tel quel. Peu importe la méthode utilisée pour analyser les problèmes comportant une double hétérogénéité, une seconde homogénéisation est toujours requise afin de pouvoir résoudre le problème.

L'approche que nous proposons ici est plutôt d'établir de façon aléatoire une distribution de sphères, puis d'analyser la géométrie résultante. Cela permettra à la fois de proposer une alternative aux méthodes traditionnelles, tout en nous donnant la chance de valider certains de leurs résultats. La technique utilisée est l'addition séquentielle aléatoire des positions des sphères une par une, en testant à chaque étape la superposition avec celles dont la position est déjà déterminée, et ce, jusqu'à l'atteinte de la densité désirée.

Pour de faibles densités, la représentation est généralement adéquate. Par contre, on ne pourra pas atteindre, ou même approcher, la densité volumique maximale (arrangement en cellule hexagonale ou cubique à faces centrées, $d_m = \frac{\pi}{3\sqrt{2}} \simeq 0.74048\dots$) des sphères avec cette méthode, considérant la génération stochastique des positions et le volume non-nul des grains (Widom, 1965). Le champ de positions des sphères ainsi généré différera d'un autre dont les sphères seront en équilibre thermodynamique, du fait que les sphères générées à une position rapprochée de deux autres sphères verra son volume de positions disponibles diminuer d'un facteur plus important que si on avait simplement « laissé tomber » la sphère, ou si on avait fait un empilement. Pour remédier au problème et utiliser des densités plus élevées, on notera entre autres la possibilité de spécifier explicitement les positions des sphères au logiciel, positions qui pourront avoir été générées indépendamment avant le début du calcul.

Chapitre 3

ÉQUATIONS DE NEUTRONIQUE

3.1 Dérivation de l'équation de transport

Le point de départ de la majorité des problèmes de neutronique est l'équation de transport. Celle-ci est dérivée de la forme linéaire de l'équation de Boltzmann de la dynamique des gaz. On suppose donc que les neutrons dans le réacteur n'interagissent pas entre eux, et qu'ils se déplacent linéairement entre les collisions avec les noyaux. Le comportement de la population neutronique à l'intérieur d'un réacteur est généralement dépendant de sept variables : trois coordonnées spatiales ($\vec{r} = x\vec{i} + y\vec{j} + z\vec{k}$), trois pour la vitesse du neutron ($\vec{V}_n = \frac{d\vec{r}}{dt}$), et une de temps (t). Nous utiliserons ici le formalisme présenté par Hébert (A. Hébert, 2009).

Nous commencerons par définir les quantités suivantes :

- $V_n = \|\vec{V}_n\|$,
- $\hat{\Omega} = \mu\vec{i} + \eta\vec{j} + \xi\vec{k}$, μ , η et ξ étant les cosinus directeurs de la vitesse V_n . On peut établir la relation $\vec{V}_n = V_n\hat{\Omega}$,
- $n(\vec{r}, V_n, \hat{\Omega}, t)$, la densité de population neutronique, avec $n(\vec{r}, V_n, \hat{\Omega}, t) d^2\Omega d^3r dV_n$ la grandeur physique représentant le nombre de neutrons dans le volume d^3r autour de \vec{r} , avec une vitesse V_n dans un intervalle dV_n , dans la direction $\hat{\Omega}$, dans un intervalle de $d^2\Omega$, au temps t .

L'équation de bilan pourra s'écrire en considérant le nombre de neutrons produits et le nombre de neutrons perdus, dans l'intervalle de temps $[t, t + \Delta t]$. La variation du nombre de neutrons sera donnée par :

$$\left[n(\vec{r}, V_n, \hat{\Omega}, t + \Delta t) - n(\vec{r}, V_n, \hat{\Omega}, t) \right] d^3r d^2\Omega dV_n \quad (3.1)$$

De la même façon, le nombre de neutrons s'échappant de l'hypervolume $d^3r d^2\Omega dV_n$ sera donné par :

$$\vec{\nabla} \cdot \Omega \phi(\vec{r}, V_n, \hat{\Omega}, t) d^3r d^2\Omega dV_n \Delta t \quad (3.2)$$

On aura le nombre de collisions entre les neutrons et le milieu, donné par :

$$\Sigma(\vec{r}, V_n) V_n n(\vec{r}, V_n, \hat{\Omega}, t) d^3r d^2\Omega dV_n \Delta t \quad (3.3)$$

Finalement, le nombre de nouveaux neutrons émis, à la fois par diffusion et par fission est donné par :

$$Q(\vec{r}, V_n, \hat{\Omega}, t) d^3r d^2\Omega dV_n \Delta t \quad (3.4)$$

où l'on définit :

- $\phi(\vec{r}, V_n, \hat{\Omega}, t)$, le flux neutronique. Notons que celui-ci n'est pas un flux proprement dit (flux de chaleur) mais est plutôt défini, à l'aide de la densité de population neutronique $n(\vec{r}, V_n, \hat{\Omega}, t)$, comme :

$$\phi(\vec{r}, V_n, \hat{\Omega}, t) = V_n n(\vec{r}, V_n, \hat{\Omega}, t) \quad (3.5)$$

- $\Sigma(\vec{r}, V_n)$, la section efficace totale macroscopique, que nous supposons indépendante de $\hat{\Omega}$ (milieu isotrope), et de t , afin d'alléger la notation,
- $Q(\vec{r}, V_n, \hat{\Omega}, t)$, le terme source, sur lequel nous élaborerons dans la prochaine section,
- $\hat{\Omega} \cdot \vec{\nabla} \phi(\vec{r}, V_n, \hat{\Omega}, t)$, le bilan des neutrons entrant et sortant du volume considéré.

L'équation globale de bilan sera alors :

$$\frac{n(\vec{r}, V_n, \hat{\Omega}, t + \Delta t) - n(\vec{r}, V_n, \hat{\Omega}, t)}{\Delta t} = Q(\vec{r}, V_n, \hat{\Omega}, t) - \vec{\nabla} \cdot \hat{\Omega} \phi(\vec{r}, V_n, \hat{\Omega}, t) - \Sigma(\vec{r}, V_n) V_n n(\vec{r}, V_n, \hat{\Omega}, t) \quad (3.6)$$

que l'on peut exprimer, en prenant la limite lorsque $\Delta t \rightarrow 0$, sous la forme :

$$\frac{1}{V_n} \frac{\partial}{\partial t} \phi(\vec{r}, V_n, \hat{\Omega}, t) + \hat{\Omega} \cdot \vec{\nabla} \phi(\vec{r}, V_n, \hat{\Omega}, t) + \Sigma(\vec{r}, V_n) \phi(\vec{r}, V_n, \hat{\Omega}, t) = Q(\vec{r}, V_n, \hat{\Omega}, t) \quad (3.7)$$

où l'on a utilisé l'identité $\vec{\nabla} \cdot (\hat{\Omega} \phi(\vec{r})) = \hat{\Omega} \cdot \vec{\nabla} \phi(\vec{r}) + \phi(\vec{r}) \vec{\nabla} \cdot \hat{\Omega}$, avec $\vec{\nabla} \cdot \hat{\Omega} = 0$. Cette équation est dite la forme différentielle de l'équation de transport. En règle générale, on utilisera l'approximation quasi-statique, qui implique que l'on considérera l'équation en mode stationnaire, les variations temporelles étant traitées, par exemple, en utilisant des méthodes dérivées de la théorie des perturbations. L'équation en mode stationnaire sera exprimée comme suit :

$$\hat{\Omega} \cdot \vec{\nabla} \phi(\vec{r}, E, \hat{\Omega}) + \Sigma(\vec{r}, E) \phi(\vec{r}, E, \hat{\Omega}) = Q(\vec{r}, E, \hat{\Omega}) \quad (3.8)$$

où nous avons remplacé V_n par l'énergie $E = \frac{mV_n^2}{2}$ du neutron se déplaçant à la vitesse V_n . Il est ardu de résoudre cette équation analytiquement sauf pour des cas extrêmement simples.

On utilisera donc généralement des approximations et des méthodes numériques afin d'isoler la quantité d'intérêt, c'est-à-dire le flux ϕ .

3.1.1 Le terme source

Dans l'équation (3.8), le terme $Q(\vec{r}, E, \hat{\Omega})$ est le terme comprenant les sources de neutrons, tant celles provenant de la fission et de la diffusion (de $\hat{\Omega}'$ vers $\hat{\Omega}$, et de E' vers E par exemple), que les sources extérieures. Cependant, celles-ci n'interviennent généralement pas dans les cas qui nous intéressent.

Pour les sources de diffusion, nous devons considérer les neutrons débutant avec une énergie E' et une direction $\hat{\Omega}'$ (dans un intervalle $dE d^2\Omega$), pour toutes les valeurs de E' , et pour toutes les directions $\hat{\Omega}'$, qui subissent soit une collision, soit une réaction (n, xn) (un neutron incident et x neutrons émis). Ces nouveaux neutrons seront exprimés par :

$$\int_0^\infty \int_{4\pi} \tilde{\Sigma}_S(\vec{r}, E \leftarrow E', \hat{\Omega} \leftarrow \hat{\Omega}') \phi(\vec{r}, E', \hat{\Omega}') d^2\Omega' dE' \quad (3.9)$$

où nous avons utilisé $\tilde{\Sigma}_S$ comme la section efficace macroscopique de diffusion (*scattering*), et qui comprend aussi les termes pour les réactions (n, xn). Notons ici que si les milieux qui composent le domaine spatial sont isotropes, la section efficace de diffusion sera une fonction seulement de l'angle $\hat{\Omega} \cdot \hat{\Omega}'$.

La production de neutrons par fission étant isotrope, leur distribution sera donnée par :

$$Q_f(\vec{r}, E, \hat{\Omega}) = \frac{1}{4\pi k_{\text{eff}}} Q_f(\vec{r}, E) \quad (3.10)$$

Il convient ici de mentionner le paramètre k_{eff} , le facteur de multiplication effectif de neutrons. Il représente une proportion nécessaire entre le nombre de neutrons produits et ceux perdus (absorption et fuites), afin de conserver l'approximation stationnaire – c'est en quelque sorte la valeur propre du système. Sa valeur variera autour de 1, et on dira que le réacteur est *sur-critique* si $k_{\text{eff}} > 1$, alors que la population de neutrons augmente, et qu'il est *sous-critique* lorsque $k_{\text{eff}} < 1$, dans quel cas la population de neutrons diminue dans le temps.

On peut développer les sources isotropes, en fonction du taux d'émission $\nu\Sigma_f$ et du spectre d'émission χ_j :

$$Q_f(\vec{r}, E) = \sum_{j=1}^J \chi_j(E) \int_0^\infty \nu\Sigma_{f,j}(\vec{r}, E') \phi(\vec{r}, E') dE' \quad (3.11)$$

où l'on a utilisé

- $\phi(\vec{r}, E)$, le flux intégré, défini comme $\phi(\vec{r}, E) = \int_{4\pi} \phi(\vec{r}, E, \hat{\Omega}) d^2\Omega$,
- J , le nombre d'isotopes fissiles, chacun représenté par $j = 1 \dots J$,
- $\nu\Sigma_{f,j}$, le taux de fission de l'isotope j , avec $\Sigma_{f,j}$, sa section efficace macroscopique de fission, et ν le nombre moyen de neutrons émis par fission,
- $\chi_j(E)$, le spectre d'émission des neutrons, c'est-à-dire la probabilité d'un neutron émis par l'isotope j d'avoir une énergie E , à dE près.

On peut finalement écrire le terme source total, de la manière suivante :

$$Q(\vec{r}, E, \hat{\Omega}) = \int_0^\infty \int_{4\pi} \tilde{\Sigma}_S(\vec{r}, E \leftarrow E', \hat{\Omega} \cdot \hat{\Omega}') \phi(\vec{r}, E', \hat{\Omega}') d^2\Omega' dE' + \frac{1}{4\pi k_{\text{eff}}} \sum_{j=1}^J \chi_j(E) \int_0^\infty \nu\Sigma_{f,j}(\vec{r}, E') \phi(\vec{r}, E') dE' \quad (3.12)$$

3.1.2 Discrétisation multigroupe

La discrétisation multigroupe consiste en une séparation du continuum énergétique, en G groupes d'énergie. On considère alors que les neutrons sont monoénergétiques dans chacun des sous-groupes – on peut donc représenter la distribution énergétique par une fonction continue par parties. On transformera donc les intégrales en énergie de l'équation (3.12) en sommation sur chacun des groupes d'énergie considérés. Notons aussi l'utilisation de E_0 comme borne supérieure d'énergie plutôt que ∞ . Une valeur de $E_0 \sim 10$ MeV permet généralement de couvrir l'ensemble des énergies des neutrons dans un réacteur à fission, surtout thermique. On devra par conséquent redéfinir les quantités que nous avons utilisées dans l'équation de transport précédemment, en les moyennant sur l'intervalle énergétique du groupe considéré :

$$\phi_g(\vec{r}, \hat{\Omega}) = \int_{E_g}^{E_{g-1}} \phi(\vec{r}, E, \hat{\Omega}) dE \quad (3.13)$$

$$\Sigma_g(\vec{r}) = \frac{1}{\phi_g(\vec{r})} \int_{E_g}^{E_{g-1}} \Sigma(\vec{r}, E) \phi(\vec{r}, E) dE \quad (3.14)$$

Où nous remarquons la pondération de Σ_g par ϕ_g , qui permet de conserver le taux de réaction. Nous pouvons donc écrire la forme différentielle de l'équation de transport multigroupe à l'état stationnaire :

$$\hat{\Omega} \cdot \vec{\nabla} \phi_g(\vec{r}, \hat{\Omega}) + \Sigma_g(\vec{r}) \phi_g(\vec{r}, \hat{\Omega}) = Q_g(\vec{r}, \hat{\Omega}) \quad (3.15)$$

Où nous avons défini le terme source multigroupe comme :

$$Q_g(\vec{r}, \hat{\Omega}) = \sum_{h=1}^G \int_{4\pi} \tilde{\Sigma}_{S,g \leftarrow h}(\vec{r}, \hat{\Omega} \cdot \hat{\Omega}') \phi_h(\vec{r}, \hat{\Omega}') d^2\Omega' + \frac{1}{4\pi k_{\text{eff}}} \sum_{j=1}^J \chi_{j,g}(E) \sum_{h=1}^G \nu_{\Sigma_{f,j,h}}(\vec{r}) \phi_h(\vec{r}) \quad (3.16)$$

avec :

$$\chi_{j,g} = \int_{E_g}^{E_{g-1}} \chi_j(E) dE \quad (3.17)$$

$$\phi_g(\vec{r}) = \int_{E_g}^{E_{g-1}} \phi(\vec{r}, E) dE \quad (3.18)$$

$$\nu_{\Sigma_{f,j,g}}(\vec{r}) = \frac{1}{\phi_g(\vec{r})} \int_{E_g}^{E_{g-1}} \nu_{\Sigma_{f,j}}(\vec{r}, E) \phi(\vec{r}, E) dE \quad (3.19)$$

$$\tilde{\Sigma}_{S,g \leftarrow h}(\vec{r}, \hat{\Omega} \cdot \hat{\Omega}') = \frac{1}{\phi_h(\vec{r})} \int_{E_g}^{E_{g-1}} \int_{E_h}^{E_{h-1}} \tilde{\Sigma}_S(\vec{r}, E \leftarrow E', \hat{\Omega} \cdot \hat{\Omega}') \phi(\vec{r}, E') dE' dE \quad (3.20)$$

3.2 Résolution de l'équation de transport

L'équation de transport peut être résolue en utilisant nombre de techniques. Nous présentons ici rapidement trois méthodes : la méthode des probabilités de collision, celle des caractéristiques et, rapidement, les méthodes stochastiques (Monte-Carlo). Toutes les trois ont la particularité d'utiliser un procédé de *tracking*, ou de définition et d'analyse de lignes d'intégration, afin d'intégrer l'équation de transport sur des régions définies de la géométrie pour par la suite pouvoir en tirer le flux.

La position \vec{r} d'une particule se déplaçant dans une direction $\hat{\Omega}$ pourra toujours être exprimée en fonction d'une position de référence \vec{p} sur un plan $\Pi_{\hat{\Omega}}$ perpendiculaire à $\hat{\Omega}$ comme $\vec{p} - s\hat{\Omega}$, avec s la distance parcourue par la particule dans la direction $\hat{\Omega}$. Il s'agit de la « caractéristique » d'une particule, qu'on peut visualiser comme une ligne dans l'espace correspondant à la trajectoire de la particule.

Afin de formaliser ce procédé de traçage de rayons (*ray tracing*) au travers de la géométrie, nous établirons les notions de densité surfacique de lignes sur chacun des plans $\Pi_{\hat{\Omega}}$, ainsi qu'un nombre défini de directions $\hat{\Omega}$ pour identifier les trajectoires caractéristiques, sur lesquelles les solutions de l'équation de transport seront constantes. Nous aurons les définitions

suivantes :

- \mathcal{T} comme l'ensemble des caractéristiques, comprenant toutes les directions et tous les points de départ ;
- \vec{T} , chaque caractéristique sera représentée complètement par son orientation $\hat{\Omega}$ et son point de départ \vec{p} sur le plan $\Pi_{\hat{\Omega}}$ perpendiculaire à $\hat{\Omega}$. La caractéristique \vec{T} aura une différentielle $d^4T = d^2\Omega d^2p$;

Nous introduirons aussi le chemin optique dans le groupe g , défini comme :

$$\tau_g(s) = \int_0^s \Sigma_g(\vec{p} - s'\hat{\Omega}) ds' \quad (3.21)$$

qui servira de facteur intégrant. Dans un milieu homogène par parties, comme c'est généralement le cas dans un réacteur, il s'agira donc de la somme des produits des distances par les sections efficaces totales des milieux k traversés par une ligne \vec{T} . On peut établir la relation suivante :

$$\frac{d}{ds} \left[e^{-\tau_g(s)} \phi_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) \right] = e^{-\tau_g(s)} \left[-\Sigma_g(\vec{p} - s\hat{\Omega}) \phi_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) + \frac{d}{ds} \phi_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) \right] \quad (3.22)$$

On peut aussi faire l'observation que l'opérateur de migration des neutrons, $\hat{\Omega} \cdot \vec{\nabla}$, est équivalent – dans un système de coordonnées cartésien – à :

$$\begin{aligned} \hat{\Omega} \cdot \vec{\nabla} &= (\hat{\Omega} \cdot \vec{i}) \frac{\partial}{\partial x} + (\hat{\Omega} \cdot \vec{j}) \frac{\partial}{\partial y} + (\hat{\Omega} \cdot \vec{k}) \frac{\partial}{\partial z} \\ &= -\frac{\partial}{\partial x} \frac{dx}{ds} - \frac{\partial}{\partial y} \frac{dy}{ds} - \frac{\partial}{\partial z} \frac{dz}{ds} \\ &= -\frac{d}{ds} \end{aligned} \quad (3.23)$$

en utilisant $\hat{\Omega} \cdot \vec{i} = -\frac{dx}{ds}$, $\hat{\Omega} \cdot \vec{j} = -\frac{dy}{ds}$ et $\hat{\Omega} \cdot \vec{k} = -\frac{dz}{ds}$.

Ces relations nous permettent de réécrire l'équation de transport (3.15) en tenant compte des caractéristiques. Si on utilise l'équation (3.22), et en multipliant les deux côtés par le facteur

intégrant $e^{-\tau_g(s)}$, on obtient :

$$\begin{aligned}
e^{-\tau_g(s)} Q_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) &= e^{-\tau_g(s)} \left[\Sigma_g(\vec{p} - s\hat{\Omega}) \phi_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) + \hat{\Omega} \cdot \vec{\nabla} \phi_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) \right] \\
&= e^{-\tau_g(s)} \left[\Sigma_g(\vec{p} - s\hat{\Omega}) \phi_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) - \frac{d}{ds} \phi_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) \right] \\
&= -\frac{d}{ds} \left[e^{-\tau_g(s)} \phi_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) \right]
\end{aligned} \tag{3.24}$$

En tenant compte du fait que $e^{-\tau_g(0)} = 1$ et $e^{-\tau_g(\infty)} = 0$, on trouve après l'intégration sur s de 0 à ∞ :

$$\begin{aligned}
\int_0^\infty e^{-\tau_g(s)} Q_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) ds &= - \int_0^\infty \frac{d}{ds} \left[e^{-\tau_g(s)} \phi_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) \right] ds \\
&= \phi_g(\vec{r}, \hat{\Omega})
\end{aligned} \tag{3.25}$$

qui est la forme intégrale multigroupe de l'équation de transport.

3.2.1 Probabilités de collision

La méthode des probabilités de collision nous vient de l'équation de transport multigroupe que nous exprimerons sous la forme intégrale (équation (3.25)), en supposant une source isotrope, donc indépendante de $\hat{\Omega}$, et répartie sur les 4π de l'angle solide :

$$\begin{aligned}
\phi_g(\vec{r}) &= \int_{4\pi} \phi_g(\vec{r}, \hat{\Omega}) d^2\Omega \\
&= \int_{4\pi} \int_0^\infty e^{-\tau_g(s)} Q_g(\vec{p} - s\hat{\Omega}, \hat{\Omega}) ds d^2\Omega
\end{aligned} \tag{3.26}$$

En introduisant le changement de variable $\vec{r}' = \vec{p} - s\hat{\Omega}$ et donc $d^3r' = d^2p ds = s^2 d^2\Omega ds$, on obtient :

$$\phi_g(\vec{r}) = \frac{1}{4\pi} \int_\infty \frac{e^{-\tau_g(s)}}{s^2} Q_g(\vec{r}', \hat{\Omega}) d^3r' \tag{3.27}$$

On effectue par la suite une discrétisation spatiale en découpant la cellule unitaire, sur laquelle on devra aussi supposer des conditions frontières, en N sous-régions de volumes V_i , $i = 1 \dots N$. Dans chacune de celles-ci, nous supposerons les sources uniformes $Q_{i,g}$, sous la forme :

$$Q_{i,g} = \sum_{h=1}^G \tilde{\Sigma}_{S,0,i,g \leftarrow h} \phi_{i,h} + \frac{1}{k_{\text{eff}}} \sum_{j=1}^J \chi_{j,g} \sum_{h=1}^G \nu \Sigma_{f,j,h} \phi_{i,h} \tag{3.28}$$

en utilisant le premier terme de l'expansion en polynômes de Legendre de la section efficace de diffusion isotrope, comme $\tilde{\Sigma}_{S,0,i,g \leftarrow h} = \frac{1}{4\pi} \tilde{\Sigma}_{S,i,g \leftarrow h, \Omega \leftarrow \Omega'}$. On peut multiplier l'équation (3.27) par $\Sigma_g(\vec{r})$ et intégrer sur chaque sous-volume V_j , pour obtenir :

$$\int_{V_j} \Sigma_g(\vec{r}) \phi_g(\vec{r}) d^3r = \frac{1}{4\pi} \int_{V_j} \Sigma_g(\vec{r}) \sum_i Q_{i,g} \int_{V_i} \frac{e^{-\tau_g(s)}}{s^2} d^3r' d^3r \quad (3.29)$$

Qu'on peut exprimer sous la forme :

$$\phi_{j,g} = \frac{1}{V_j \Sigma_{j,g}} \sum_i Q_{i,g} V_i P_{ij,g} \quad (3.30)$$

avec :

$$\phi_{j,g} = \frac{1}{V_j} \int_{V_j} \phi_g(\vec{r}) d^3r \quad (3.31)$$

$$\Sigma_{j,g} = \frac{1}{V_j \phi_{j,g}} \int_{V_j} \Sigma_g(\vec{r}) \phi_g(\vec{r}) d^3r \quad (3.32)$$

$$P_{ij,g} = \frac{1}{4\pi V_i} \int_{V_i} \int_{V_j} \Sigma_g(\vec{r}) \frac{e^{-\tau_g(s)}}{s^2} d^3r d^3r' \quad (3.33)$$

où on suppose $\Sigma_{j,g}$ constant sur la région V_j . L'expression $P_{ij,g}$ représente la probabilité de collision pour un neutron né dans une des régions V_i dans le groupe d'énergie g , de subir sa première collision dans la région V_j de la cellule. On utilisera généralement les probabilités de collision sous leur forme réduite, $p_{ij,g}$:

$$p_{ij,g} = \frac{P_{ij,g}}{\Sigma_{j,g}} = \frac{1}{4\pi V_i} \int_{V_i} \int_{V_j} \frac{e^{-\tau_g(s)}}{s^2} d^3r d^3r' \quad (3.34)$$

Ces probabilités de collision réduites ont les propriétés de rester finies alors que $\Sigma_{j,g}$ tend vers zéro, et d'être réciproques selon les régions l'espace, c'est-à-dire $p_{ij,g} V_i = p_{ji,g} V_j$, ce qui nous permet de simplifier l'équation (3.30) comme :

$$\phi_{i,g} = \sum_j Q_{j,g} p_{ij,g} \quad (3.35)$$

Cette forme de l'équation (3.35) nous montre clairement qu'il est possible de calculer le flux $\phi_{i,g}$, dans la région i et dans le groupe d'énergie g , en fonction seulement du terme source et des probabilités de collision réduites. Afin d'évaluer ces quantités, on effectue une discrétisation angulaire selon les directions $\hat{\Omega}$, en fonction de quadratures qui permettent d'établir les directions sur la sphère unité. Cette discrétisation angulaire nous permet de

réécrire les intégrales de volumes en intégrales sur le *tracking* comme :

$$\int_{4\pi} \int_{V_j} \mathcal{F}(\vec{r}, \hat{\Omega}) d^3r d^2\Omega = \int_{4\pi} \int_{\Pi_{\hat{\Omega}}} \int_{-\infty}^{\infty} \mathcal{W}_{V_j}(s', \hat{\Omega}) \mathcal{F}(s', \hat{\Omega}) ds' d^2p' d^2\Omega \quad (3.36)$$

où on a introduit la fonction caractéristique $\mathcal{W}_{V_j}(s, \hat{\Omega})$, qui vaut 1 si le point s de la trajectoire dans la direction $\hat{\Omega}$ est dans la région V_j , et 0 sinon.

Pour évaluer le flux, il importe donc de calculer, les volumes des sous-régions V_i , ainsi que les longueurs des trajets des caractéristiques $s\hat{\Omega}$ dans chacune des régions, pour en déduire les probabilités de collision, si l'on désire utiliser cette méthode pour résoudre l'équation de transport. Ces deux quantités pourront donc être évaluées comme suit :

$$V_{j,\Omega} = \int_{\Pi_{\hat{\Omega}}} \int_{-\infty}^{\infty} \mathcal{W}_{V_j}(s', \hat{\Omega}) ds' d^2p' \quad (3.37)$$

$$p_{ij,g} = \frac{1}{4\pi V_i} \int_{4\pi} \int_{-\infty}^{\infty} \left[\int_{\Pi_{\hat{\Omega}}} \int_{-\infty}^{\infty} \mathcal{W}_{V_j}(s, \hat{\Omega}) \mathcal{W}_{V_i}(s', \hat{\Omega}) \frac{e^{-\tau_g(s)}}{s^2} ds' d^2p' \right] ds d^2\Omega \quad (3.38)$$

En général, on définira les trajectoires de manière à découper l'espace, tant en angle et en densité linéaire (2-D) ou en angle solide et en densité surfacique (3-D), assez finement pour représenter suffisamment de trajectoires neutroniques. On découpera en premier lieu le domaine angulaire selon les directions, associant à chacune un poids, ainsi qu'un plan orthogonal, sur lequel les points de départ des trajectoires seront définis. Celles-ci seront par la suite analysées pour déterminer leurs intersections avec les domaines du problème. En connaissant les sections efficaces macroscopiques de chacune des régions, on peut alors explicitement calculer le parcours optique selon chacune des *tracks*, et donc les probabilités de collision.

En pratique, comme une intégration purement analytique ne peut être effectuée pour le processus de *tracking* de la géométrie, on procédera en choisissant un nombre fini de directions $\hat{\Omega}_m$, chacune associée à un poids ω_m tel que $\int_{4\pi} d^2\Omega = \sum \omega_m \hat{\Omega}_m = 4\pi$, ainsi qu'une densité surfacique sur chaque plan $\Pi_{\hat{\Omega}_m}$, chaque trajectoire étant aussi pondérée par un poids \mathcal{P}_n , tel que $\sum_n \mathcal{P}_n = 1$. Chacune des trajectoires croisera des régions identifiées par k (région V_i) ou h (région V_j). Une évaluation numérique des équations 3.37 et 3.38 peut donc être

effectuée :

$$V_{j,m} \simeq \sum_n \mathcal{P}_n \sum_h \delta_{h,j} l_h \quad (3.39)$$

$$p_{ij} \simeq \frac{1}{\sum_i \sum_j V_i} \sum_m \omega_m \sum_n \mathcal{P}_n \sum_k \delta_{k,i} \sum_h \delta_{h,j} e^{-\tau_{k,h}} [1 - e^{-\Sigma_i l_k}] [1 - e^{-\Sigma_j l_h}] \quad (3.40)$$

où $\delta_{i,j}$ est le delta de Kronecker qui vaut 1 si $i = j$ et 0 sinon, et l_k est la distance traversée par la trajectoire à l'intérieur du volume V_k .

Les volumes décrits à l'équation (3.39) sont en principe indépendants des directions $\hat{\Omega}$ utilisées pour les évaluer. En pratique, cependant, on peut songer à certaines géométries qui pourraient être mal évaluées numériquement par un tel procédé, notamment les géométries qui ne seront pas traversées par un nombre représentatif de trajectoires, comme des géométries de très petite dimension, ou des éléments très étendus dans une seule dimension. On moyenne donc les volumes obtenus numériquement en fonction de toutes les directions utilisées.

Les volumes peuvent aussi, en règle générale, être calculés analytiquement. Le module NXT: de *tracking* de DRAGON permet d'utiliser les deux méthodes, et de comparer les résultats entre eux. Une différence entre les deux méthodes peut vouloir dire, de deux choses, une densité (angulaire, surfacique ou généralement les deux) trop faible pour bien analyser la totalité de la géométrie, ou encore des erreurs dues à la précision numérique de la machine utilisée, lors de l'analyse de régions géométriques de très petites dimensions. Ainsi, l'analyse géométrique analytique des volumes sphériques est présentée dans le chapitre suivant, afin de permettre le traitement des géométries particulières présentées au chapitre précédent.

3.2.2 Méthode des caractéristiques

La méthode des caractéristiques (MOC, *method of characteristics*) (Askew, 1972), implantée par LeTellier dans DRAGON (LeTellier, 2006), se sert aussi, comme son nom l'indique, des trajectoires caractéristiques $\vec{T} = \vec{p} - s\hat{\Omega}$, ce qui en fait une méthode assez proche de la méthode des probabilités de collision. Nous laisserons ici de côté le formalisme multi-groupe afin d'alléger la notation. Pour commencer, considérons chaque trajectoire, qui s'étend de $-\infty$ jusqu'à ∞ . Elle rencontrera, sur une géométrie définie, un certain nombre de régions spatiales, qui découperont la droite en K régions, identifiées par l'indice $k = 1 \dots K$, qui auront une section efficace macroscopique totale Σ_k constante par morceaux sur les K segments, de longueurs respectives $l_k(\vec{T})$.

Nous utiliserons la forme caractéristique de l'équation de transport, à partir de (3.15) et (3.23) :

$$-\frac{d}{ds}\phi(\vec{p}-s\hat{\Omega},\hat{\Omega})+\Sigma(\vec{p}-s\hat{\Omega})\phi(\vec{p}-s\hat{\Omega},\hat{\Omega})=Q(\vec{p}-s\hat{\Omega},\hat{\Omega}) \quad (3.41)$$

Nous considérerons aussi le flux moyen sur un segment, défini comme :

$$\bar{\phi}_k(\vec{T})=\frac{1}{l_k(\vec{T})}\int_0^{l_k(\vec{T})}\phi(\vec{r}_k+s\hat{\Omega},\Omega)ds \quad (3.42)$$

où \vec{r}_k est défini comme le point de l'intersection de la caractéristique avec la frontière entrante de la région k , d'où la relation $\vec{r}_{k+1}=\vec{r}_k+l_k\hat{\Omega}$. En intégrant l'équation (3.41) en s sur un seul segment l_k , on obtiendra :

$$\phi_{k+1}(\vec{T})-\phi_k(\vec{T})+\Sigma_k l_k(\vec{T})\bar{\phi}_k(\vec{T})=l_k(\vec{T})Q_k(\vec{T}) \quad (3.43)$$

avec

$$Q_k(\vec{T})=\frac{1}{l_k(\vec{T})}\int_0^{l_k(\vec{T})}Q(\vec{r}_k+s\hat{\Omega},\hat{\Omega})ds \quad (3.44)$$

Cette équation nous pousse à poser une hypothèse sur la dépendance spatiale du terme source. L'approximation la plus répandue est celle du *step characteristic* (SC), qui suppose que le terme source est constant sur chaque segment de la trajectoire :

$$Q(\vec{r}_k+s\hat{\Omega},\hat{\Omega})=Q_k(\hat{\Omega})=\frac{Q_k}{4\pi} \quad \forall s \in [0, l_k] \quad (3.45)$$

dans le cas d'une source isotrope. En définissant le parcours optique τ qu'on peut expliciter comme $\tau_k(\vec{T})=\Sigma_k l_k(\vec{T})$ sur le segment k , on utilisera aussi un facteur intégrant $e^{-\tau_k(\vec{T})}$ comme précédemment, pour ensuite résoudre analytiquement l'équation. On obtient :

$$\phi_{k+1}(\vec{T})=e^{-\tau_k(\vec{T})}\phi_k(\vec{T})+\frac{Q_k(\vec{T})}{4\pi\Sigma_k}[1-e^{-\tau_k(\vec{T})}] \quad (3.46)$$

$$\bar{\phi}_k(\vec{T})=\frac{\phi_k(\vec{T})}{\tau_k(\vec{T})}[1-e^{-\tau_k(\vec{T})}]+\frac{Q_k(\vec{T})}{4\pi\Sigma_k}\left[1-\frac{1-e^{-\tau_k(\vec{T})}}{\tau_k(\vec{T})}\right] \quad (3.47)$$

Maintenant, en utilisant le flux moyenné dans le volume V_i (équation (3.31)), mais en utilisant les notations caractéristiques introduites au début de la présente section, on peut le redéfinir

avec un procédé similaire à l'équation (3.36) comme :

$$\begin{aligned}\phi_i &= \frac{1}{V_i} \int_{4\pi} \int_{V_i} \phi(\vec{r}) d^3r d^2\Omega \\ &= \frac{1}{V_i} \int_{\mathcal{T}} \int_{-\infty}^{\infty} \mathcal{W}_{V_i}(\vec{T}, s) \phi(\vec{p} - s\hat{\Omega}, \hat{\Omega}) ds d^4T\end{aligned}\quad (3.48)$$

On voit qu'on peut réécrire cette équation comme :

$$\phi_i = \frac{1}{V_i} \int_{\mathcal{T}} \sum_k \delta_{i,k} l_k(\vec{T}) \bar{\phi}(\vec{T}) d^4T \quad (3.49)$$

que l'on remplacera de nouveau par une somme pondérée sur les trajectoires comme à l'équation (3.40), de la manière suivante :

$$\phi_i = \frac{1}{V_i} \sum_m \omega_m \sum_n \mathcal{P}_n \sum_k \delta_{i,k} l_k(\vec{T}_m) \bar{\phi}(\vec{T}_m) \quad (3.50)$$

Ainsi, à l'aide des équations (3.46), (3.47) et (3.50), on peut évaluer le flux. Le processus demandera cependant d'établir des conditions frontières aux extrémités des caractéristiques sortant de la géométrie, qui ne seront pas présentées ici – le but étant surtout de démontrer l'utilité du processus de *tracking* – afin de pouvoir obtenir une solution complète. De même, encore une fois, à l'aide de (3.39), on peut évaluer numériquement les volumes et encore comparer avec ceux obtenus analytiquement. Il est évident, en observant les équations qu'il s'agira d'un processus itératif, notamment sur les termes source, afin d'obtenir le flux intégré sur chaque sous-volume.

La méthode des caractéristiques présente à la fois l'avantage d'utiliser le même procédé de *tracking* que la méthode des probabilités de collision, ce qui permet dans un code comme DRAGON d'utiliser le même objet de *tracking* pour les p_{ij} que pour MOC, et d'avoir un besoin en mémoire et en capacité de calcul de l'ordre de $n + l$ (n le nombre de régions et l le nombre de surfaces), comparativement à $n \times n$ pour les p_{ij} . Par contre, au lieu d'utiliser une sommation directe spatiale comme les PC, elle implique un processus itératif, qui requiert de relire en entier les informations de *tracking* à chaque itération, ce qui peut soit être long, soit nécessiter un espace de mémoire vive très important - d'où la nécessité d'implanter des méthodes d'accélération pour la convergence des itérations.

3.2.3 Méthodes Monte-Carlo

Les méthodes Monte-Carlo font partie des méthodes de calcul stochastiques, car elles font appel à un générateur de nombres aléatoire (généralement pseudo-aléatoire), afin d'évaluer le comportement de la population neutronique. On sélectionne un isotope fissile qui génère un neutron dans un certain milieu. Puis, on lui assigne aléatoirement une direction angulaire, ainsi qu'une énergie en fonction du spectre d'émission de l'isotope. On simule par la suite son comportement dans la géométrie, en fonction des milieux qu'il rencontre, des conditions frontières définies, et des collisions qui peuvent survenir.

Ces méthodes ont l'avantage majeur de pouvoir représenter presque n'importe quel niveau de complexité de géométrie et de propriétés matérielles, sans avoir besoin de discrétiser la géométrie, et peuvent aussi, si on leur donne un nombre assez important d'« historiques » de neutrons à simuler, être extrêmement précises. Par contre, elles requièrent pour ce faire, un temps machine et des capacités de calcul très importantes. Pour cette raison, elles servent généralement à effectuer des calculs de référence et à valider les codes déterministes.

Les méthodes Monte-Carlo n'utilisent donc pas un *tracking* tel que défini pour les méthodes déterministes comme PC ou MOC. Par contre, un des calculs les plus fréquemment considérés lors d'une simulation est de déterminer le libre parcours du neutron dans le milieu, et cela implique généralement de trouver la distance :

- jusqu'à toutes les surfaces dans la direction de la trajectoire ;
- jusqu'à la prochaine collision, en fonction de la section efficace macroscopique totale.

La plus courte de ces distances décidera de la suite du comportement du neutron. Certaines méthodes ont été développées pour les géométries à caractère aléatoire permettant d'accélérer le calcul des distances, en supposant des subdivisions virtuelles de l'espace, pour calculer ensuite les distances entre les surfaces de ce sous-volume et le neutron (Reinert, 2008). Cependant, il faudra toujours calculer les distances avec certains éléments de volume réels, d'où l'intérêt de la routine permettant de calculer les distances entre les éléments sphériques et une position arbitraire.

Une routine a aussi été développée afin de permettre la génération aléatoire de positions (voir section 4.2), et d'en garder la position en mémoire pour ensuite l'analyser, ce qui pourra être effectué par des méthodes soit déterministes, soit stochastiques.

Chapitre 4

ANALYSE GÉOMÉTRIQUE

Pour résoudre le problème de la double hétérogénéité présentée précédemment, il faut, en plus de la génération séquentielle aléatoire des positions des sphères, ajouter une capacité de traitement des géométries sphériques au logiciel DRAGON. L'analyse géométrique contiendra donc trois sections, soit une concernant l'analyse des volumes des sphères en tant que telles, une concernant la distribution spatiale des sphères, et une concernant le tracking proprement dit.

4.1 Analyse des sphères

L'analyse des sphères se base sur le même principe que l'analyse 2-D des disques, telle que démontrée par Marleau (Marleau, 2005) et par Zkiek (Zkiek et Marleau, 2007). Le modèle tient compte successivement des huit coins du parallélépipède formé par les intersections mutuelles des six plans dont on désire trouver l'intersection avec la sphère. On trouve par la suite le volume de la sphère situé sous (z), derrière (y) et à gauche (x) de l'intersection des trois plans qui forment le coin considéré du parallélépipède. Le volume total de l'intersection sera donné par une combinaison linéaire des volumes ainsi calculés.

Nous considérerons une sphère centrée à (x_c, y_c, z_c) , de rayon R , traversée par des plans x_+, x_-, y_+, y_-, z_+ et z_- , ou les indices $+$ et $-$ représentent respectivement le plan situé dans la direction croissante de l'axe par rapport au centre du parallélépipède, et celui dans la direction décroissante.

Le coin (x_+, y_+, z_+) aura une contribution positive au volume total de l'intersection, enveloppant tout le volume de l'intersection, mais aussi le reste de la sphère. Les coins (x_-, y_+, z_+) , (x_+, y_-, z_+) et (x_+, y_+, z_-) auront tous une contribution négative, afin d'enlever les volumes qui n'ont pas à être considérés dans l'intersection. Les volumes associés aux coins (x_-, y_-, z_+) , (x_-, y_+, z_-) et (x_+, y_-, z_-) ayant été soustraits deux fois chacun, auront une contribution positive. Finalement, le volume associé au coin (x_-, y_-, z_-) devra être soustrait, ayant été soustrait trois fois et ajouté quatre fois lors des opérations précédentes. Le volume total de

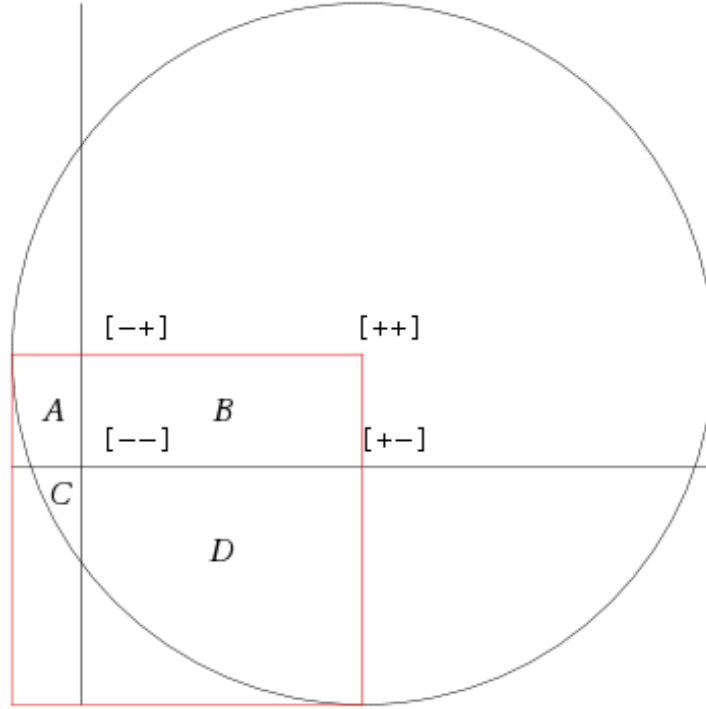


FIGURE 4.1 Analogie 2-D de la méthode de calcul des volumes

l'intersection entre le parallélépipède et la sphère pourra donc être calculé comme suit :

$$V_{\text{Inter}} = V_{[+,+,+]} - V_{[-,+,+]} - V_{[+,-,+]} - V_{[+,+,-]} + V_{[-,-,+]} + V_{[-,+,-]} + V_{[+,-,-]} - V_{[-,-,-]} \quad (4.1)$$

On peut visualiser cette méthode en considérant une analogie simplifiée en 2-D, présenté à la figure 4.1, en considérant évidemment les surfaces au lieu des volumes.

On voit ici que si l'on connaît les surfaces définies par $S_{[+,+]}$, $S_{[+,-]}$, $S_{[-,+]}$ et $S_{[-,-]}$ et qu'on désire calculer la surface du rectangle identifié par la lettre B , on considèrera la surface $S_{[+,+]}$, qui ici représente les surfaces A , B , C et D réunies. On soustraira ensuite de ce total les surfaces $S_{[+,-]}$ et $S_{[-,+]}$ qui représentent respectivement les sommes $C + D$ et $A + C$. Il faudra finalement ajouter la surface C , qui est ici représentée par $S_{[-,-]}$. On trouve donc l'analogie 2-D de l'équation (4.1) : $B = S_{[+,+]} - S_{[+,-]} - S_{[-,+]} + S_{[-,-]}$.

La méthode de calcul de chacun des sous-volumes constituant V_{Inter} différera selon le type d'intersection : Le cas le plus simple survient lorsqu'aucun des trois plans du coin considéré n'intersecte la sphère. Dans ce cas, le sous-volume correspondant au coin sera nul dans tous les cas sauf si l'intersection survient dans l'octant $[+, +, +]$ d'une sphère située à l'origine,

auquel cas nous aurons $V_{(x,y,z)} = V_S$, où $V_S = \frac{4\pi R^3}{3}$ est le volume total de la sphère. Dans tous les autres cas, il faudra analyser plus en détail les types d'intersection qui surviennent. Pour chaque coin, on pourra en distinguer trois types :

- Intersection d'un ou plusieurs plans avec la sphère, mais sans intersection entre les plans à l'intérieur de la sphère ;
- Une, deux ou trois intersections de deux plans entre eux à l'intérieur de la sphère, mais l'intersection des trois plans se situant à l'extérieur de la sphère ;
- Intersection des trois plans à l'intérieur de la sphère.

4.1.1 Pas d'intersection des plans à l'intérieur de la sphère

Le cas le plus simple survient lorsqu'aucune intersection ne survient entre les plans à l'intérieur de la sphère, les plans pouvant cependant intersecter individuellement la sphère, tant que la condition suivante est respectée :

$$\exists (j_p^2 + k_p^2) \in [(x_p, y_p), (x_p, z_p), (y_p, x_p)] \mid (j_p^2 + k_p^2) > R^2 \wedge (|j_p| < R \vee |k_p| < R) \quad (4.2)$$

où x_p, y_p , et z_p représentent les positions des trois plans considérés, respectivement selon les axes X, Y et Z . Le volume à calculer n'est alors qu'une coupe de la sphère par un plan, telle qu'illustrée à la figure 4.2. En définissant V_i comme le volume situé sous (z), derrière (y) ou à gauche (x) du plan $i = i_p$, et en prenant $h = R - i_p$, on peut trouver le volume au dessus du plan, calculé de la façon suivante :

$$\begin{aligned} V_{Cap} &= \int_{R-h}^R \pi(R^2 - z^2) dz \\ &= \pi \left[R^2 z - \frac{z^3}{3} \right]_{R-h}^R \\ &= \frac{\pi h^2(3R - h)}{3} \end{aligned} \quad (4.3)$$

Le volume désiré, V_i , est alors simplement :

$$\begin{aligned} V_i &= V_S - V_{Cap_i} \\ &= \frac{\pi}{3} [4R^3 - 3Rh^2 + h^3] \quad \text{pour } 0 \leq h \leq 2R \end{aligned} \quad (4.4)$$

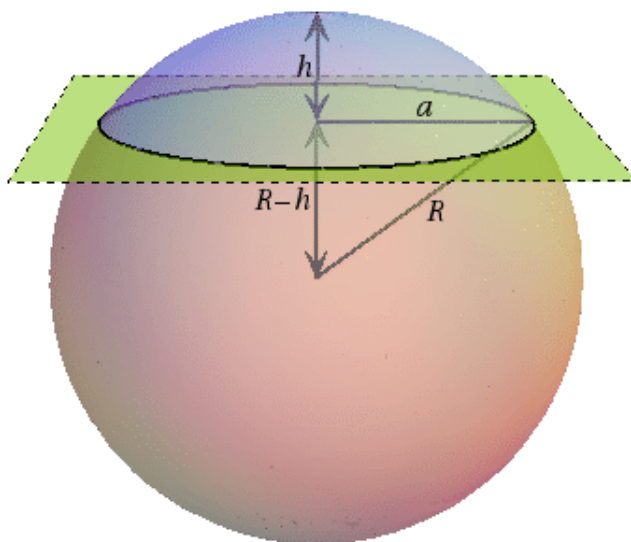


FIGURE 4.2 Intersection d'un plan avec la sphère (ici, $R^2 = a^2 + z^2$)

D'où :

$$V_i = \begin{cases} \frac{4}{3}\pi R^3 & \text{si } h \leq 0 \\ \frac{\pi}{3} [4R^3 - 3Rh^2 + h^3] & \text{si } 0 < h < 2R \\ 0 & \text{si } h \geq 2R \end{cases} \quad (4.5)$$

Le volume recherché est alors une combinaison des V_i selon x, y ou z , dépendant de la position de l'intersection des trois plans. Le tableau 4.1 donne les volumes à considérer en fonction de la position de l'intersection des trois plans. Dans celui-ci, les notations utilisées pour les indices désignent la position des plans par rapport au centre de la sphère, et représentent le signe de $(x_p - x_c)$, $(y_p - y_c)$ ou $(z_p - z_c)$. Par exemple, pour une sphère dont le centre est situé en $(1, -2, 3)$ intersectée par des plans en $x_p = 0, y_p = -1$ et $z_p = 2$, on aura l'intersection $[-, +, -]$.

4.1.2 Intersection de deux des trois plans à l'intérieur de la sphère

L'intersection de deux plans à l'intérieur de la sphère crée la région illustrée à la figure 4.3. Pour calculer le volume d'une telle intersection, nous considérerons le volume V_i défini précédemment à l'équation 4.5, ainsi que le volume identifié par V_{Int} sur la figure 4.3. Nous

TABLEAU 4.1 Volumes selon la position des plans - Aucune intersection dans la sphère

Position des plans par rapport au centre de la sphère	$V_{(x,y,z)}$
[+, +, +]	$V_x + V_y + V_z - 2V_S$
[-, +, +]	V_x
[+, -, +]	V_y
[+, +, -]	V_z
[+, -, -]	0
[-, +, -]	
[-, -, +]	
[-, -, -]	

utiliserons ici une sphère centrée à l'origine pour simplifier les calculs. Les volumes calculés seront tout de même valides, si l'on applique une translation équivalente aux coordonnées des plans. Ce volume pourra être calculé de la manière suivante :

$$\begin{aligned}
V_{Int(x,y)} &= \int_0^{y_p} \int_0^{x_p} \sqrt{R^2 - x^2 - y^2} \, dx dy \\
&= \int_0^{y_p} \frac{x_p}{2} \sqrt{R^2 - x_p^2 - y^2} + \frac{R^2 - y^2}{2} \arctan \left(\frac{x_p}{\sqrt{R^2 - x_p^2 - y^2}} \right) dy \\
&= \frac{1}{6} \left[2x_p y_p \sqrt{R^2 - x_p^2 - y_p^2} \right. \\
&\quad - y_p (y_p^2 - 3R^2) \arctan \left(\frac{x_p}{\sqrt{R^2 - x_p^2 - y_p^2}} \right) \\
&\quad - x_p (x_p^2 - 3R^2) \arctan \left(\frac{y_p}{\sqrt{R^2 - x_p^2 - y_p^2}} \right) \\
&\quad \left. - 2R^3 \arctan \left(\frac{x_p y_p}{R \sqrt{R^2 - x_p^2 - y_p^2}} \right) \right] \quad (4.6)
\end{aligned}$$

Notons que ce volume $V_{Int(x,y)}$ a une valeur nulle, telle que prévue, lorsque $x_p = 0$ ou $y_p = 0$, et aura une valeur négative lorsque seulement $x_p < 0$ ou $y_p < 0$. Dans ce dernier cas, on ne parlera pas de volume ayant une signification physique, mais remarquons que $\|V_{Int(x,y)}\|$ représentera le volume à considérer lorsque l'intersection survient dans l'un des deux quadrants menant à ces valeurs. On peut aussi, de manière équivalente, trouver $V_{Int(x,z)}$ et $V_{Int(y,z)}$, ou, de manière plus générale, $V_{Int(i,j)}$.

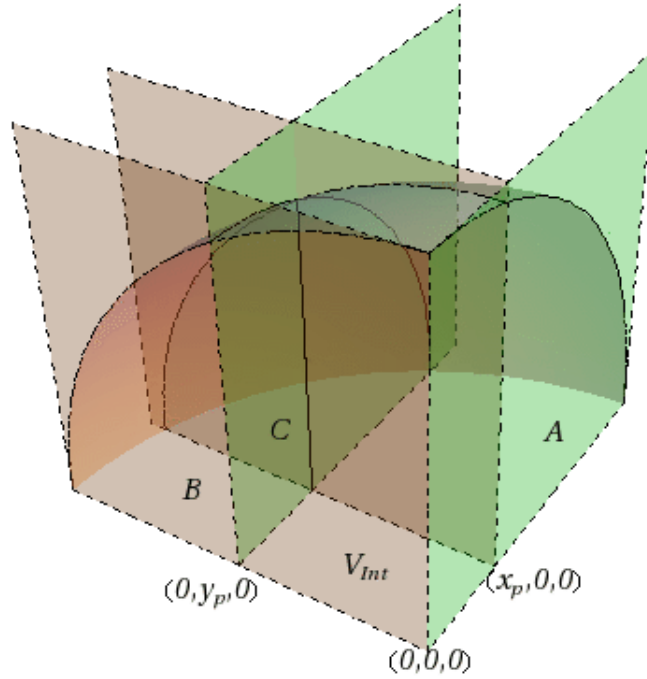


FIGURE 4.3 Volume à intégrer

On obtient, en observant les volumes V_A , V_B et V_C des régions représentées respectivement par les lettres A , B et C illustrées à la figure 4.3 :

$$\frac{V_{Sph}}{8} = V_A + V_B + V_C + V_{Int} \quad (4.7a)$$

$$V_{(x,y)} = 2 \times V_C \quad (4.7b)$$

$$\frac{V_x}{4} = V_A + V_C \quad (4.7c)$$

$$\frac{V_y}{4} = V_B + V_C \quad (4.7d)$$

d'où :

$$\frac{V_{Sph}}{8} = \frac{V_x}{4} + \frac{V_y}{4} + V_{Int(x,y)} - \frac{V_{(x,y)}}{2} \quad (4.8)$$

On trouve donc, en généralisant pour les axes i et j :

$$V_{(i,j)} = 2V_{Int(i,j)} + \frac{V_i + V_j}{2} - \frac{V_S}{4} \quad (4.9)$$

On peut vérifier que cette équation est valide pour les positions de i_p et j_p peu importe l'axe considéré, ou le signe de i_p ou de j_p .

Une seule intersection

En fonction de la position des plans entre lesquels survient l'intersection illustrée à la figure 4.4, le volume à considérer ne sera évidemment pas le même. Par simple observation, on peut définir quelle combinaison des volumes aura à être utilisée. Le tableau 4.2 donne la technique pour calculer le volume en fonction de la position des plans par rapport au centre de la sphère, en utilisant la même notation que pour le tableau 4.1.

Deux intersections

Lorsque deux intersections surviennent entre deux plans à l'intérieur de la sphère, tel qu'illustré à la figure 4.5, les volumes à considérer sont donnés par le tableau 4.3.

Trois intersections

Finalement, lorsqu'il y a intersection entre chaque couple de plans possible, sans pour autant que l'intersection des trois plans se situe à l'intérieur de la sphère, tel que montré à la figure 4.6, le volume situé sous, derrière et à gauche des trois plans sera donné tel que décrit au tableau 4.4.

4.1.3 Intersection des trois plans à l'intérieur de la sphère

L'intersection de trois plans à l'intérieur de la sphère, représentée à la figure 4.7, crée un volume $V_{(x,y,z)}$ qui peut être calculé en utilisant les volumes déjà définis, V_i et $V_{Int(i,j)}$. En considérant une sphère centrée à l'origine, avec l'intersection des plans située dans l'octant $[-,-,-]$ – donc $x_p < 0, y_p < 0$ et $z_p < 0$ – on peut voir, à l'aide des régions de A à H identifiées à la figure 4.8 que le volume à calculer $V_{(x,y,z)}$ peut être exprimé en commençant

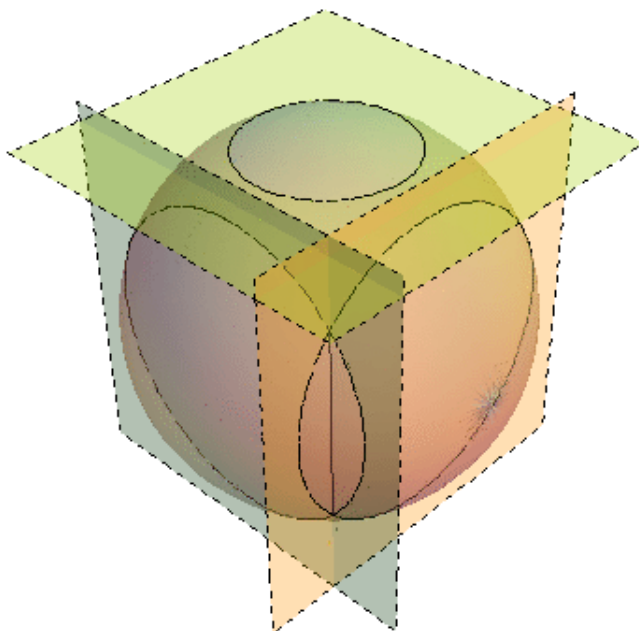


FIGURE 4.4 Intersection de deux plans à l'intérieur de la sphère

TABLEAU 4.2 Volumes selon la position des plans - Une seule intersection de deux plans dans la sphère

Position du plan par rapport au centre de la sphère	$V_{(x,y,z)}$		
	(y, z)	(x, z)	(x, y)
$[+, +, +]$	$V_{(y,z)} + V_x - V_S$	$V_{(x,z)} + V_y - V_S$	$V_{(x,y)} + V_z - V_S$
$[-, +, +]$	V_x	$V_{(x,z)}$	$V_{(x,y)}$
$[+, -, +]$	$V_{(y,z)}$	V_y	$V_{(x,y)}$
$[+, +, -]$	$V_{(y,z)}$	$V_{(x,z)}$	V_z
$[+, -, -]$	$V_{(y,z)}$	0	0
$[-, +, -]$	0	$V_{(x,z)}$	0
$[-, -, +]$	0	0	$V_{(x,y)}$
$[-, -, -]$	0		

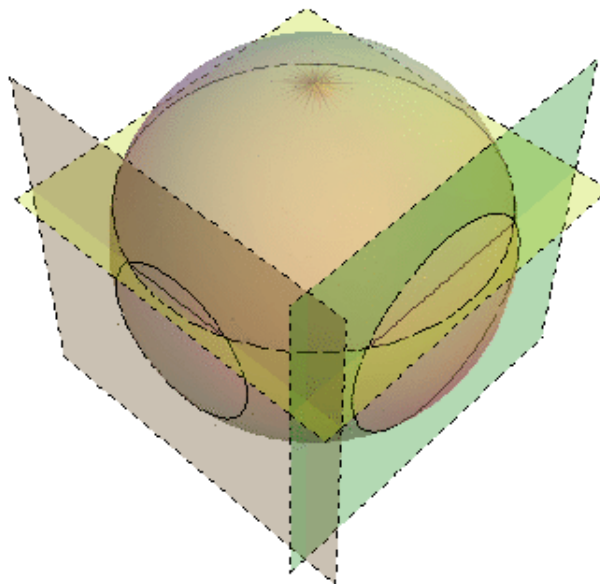


FIGURE 4.5 Deux intersections de deux plans à l'intérieur de la sphère

TABLEAU 4.3 Volumes selon la position des plans - Deux intersections de deux plans à l'intérieur de la sphère

Position du plan par rapport au centre de la sphère	$V_{(x,y,z)}$		
	$(x, y), (x, z)$	$(x, y), (y, z)$	$(x, z), (y, z)$
[+, +, +]	$V_{(x,y)} + V_{(x,z)} - V_x$	$V_{(x,y)} + V_{(y,z)} - V_y$	$V_{(x,z)} + V_{(y,z)} - V_z$
[-, +, +]	$V_{(x,y)} + V_{(x,z)} - V_x$	$V_{(x,y)}$	$V_{(x,z)}$
[+, -, +]	$V_{(x,y)}$	$V_{(x,y)} + V_{(y,z)} - V_y$	$V_{(y,z)}$
[+, +, -]	$V_{(x,z)}$	$V_{(y,z)}$	$V_{(x,z)} + V_{(y,z)} - V_z$
[+, -, -]	0	$V_{(y,z)}$	$V_{(y,z)}$
[-, +, -]	$V_{(x,z)}$	0	$V_{(x,z)}$
[-, -, +]	$V_{(x,y)}$	$V_{(x,y)}$	0
[-, -, -]		0	

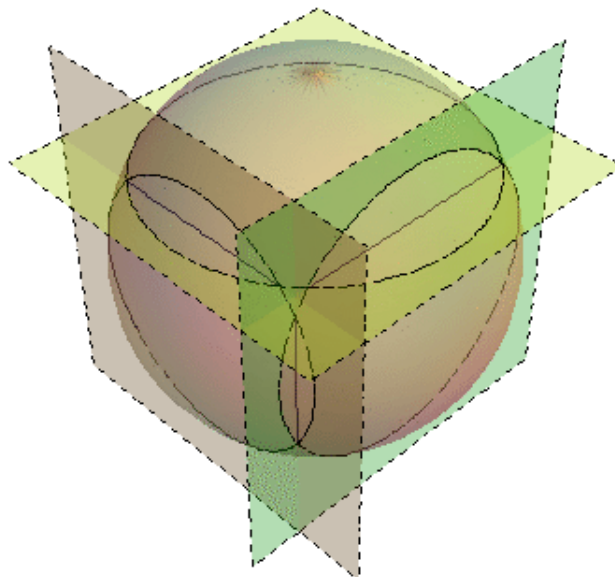


FIGURE 4.6 Trois intersections de deux plans à l'intérieur de la sphère

TABLEAU 4.4 Volumes selon la position des plans - Trois intersections de deux plans à l'intérieur de la sphère

Position du plan par rapport au centre de la sphère	$V_{(i,j,k)}$
[+, +, +]	$V_{(x,y)} + V_{(x,z)} + V_{(y,z)} - V_x - V_y - V_z + V_S$
[-, +, +]	$V_{(x,y)} + V_{(x,z)} - V_x$
[+, -, +]	$V_{(x,y)} + V_{(y,z)} - V_y$
[+, +, -]	$V_{(x,z)} + V_{(y,z)} - V_z$
[+, -, -]	$V_{(y,z)}$
[-, +, -]	$V_{(x,z)}$
[-, -, +]	$V_{(x,y)}$
[-, -, -]	0

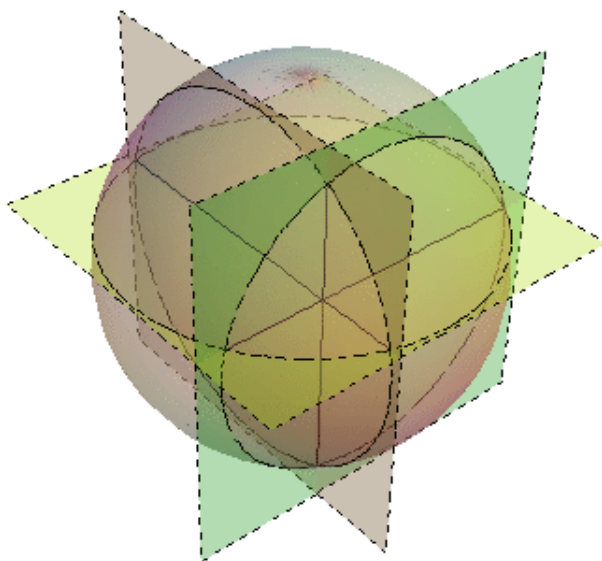


FIGURE 4.7 Intersection des trois plans à l'intérieur de la sphère

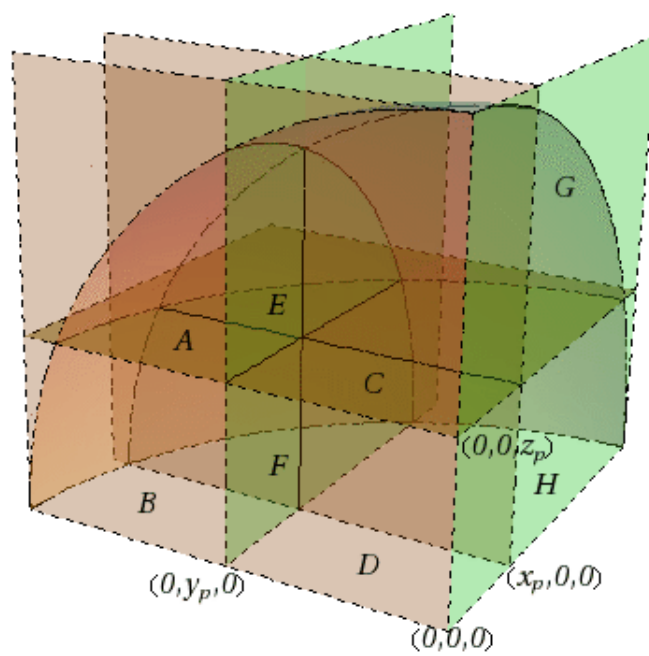


FIGURE 4.8 Volumes découpés par l'intersection des trois plans à l'intérieur de la sphère

par établir les relations suivantes :

$$V_{Int(x,z)} = V_B + V_D \quad (4.10a)$$

$$V_{Int(x,y)} = V_C + V_D \quad (4.10b)$$

$$V_{Int(y,z)} = V_H + V_D \quad (4.10c)$$

$$\frac{V_x}{4} = V_E + V_F + V_G + V_H \quad (4.10d)$$

$$\frac{V_y}{4} = V_A + V_B + V_E + V_F \quad (4.10e)$$

$$\frac{V_z}{4} = V_A + V_C + V_E + V_G \quad (4.10f)$$

$$\frac{V_S}{8} = V_A + V_B + V_C + V_D + V_E + V_F + V_G + V_H \quad (4.10g)$$

En additionnant les équations (4.10a) à (4.10f), et à l'aide de (4.10g) on obtient :

$$2\frac{V_S}{8} + V_D + V_E = \frac{V_x + V_y + V_z}{4} + V_{Int(x,y)} + V_{Int(x,z)} + V_{Int(y,z)} \quad (4.11)$$

avec :

$$x_p y_p z_p = -V_D \quad (4.12)$$

$$V_{(x,y,z)} = V_E \quad (4.13)$$

on trouve :

$$2\frac{V_S}{8} = \frac{V_x + V_y + V_z}{4} + x_p y_p z_p - V_{(x,y,z)} + V_{Int(x,y)} + V_{Int(x,z)} + V_{Int(y,z)} \quad (4.14)$$

d'où :

$$V_{(x,y,z)} = x_p y_p z_p + V_{Int(x,y)} + V_{Int(x,z)} + V_{Int(y,z)} + \frac{V_x + V_y + V_z - V_S}{4} \quad (4.15)$$

On peut aussi vérifier que cette relation est valide pour tous les octants, donc pour tous les x_p, y_p et z_p , tant que $x_p^2 + y_p^2 + z_p^2 < R^2$.

4.2 Distribution spatiale des sphères

Afin de pouvoir effectuer une comparaison adéquate avec la méthode de la double hétérogénéité, il est nécessaire de pouvoir générer stochastiquement des sphères dans la cellule pour

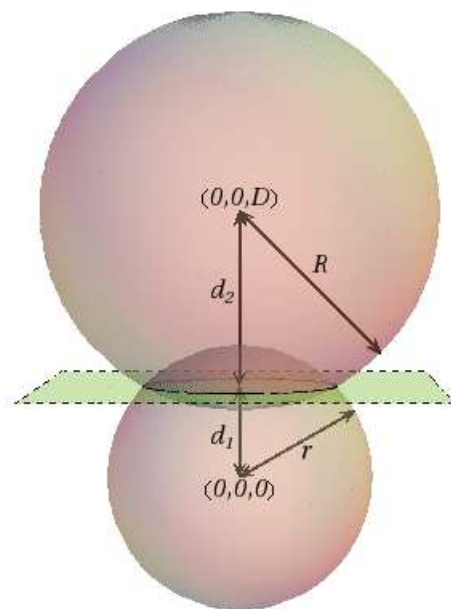


FIGURE 4.9 Intersection de deux sphères

pouvoir par la suite les analyser de façon déterministe. Pour ce faire, nous avons utilisé la fonction `RANDF` de la `GANLIB` (Roy et Hébert, 2000). Il s'agit d'un générateur de nombres pseudo-aléatoires, disponible notamment pour des calculs Monte-Carlo à l'aide de `DRAGON`, afin de générer les positions des centres des sphères aléatoirement à l'intérieur des limites de la cellule.

Afin de pouvoir éventuellement utiliser ce module pour une analyse des réacteurs à lits de boulets, il sert aussi à tester la superposition des particules sphériques et permet de calculer le volume correspondant à l'intersection de deux sphères. Cette intersection n'est en fait que la somme des volumes des deux coupes V_i des sphères par le plan contenant le cercle de l'intersection, tel qu'illustré à la figure 4.9.

On peut aisément calculer la distance entre le plan d'intersection et le centre des cercles, et donc la hauteur des coupes sphériques, en prenant D comme la distance entre les centres des sphères, et r et R les rayons :

$$R^2 = x^2 + y^2 + (z - D)^2 \quad (4.16)$$

$$r^2 = x^2 + y^2 + z^2 \quad (4.17)$$

en combinant, on obtient :

$$R^2 - r^2 = (z - D)^2 - z^2 = D^2 - 2Dz \quad (4.18)$$

ce qui donne, pour la première section :

$$h_1 = r - d_1 = r - \frac{D^2 - R^2 + r^2}{2D} = \frac{(R - r + D)(R + r - D)}{2D} \quad (4.19)$$

et pour la seconde :

$$h_2 = R - d_2 = R - (D - d_1) = \frac{(r - R + D)(r + R - D)}{2D} \quad (4.20)$$

On trouve donc le volume d'intersection total, à l'aide de (4.3)

$$\begin{aligned} V_{Int} &= V_{cap1} + V_{cap2} \\ &= \frac{\pi(3rh_1^2 - h_1^3)}{3} + \frac{\pi(3Rh_2^2 - h_2^3)}{3} \\ &= \frac{\pi(R + r - D)^2 [D^2 + 2D(r + R) - 3(r - R)]}{12D} \end{aligned} \quad (4.21)$$

Généralement, dans le cas qui nous intéresse, les sphères auront le même rayon $r = R$, d'où un volume d'intersection de :

$$V_{Int} = \frac{\pi(2R + D)^2(D + 4R)}{12} \quad (4.22)$$

4.3 Lignes d'intégration

Le module de tracking cherche à établir le chemin optique en fonction de trajectoires prédéfinies selon des orientations discrétisées avec une certaine densité surfacique de lignes d'intégration. Les milieux ensuite traversés par ces lignes d'intégration fourniront au chemin optique une certaine contribution en fonction de la composition du milieu et de la longueur traversée par le segment.

Afin de trouver ces distances lorsque le milieu considéré présente des sphères, il importe de déterminer l'intersection entre une droite (la ligne d'intégration) et une sphère. En termes d'algèbre linéaire, la distance pourra être aisément calculée en connaissant les points d'intersection de la droite avec la coquille sphérique délimitant le milieu considéré. Ces points

d'intersection peuvent être déterminés en considérant l'équation d'une sphère et l'équation paramétrique d'une droite :

$$\text{Droite : } \quad \frac{x - x_1}{a_1} = \frac{y - y_1}{a_2} = \frac{z - z_1}{a_3} = t \quad (4.23)$$

$$\text{Sphère : } \quad (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = R^2 \quad (4.24)$$

où (x_0, y_0, z_0) est la position du centre de la sphère, R son rayon, (x_1, y_1, z_1) est un point sur la droite, et (a_1, a_2, a_3) est un vecteur parallèle à la cette dernière – nous utiliserons les cosinus directeurs.

En combinant les équations (4.23) et (4.24), on obtient une équation quadratique aisément soluble permettant d'obtenir les coordonnées des points d'intersection :

$$\begin{aligned} 0 &= (a_1 t + x_1 - x_0)^2 + (a_2 t + y_1 - y_0)^2 + (a_3 t + z_1 - z_0)^2 - R^2 \\ &= At^2 + Bt + C \end{aligned} \quad (4.25)$$

qu'on peut résoudre pour t , avec :

$$A = a_1^2 + a_2^2 + a_3^2 \quad (4.26a)$$

$$B = 2(a_1 x_1 - a_1 x_0 + a_2 y_1 - a_2 y_0 + a_3 z_1 - a_3 z_0) \quad (4.26b)$$

$$C = x_0^2 + y_0^2 + z_0^2 + x_1^2 + y_1^2 + z_1^2 - 2x_0 x_1 - 2y_0 y_1 - 2z_0 z_1 - R^2 \quad (4.26c)$$

Pour trouver les coordonnées d'intersection grâce à (4.23) :

$$[x_{i_1}, y_{i_1}, z_{i_1}] = [t_1 a_1 + x_1, t_1 a_2 + y_1, t_1 a_3 + z_1] \quad (4.27a)$$

$$[x_{i_2}, y_{i_2}, z_{i_2}] = [t_2 a_1 + x_1, t_2 a_2 + y_1, t_2 a_3 + z_1] \quad (4.27b)$$

4.4 Implémentation

Des routines FORTRAN (présentées à l'annexe A) ont été écrites pour effectuer les calculs présentés dans ce chapitre, et validées avec des routines équivalentes en *Matlab* (MATLAB, 2006), puis intégrées dans le module NXT: de DRAGON, afin de permettre l'analyse des géométries sphériques. Une étape de vérification et de validation a par la suite été effectuée, et sera présentée dans le chapitre suivant.

Chapitre 5

RÉSULTATS

Afin de tester l'implémentation des nouvelles routines (présentées à l'annexe A) dans le module `NXT`, une série de cas et de données ont été analysés. Les premiers tests effectués concernaient une seule sphère centrée à l'origine dans une cellule unitaire, elle aussi centrée à l'origine.

Les tests ont continué en augmentant la complexité afin de déceler les erreurs potentielles dans l'intégration des nouvelles routines, en augmentant donc, séquentiellement, la complexité des cas, notamment en déplaçant le centre de la sphère de l'origine, en ajoutant des couches à la sphère, en ajoutant un maillage cartésien dans la cellule, en ajoutant plus d'une sphère dans le *cluster*, et en ajoutant plus d'un *cluster* dans la géométrie.

À chaque étape de la vérification, un échantillon de quelques lignes a été choisi au hasard dans l'ensemble de l'élément *tracking* et analysées afin d'en assurer la validité, en plus des volumes analysés analytiquement, qui ont été vérifiés, et comparés aux volumes calculés numériquement par le *tracking*.

Un test a été effectué, où une comparaison a été effectuée entre une cellule cartésienne contenant des *pins* cartésiennes et une contenant des *pins* sphériques. De plus, une validation préliminaire consistant en une comparaison, sur une géométrie simple, des résultats obtenus à l'aide de la nouvelle méthode à ceux obtenus par des méthodes Monte-Carlo et par deux méthodes de traitement de la double hétérogénéité déjà disponibles dans `DRAGON` a été effectuée. Ce chapitre présentera les résultats de la vérification et de la validation.

5.1 Utilisation des géométries sphériques

Une nouvelle option dans le module `GEO` a été ajoutée afin de pouvoir considérer les géométries de type `CLUSTER` sphériques. Le format pour définir les géométries sphériques est présenté à l'annexe B. Mentionnons seulement que les nouvelles options permettent de définir explicitement les positions de pins sphériques dans la cellule à l'aide des mots clés `CPINX`,

CPINY et CPINZ, ou de générer aléatoirement les positions lui-même à l'aide du mot-clé DPIN. Cette dernière option calcule un nombre de *pins* N_p selon la formule :

$$N_p = \text{MAX} \left(1, \text{NINT} \left(\frac{d_{pin} V_c}{V_p} \right) \right) \quad (5.1)$$

avec la densité de pins d_{pin} , le volume de la cellule V_c et le volume d'une *pin* V_p . Le module génère par la suite aléatoirement les positions de leurs centres séquentiellement, en testant à chaque nouvelle position générée la superposition avec les *pins* précédentes.

Ces options ont été ajoutées et testées pour l'utilisation avec le module NXT:, lui permettant donc le traitement de géométries sphériques tri-dimensionnelles.

5.2 Vérification des volumes analytiques

Les vérifications faites dans cette section l'ont été à l'aide de l'option NOTR de NXT:, qui permet d'effectuer seulement l'analyse de la géométrie, sans *tracking*, et, par la suite, en observant l'objet créé, on peut lire les volumes calculés. Les schémas de numérotation utilisés tant dans cette section que dans la programmation des routines sont cohérents avec ceux utilisés pour les géométries cylindriques déjà présentes dans NXT:, c'est à dire :

1. radialement, de l'intérieur vers l'extérieur ;
2. dans l'ordre croissant de X ;
3. dans l'ordre croissant de Y ;
4. dans l'ordre croissant de Z.

5.2.1 Cellule unitaire

Le premier cas, trivial, d'une seule sphère comprise dans une cellule cartésienne unitaire calcule les volumes bien établis, d'une sphère et du restant du cube dans lequel elle est située. On vérifiera aussi les volumes créés par l'imbrication de coquilles sphériques les unes à l'intérieur des autres. On considère donc une cellule cartésienne cubique ($1 \times 1 \times 1$), de côté 2 cm, avec une sphère centrée dans celle-ci, composée de 8 couches. Les résultats de l'analyse géométrique et les rayons sont présentés dans le tableau 5.1, mis en relations avec les volumes réels, calculés extérieurement en double précision. Ces résultats ont été testés avec différentes positions de la sphère dans la cellule, et avec d'autres combinaisons de rayons, non présentées ici.

TABLEAU 5.1 Volumes calculés analytiquement par NXT :

Région	Rayon (cm)	Volume NXT : (cm ³)	Volume réel (cm ³)	Différence Relative
1	0.10	0.004188790392	0.004188790205	4.47E-08
2	0.15	0.009948378234	0.009948376736	1.51E-07
3	0.17	0.006442358299	0.006442359335	1.61E-07
4	0.20	0.01293079621	0.01293079536	6.56E-08
5	0.37	0.1786644753	0.1786644686	3.75E-08
6	0.40	0.05590778664	0.05590778286	6.76E-08
7	0.50	0.2555161905	0.2555162025	4.69E-08
8	0.80	1.621061905	1.621061809	5.91E-08
9	Cartésienne (2.0 × 2.0 × 2.0)	5.8553393193	5.8553394151	1.64E-08
Total		7.999999999875	8.000000000000	1.56E-11

Les différences observées entre les volumes calculés de manière extérieure peuvent être expliquées par la précision numérique impliquée, car bien que les calculs dans NXT: soient effectués en double précision, la lecture dans le fichier d'entrée et le stockage des données se fait en simple précision. On observe donc des différences de l'ordre de la simple précision numérique, avec une différence absolue maximale sur les valeurs obtenue pour la région cartésienne, d'une valeur de $9.58 \times 10^{-9} \text{cm}^{-3}$.

5.2.2 Maillage cartésien dans la cellule

En deuxième lieu, des tests ont été effectués en imposant un maillage cartésien dans la cellule, afin de vérifier la validité des volumes calculés lors de l'intersection d'une *pin* sphérique avec les régions cartésiennes. Pour ce faire, nous avons utilisé le maillage présenté au tableau 5.2, afin d'avoir un échantillon représentatif des types d'intersections qui peuvent survenir, avec un total de 27 sous-régions cartésiennes, certaines étant amputées d'une section de par la présence de la *pin* sphérique, et une région sphérique.

TABLEAU 5.2 Maillage pour vérification des volumes d'intersection

Direction	Position des surfaces
X	-1.0 -0.7 0.6 1.0
Y	-1.0 -0.7 0.5 1.0
Z	-1.0 -0.7 0.4 1.0
R	0.9

TABLEAU 5.3 Volumes calculés analytiquement par NXT : (Cellule $3 \times 3 \times 3$)

Région	Volume NXT : (cm ³)	Volume réel (cm ³)	Différence Relative
1 (Sphérique)	3.05362773	3.05362806	1.08E-07
2	0.0270000026	0.0270000000	9.63E-08
3	0.117000014	0.117000000	1.20E-07
4	0.0360000022	0.0360000000	6.11E-08
5	0.108000010	0.108000000	9.26E-08
6	0.363746703	0.363746656	1.29E-07
7	0.143999994	0.144000000	4.17E-08
8	0.0450000018	0.0450000000	4.00E-08
9	0.194533601	0.194533589	6.20E-08
10	0.0599999987	0.0600000000	2.17E-08
11	0.0990000069	0.0990000000	6.97E-08
12	0.328435183	0.328435154	8.90E-08
13	0.131999999	0.132000000	7.58E-09
14	0.2959015970	0.2959015648	1.09E-07
15	0.0844461694	0.0844461558	1.61E-07
16	0.327427775	0.327427762	4.03E-08
17	0.164533600	0.164533589	6.72E-08
18	0.382021248	0.382021221	7.20E-08
19	0.213284895	0.213284903	3.94E-08
20	0.0540000051	0.0540000000	9.44E-08
21	0.229845107	0.229845091	6.91E-08
22	0.0719999969	0.0720000000	4.31E-08
23	0.211845100	0.211845091	4.19E-08
24	0.431990117	0.431990065	1.21E-07
25	0.269103080	0.269103101	7.61E-08
26	0.0900000036	0.0900000000	4.00E-08
27	0.345268458	0.345268436	6.33E-08
28	0.119989552	0.119989563	9.38E-08
Total	7.9999999502	8.0000000000	6.23E-09

TABLEAU 5.4 Propriétés des lignes de l'échantillon

Ligne		Origine	Direction (cosinus)
1	X	-2.220446049250E-16	0.1971379965544
	Y	0.000000000000E+00	0.1971379965544
	Z	0.000000000000E+00	0.9603505730629
2	X	4.704737833658E-01	0.1971379965544
	Y	4.704737833658E-01	0.1971379965544
	Z	-1.931549970167E-01	0.9603505730629
3	X	6.786687599642E-01	0.1971379965544
	Y	-4.440892098501E-16	0.1971379965544
	Z	-1.393151661655E-01	0.9603505730629
4	X	2.220446049250E-16	0.2133980989456
	Y	0.000000000000E+00	0.5512958168983
	Z	0.000000000000E+00	0.8065569996834
5	X	5.130126867046E-01	-0.8065569996834
	Y	6.768614338665E-01	0.2133980989456
	Z	4.885453849886E-01	0.5512958168983

Les volumes ainsi calculés sont présentés dans le tableau 5.3. On peut encore une fois vérifier que l'écart est minime, et tombe à l'intérieur de la précision numérique établie pour la simple précision.

La cellule représentée ici est un cas test où tous les types d'intersection présentés plus tôt (section 4.1) sont représentés. Étant donné la précision obtenue, on pourra se servir de ces volumes pour vérifier les volumes calculés numériquement par le *tracking*, présenté dans la section suivante.

5.3 Vérification du *tracking*

5.3.1 Vérification des longueurs des lignes

En premier lieu, pour effectuer la vérification du *tracking*, nous avons étudié une cellule semblable à celle présentée à la section 5.2.1, en prenant un échantillon de lignes d'intégration générées, et en analysant celles qui intersectent la sphère, et en vérifiant les longueurs et les points d'intersection générés. Un échantillon est présenté au tableau 5.5, les informations relatives aux lignes elles-mêmes étant présentées dans le tableau 5.4. On analyse une cellule cubique cartésienne de côté 2 cm (région 3), comprenant une sphère centrée en (0.3, 0.3, 0.3), composée de deux couches, de rayons respectifs 0.4 cm (région 1) et 0.6 cm (région 2).

TABLEAU 5.5 Échantillon de lignes d'intégration

Ligne	Région	Longueur calculée (cm)	Longueur réelle (cm)	Différence Relative
1	3	0.9425494581001341	0.9425494593034130	5.44E-09
	2	0.2702820574127690	0.2702820297522843	1.02E-07
	1	0.4696857590095490	0.4696857741868145	3.23E-08
	2	0.2702820574127690	0.2702820297522843	1.02E-07
	3	0.1297734881754953	0.1297734883411650	2.80E-07
2	3	0.7734414054464570	0.7734414305581790	3.25E-08
	2	0.3187442539765440	0.3187442599093905	1.86E-08
	1	0.3087181123146964	0.3087180199034411	2.99E-07
	2	0.3187442539765440	0.3187442599093905	1.86E-08
	3	0.3629247943147180	0.3629248511992100	1.57E-07
3	3	0.9883410062729130	0.9883410441070590	3.83E-08
	2	0.6285327986964347	0.6285326932547344	3.83E-08
	3	0.4656990146540730	0.4656990841178400	1.49E-07
4	3	1.152469435894660	1.152469450949460	1.31E-08
	2	0.2237891395437119	0.2237891187879097	9.27E-08
	1	0.6699093851015967	0.6699093822082209	4.32E-09
	2	0.2237891395437119	0.2237891187879097	9.27E-08
	3	0.2097188680673880	0.2097189111084130	2.05E-07
5	3	0.2208084855568860	0.2208085519623310	3.01E-07
	2	0.7408359237228181	0.7408357963485904	1.72E-07
	3	0.5698726816267330	0.5698727505425240	1.21E-07

Les lignes choisies ici pour illustrer la validation ne concernent que les lignes qui intersectent les couches sphériques. On observe que les lignes 3 et 5 n'intersectent que la coquille extérieure de la *pin* sphérique, ne produisant qu'un segment, alors que les autres croisent les deux couches de la sphère, produisant des segments de longueurs symétriques.

5.3.2 Vérification de l'emplacement des lignes à l'aide de TLM :

Le module TLM: de DRAGON a été très utile dans l'étape de vérification des lignes d'intégration, permettant de générer un fichier *Matlab* illustrant certaines lignes d'intégration au complet, ou encore leurs intersections avec certains plans (spécifiés par l'utilisateur). En spécifiant une densité de lignes assez faible, et en traçant les lignes dans toutes les directions, on peut se représenter explicitement une géométrie assez simple (figure 5.1), ce qui est utile afin de déterminer la validité des nouvelles routines de *tracking*.

Cependant, dès que vient le temps d'utiliser des géométries plus complexes, ou une analyse plus complète de la géométrie, les figures générées par le tracage de l'ensemble des lignes deviennent rapidement trop encombrées pour pouvoir y discerner la géométrie. On utilise donc l'option permettant de tracer les points d'intersection des lignes avec certains plans, afin de vérifier la validité des lignes. En prenant une géométrie cartésienne ($3 \times 3 \times 3$) comme celle définie dans la section 5.2.2, avec des coquilles sphériques en $R = 0.3, 0.6$ et 0.9 cm, on peut générer la figure 5.2, qui permet d'observer visuellement les régions, dans l'ensemble de la géométrie, ou individuellement dans chacune des sous-régions.

On peut donc observer visuellement les régions de la géométrie, et ainsi déterminer si les lignes d'intégration qui sont générées sont bel et bien dans la région appropriée de la géométrie, et donc si le module NXT: a généré un *tracking* adéquat.

5.3.3 Vérification des volumes numériques

Finalement, une dernière vérification qui peut être effectuée pour valider le *tracking* est d'observer les volumes calculés numériquement par NXT: en fonction des longueurs respectives des segments qui traversent chaque région, et d'en effectuer la comparaison avec les volumes analytiques calculés préalablement. Évidemment, ces volumes seront fortement dépendants de la densité de lignes générées, car il faudra qu'une quantité suffisante de lignes traverse chaque région dont on veut calculer le volume. On peut cependant vérifier la convergence des volumes calculés vers les volumes analytiques en fonction du raffinement sur la géométrie

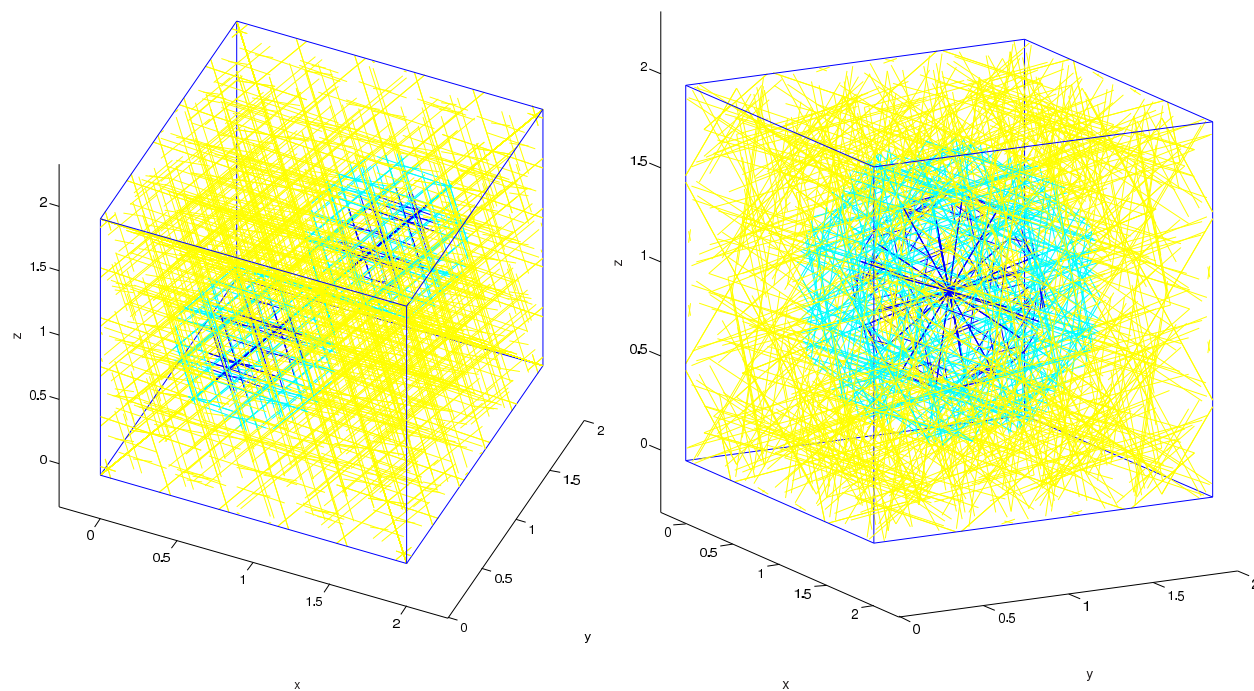


FIGURE 5.1 Visualisation du *tracking* à l'aide du module TLM :

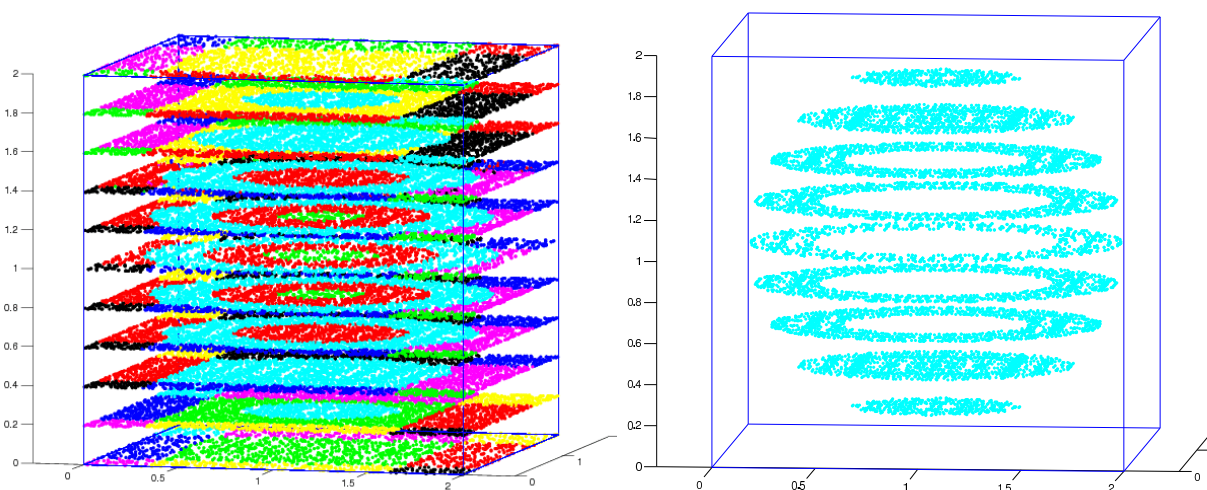


FIGURE 5.2 Intersection du *tracking* avec des plans à intervalles réguliers

TABLEAU 5.6 Volumes calculés numériquement par NXT :

Quadrature	Densité (cm^{-2})	Ordre	Erreur RMS (%)	Erreur Max. (%)	Erreur Moy. (%)
Poids Égaux	10.0	2	18.24550	90.06638	1.48001
		8	16.64095	90.06638	-2.55817
		16	16.85245	90.06638	-1.58190
	20.0	2	2.48498	10.06485	0.47423
		8	3.53168	10.06484	-1.35797
		16	3.72187	10.06482	1.65092
	50.0	2	1.92751	7.78483	-0.04890
		8	1.79627	7.78483	-0.52046
		16	1.78751	7.78484	-0.64180
	100.0	2	0.75645	2.73456	0.29326
		8	0.74120	2.86485	-0.00515
		16	0.53025	2.73458	0.01692
Legendre Trapézoïdale	10.0	10	16.92533	90.06638	-1.13577
		20	17.12331	90.06638	-1.04205
		40	17.02279	90.04672	-0.44984
	20.0	10	2.72407	10.06484	0.67340
		20	2.48894	10.06484	-0.74967
		40	2.51304	10.07414	-0.48961
	50.0	10	1.87569	7.78483	0.11505
		20	1.70571	7.78483	-0.59951
		40	1.58153	7.77368	-0.29844
	100.0	10	0.58333	2.73456	0.09540
		20	0.61330	2.73456	-0.03852
		40	0.64838	2.72393	0.11514
Legendre- Techbychev	10.0	10	16.74952	90.06638	-2.80308
		20	17.03106	90.06638	-1.50008
		40	17.06606	90.04592	-0.60590
	20.0	10	2.53333	10.06484	0.36693
		20	2.60308	10.06484	0.39659
		40	2.41308	10.07452	-0.12476
	50.0	10	1.85345	7.78483	-0.14254
		20	1.91013	7.78483	-1.01420
		40	1.68811	7.77323	-0.55937
	100.0	10	0.60375	2.73456	0.06960
		20	0.52499	2.73456	-0.01735
		40	0.56280	2.72350	-0.05476

($3 \times 3 \times 3$) considérée à la section 5.3.2. Le tableau 5.6 en présente les résultats, selon les trois quadratures angulaires tri-dimensionnelles disponibles : à poids égaux (option EQW de NXT:) (Carlson, 1971), de Legendre trapézoïdale (option SMS) (Sanchez *et al.*, 2002), et de Legendre-Techbychev (option PNTN) (Longoni et Haghghat, 2001).

On voit bien que l'erreur diminue généralement avec la densité de *tracks* choisie, ainsi qu'avec, bien que de manière moins prononcée, l'augmentation de l'ordre de quadrature angulaire. Cet effet peut s'expliquer du fait de l'indépendance angulaire de l'équation 3.37, et de la très faible dépendance angulaire lorsque vient le temps de la discrétiser (équation 3.39). Cet effet est amplifié pour des géométries sphériques, qui ne présentent aucune direction privilégiée.

Aux deux extrêmes de la discrétisation, des effets d'imprécision numérique peuvent cependant commencer à entrer en jeu, et induire des erreurs indésirables dans le calcul des volumes, soit en sous-représentant les régions traversées par peu de lignes, soit en raison de sommes ou de différences de valeurs se rapprochant de la précision machine.

5.4 Comparaison cellule à 8 pins

Le test final a été effectué par une comparaison entre une cellule cartésienne ($5 \times 5 \times 5$), de côté 2 cm, comprenant huit *pins* cartésiennes centrées en $(\pm\frac{1}{3}, \pm\frac{1}{3}, \pm\frac{1}{3})$ contenant du combustible, tel qu'illustré à la figure 5.3, et le même maillage avec huit *pins* sphériques situées aux mêmes coordonnées. On a utilisé des *pins* de rayon 0.1 cm, et établi les longueurs des côtés des cubes pour obtenir des volumes équivalents, de façon à ce que les différences observées ne soient dues majoritairement qu'au changement de géométrie.

Pour évaluer la différence en termes de k_{eff} , une géométrie bi-dimensionnelle équivalente, illustrée à la figure 5.4, a aussi été testée, en utilisant un maillage (5×5), avec des cylindres de rayons 0.1 cm (et des carrés de surface équivalentes) afin de déterminer l'ordre de grandeur de la différence apportée par les géométries cylindriques, puis comparer avec celle apportée par les géométries sphériques. Le tableau 5.7 présente les valeurs et les différences de k_{eff} obtenues pour les différents cas. Les cellules considérées sont composées de graphite, avec des conditions aux frontières de réflexion totale.

Dans les cas contenant des *pins* sphériques et cylindriques, on a conservé le même maillage que pour les cas avec *pins* cartésiennes, afin de tenter de réduire au maximum les effets dus à les différences pouvant être causée par une discrétisation spatiale non-équivalente.

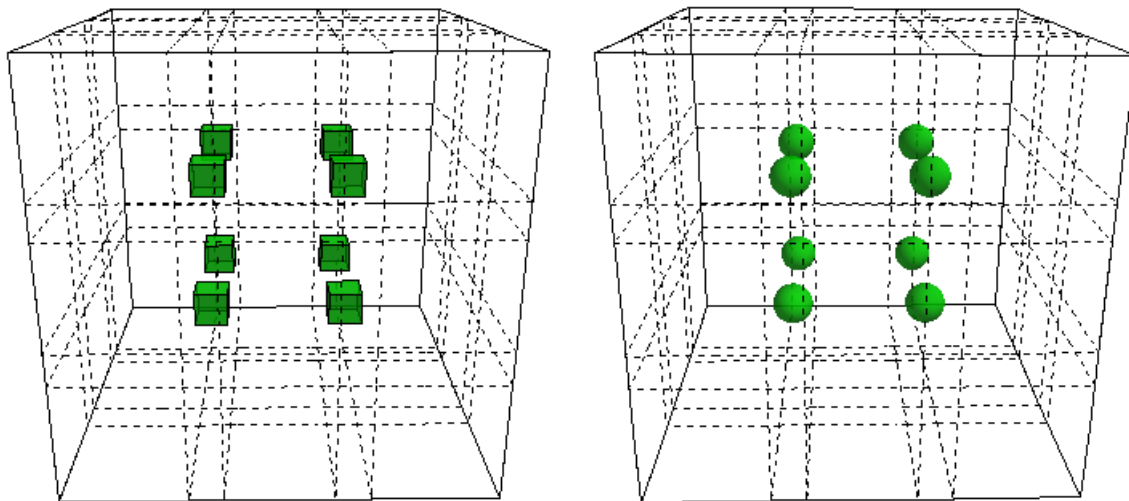


FIGURE 5.3 Cellule de test 3-D avec *pins* cartésiennes et sphériques

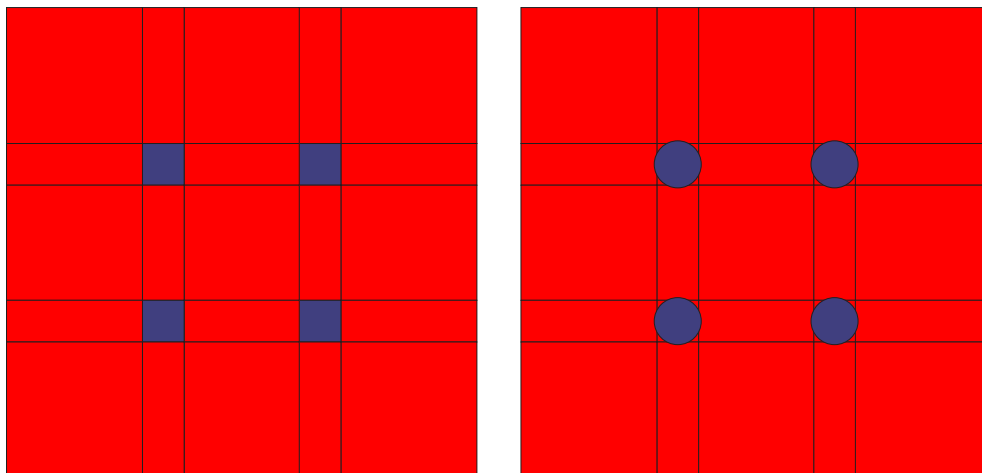


FIGURE 5.4 Cellule de test 2-D avec *pins* cartésiennes et cylindriques

Ce dernier test montre bien que la différence de k_{eff} apportée par le changement des *pins* cartésiennes en sphériques ou cylindriques est bien du même ordre de grandeur, que l'on utilise deux ou trois dimensions. Il est par ailleurs parfaitement normal qu'il y ait un écart important entre les cas bi- et tri-dimensionnels, du fait même que le cas à deux dimensions suppose implicitement une continuité géométrique dans l'axe manquant.

5.5 Génération aléatoire

Tel que mentionné dans la section 2.3, la méthode traditionnellement utilisée pour analyser les réacteurs présentant des caractéristiques de double hétérogénéité ne prend pas en compte les positions explicites des micro-éléments, mais établira plutôt une forme dérivée de l'équation de transport afin d'appliquer des éléments d'analyse statistique pour résoudre et trouver les propriétés pertinentes de la cellule.

La méthode que nous proposons passe par la routine NXXTPRD, qui sert ici à générer des positions aléatoires de *pins* dans une cellule cartésienne. La routine NXXTISS, en plus de modifications apportées à la routine NXXTGMD, sert à tester la superposition avec les *pins* placées au préalable – notamment, si plusieurs *clusters* sont définis. Toutes les *pins* dont les positions sont définies explicitement avec CPIN sont placées et testées pour éviter la superposition, avant de commencer à générer aléatoirement les *pins* définies par DPIN.

La méthode de génération aléatoire séquentielle des positions des *pins* sphériques permet l'atteinte d'une densité maximale d'environ $d_{pin} = 0.21$, variant en fonction du rapport côté de la cellule / rayon des sphères, les densités plus élevées pouvant être atteintes avec des rapports plus élevés, et donc généralement un nombre de micro-particules sphériques très élevé, ce qui est de toutes façons généralement le cas dans les réacteurs VHTR.

TABLEAU 5.7 Test de cellule à 8 *pins*

Type de cellule		k_{eff}	Différence (mk)	Différence (%)
3-D	pins cubiques	1.806186	0.058	0.0032
	pins sphériques	1.806244		
2-D	pins carrées	1.592085	0.082	0.0052
	pins cylindriques	1.592167		

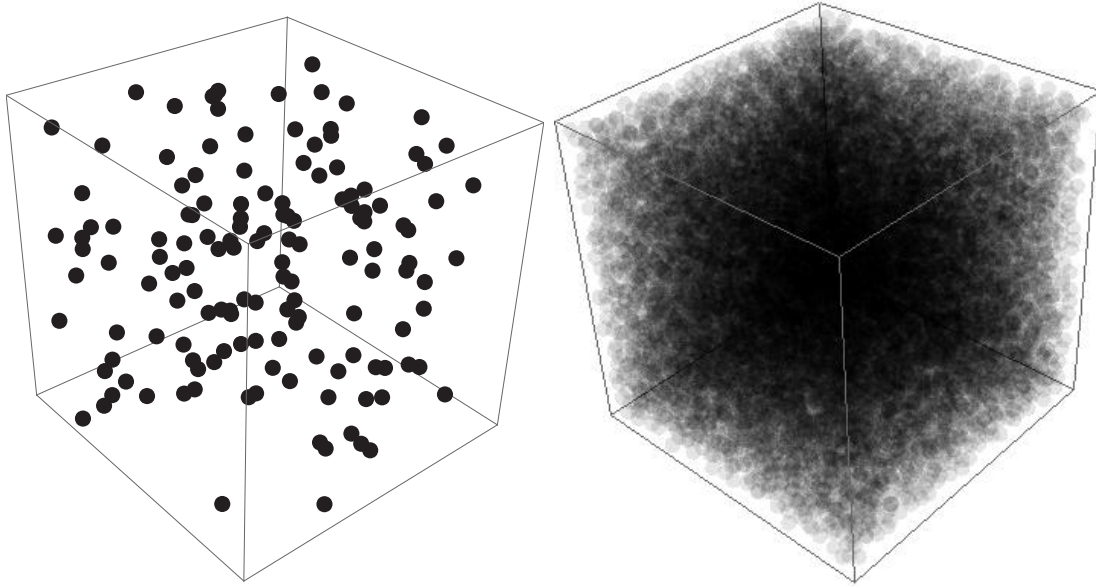


FIGURE 5.5 Pins générées aléatoirement pour des densités $d_{pin} = 0.002$ et $d_{pin} = 0.2$

Le blocage survient ici lorsqu'un essai de localisation d'une particule donnée a échoué 100 fois de suite. Il serait éventuellement possible d'atteindre une densité plus élevée en augmentant cette limite, sans toutefois espérer une augmentation trop importante, et à un coût plus important en temps de calcul lors de l'opération de génération des positions. Deux exemples de positionnement aléatoire de sphères sont donnés graphiquement à la figure 5.5, dans une cellule carrée de côté 2 cm, avec des sphères de rayon 0.03 cm, et des densités $d_{pin} = 0.002$ et $d_{pin} = 0.2$, pour des totaux respectifs de 141 et 14 147 *pins*.

Cependant, certains cas peuvent requérir une densité plus élevée que celle atteignable avec l'algorithme utilisé ici. Il sera à ce moment nécessaire soit d'utiliser un autre algorithme de génération pseudo-aléatoire ou semi-aléatoire des positions et de l'implanter, soit de générer à l'extérieur DRAGON les positions des *pins* et de les spécifier explicitement dans le fichier d'entrée, ce qui deviendrait par contre rapidement exhaustif pour des cas contenant un grand nombre de *pins*.

5.6 Comparaison avec le *benchmark* HTGR

Finalement, une comparaison a été effectuée avec la première partie d'un *benchmark* (DeHart et Ulses, 2009) préétabli à l'aide de méthodes Monte-Carlo, par le code Serpent (Leppänen, 2009), qui a donné des résultats probants. Le *benchmark* consiste en une seule

TABLEAU 5.8 Description des pastilles TRISO utilisées pour le *benchmark*

Rayon (cm)	Description
0.0250	Coeur (UO ₂ enrichi à 8%)
0.0340	Carbone poreux à faible densité
0.0380	Carbone pyrolitique (C-Gr)
0.0415	Carbure de silicium (SiC)
0.0455	Carbone pyrolitique (C-Gr)

particule TRISO (voir section 2.1.1) centrée dans une cellule cubique de graphite, de côté défini de façon à avoir une densité de 0.09043. Les rayons des coquilles sphériques sont donnés dans le tableau 5.8.

Nous ne présenterons ici que les grandes lignes de la comparaison, un article soumis pour la conférence PHYSOR-2010 est présenté à l'annexe C, donnant une description plus détaillée de la comparaison effectuée pour ce *benchmark*.

La simplicité de la géométrie étudiée nous permettra de faire une comparaison directe code à code. Nous n'effectuerons que des calculs sans évolution ($burnup=0$). Trois méthodes seront donc comparées :

- Monte-Carlo 3-D, qui sera considérée comme une solution de référence, établie à l'aide du code SERPENT.
- Probabilités de collision 3-D, à l'aide des nouvelles capacités de NXT: et de DRAGON.
- Traitement 3-D de la double hétérogénéité dans DRAGON, à l'aide des méthodes de Hébert et de Sanchez-Pomraning.

5.6.1 Comparaison avec les calculs Monte-Carlo

Une étude préliminaire a été effectuée afin de déterminer les paramètres minimaux de discrétisation à utiliser pour l'opérateur de *tracking* 3-D, dont les résultats sont présentés dans le tableau 5.9.

On observe qu'une densité de *tracks* importante, combinée à une quadrature angulaire d'ordre élevé est nécessaire pour obtenir une précision d'un niveau acceptable. Cette limitation était à prévoir, en raison des volumes minuscules impliqués, de l'ordre de 4.36×10^{-3} pour la cellule complète, et de 6.54×10^{-5} pour le noyau de la particule TRISO. La discrétisation la plus per-

TABLEAU 5.9 Influence de la discrétisation angulaire et spatiale vs. SERPENT

	k_{inf} SERPENT : 1.35035 ± 49 pcm	
$S_i\{N_{\text{angles}} \times \text{densité (tracks/cm}^2)\}$	k_{inf} CP	diff. vs. Serpent (pcm)
$S_1\{44 \times 5000\}$	1.36195	1161
$S_2\{44 \times 10000\}$	1.34307	-728
$S_3\{44 \times 20000\}$	1.34683	-352
$S_4\{44 \times 25000\}$	1.35469	435
$S_5\{44 \times 30000\}$	1.34407	-628
$S_6\{44 \times 35000\}$	1.34646	-388
$S_7\{44 \times 40000\}$	1.35068	33

formante, S_7 , est par la suite utilisée pour comparer les taux de réaction, taux d'absorption et le flux intégré à deux groupes énergétiques, avec les valeurs obtenues par SERPENT. Les résultats présentés dans le tableau 5.10, démontrent la précision de la méthode lorsque vient le temps de modéliser le comportement neutronique d'une particule TRISO individuelle, tant que la densité de *tracks* est suffisante.

5.6.2 Comparaison avec les modèles de double-hétérogénéité

Une seconde comparaison a été effectuée versus les résultats obtenus grâce aux deux modèles de double-hétérogénéité disponibles dans DRAGON. Le k_{inf} donné par les deux méthodes est de 1.35066, pour une différence avec la discrétisation S_7 présentée plus tôt de seulement 3 pcm. Les valeurs des constantes par groupes sont données au tableau 5.10. On observe encore une fois une adéquation entre les deux méthodes, ce qui vient confirmer les résultats obtenus, d'autant plus que les deux calculs utilisaient le même format de bibliothèques d'entrée.

TABLEAU 5.10 Comparaison des constantes par groupe énergétique

Parameters		CP	Serpent	diff. vs. Serpent (%)	DH	diff. vs. DH (%)
Flux intégré	1	2.27418E+02	2.27396E+02	9.7E-03	2.27419E+02	-4E-04
	2	5.54760E+01	5.55959E+01	-2.157E-01	5.54912E+01	-2.74E-02
$\nu \times$ Taux de fission	1	2.24770E-01	2.24964E-01	-8.62E-02	2.24710E-01	2.67E-02
	2	1.12591E+00	1.12425E+00	1.477E-01	1.12595E+00	-3.6E-03
Taux d'absorption	1	4.18215E-01	4.19202E-01	-2.354E-01	4.18196E-01	4.5E-03
	2	5.81785E-01	5.81055E-01	1.256E-01	5.81804E-01	-3.3E-03

Chapitre 6

CONCLUSION

Ce projet a pour but d'effectuer l'intégration dans le module `NXT`: de fonctionnalités pour analyser les géométries sphériques à plusieurs couches, caractéristiques des réacteurs de quatrième génération à très haute température (VHTR). Une analyse géométrique des volumes sphériques nous a permis d'implanter le calcul des volumes analytiques, ainsi que le calcul des distances entre les points d'intersection des lignes d'intégration avec les éléments de géométrie sphériques.

Une analyse exhaustive des combinaisons de maillages cartésiens et sphériques a par la suite permis d'effectuer une vérification de la validité des longueurs et des volumes calculés, effectuant dans les faits-même une vérification des valeurs calculées par les nouvelles routines.

De plus, de par la nature aléatoire de la distribution des éléments de combustible dans les réacteurs VHTR, une fonctionnalité permettant la génération aléatoire des positions des éléments de géométrie sphériques à l'intérieur d'une cellule a été implantée, de manière à offrir une méthode alternative aux approches généralement utilisées pour traiter les problèmes présentant des caractéristiques de double hétérogénéité.

En dernier lieu, des tests de validation préliminaires ont aussi été effectués en comparant les résultats avec des *benchmarks* établis à l'aide du code Monte-Carlo `Serpent`, pour une géométrie simple. Les résultats vont dans la direction espérée, en ce sens que plus les techniques utilisées se raffinent, plus le résultat final tend à s'approcher des résultats Monte-Carlo, pour une différence finale inférieure à 0.5 mk.

Dans le futur, des études supplémentaires pourront être effectuées concernant les nouvelles fonctionnalités implémentées dans le module `NXT`: devraient se concentrer, dans un premier temps, sur une validation plus complète des résultats obtenus à l'aide d'un calcul complet, puis une comparaison avec d'autres *benchmarks*. On pense notamment à effectuer des calculs en évolution isotopique avec une augmentation du *burnup*, ou encore comparer les résultats obtenus par la méthode proposée de positionnement aléatoire des *pins* sphériques avec des

résultats obtenus par les traitements habituels des géométries à double hétérogénéité.

Afin de pousser plus loin les capacités d'analyse des réacteurs VHTR dans DRAGON, il sera aussi nécessaire d'incorporer des capacités d'analyse de géométries plus avancées. Les *pins* ne pouvant pour l'instant être utilisées que dans des cellules à maillage cartésien, les capacités pourraient être étendues pour des combinaisons sphères-cylindres-prismes hexagonaux pour les réacteurs prismatiques et sphères-sphères pour les réacteurs à lits de boulets.

On peut aussi songer à certaines améliorations qui pourraient être apportées à l'algorithme de génération pseudo-aléatoire des positions des *pins*. Des méthodes semi-déterministes pourraient être utilisées, permettant de spécifier à l'avance une distribution désirée de pins qui permettrait à la fois d'atteindre des densités plus élevées, et d'obtenir des distributions plus semblables à celles obtenues « naturellement ».

Références

- A. HÉBERT (2009). *Applied Reactor Physics*. Presses Internationales Polytechnique, Montréal.
- ASKEW, J. (1972). A Characteristics Formulation of the Neutron Transport Equation in Complicated Geometries. Technical Report AEEW-M 1108, United Kingdom Atomic Energy Establishment, Winfrith.
- CARLSON, B. G. (1971). Tables of Equal Weight Quadrature EQ_n Over the Unit Sphere. Rapport technique LA-4734, Los Alamos Scientific Laboratory.
- DEHART, M. D. et ULSES, A. P. (2009). Benchmark Specification for HTGR Fuel Element Depletion. Benchmark specification, Nuclear Energy Agency.
- DUCO, J. (2003). Accidents Nucléaires – Windscale (Royaume-Uni). *Techniques de l'Ingénieur*.
- GENERATION IV INTERNATIONAL FORUM (2002). A Technology Roadmap for Generation IV Nuclear Energy Systems. Technical report, Generation IV International Forum, Issy-les-Moulineaux. See <http://www.gen-4.org>.
- GOTTAUT, H. et KRÜGER, K. (1990). Results of Experiments at the AVR Reactor. *Nuclear Engineering and Design*, 121, 143 – 153.
- HÉBERT, A. (2008). Development of the Subgroup Projection Method for Resonance Self-shielding Calculations. *Accepted for publication in Nuclear Science and Engineering*.
- HÉBERT, A. (2008). Scattering Reduction of the Double-Heterogeneity Treatment in Dragon. *Nuclear Science and Engineering*, 160, 261–266.
- HÉBERT, A. et SANTAMARINA, A. (2008). Refinement of the Santamarina-Hfaiedh energy mesh between 22.5 eV and 11.4 KeV. *PHYSOR-2008, Int. Conf. on the physics of reactors*. Interlaken, Switzerland.
- HÉBERT, A. et SAYGIN, H. (1992). Development of Dragr for the Formatting of DRAGON Cross Section Libraries. *Seminar on NJOY-91 and THEMIS for the Processing of Evaluated Nuclear Data Files*. Saclay, France.
- JING, X., XU, X., YANG, Y. et QU, R. (2002). Prediction Calculations and Experiments for the First Criticality of the 10 MW High Temperature Gas-cooled Reactor-Test Module. *Nuclear Engineering and Design*, 218.

- KIM, K.-S., CHO, J.-Y., LEE, H. C., NOH, J. M. et ZEE, S. Q. (2007). Development of a Physics Analysis Procedure for the Prismatic Very High Temperature Gas-cooled Reactors. *Annals of Nuclear Energy*, 34, 849 – 860.
- KIM, Y. et BAEK, M. (2005). Elimination of Double-Heterogeneity through a Reactivity-Equivalent Physical Transformation. *Proceedings of GLOBAL 2005*. CD-ROM.
- KRÜGER, K., IVENS, G. et KIRCH, N. (1988). Operational Experience and Safety Experiments with the AVR Power Station. *Nuclear Engineering and Design*, 109, 233 – 238.
- LEE, H. C., KIM, K.-S., NOH, J. M. et JOO, H. K. (2005). The Equivalent Cylinder Models for the Homogenization of Pebble Bed Reactor Cores. *Transactions of the American Nuclear Society*, 93, 961 – 962.
- LEPPÄNEN, J. (2009). PSG2 / SERPENT – A Continuous-Energy Monte-Carlo Reactor Physics Burnup Calculation Code. Rapport technique, VTT Technical research centre of Finland.
- LETELLIER, R. (2006). *Développement de la méthode des caractéristiques pour le calcul de réseau*. Thèse de doctorat, École Polytechnique de Montréal, Montréal.
- LONGONI, G. et HAGHIGHAT, A. (2001). Development of New Quadrature Sets with the “Ordinate Splitting” Technique . *M&C-2001, American Nuclear Society Topical Meeting in Mathematics and Computations*. Salt Lake City, Utah. (Proceedings available on CD-Rom).
- MARLEAU, G. (2005). New Geometries Processing in DRAGON : the NXT: Module. Technical Report IGE-260, Institut de Génie Nucléaire, École Polytechnique de Montréal, Montréal.
- MARLEAU, G., HÉBERT, A. et ROY, R. (1992). New Computational Methods Used in the Lattice Code DRAGON. *Topical Meeting on Advances in Reactor Physics*. Charleston, South Carolina, 1.177.
- MATLAB (2006). The Language of Technical Computing.
- NABIELEK, H., KAISER, G., HUSCHKA, H., RAGOSS, H., WIMMERS, M. et THEYMANN, W. (1984). Fuel for Pebble-bed HTRs. *Nuclear Engineering and Design*, 78, 155 – 166.
- NABIELEK, H., KÜHNLEIN, W., SCHENK, W., HEIT, W., CHRIST, A. et RAGOSS, H. (1990). Development of Advanced HTR Fuel Elements. *Nuclear Engineering and Design*, 121, 199 – 210.
- NOH, J. M., KIM, K.-S., KIM, Y. et LEE, H. C. (2008). Development of a Computer Code System for the Analysis of Prism and Pebble Type VHTR Cores. *Annals of Nuclear Energy*, 35, 1919 – 1928.

- REINERT, D. R. (2008). *Investigation of Stochastic Radiation Transport Methods in Random Heterogeneous Mixtures*. Thèse de doctorat, The University of Texas at Austin, Austin, Texas.
- ROY, R. (1999). The CLE-2000 Tool-Box. Rapport technique IGE-163, École Polytechnique de Montréal.
- ROY, R. et HÉBERT, A. (2000). The GAN Generalized Driver. Rapport technique IGE-158, École Polytechnique de Montréal.
- SANCHEZ, R. (1988). APOLLO II : A user oriented, portable, modular code for multigroup transport assembly calculations. *Nucl. Sci. and Eng.*, 100, 352–362.
- SANCHEZ, R., MAO, L. et SANTANDREA, S. (2002). Treatment of Boundary Conditions in Trajectory-Based Deterministic Transport Methods. *Nucl. Sci. Eng.*, 140, 23–50.
- SANCHEZ, R. et POMRANING, G. (1991). A Statistical Analysis of the Double Heterogeneity Problem. *Annals of Nuclear Energy*, 18, 371–395.
- SCHULTEN, R. (1978). Pebble-bed HTRs. *Annals of Nuclear Energy*, 5, 357 – 374. Thorium and Gas Cooled Reactors.
- SHIOZAWA, S., FUJIKAWA, S., IYOKU, T., KUNITOMI, K. et TACHIBANA, Y. (2004). Overview of HTTR Design Features. *Nuclear Engineering and Design*, 233, 11 – 21. Japan’s HTTR.
- TALAMO, A. et GUDOWSKI, W. (2005). Performance of the Gas Turbine-Modular Helium Reactor Fuelled with Different Types of Fertile TRISO Particles. *Annals of Nuclear Energy*, 32, 1719 – 1749.
- THE MATHWORKS, INC. (2005). MATLAB version 7.1.0.183 (R14) Service Pack 3. Software for the Linux operating system, Natick.
- VITART, X., DUIGOU, A. L. et CARLES, P. (2006). Hydrogen Production Using the Sulfur-Iodine Cycle Coupled to a VHTR : An Overview. *Energy Conversion and Management*, 47, 2740 – 2747. 12th International Conference on Emerging Nuclear Energy Systems.
- WIDOM, B. (1965). Random Sequential Addition of Hard Spheres to a Volume. *The Journal of Chemical Physics*, 44, 3888–3894.
- ZKIEK, A. et MARLEAU, G. (2007). Dragon Verification : the NXT: Module. Technical Report IGE-299, Revision 0, Institut de Génie Nucléaire, École Polytechnique de Montréal, Montréal.

Annexe A

Code source

A.1 Routine NXTIRS

Cette routine sert à déterminer s'il existe une intersection entre une sphère quelconque et un parallélépipède quelconque, et, le cas échéant, à en déterminer le volume.

```
*DECK NXTIRS
      FUNCTION NXTIRS(XYZCAR,POSSPH,VOLINT)
*
*-----
*
*Purpose:
* Compute the volume of intersection between
* a cartesian and a spherical region
*
*Author(s): M-A Lajoie
*
*Parameters: input
* XYZCAR Spatial description of the Cartesian region with
* XYZCAR(1) for left face, XYZCAR(2) for right face,
* XYZCAR(3) for bottom face, XYZCAR(4) for top face,
* XYZCAR(5) for front face, XYZCAR(6) for back face
* positions.
* POSSPH Spatial description of the spherical region with
* POSSPH(0) the radius,
* POSSPH(1) the $X$ position of center,
* POSSPH(2) the $Y$ position of center, and
* POSSPH(3) the $Z$ position of center.
*
*Parameters: output
* NXTIRS Type of intersection between cartesian region and
* spherical region, where
* \begin{itemize}
* \item \verb|NXTIRS| = 0 means that there is no intersection
* between the two regions;
* \item \verb|NXTIRS| = 1 means that the cartesian region
* is all located inside the spherical region;
* \item \verb|NXTIRS| = 2 means that the spherical region
* is all located inside the cartesian region;
* \item \verb|NXTIRS| = -1 means that the intersection between
* the spherical region and the cartesian region is partial.
* \end{itemize}
* VOLINT 3-D volume of intersection between Cartesian region
```

```

*           and spherical region.
*
*-----
*
*           IMPLICIT          NONE
*-----
*           Subroutine arguments
*-----
*           INTEGER          NXTIRS
*           DOUBLE PRECISION XYZCAR(6) ,POSSPH(0:3) ,VOLINT
*-----
*           Local parameters
*-----
*           INTEGER          IOUT
*           CHARACTER       NAMSBR*6
*           PARAMETER      (IOUT=6,NAMSBR='NXTIRS')
*           INTEGER          IPRINT
*           PARAMETER      (IPRINT=1)
*           DOUBLE PRECISION DCUTOF
*           PARAMETER      (DCUTOF=1.0D-8)
*-----
*           Other variables
*-----
*           DOUBLE PRECISION XDR CST, PI
*           DOUBLE PRECISION XYZFAC(6) ,VOLP(6) ,INTIND , X, Y, E, RACINE
*           DOUBLE PRECISION RAYON2, DIST2 ,VOLP2(3)
*           DOUBLE PRECISION VOLREG, VOLSPH, VOLCAR
*           INTEGER          ICNT , ILOC(6) , IDX , IDY , IDZ , ID(3) , NISPOS(3)
*           INTEGER          FACTX,FACTY,FACTZ, INTER2(3) , ISPOS(3)
*-----
*           Initialize parameters and exit values
*-----
*           IF (IPRINT .GE. 200) THEN
*               WRITE(IOUT,6000) NAMSBR
*               WRITE(IOUT,6010) (XYZCAR(ICNT) ,ICNT=1,6)
*               WRITE(IOUT,6011) (POSSPH(ICNT) ,ICNT=0,3)
*           ENDIF
*           NXTIRS = 0
*           VOLINT = 0.0D0
*           VOLCAR = (XYZCAR(2) - XYZCAR(1))*
>                 (XYZCAR(4) - XYZCAR(3))*
>                 (XYZCAR(6) - XYZCAR(5))
*           PI = XDR CST('Pi' , '_')
*           RAYON2 = POSSPH(0)*POSSPH(0)
*           VOLSPH = 4 * PI * POSSPH(0) * RAYON2 / 3
*           IF (XYZCAR(1) .GT. XYZCAR(2) .OR.
>             XYZCAR(4) .GT. XYZCAR(4) .OR.
>             XYZCAR(5) .GT. XYZCAR(6)) THEN
*               CALL XABORT ('Planes _not_in_the_good_order')
*           ENDIF
*           IF (POSSPH(0) .LT. 0) CALL XABORT ('Negative_sphere_radius')
*-----

```

```

* Recenter sphere at origin and verify intersections
*-----
      DO 100 ICNT=1,6
        XYZFAC(ICNT)=XYZCAR(ICNT)-POSSPH(NINT(DBLE(ICNT)/2))
        IF (XYZFAC(ICNT) .LE. -POSSPH(0)) THEN
C           The plane is under, to the left or behind the sphere
            ILOC(ICNT) = (-1)**ICNT
        ELSE IF (XYZFAC(ICNT) .GE. POSSPH(0)) THEN
C           The plane is over, to the right or in front of the sphere
            ILOC(ICNT) = (-1)**(ICNT+1)
        ELSE
C           The plane crosses the sphere
            ILOC(ICNT) = 0
        ENDIF
100 CONTINUE
*-----
* Calculate volumes
*-----
      IF (ILOC(1).NE.1 .AND. ILOC(2).NE.1 .AND. ILOC(3).NE.1 .AND.
>         ILOC(4).NE.1 .AND. ILOC(5).NE.1 .AND. ILOC(6).NE.1 ) THEN
C           Volume of the sphere which is under / to the left / behind
C           the considered plane
      DO 101 ICNT = 1,6
        IF (XYZFAC(ICNT) .LT. -POSSPH(0))THEN
          VOLP(ICNT) = 0
        ELSE IF (XYZFAC(ICNT) .GT. POSSPH(0))THEN
          VOLP(ICNT) = VOLSPH
        ELSE
          VOLP(ICNT) = VOLSPH - PI * (POSSPH(0)-XYZFAC(ICNT))**2
>                * (3*POSSPH(0)-(POSSPH(0)-XYZFAC(ICNT)))/3
        ENDIF
101 CONTINUE
      FACTX = -1.0
      DO 110 IDX = 1,2
        ID(1) = IDX
        FACTY = -1.0
      DO 120 IDY = 3,4
        ID(2) = IDY
        FACTZ = -1.0
      DO 130 IDZ = 5,6
        ID(3) = IDZ
        VOLREG = 0
      DO 131 ICNT = 1,3
        INTER2(ICNT) = 0
        VOLP2 (ICNT) = 0
131 CONTINUE
      IF (IPRINT .GE. 300) THEN
        WRITE (IOUT,6020) IDX, IDY, IDZ
      IF (IPRINT .GE. 500) THEN
        WRITE (IOUT,6033) ILOC(IDX), ILOC(IDY), ILOC(IDZ),
>                XYZCAR(IDX), XYZCAR(IDY), XYZCAR(IDZ)
      ENDIF

```

```

ENDIF
IF ( ILOC(IDX)*XYZFAC(IDX) .LE. 0 .AND.
>      ILOC(IDY)*XYZFAC(IDY) .LE. 0 .AND.
>      ILOC(IDZ)*XYZFAC(IDZ) .LE. 0 ) THEN
C      DIST2 = XYZFAC(IDX)**2+XYZFAC(IDY)**2+XYZFAC(IDZ)**2
C      Intersection of the three planes inside the sphere
IF (DIST2 .LT. RAYON2) THEN
C      IF (IPRINT .GE. 500) WRITE (IOUT,6030)
C      INTIND = 0.0D0
C      DO 132 ICNT = 1,3
C          X = XYZFAC(ID(MOD(ICNT ,3)+1))
C          Y = XYZFAC(ID(MOD(ICNT+1,3)+1))
C          RACINE = SQRT(RAYON2-X*X-Y*Y)
C
C          Defined integral corresponding to the volume of a
C          region bounded by 5 cartesian planes, and one
C          spherical segment boundary
C          INTIND = INTIND + X*Y/3*RACINE
C          >          - Y * (Y*Y-3*RAYON2) / 6 * ATAN(X/RACINE)
C          >          - X * (X*X-3*RAYON2) / 6 * ATAN(Y/RACINE)
C          >          - POSSPH(0)**3/3*ATAN(X*Y/POSSPH(0)/RACINE)
132      CONTINUE
C
C          Volume of the sphere which is under, to the left and
C          behind the intersection
C          VOLREG = INTIND - VOLSPH/4
C          >          + XYZFAC(IDX)*XYZFAC(IDY)*XYZFAC(IDZ)
C          >          + (VOLP(IDX) + VOLP(IDY) + VOLP(IDZ))/4
C          Intersection of the three planes is outside the sphere
ELSE
C      DO 133 ICNT = 1,3
C          Sign flag of the considered plane :
C          ISPOS 1 if < 0, 0 otherwise
C          NISPOS is the opposite.
C          ISPOS (ICNT) = (SIGN(0.5 ,XYZFAC(ID (ICNT))))+0.5)
C          NISPOS(ICNT) =-(SIGN(0.5 ,XYZFAC(ID (ICNT))))-0.5)
C
C          The two planes intersect inside the sphere
C          IF(XYZFAC(ID(MOD(ICNT ,3)+1))**2 +
>          XYZFAC(ID(MOD(ICNT+1,3)+1))**2 .LT. RAYON2)THEN
C          Flag of the intersection of the two planes
C          INTER2(ICNT) = 1
C          IF (IPRINT .GE. 500) WRITE(IOUT,6031) ICNT
C          X = XYZFAC(ID(MOD(ICNT ,3)+1))
C          Y = XYZFAC(ID(MOD(ICNT+1,3)+1))
C          RACINE = SQRT(RAYON2 - X*X - Y*Y)
C
C          Same integral as before
C          INTIND= X*Y/3*RACINE
C          >          - Y*(Y*Y-3*RAYON2)/6*ATAN(X/RACINE)
C          >          - X*(X*X-3*RAYON2)/6*ATAN(Y/RACINE)
C          >          - ( POSSPH(0)**3 / 3

```

```

> * ATAN(X*Y/POSSPH(0)/RACINE) )
C
C      Volume under / to the left / behind the
C      intersection of the two planes,
C      whose direction is ICNT
      VOLP2(ICNT) = 2 * INTIND - VOLSPH / 4
>                + (VOLP(ID(MOD(ICNT ,3)+1)))
>                + VOLP(ID(MOD(ICNT+1,3)+1))) / 2
      ENDIF
133 CONTINUE
C
C      No intersection between the planes inside the sphere
      IF (INTER2(1) + INTER2(2) + INTER2(3) .EQ. 0) THEN
      IF (IPRINT .GE. 500) WRITE(IOUT,6032)
      IF (ISPOS(1) .EQ. 1 .AND. ISPOS(2) .EQ. 1 .AND.
>        ISPOS(3) .EQ. 1) THEN
      VOLREG = VOLP(IDX) + VOLP(IDY) + VOLP(IDZ)
>            - 2 * VOLSPH
      ELSE IF (NISPOS(1) + NISPOS(2) +
>            NISPOS(3) .EQ. 1) THEN
      VOLREG = VOLP(IDX) * NISPOS(1)
>            + VOLP(IDY) * NISPOS(2)
>            + VOLP(IDZ) * NISPOS(3)
      ELSE
      VOLREG = 0
      ENDIF
C
C      One intersection of two planes inside the sphere
      ELSE IF (INTER2(1)+INTER2(2)+INTER2(3) .EQ. 1) THEN
      IF (ISPOS(1) .EQ. 1 .AND. ISPOS(2) .EQ. 1 .AND.
>        ISPOS(3) .EQ. 1) THEN
      VOLREG = INTER2(1) * (VOLP2(1) + VOLP(IDX))
>            + INTER2(2) * (VOLP2(2) + VOLP(IDY))
>            + INTER2(3) * (VOLP2(3) + VOLP(IDZ))
>            - VOLSPH
      ELSE IF ((INTER2(1)*NISPOS(1)
>            + INTER2(2)*NISPOS(2)
>            + INTER2(3)*NISPOS(3) .EQ. 1)
>            .AND. (NISPOS(1)+NISPOS(2)+NISPOS(3) .EQ. 1)) THEN
      VOLREG = NISPOS(1)*VOLP(IDX)
>            + NISPOS(2)*VOLP(IDY)
>            + NISPOS(3)*VOLP(IDZ)
      ELSE IF ((ISPOS(1)*INTER2(1) + ISPOS(2)*INTER2(2)
>            + ISPOS(3)*INTER2(3)) .EQ. 1) THEN
      VOLREG = VOLP2(1)*INTER2(1) + VOLP2(2)*INTER2(2)
>            + VOLP2(3)*INTER2(3)
      ELSE
      VOLREG = 0
      ENDIF
C
C      Two intersections of two planes inside the sphere
      ELSE IF (INTER2(1)+INTER2(2)+INTER2(3) .EQ. 2) THEN

```

```

      IF ( ISPOS(1)*INTER2(1) + ISPOS(2)*INTER2(2)
>      + ISPOS(3)*INTER2(3) .EQ. 2 ) THEN
      DO 134 ICNT = 1,3
        IF (INTER2(ICNT) .EQ. 1) THEN
          VOLREG = VOLREG + VOLP2(ICNT)
        ELSE
          VOLREG = VOLREG - VOLP(ID(ICNT))
        ENDIF
      CONTINUE
134      VOLREG=INTER2(1)*VOLP2(1)-~INTER2(1)*VOLP(IDX)
C      +INTER2(2)*VOLP2(2)-~INTER2(2)*VOLP(IDY)
C      +INTER2(3)*VOLP2(3)-~INTER2(3)*VOLP(IDZ)
      ELSE IF ( INTER2(1)*ISPOS(1) + INTER2(2)*ISPOS(2)
>      + INTER2(3)*ISPOS(3) .EQ. 0) THEN
        VOLREG = 0
      ELSE
        VOLREG = ISPOS(1) * INTER2(1) * VOLP2(1)
>      + ISPOS(2) * INTER2(2) * VOLP2(2)
>      + ISPOS(3) * INTER2(3) * VOLP2(3)
      ENDIF
C
C      Three intersections of two planes inside the sphere
      ELSE IF (INTER2(1)+INTER2(2)+INTER2(3) .EQ. 3) THEN
        IF (ISPOS(1) .EQ. 1 .AND. ISPOS(2) .EQ. 1 .AND.
>      ISPOS(3) .EQ. 1) THEN
          VOLREG = VOLSPH + VOLP2(1) + VOLP2(2) + VOLP2(3)
>      - VOLP(IDX) - VOLP(IDY) - VOLP(IDZ)
        ELSE IF (NISPOS(1) + NISPOS(2) +
>      NISPOS(3) .EQ. 1) THEN
          VOLREG = ISPOS(1)*VOLP2(1) - NISPOS(1)*VOLP(IDX)
>      + ISPOS(2)*VOLP2(2) - NISPOS(2)*VOLP(IDY)
>      + ISPOS(3)*VOLP2(3) - NISPOS(3)*VOLP(IDZ)
        ELSE IF (NISPOS(1) + NISPOS(2) +
>      NISPOS(3) .EQ. 2) THEN
          VOLREG = ISPOS(1)*VOLP2(1) + ISPOS(2)*VOLP2(2)
>      + ISPOS(3)*VOLP2(3)
        ELSE
          VOLREG = 0
        ENDIF
      ENDIF
    ENDIF
  ENDIF
  VOLINT = VOLINT + FACTX * FACTY * FACTZ * VOLREG
  IF (IPRINT .GE. 300) THEN
    WRITE (IOUT,6021) FACTX*FACTY*FACTZ*VOLREG
    WRITE (IOUT,6022) VOLINT
  ENDIF
  FACTZ = -FACTZ
130  CONTINUE
      FACTY = -FACTY
120  CONTINUE
      FACTX = -FACTX

```



```

110     CONTINUE
      ENDIF
*-----
*   Establish exit values
*-----
      IF (ABS(VOLINT - VOLSPH) .LT. DCUTOF) THEN
          VOLINT = VOLSPH
          NXTIRS = 2
      ELSE IF (ABS(VOLINT - VOLCAR) .LT. DCUTOF) THEN
          VOLINT = VOLCAR
          NXTIRS = 1
      ELSE IF (VOLINT .LT. DCUTOF) THEN
          VOLINT = 0.0D0
          NXTIRS = 0
      ELSE
          NXTIRS = -1
      ENDIF
      IF (IPRINT .GE. 200) THEN
          WRITE(IOUT,6012) VOLINT
          WRITE(IOUT,6001) NAMSBR
      ENDIF
*-----
*   Output formats
*-----
6000 FORMAT( '(*_Output_from_---', A6, '---_follows ', /)
6001 FORMAT( '---_Output_from_---', A6, '---_completed_* ) ')
6010 FORMAT( '---XYZCAR_={', 2(2(F20.10, ', ')/12X),
>           F20.10, ', ', F20.10, '}; ')
6011 FORMAT( '---POSSPH_={', 3(F20.10, ', ')/12X, F20.10, '}; ')
6012 FORMAT( '---Total_volume_of_the_intersection_:', F20.10, /)
6020 FORMAT(6X, 'Analysing_planes ', I2, ', ', I2, ' and ', I2)
6021 FORMAT(6X, 'VOLREG_=_', F20.10)
6022 FORMAT(6X, 'VOLINT_=_', F20.10, /)
6030 FORMAT(9X, 'Intersection_of_the_three_planes_inside_the_sphere ')
6031 FORMAT(9X, 'Intersection_of_two_planes_inside_the_sphere ',
>           ' in_direction_:', I1)
6032 FORMAT(9X, 'No_intersection_of_the_planes_inside_the_sphere ')
6033 FORMAT(9X, 'ILOC_=_', I8, I8, ' and_', I8, '/', 9X,
>           'XYZCAR_=_', F8.4, F8.4, ' and_', F8.4)
      END

```

A.2 Routine NXTISS

Cette routine sert à déterminer s'il existe une intersection entre deux sphères quelconques, et, le cas échéant, à en déterminer le volume.

```

*DECK NXTISS
      FUNCTION NXTISS(SPHER1, SPHER2, VOLINT)
*
*-----

```

```

*
* Purpose:
* Determine if an intersection exists between two spheres
*
* Author(s): M-A Lajoie
*
* Parameters: input
* SPHER1 Spatial description of the first spherical region with
*         SPHER1(0) the radius,
*         SPHER1(1) the $X$ position of center,
*         SPHER1(2) the $Y$ position of center, and
*         SPHER1(3) the $Z$ position of center.
* SPHER2 Spatial description of the second spherical region with
*         SPHER2(0) the radius,
*         SPHER2(1) the $X$ position of center,
*         SPHER2(2) the $Y$ position of center, and
*         SPHER2(3) the $Z$ position of center.
*
* Parameters: output
* NXTISS Type of intersection between cartesian region and
*         spherical region, where
*         \begin{itemize}
*         \item \verb|NXTISS| = 0 means that there is no intersection
*         between the two regions;
*         \item \verb|NXTISS| = 1 means that the first (SPHER1)
*         spherical region is all located inside the second (SPHER2)
*         spherical region;
*         \item \verb|NXTISS| = 2 means that the second (SPHER2)
*         spherical region is all located inside the first (SPHER1)
*         spherical region;
*         \item \verb|NXTISS| = -1 means that the intersection between
*         the two spherical regions is partial.
*         \end{itemize}
* VOLINT 3-D volume of intersection between the two spherical regions
*
*-----
*
*      IMPLICIT      NONE
*-----
* Subroutine arguments
*-----
*      INTEGER      NXTISS
*      DOUBLE PRECISION SPHER1(0:3), SPHER2(0:3), VOLINT
*-----
* Local parameters
*-----
*      INTEGER      IOUT
*      CHARACTER    NAMSBR*6
*      PARAMETER    (IOUT=6, NAMSBR='NXTISS')
*      INTEGER      IPRINT
*      PARAMETER    (IPRINT=1)
*      DOUBLE PRECISION DCUTOF

```

```

      PARAMETER      (DCUTOF=1.0D-8)
*-----
*  Other variables
*-----
      DOUBLE PRECISION XDR CST, PI
      DOUBLE PRECISION DIST ,R1 ,R2
      INTEGER          ICNT
*-----
*  Initialize parameters and exit values
*-----
      IF (IPRINT .GE. 200) THEN
        WRITE(IOUT,6000)  NAMSBR
        WRITE(IOUT,6010)  1 ,(SPHER1(ICNT) ,ICNT=0,3)
        WRITE(IOUT,6010)  2 ,(SPHER2(ICNT) ,ICNT=0,3)
      ENDIF
*-----
*  Default : No intersection
*-----
      NXTISS = 0
      VOLINT = 0.0D0
      PI = XDR CST('Pi' ,'_')
      R1 = SPHER1(0)
      R2 = SPHER2(0)
      IF (R1 .LT. 0 .OR. R2 .LT. 0)
        > CALL XABORT (NAMSBR/' : Negative sphere radius ')
*-----
*  Distance between the two centers
*-----
      DIST = SQRT((SPHER1(1)-SPHER2(1))**2
        >      + (SPHER1(2)-SPHER2(2))**2
        >      + (SPHER1(3)-SPHER2(3))**2)
*-----
*  No intersection
*-----
      IF (R1 + R2 - DIST .LT. DCUTOF) THEN
        IF (IPRINT .GE. 200) THEN
          IF (IPRINT .GE. 500) WRITE(IOUT,6032)
          WRITE(IOUT,6001) NAMSBR
        ENDIF
      RETURN
*-----
*  Intersection
*-----
      ELSEIF (DIST + R1 - R2 .LT. DCUTOF) THEN
*-----
*  Sphere 1 is inside sphere 2
*-----
      IF (IPRINT .GE. 500) WRITE (IOUT,6031) 1,2
      VOLINT = 4/3*PI*R1**3
      NXTISS = 1
      ELSEIF (DIST + R2 - R1 .LT. DCUTOF) THEN
*-----

```

```

* Sphere 2 is inside sphere 1
*-----
      IF (IPRINT .GE. 500) WRITE (IOUT,6031) 2,1
      VOLINT = 4/3*PI*R2**3
      NXTISS = 2
      ELSE
*-----
* Partial intersection between the spheres
*-----
      Formula taken from Wolfram Mathworld :
*
*  $V_{ISS} = \frac{\pi (R+r-d)^2 (d^2+2dr-3r^2+2dR+6rR-3R^2)}{12d}$ 
*
      IF (IPRINT .GE. 500) WRITE (IOUT,6030)
      VOLINT = (DIST**2 + 6*R1*R2
>             + 2*DIST*(R1 + R2) - 3*(R1**2 + R2**2))
>             * (R1 + R2 - DIST)**2 * PI / (12 * DIST)
      NXTISS = -1
      ENDIF
*-----
* Write output
*-----
      IF (IPRINT .GE. 200) THEN
        WRITE(IOUT,6012) VOLINT
        WRITE(IOUT,6001) NAMSBR
      ENDIF
*-----
* Output formats
*-----
6000 FORMAT('(* Output from ___',A6,'___ follows ',/)
6001 FORMAT('___ Output from ___',A6,'___ completed_*')
6010 FORMAT('___ Sphere ___',I1,' position _={',
>          3(F20.10,',',',',/23X),F20.10,'};')
6012 FORMAT('___ Total volume of the intersection :',F20.10,/)
6022 FORMAT(6X,'VOLINT_=_',F20.10,/)
6030 FORMAT(9X,'Partial intersection between the two spheres')
6031 FORMAT(9X,'Sphere ___',I1,' is located inside sphere ___',I1,/)
6032 FORMAT(9X,'No intersection between the spheres')
      END

```

A.3 Routine NXTVCS

Cette routine sert à calculer le volume de chaque région d'un MESH cartésien à trois dimensions contenant des PINS sphériques. Elle fait appel à NXTIRS.

```

*DECK NXTVCS
      SUBROUTINE NXTVCS(IPRINT ,MXMESH,MAXREG,MESH ,
>                      DMESH ,NBREG ,INDXSR,SURVOL)
*
*-----
*
* Purpose:

```

```

* Compute the volume of each region for a mixed Spherical/Cartesian
* 3-D geometry using the NXT tracking procedure.
*
* Copyright:
* Copyright (C) 2005 Ecole Polytechnique de Montreal.
*
* Author(s):
* M-A Lajoie
*
* Reference:
* G. Marleau,
* \textsl{New Geometries Processing in DRAGON: The NXT: Module},
* Report IGE-260, \'{E}cole Polytechnique de Montr\ 'e}al,
* Montr\ 'e}al, 2005.
*
* Parameters: input
* IPRINT print level.
* MXMESH maximum number of spatial subdivision in
* $R$ and $X$, $Y$ or $Z$.
* MAXSUR maximum number of surfaces in the geometry.
* MAXREG maximum number of regions in the geometry.
* MESH effective number of spatial subdivision in $R$
* and $X$, $Y$ or $Z$.
* DMESH spatial description of the sphere.
*
* Parameters: output
* NBREG number of regions in the geometry.
* INDXSR local indexing of surfaces/regions.
* SURVOL area/volume of regions.
*
* Comments:
* 1- Contents of the DMESH array:
* mesh in $X$ is  $x(i)=DMESH(i,1)$  for  $i=0,MESH(1)$ ;
* mesh in $Y$ is  $y(j)=DMESH(j,2)$  for  $j=0,MESH(2)$ ;
* mesh in $Z$ is  $z(k)=DMESH(k,3)$  for  $k=0,MESH(3)$ ;
* centre of sphere in  $(x,y,z)=(DMESH(-1,1),$ 
*  $DMESH(-1,2),$ 
*  $DMESH(-1,3))$ 
* radius of shells  $r(l)=DMESH(l,4)$ ,  $l=1,MESH(4)$ 
* 2- Contents of the INDXSR array:
* For  $i>0$ 
*  $INDXSR(1,i)=ix$  is the $X$ location of region  $i$ 
*  $INDXSR(2,i)=iy$  is the $Y$ location of region  $i$ 
*  $INDXSR(3,i)=iz$  is the $Z$ location of region  $i$ 
*  $INDXSR(4,i)=ir=0$  is the $R$ location of region  $i$ .
* For  $i<0$ 
* Surface analysis of spherical geometries is not implemented
*
*-----
*
* IMPLICIT NONE
*-----

```

* *Subroutine arguments*

```
*-----
      INTEGER          IPRINT ,MXMESH,MAXREG
      INTEGER          MESH(4)
      DOUBLE PRECISION DMESH(-1:MXMESH,4)
      INTEGER          NBREG,INDXSR(4,-1:MAXREG)
      DOUBLE PRECISION SURVOL(-1:MAXREG)
```

*-----
* *Local parameters*

```
*-----
      INTEGER          IOUT
      CHARACTER        NAMSBR*6
      PARAMETER        (IOUT=6,NAMSBR='NXTVCS')
      INTEGER          MAXDIM
      PARAMETER        (MAXDIM=4)
      DOUBLE PRECISION DCUTOF
      PARAMETER        (DCUTOF=1.0D-8)
      DOUBLE PRECISION DZERO,DONE,DIWO
      PARAMETER        (DZERO=0.0D0,DONE=1.0D0,DIWO=2.0D0)
```

*-----
* *Functions*

```
*-----
      DOUBLE PRECISION XDR CST, PI
      INTEGER          NXTIRS,ITYIRS
      DOUBLE PRECISION VOLINT
```

*-----
* *Local variables*

```
*-----
      INTEGER          ID,IR,IXYZ,IX,IY,IZ,IVOL,IVOLP

      DOUBLE PRECISION XYZCAR(6),POSPH(0:3)
```

*-----
* *Data*

```
*-----
      CHARACTER        CDIR(MAXDIM)*1
      SAVE             CDIR
      DATA            CDIR / 'X', 'Y', 'Z', 'R' /
```

*-----
* *Print header (if required) and initialize variables*

```
*-----
      IF (IPRINT .GE. 100) THEN
        WRITE(IOUT,6000) NAMSBR
        WRITE(IOUT,6002) 'CENTERXYZ'
        WRITE(IOUT,6006) (DMESH(-1,ID),ID=1,3)
        WRITE(IOUT,6003)
        WRITE(IOUT,6002) 'RADIUS'
        WRITE(IOUT,6006) (DMESH(IR,4),IR=1,MESH(4))
        WRITE(IOUT,6003)
        DO 100 ID=1,3
          WRITE(IOUT,6002) 'MESH'//CDIR(ID)
          WRITE(IOUT,6006) (DMESH(IXYZ,ID),IXYZ=0,MESH(ID))
          WRITE(IOUT,6003)
```

```

100    CONTINUE
      ENDIF
      CALL XDISET (INDXSR,4*(MAXREG+1),0)
      CALL XDDSET (SURVOL,(MAXREG+1),DZERO)
      NBREG=MESH(1)*MESH(2)*MESH(3)*MESH(4)
      IF (MAXREG .LT. NBREG) CALL XABORT(NAMSB//
>':_Insufficient_space_to_store_region_volumes')
*-----
*   Loop over radial regions for cartesian/spherical region
*   intersection
*-----
C     WRITE(*,*) 'SURVOL',SURVOL
      DO 101 IR=MESH(4),1,-1
        POSSPH(0)=DMESH(IR,4)
        POSSPH(1)=DMESH(-1,1)
        POSSPH(2)=DMESH(-1,2)
        POSSPH(3)=DMESH(-1,3)
        DO 111 IX=1,MESH(1)
          XYZCAR(1)=DMESH(IX-1,1)
          XYZCAR(2)=DMESH(IX,1)
          DO 121 IY=1,MESH(2)
            XYZCAR(3)=DMESH(IY-1,2)
            XYZCAR(4)=DMESH(IY,2)
            DO 131 IZ=1,MESH(3)
              XYZCAR(5)=DMESH(IZ-1,3)
              XYZCAR(6)=DMESH(IZ,3)
              ITYIRS=NXTIRS(XYZCAR,POSSPH,VOLINT)
              IF (ITYIRS .NE. 0) THEN
                IVOL = IR + MESH(4)*(IX-1)+MESH(4)*MESH(1)*(IY-1)
                >      + MESH(4)*MESH(1)*MESH(2)*(IZ-1)
                IVOLP = IVOL + 1
                IF (IR .NE. MESH(4)) THEN
                  SURVOL(IVOLP) = SURVOL(IVOLP)-VOLINT
                ENDIF
                SURVOL(IVOL)=VOLINT
                INDXSR(1,IVOL)=IX
                INDXSR(2,IVOL)=IY
                INDXSR(3,IVOL)=IZ
                INDXSR(4,IVOL)=IR
              ENDIF
            DO 131 IZ=1,MESH(3)
          DO 121 IY=1,MESH(2)
        DO 111 IX=1,MESH(1)
      DO 101 IR=MESH(4),1,-1
131    CONTINUE
121    CONTINUE
111    CONTINUE
101    CONTINUE
*-----
*   Print output
*-----
C     WRITE(*,*) 'SURVOL',SURVOL, ' INDXSR',INDXSR
C, ' IDREG',IDREG
C     WRITE(*,*) 'SURVOL',SURVOL(0)
      IF (IPRINT .GE. 100) THEN
        WRITE(IOUT,6002) 'SurVol'

```

```

        WRITE(IOUT,6005) (ID,(INDXSR(IR, ID),IR=1,4),SURVOL(ID),
>
        ID=0,NBREG)
        WRITE(IOUT,6003)
        WRITE(IOUT,6001) NAMSBR
    ENDIF
    RETURN
*-----
*   Output formats
*-----
6000 FORMAT('(*_Output_from_---',A6,'---_follows_')
6001 FORMAT('---_Output_from_---',A6,'---_completed_*)')
6002 FORMAT(A12,'={')
6003 FORMAT('};')
6005 FORMAT((5(I10,' ',' '),D20.10,':, ',' '))
6006 FORMAT(4(F20.10,':, ',' '))
6007 FORMAT((5(I10,' ',' '),F20.10,':, ',' '))
    END

```

A.4 Routine NXTLSP

Cette routine sert à déterminer la longueur des segments de droite formés par l'intersection d'une ligne dans une direction quelconque intersectant une sphère composée de plusieurs couches, et à les situer sur la droite originale.

```

*DECK NXTLSP
    FUNCTION NXTLSP(IPRINT,ITST ,NDIM ,MXMESH,LINMAX,
>
    MESH ,ORITRK,DIRTRK,DMESH ,
>
    NBCOR ,NBSINT,ISINT ,TRKLSI)
*
*-----
*
* Purpose:
* To track a spherical 3-D geometry
* using the NXT tracking procedure
*
* Copyright:
* Copyright (C) 2005 Ecole Polytechnique de Montreal
*
* Author(s): M-A Lajoie
*
* Reference:
* G. Marleau,
* \textsl{New Geometries Processing un DRAGON: The NXT: Module},
* Report IGE-260,
* \ 'E}cole Polytechnique de Montr\ ' {e}al, Montr\ ' {e}al,
* 2005.
*
* Parameters: input
* IPRINT print level
* ITST type of tracking, where

```



```

*      \begin{itemize}
*      \item \moc{ITST}=-1 means that only the exact geometry
*            is considered taking into account the submesh in
*            each direction.
*      \item \moc{ITST}= 0 means that only the global geometry
*            is considered without taking into account the
*            submesh in each direction.
*      \item \moc{ITST}= 1 means that both the global geometry
*            (as first step) ant the exact geometry are
*            considered taking into account the submesh in each
*            direction.
*      \end{itemize}
*
* NDIM      number of dimension *Always 3 in this case*
* MXMESH    maximum number of spatial subdivisions in $R$, $X$, $Y$
*            and $Z$.
* LINMAX    maximum number of segments in a track.
* MESH      effective number of spatial subdivisions in $X$, $Y$, $Z$
*            and $R$.
* ORITRK    a point on the track (origin).
* DIRTRK    the track direction (director cosines).
* DMESH     spatial description of the sphere.
*
* Parameters: output
* NXTLSP    Intersection flag between the track and the sphere, where
*            \begin{itemize}
*            \item \verb|NXTLSP| = 0 means there is no intersection;
*            \item \verb|NXTLSP| = 2 means there is an intersection;
*            \end{itemize}
* NBCOR    number of corners found for each external faces.* Obsolete*
* NBSINT    number of surfaces crossed by track.
* ISINT     the surfaces crossed by track.
* TRKLSI    the surface intersection distance.
*
*-----
*
*      IMPLICIT          NONE
*
*-----
* Subroutine arguments
*-----
*      INTEGER          MESH(4) ,NXTLSP ,NDIM ,IPRINT ,MXMESH
*      INTEGER          NBCOR(2) ,NBSINT ,ITST ,LINMAX ,ISINT  (0:4 ,LINMAX)
*      DOUBLE PRECISION ORITRK(3) ,DIRTRK(3)
*      DOUBLE PRECISION DMESH(-1:MXMESH,4) ,TRKLSI(LINMAX)
*
*-----
* Local parameters
*-----
*      INTEGER          IOUT
*      CHARACTER        NAMSBR*6
*      DOUBLE PRECISION DCUTOF
*      PARAMETER        (IOUT=6 ,NAMSBR='NXTLSP' ,DCUTOF=1.0D-8)
*
*-----
* Local variables

```

```

*-----
DOUBLE PRECISION TRKDIS,BEGEND(2),COMPAR(2)
DOUBLE PRECISION DISCR,PARAM1,PARAM2,A,B,C,CTRSPPH(3)
DOUBLE PRECISION DIST2,DIST2P,RDSSPH,PRECIS,SGMNTS(2)
DOUBLE PRECISION INTER(2,3),INTERP(2,3)
INTEGER ICNT,IFACE,IFIRST,NR,CNTOUT,DIR
CHARACTER STR*28
*-----
* Verify ITST option and reset to complete analysis if invalid
*-----
IF (ITST .LT. -1 .OR. ITST .GT. 1) ITST = 1
*-----
* Initialise variables
*-----
NXTLSP = 0
NBSINT = 0
NR = MESH(4)
CALL XDISET (NBCOR,2,0)
CALL XDISET (ISINT,5*(LINMAX),0)
CALL XDDSET (TRKLSI,(LINMAX),0.0D0)
IF (DIRTRK(1).LT.DCUTOFF .AND.
> DIRTRK(2).LT.DCUTOFF .AND.
> DIRTRK(3).LT.DCUTOFF) CALL XABORT('Invalid_track_direction')
*-----
* Print header, if required
*-----
IF(IPRINT .GT. 1000) THEN
WRITE(IOUT,6000) NAMSBR
WRITE(IOUT,6011) 'radialXYZ={_____}'
WRITE(IOUT,6012) (DMESH(ICNT,4),ICNT=1,NR)
WRITE(IOUT,6013)
WRITE(IOUT,6011) 'centerXYZ={_____}'
WRITE(IOUT,6012) DMESH(-1,1),DMESH(-1,2),DMESH(-1,3)
WRITE(IOUT,6013)
C WRITE(IOUT,6011) 'axial={_____}'
C WRITE(IOUT,6012) (DMESH(IC1,IDGP),IC1=0,NP)
C WRITE(IOUT,6013)
WRITE(IOUT,6011) 'trackorigin={_____}'
WRITE(IOUT,6012) ORITRK
WRITE(IOUT,6013)
WRITE(IOUT,6011) 'trackdirection={_____}'
WRITE(IOUT,6012) DIRTRK
WRITE(IOUT,6013)
ENDIF
C IF (IPRINT .GE. 100) THEN
C WRITE (IOUT,6000)NAMSBR
C WRITE (IOUT,6010)'Radial positions',
C > (DMESH(CNTOUT,4),CNTOUT=0,NR)
C WRITE (IOUT,6011)
C WRITE (IOUT,6010)'Center of sphere',
C > (DMESH(-1,CNTOUT),CNTOUT=1,3)
C WRITE (IOUT,6011)

```

```

C      WRITE (IOUT,6010) 'Track origin      ',ORITRK
C      WRITE (IOUT,6011)
C      WRITE (IOUT,6010) 'Track direction  ',DIRTRK
C      WRITE (IOUT,6011)
C      ENDIF
*-----
*   If ITST = 0,1, track global geometry
*-----
      IF (ITST .EQ. 0 .OR. ITST .EQ. 1) THEN
*-----
*   Calculate the parameters of the equation  $AX^{**2} + BX + C = 0$ 
*-----
      RDSSPH = DMESH(NR,4)
      A = 0.0D0
      B = 0.0D0
      C = -RDSSPH**2
      DO 100 ICNT = 1,3
        CTRSPH(ICNT) = DMESH(-1,ICNT)
        A = A + DIRTRK(ICNT)**2
        B = B + 2 * DIRTRK(ICNT) * (ORITRK(ICNT) - CTRSPH(ICNT))
        C = C + ORITRK(ICNT)**2 + CTRSPH(ICNT)**2
      >      - 2 * ORITRK(ICNT)      * CTRSPH(ICNT)
100    CONTINUE
      DISCR = B**2 - 4*A*C
*-----
*   If the discriminant is positive, then
*   the track intersects the global geometry
*-----
      IF (DISCR .GT. DCUTOF) THEN
        NBSINT = 2
      ENDIF
*-----
*   If ITST = -1, assume intersection
*-----
      ELSE
        NBSINT = 2
      ENDIF
      NXTLSP = NBSINT
      IF (NBSINT .EQ. 0) THEN
        IF (IPRINT .GT. 1000) THEN
          WRITE (IOUT,6002)
          WRITE (IOUT,6001) NAMSBR
        ENDIF
        RETURN
      ENDIF
      IF (NBSINT .NE. 2) CALL XABORT (NAMSBR//
      > ' : Invalid track : _NBSINT_not_equal_to_0_or_2 ')
      NBSINT = 0
*-----
*   Loop over $$
*-----
      DO 200 IFACE = 1,NR

```

```

*-----
*   Calculate the parameters of the equation  $AX^{**2} + BX + C = 0$ 
*-----
      RDSSPH = DMESH(IFACE,4)
      IF (IFACE.NE.1 .AND. RDSSPH-DMESH(IFACE-1,4) .LT. DCUTOF)
    >   CALL XABORT ('Sphere_radii_not_in_the_good_order')
      A = 0.0D0
      B = 0.0D0
      C = -RDSSPH**2
      DO 201 ICNT = 1,3
        CTRSPH(ICNT) = DMESH(-1,ICNT)
        A = A + DIRTRK(ICNT)**2
        B = B + 2 * DIRTRK(ICNT) * (ORITRK(ICNT) - CTRSPH(ICNT))
        C = C + ORITRK(ICNT)**2 + CTRSPH(ICNT)**2
    >       - 2 * ORITRK(ICNT) * CTRSPH(ICNT)
201   CONTINUE
      DISCR = B**2 - 4*A*C
*-----
*   If the discriminant is positive, then there is
*   an intersection with the considered shell.
*   In that case, compute the intersection points
*   Else, the track doesn't intersect, go to next shell
*-----
      IF (DISCR .GT. DCUTOF) THEN
C      IF (IPRINT .GE. 300)
C    >   WRITE (IOUT,6014) 'Intersection with shell      ',IFACE
      PARAM1 = (-B + SQRT(DISCR)) / (2*A)
      PARAM2 = (-B - SQRT(DISCR)) / (2*A)
      DO 202 ICNT = 1,3
        INTERP(1,ICNT) = INTER(1,ICNT)
        INTER (1,ICNT) = PARAM1 * DIRTRK(ICNT) + ORITRK(ICNT)
        INTERP(2,ICNT) = INTER(2,ICNT)
        INTER (2,ICNT) = PARAM2 * DIRTRK(ICNT) + ORITRK(ICNT)
202   CONTINUE
      IF (IPRINT .GE. 400) THEN
        DO 203 ICNT = 1,2
          WRITE (STR, '(A13,I1,A12,I2)') 'Intersection_',ICNT,
    >           '_with_shell_',IFACE
          WRITE (IOUT,6017) STR,(INTER(ICNT,CNTOUT),CNTOUT=1,3)
          WRITE (IOUT,6018)
203   CONTINUE
      ENDIF
*-----
*   Calculate the distance between the points
*-----
      DIST2P   = DIST2
      DIST2    = (INTER (1,1) - INTER (2,1))**2
    >           + (INTER (1,2) - INTER (2,2))**2
    >           + (INTER (1,3) - INTER (2,3))**2
      SGMNTS(1) = (INTER (1,1) - INTERP(1,1))**2
    >           + (INTER (1,2) - INTERP(1,2))**2
    >           + (INTER (1,3) - INTERP(1,3))**2

```

```

      SGMNTS(2) = (INTER (2,1) - INTERP(2,1))**2
>              + (INTER (2,2) - INTERP(2,2))**2
>              + (INTER (2,3) - INTERP(2,3))**2
      IF (NBSINT .EQ. 0) THEN
          TRKLSI(NR-IFACE+2) = SQRT(DIST2)
          IFIRST = IFACE
      ELSE
          IF (ABS(SGMNTS(1) - SGMNTS(2)) .GT. DCUTOF)
>              CALL XABORT('No_segment_symmetry')
          TRKLSI(NR-IFACE+2) = SQRT(SGMNTS(1))
          TRKLSI(NR+IFACE+2-2*IFIRST) = SQRT(SGMNTS(2))
*-----
*   Verify that all segments lengths add up right
*-----
          PRECIS = SQRT(DIST2 ) - SQRT(SGMNTS(2))
>              - SQRT(DIST2P) - SQRT(SGMNTS(2))
          IF (ABS(PRECIS) .GT. DCUTOF)
>              CALL XABORT ('Problem_calculating_distances')
          ENDIF
          NBSINT = NBSINT + 2
      ELSE
          IF (IPRINT .GE. 300)
>              WRITE(IOUT,6016) 'No_intersection_with_shell_',IFACE
          ENDIF
200  CONTINUE
*-----
*   Reorganise segments, and add delimiting beginning/ending
*-----
      IF (NBSINT .GE. 2) THEN
          DO 300 ICNT=1,2
              DO 301 CNTOUT=1,3
                  IF (DIRTRK(CNTOUT) .GE. DCUTOF) THEN
                      BEGEND(ICNT) = ( INTER(ICNT,CNTOUT) - ORITRK(CNTOUT) )
>                                  / DIRTRK(CNTOUT)
                  ENDIF
301  CONTINUE
              DO 302 CNTOUT=1,3
                  IF (DIRTRK(CNTOUT) .GE. DCUTOF) THEN
                      IF (BEGEND(ICNT) - (INTER(ICNT,CNTOUT) - ORITRK(CNTOUT))
>                          / DIRTRK(CNTOUT) .GT. DCUTOF)
>                          CALL XABORT('Problem_calculating_BEGEND')
                  ENDIF
302  CONTINUE
300  CONTINUE
          IF (BEGEND(1) .LT. BEGEND(2)) THEN
              TRKLSI(1) = BEGEND(1)
              TRKLSI(NBSINT+1) = BEGEND(2)
          ELSE
              TRKLSI(1) = BEGEND(2)
              TRKLSI(NBSINT+1) = BEGEND(1)
          ENDIF
*-----

```

```

* Find number of corners for initial surface intersections
*-----
      NBCOR(1)=1
C      DO 160 ICNT=2,NBSINT+1
C          IF (TRKLSI(ICNT) .GT. TRKLSI(1)) GO TO 165
C          NBCOR(1)=NBCOR(1)+1
C 160    CONTINUE
C 165    CONTINUE
*-----
* Find number of corners for final surface intersections
*-----
      NBCOR(2)=1
C      DO 170 ICNT=NBSINT-1,1,-1
C          IF (TRKLSI(ICNT) .LT. TRKLSI(NBSINT+1)) GO TO 175
C          NBCOR(2)=NBCOR(2)+1
C 170    CONTINUE
C 175    CONTINUE
      ENDIF
*-----
* Identify segments
*-----
      ISINT(4,1)          = -2
      ISINT(4,NBSINT+1) = -2
      DO 400 ICNT = 2,NBSINT/2+1
          ISINT(4,ICNT)          = NR-ICNT+2
          ISINT(4,NBSINT-ICNT+2) = NR-ICNT+2
400    CONTINUE
*-----
* Verify number of intersections and number of segments
*-----
      IF (NBSINT .NE. 2*(NR-IFIRST+1)) CALL XABORT ('NBSINT_not_OK')
*-----
* Print intersection points
*-----
C      WRITE(*,*) 'NBCOR',NBCOR
C      WRITE(*,*) 'TRKLSI',TRKLSI
      IF (IPRINT .GT. 1000) THEN
          WRITE(IOUT,6011) 'Initial_planes.....'
          DO 500 ICNT=1,NBCOR(1)
              WRITE(IOUT,6010) TRKLSI(ICNT),
> (ISINT(IDIR,ICNT),IDIR=0,4)
500    CONTINUE
          IF (NBCOR(1)+1 .LE. NBSINT-NBCOR(2)) THEN
              WRITE(IOUT,6011) 'Intermediate_planes_'
              DO 501 ICNT=NBCOR(1)+1,NBSINT-NBCOR(2)
                  WRITE(IOUT,6010) TRKLSI(ICNT),
> (ISINT(IDIR,ICNT),IDIR=0,4)
501    CONTINUE
          ENDIF
          WRITE(IOUT,6011) 'Final_planes.....'
          DO 502 ICNT=NBSINT-NBCOR(2)+1,NBSINT
              WRITE(IOUT,6010) TRKLSI(ICNT),

```

```

      > (ISINT(IDIR,ICNT),IDIR=0,4)
502  CONTINUE
      ENDIF
*-----
*   Finish output
*-----
*-----
*   Print final track information
*-----
      IF(IPRINT .GT. 1000) THEN
        WRITE(IOUT,6011) 'Initial_face.....'
        DO ICNT=1,NBCOR(1)
          WRITE(IOUT,6010) TRKLSI(ICNT),
      > (ISINT(IDIR,ICNT),IDIR=1,4)
          ENDDO
        WRITE(IOUT,6011) 'Regions.....'
        DO ICNT=NBCOR(1)+1,NBSINT-NBCOR(2)+1
          WRITE(IOUT,6010) TRKLSI(ICNT),
      > (ISINT(IDIR,ICNT),IDIR=1,4)
          ENDDO
        WRITE(IOUT,6011) 'Final_face.....'
        DO ICNT=NBSINT-NBCOR(2)+2,NBSINT+1
          WRITE(IOUT,6010) TRKLSI(ICNT),
      > (ISINT(IDIR,ICNT),IDIR=1,4)
          ENDDO
        WRITE(IOUT,6001) NAMSBR
      ENDIF
      RETURN
C     IF (IPRINT .GE. 200) THEN
C       WRITE(IOUT,6010) 'Segments lengths ',
C     > (TRKLSI(ICNT),ICNT=1,NBSINT+1)
C       WRITE(IOUT,6011)
C       WRITE(IOUT,6001) NAMSBR
C     ENDIF
*-----
*   Output formats
*-----
6000 FORMAT('*_Output_from_---',A6,'---_follows')
6001 FORMAT('---Output_from_---',A6,'---_completed_*')
6002 FORMAT('---No_region_crossed')
6010 FORMAT(1X,F25.16,5I10)
6011 FORMAT(A20)
6012 FORMAT(6(1X,F25.16,:',','))
6013 FORMAT('};')
6014 FORMAT(6I10)
6015 FORMAT(6(1X,F25.16))
C 6010 FORMAT(3X,A16,'={',F25.13,',', '/(22X,F25.13,:',','),$)
6018 FORMAT('};')
6017 FORMAT(9X,A28,'_={',F25.13,',', '/(40X,F25.13,:',','),$)
6016 FORMAT(6X,A27,I2/)
      END

```

A.5 Routine NXXTPRD

Cette routine sert à générer aléatoirement les positions des éléments sphériques lorsque le mot-clé DPIN est utilisé.

```

*DECK NXXTPRD
      SUBROUTINE NXXTPRD(NDIM ,XYZCAR, ISEED ,PINCEN)
*
*-----
*
* Purpose:
* To generate the position of a randomly distributed pin inside a
* cylindrical cell
*
* Copyright:
* Copyright (C) 2005 Ecole Polytechnique de Montreal.
*
* Author(s):
* M-A Lajoie
*
* Update(s):
* None.
*
* Reference:
* G. Marleau,
* \textsl{New Geometries Processing in DRAGON: The NXX: Module},
* Report IGE-260, \ 'E}cole Polytechnique de Montr\ '{e}al,
* Montr\ '{e}al, 2005.
*
* Parameters: input
* NDIM number of dimensions for the problem
* XYZCAR cell limits
* ISEED Seed for pseudo-random number generation
* PINCEN(0) Radius of the pin
*
* Parameters: output
* PINCEN center of randomly generated pin
*
*-----
*
*      IMPLICIT          NONE
*
* Subroutine arguments
*-----
*      INTEGER          NDIM, ISEED
*      DOUBLE PRECISION XYZCAR(6) ,PINCEN(0:NDIM)
*
* Local parameters
*-----
*      INTEGER          IOUT
*      CHARACTER        NAMSBR*6
*      PARAMETER        (IOUT=6, NAMSBR= 'NXXTPRD' )

```



```

      INTEGER          IPRINT
      PARAMETER       (IPRINT=1)
*-----
*  Local variables
*-----
      INTEGER          IDIM,IFIRST
      DOUBLE PRECISION RAND,RADIUS
*-----
*  Processing starts
*-----
      IF (IPRINT .GE. 500) WRITE(IOUT,6000) NAMSBR
      RADIUS = DBLE(PINCEN(0))
c      ISEED = SECNDS(0.0)
      CALL RANDF (ISEED,IFIRST,RAND)
      DO IDIM = 1,NDIM
          CALL RANDF(ISEED,IFIRST,RAND)
          PINCEN(IDIM) = ((XYZCAR(2*IDIM)-XYZCAR(2*IDIM-1)) - 2*RADIUS)
>          * (RAND-DBLE(0.5))
      ENDDO
C      WRITE(*,*) PINCEN
      IF (IPRINT .GE. 500) WRITE(IOUT,6001) NAMSBR
      RETURN
*-----
*  Output formats
*-----
6000 FORMAT('(*_Output_from_---',A6,'---_follows_')
6001 FORMAT('---_Output_from_---',A6,'---_completed_*)')
      END

```

Annexe B

Guide d'utilisation pour les géométries sphériques

Le format à utiliser pour définir une géométrie sphérique selon le format CLE-2000 (Roy, 1999) est donné comme suit :

```

CLUSTER PINSPH
::: PINSPH := GEO: SPHERE lr
    {NPIN npins
      CPINX (xpins(i), i=1,npins)
      CPINY (ypins(i), i=1,npins)
      CPINZ (zpins(i), i=1,npins) |
      DPIN dpins}
    RADIUS (rrr(i), i=1,lr+1)
    MIX (imix(i), i=1,lr)
;

```

où l'on utilisera les variables et mots-clés suivants :

<i>PINSPH</i>	nom du CLUSTER ;
SPHERE	mot-clé pour indiquer une géométrie sphérique ;
<i>lr</i>	nombre de coquilles composant l'élément de géométrie sphérique ;
NPIN	mot-clé pour indiquer le nombre d'éléments dans le CLUSTER ;
<i>npins</i>	nombre de pins dans la sous-géométrie <i>PINSPH</i> ;
CPINX	mot-clé pour indiquer les positions <i>x</i> des centres des pins dans le CLUSTER ;
<i>xpinx</i>	vecteur de dimension <i>npins</i> contenant les positions <i>x</i> (cm) des centres pins dans la sous-géométrie ;
CPINY	mot-clé pour indiquer les positions <i>y</i> des centres des pins dans le CLUSTER ;
<i>ypinx</i>	vecteur de dimension <i>npins</i> contenant les positions <i>y</i> (cm) des centres pins dans la sous-géométrie ;
CPINZ	mot-clé pour indiquer les positions <i>z</i> des centres des pins dans le CLUSTER ;
<i>zpins</i>	vecteur de dimension <i>npins</i> contenant les positions <i>z</i> (cm) des centres pins dans la sous-géométrie ;
DPIN	mot-clé pour spécifier qu'une distribution aléatoire des pins dans le CLUSTER suit, déterminée par une densité $d_{p,r}$;
<i>dpins</i>	la densité de pins désirée, $d_{p,r}$. Un nombre $N_{p,r}$ de pins sera généré par addition séquentielle aléatoire, dans la géométrie de niveau précédent, donné par

$$N_{p,r} = \text{MAX} \left(1, \text{NINT} \left(\frac{d_{p,r} V_c}{V_p} \right) \right) \quad (\text{B.1})$$

avec V_c le volume de la cellule dans laquelle les pins sont générées, et V_p le volume d'une pin ;

RADIUS mot-clé pour spécifier les rayons de chacune des couches des sphères qui composent le **CLUSTER** ;

rrr vecteur de dimension $lr+1$ contenant les rayons (cm) croissants successifs des sphères. Il est à noter qu'obligatoirement, $rrr(1) = 0.0$;

MIX mot-clé pour indiquer les compositions des sphères successives composant les pins ;

imix vecteur de dimension $lr+1$ contenant les mélanges composant les sphères imbriquées.

Annexe C

Article : Application of 3D CP Theory to VHTR Spherical Geometries

Cet annexe présente un article écrit pour être présenté à la conférence PHYSOR 2010, et a été soumis pour publication au moment de l'écriture du présent mémoire.

Lajoie M-A., Marleau G., Martin. N., Hébert A.

École Polytechnique de Montréal, Institut de Génie Nucléaire

P.O. Box 6079, Station "Centre-Ville", Montréal, Québec, H3C 3A7, Canada

C.1 ABSTRACT

The analysis of VHTR fuel tends to be difficult, when using deterministic methods currently employed in lattice codes, notably because of limitations on geometry representation, and the stochastic positioning of spherical elements. The method proposed here and implemented in the lattice code DRAGON is to generate the positions of multi-layered spheres using random sequential addition, and to analyze the resulting geometry using a full three-dimensional spherical collision probability method. The preliminary validation runs are consistent with results obtained through the Monte-Carlo method. Comparisons have also been made with two double-heterogeneity models already available in DRAGON.

C.2 INTRODUCTION

Fourth generation very high temperature reactors (VHTRs) present characteristics that tend to be harder than conventional geometries to analyze using methods available in traditional deterministic neutronics computer codes. Worthy of interest in the scope of this document are the random character of fuel elements in a uniform matrix, and the multi-layered, spherical geometry of these fuel elements.

The goal of this project has been to integrate the capacity to analyze such properties in the transport lattice code DRAGON (Marleau *et al.*, 1992). First, a geometrical analysis of spherical volumes has allowed us to develop analytical volumes calculation capacities. Then, integration line tracking for spherical geometries has been implemented. Routines enabling this exact geometrical analysis of multi-layered, randomly disposed spherical particles have been successfully incorporated and tested in this code. A thorough analysis of combinations of spherical and Cartesian meshes has also been performed, to ensure the validity of the lengths and volumes calculated by the new routines.

Finally, preliminary validation runs have been carried out, comparing results obtained using the new module on a simple, three dimensional, spherical geometry for a benchmark established using the Monte-Carlo code Serpent (Leppänen, 2009) and two double-heterogeneity treatment models already implemented in DRAGON. The new capabilities have been tested with an increasing angular and spatial discretization, leading to results approaching those generated by the Monte-Carlo method and the double-heterogeneity models.

This paper will first present a quick theoretical review of the collision probability method, followed by the geometrical analysis required for volumes calculation and tracking lines analysis. Then, the verification process will be presented, followed by the results of the validation for an infinite grain lattice benchmark.

C.3 THEORY

The neutron transport equation can be solved using various methods, some of which use a geometrical discretization analysis technique known as tracking. We will here present the collision probability method and formalism, as presented by Hébert (A. Hébert, 2009).

C.3.1 The Collision Probability Method

The integral form of the multigroup steady-state neutron transport equation can be written, assuming isotropic sources and scattering, as:

$$\phi_g(\vec{r}) = \frac{1}{4\pi} \int_{4\pi} \int_0^\infty e^{-\tau_g(s)} Q_g(\vec{r} - s\hat{\Omega}) ds d^2\Omega \quad (\text{C.1})$$

where $\phi_g(\vec{r})$ is the neutron flux in energy group g at position \vec{r} , and where, using Σ_g as the transport corrected total cross section, the optical path is $\tau_g(s) = \int_0^s \Sigma_g(\vec{r} - s'\hat{\Omega}) ds'$. The

isotropic neutron source term, Q_g can be expressed as a sum of scattered and fission-produced neutrons, as:

$$Q_g(\vec{r}) = \sum_{h=1}^G \Sigma_{s,0,g \leftarrow h}(\vec{r}) \phi_h(\vec{r}) + \frac{1}{k_{\text{eff}}} \sum_{j=1}^J \chi_{j,g} \sum_{h=1}^G \nu \Sigma_{f,j,h}(\vec{r}) \phi_h(\vec{r}) \quad (\text{C.2})$$

with $\Sigma_{s,0,g \leftarrow h}$ the transport corrected scattering cross section, $\nu \Sigma_{f,j,h}$ the fission rate of isotope j , and $\chi_{j,g}$ its fission spectrum.

A volume discretization is then considered, where the unit cell is subdivided in N subregions, each with volume V_i , $i = 1 \dots N$. Each of the subregions will be assumed to contain a source $Q_{i,g}$ that is uniform over its volume, as well as a uniform total macroscopic cross-section $\Sigma_{i,g}$. Using $\phi_{i,g} = \frac{1}{V_i} \int_{V_i} \phi_g(\vec{r}) d^3r$ as the averaged flux in region V_i , we can reformulate equation C.1, after multiplication by $\Sigma_{i,g}$ and integration over V_j , as:

$$\phi_{i,g} = \sum_j Q_{j,g} p_{ij,g} \quad (\text{C.3})$$

where, using s as an index representing the distance traveled on a track in direction $\hat{\Omega}$, we have defined the reduced collision probability $p_{ij,g}$, as:

$$p_{ij,g} = \frac{1}{4\pi V_i} \int_{V_j} \int_{4\pi} \int_{-\infty}^{\infty} \mathcal{W}_{V_i}(s, \hat{\Omega}) e^{-\tau_g(s)} ds d^2\Omega d^3r \quad (\text{C.4})$$

with $\mathcal{W}_{V_i}(s, \hat{\Omega})$ being a characteristic function worth 1 if point s of track trajectory $\hat{\Omega}$ is in volume V_i , and 0 otherwise. This reduced collision probability is defined in such a way that it remains finite as $\Sigma_{j,g}$ tends to zero, and so that $\Sigma_{j,g} p_{ij,g}$ is the probability for a neutron born in region i , in energy group g , to have its first collision in region j .

C.3.2 Tracking

An angular discretization is then performed, where planes $\Pi_{\hat{\Omega}}$ are defined orthogonal to direction $\hat{\Omega}$. Each plane is characterized by orthogonal lines (*tracks*), each with a starting point \vec{p}' on plane $\Pi_{\hat{\Omega}}$, and also propagating in direction $\hat{\Omega}$. This allows us to rewrite volume integrals as integrals over the whole tracking as:

$$\int_{4\pi} \int_{V_j} \mathcal{F}(\vec{r}, \hat{\Omega}) d^3r d^2\Omega = \int_{4\pi} \int_{\Pi_{\hat{\Omega}}} \int_{-\infty}^{\infty} \mathcal{W}_{V_j}(s', \hat{\Omega}) \mathcal{F}(s', \hat{\Omega}) ds' d^2p' d^2\Omega \quad (\text{C.5})$$

We can therefore evaluate the volumes and reduced collision probabilities as:

$$V_{j,\Omega} = \int_{\Pi_{\hat{\Omega}}} \int_{-\infty}^{\infty} \mathcal{W}_{V_j}(s', \hat{\Omega}) ds' d^2p' \quad (\text{C.6})$$

$$p_{ij,g} = \frac{1}{4\pi V_i} \int_{4\pi} \int_{-\infty}^{\infty} \left[\int_{\Pi_{\hat{\Omega}}} \int_{-\infty}^{\infty} \mathcal{W}_{V_j}(s, \hat{\Omega}) \mathcal{W}_{V_i}(s', \hat{\Omega}) e^{-\tau_g(s)} ds' d^2p' \right] ds d^2\Omega \quad (\text{C.7})$$

In practice, since an analytic integration cannot be performed, the tracking of the geometry will be done by supposing a finite number of directions $\hat{\Omega}_m$, each with an associated weight ω_m so that $\int_{4\pi} d^2\Omega = \sum \omega_m \hat{\Omega}_m = 4\pi$, and a surfacic track density on each plane defined by $\hat{\Omega}_m$, each track also being weighted with weight \mathcal{T}_n so that $\sum_n \mathcal{T}_n = 1$, and crossing regions identified by indices k or l . We can then evaluate numerically equations C.6 and C.7 as:

$$V_{j,m} \simeq \sum_n \mathcal{T}_n \sum_k \delta_{k,j} \quad (\text{C.8})$$

$$p_{ij} \simeq \frac{1}{\sum_i \sum_j V_i} \sum_m \omega_m \sum_n \mathcal{T}_n \sum_k \delta_{k,i} \sum_l \delta_{h,j} e^{-\tau_{k,l}} [1 - e^{-\Sigma_i L_k}] [1 - e^{-\Sigma_j L_h}] \quad (\text{C.9})$$

C.4 GEOMETRICAL ANALYSIS

The implementation in DRAGON of the capacity for the analysis of multi-layered spherical particles has two components. First, the tracking itself, which is the main component, although mathematically quite trivial, and second, the analytical volumes calculation, which serves as a tracking verification, by comparing analytical and numerical volumes, computed using equation C.8. Both steps are performed by the module `NXT`: (Zkiek et Marleau, 2007).

C.4.1 Spherical 3-D Tracking

The `NXT`: module generates a number of tracking lines, based on user-specified track density and angular quadrature, each line being characterized by (x_l, y_l, z_l) , its starting point, and (a_x, a_y, a_z) , its orientation or direction cosines. Each sphere is also characterized by the coordinates of its center (x_s, y_s, z_s) , and its radius R . Hence, we have the following equations, respectively for the line and the sphere:

$$\frac{x - x_l}{a_x} = \frac{y - y_l}{a_y} = \frac{z - z_l}{a_z} = t \quad (\text{C.10})$$

$$(x - x_s)^2 + (y - y_s)^2 + (z - z_s)^2 = R^2 \quad (\text{C.11})$$

These can easily be solved for the intersections coordinates $[x_1, y_1, z_1]$ and $[x_2, y_2, z_2]$:

$$[x_1, y_1, z_1] = [t_1 a_x + x_l, t_1 a_y + y_l, t_1 a_z + z_l] \quad (\text{C.12})$$

$$[x_2, y_2, z_2] = [t_2 a_x + x_l, t_2 a_y + y_l, t_2 a_z + z_l] \quad (\text{C.13})$$

where t_1 and t_2 are the roots of the quadratic equation

$$(a_x t + x_l - x_s)^2 + (a_y t + y_l - y_s)^2 + (a_z t + z_l - z_s)^2 = R^2 \quad (\text{C.14})$$

C.4.2 Volumes

The volumes that need to be computed here are the volumes of spherical shells (which are trivial) and those corresponding to the intersection of a Cartesian and spherical region. The methodology employed for the latter relies on a linear combination of the volumes of the intersection situated *below*, to the *left* and *behind* each corner of the parallelepiped formed by the intersection of the six planes delimiting the Cartesian region. Using indices + and – to identify the two planes relative to each axis, the volume of the total intersection can be given by:

$$V_{\text{Inter}} = V_{[+,+,+]} - V_{[-,+,+]} - V_{[+,-,+]} - V_{[+,-,-]} + V_{[-,-,+]} + V_{[-,+,-]} + V_{[+,-,-]} - V_{[-,-,-]} \quad (\text{C.15})$$

Recentering the sphere to the origin, and applying the same translation to each of the planes, we can see that a number of types of intersections can be found. The most trivial case occurs when no intersection occurs between the three concerned planes and the sphere, where the volume is null except if the intersection of the three planes occurs in the x_+ , y_+ and z_+ octant, in which case the desired volume is that of the whole sphere $V_S = \frac{4\pi R^3}{3}$. Otherwise, if an intersection between at least one of the three Cartesian planes and the sphere takes place, one of the following three cases has to be considered:

- No intersection between any of the three planes occurs inside the sphere;
- One, two or three intersections of two of the three planes is found inside the sphere, but the intersection of the three planes is located outside the sphere;
- The intersection of the three planes is found inside the sphere.

No intersection between the planes inside the sphere

When no intersection of the planes takes place inside the sphere, the only volumes that need to be considered are spherical caps (see Figure C.1). These volumes can be calculated using

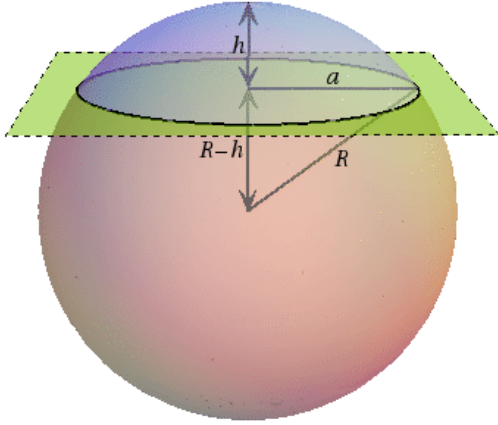


Figure C.1 Intersection of a plane with a sphere. Here, $a^2 = R^2 - z^2$

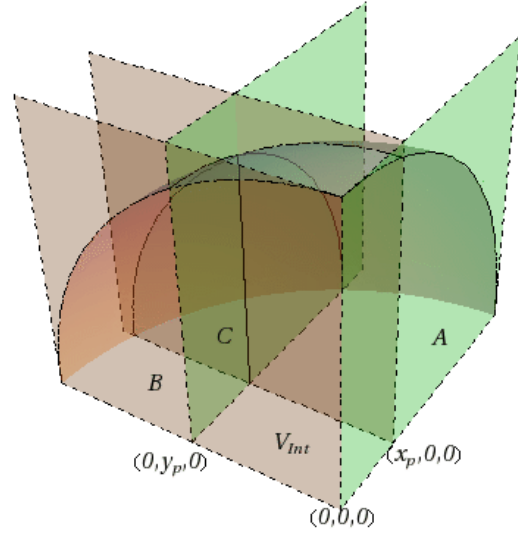


Figure C.2 Region formed by the intersection of two planes inside the sphere

an integral over the axis orthogonal to the plane:

$$V_{Cap} = \int_{R-h}^R \pi(R^2 - z^2) dz \quad (\text{C.16})$$

$$= \frac{\pi h^2(3R - h)}{3} \quad (\text{C.17})$$

This gives, for the volume delimited by a plane at position h in the negative direction of axis i (x , y or z):

$$V_i = V_S - V_{Cap} = \begin{cases} \frac{4}{3}\pi R^3 & \text{for } h \leq 0 \\ \frac{\pi}{3} [4R^3 - 3Rh^2 + h^3] & \text{for } 0 < h < 2R \\ 0 & \text{for } h \geq 2R \end{cases} \quad (\text{C.18})$$

Once again, it is important to consider where the intersection between the three planes occur, the desired volume being null, or given by a combination of V_x , V_y , V_z and V_S in some combinations of the planes positions.

Intersection of two planes inside the sphere

The intersection of two planes inside the sphere creates the region illustrated in Figure C.2.

The volume represented by V_{Int} can be calculated again by an integral:

$$V_{Int(x,y)} = \int_0^{y_p} \int_0^{x_p} \sqrt{R^2 - x^2 - y^2} dx dy \quad (C.19)$$

$$= \int_0^{y_p} \frac{x_p}{2} \sqrt{R^2 - x_p^2 - y^2} + \frac{R^2 - y^2}{2} \arctan \left(\frac{x_p}{\sqrt{R^2 - x_p^2 - y^2}} \right) dy \quad (C.20)$$

$$= \frac{1}{6} \left[2x_p y_p \sqrt{R^2 - x_p^2 - y_p^2} \right. \\ \left. - y_p (y_p^2 - 3R^2) \arctan \left(\frac{x_p}{\sqrt{R^2 - x_p^2 - y_p^2}} \right) \right. \\ \left. - x_p (x_p^2 - 3R^2) \arctan \left(\frac{y_p}{\sqrt{R^2 - x_p^2 - y_p^2}} \right) \right. \\ \left. - 2R^3 \arctan \left(\frac{x_p y_p}{R \sqrt{R^2 - x_p^2 - y_p^2}} \right) \right] \quad (C.21)$$

The desired volume can now be calculated by simple observation of Figure C.2, and establishing:

$$\frac{V_S}{8} = \frac{V_i}{4} + \frac{V_j}{4} + V_{Int(i,j)} - \frac{V_{(i,j)}}{2} \quad (C.22)$$

leading to:

$$V_{(i,j)} = 2V_{Int(i,j)} + \frac{V_i + V_j}{2} - \frac{V_S}{4} \quad (C.23)$$

The required volume will, once again depending on the position of the intersection of the three planes and the number of intersections of two planes occurring inside the sphere, be expressed as a combination of $V_{(i,j)}$ and V_i .

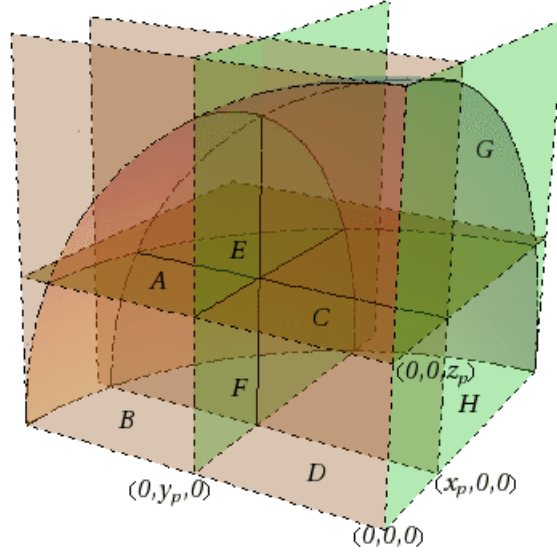


Figure C.3 Volumes created by the intersection of three planes inside the sphere

Intersection of the three planes inside the sphere

The intersection of the three planes inside the sphere creates the region illustrated in Figure C.3. By studying the figure, we can find the desired volume by observing:

$$\|x_p y_p z_p\| = V_D \quad (\text{C.24})$$

$$V_{Int(x,z)} = V_B + V_D \quad (\text{C.25})$$

$$V_{Int(x,y)} = V_C + V_D \quad (\text{C.26})$$

$$V_{Int(y,z)} = V_H + V_D \quad (\text{C.27})$$

$$\frac{V_x}{4} = V_E + V_F + V_G + V_H \quad (\text{C.28})$$

$$\frac{V_y}{4} = V_A + V_B + V_E + V_F \quad (\text{C.29})$$

$$\frac{V_z}{4} = V_A + V_C + V_E + V_G \quad (\text{C.30})$$

$$\frac{V_S}{8} = V_A + V_B + V_C + V_D + V_E + V_F + V_G + V_H \quad (\text{C.31})$$

This gives, in the case where $x_p < 0$, $y_p < 0$ and $z_p < 0$, for the desired volume $V_E = V_{(x,y,z)}$:

$$V_{(x,y,z)} = x_p y_p z_p + V_{Int(x,y)} + V_{Int(x,z)} + V_{Int(y,z)} + \frac{V_x + V_y + V_z - V_S}{4} \quad (\text{C.32})$$

It can also be verified that this relation is valid for any position of the planes, as long as $(x_p^2 + y_p^2 + z_p^2 \leq R^2)$. Fortran routines have been written to calculate the volumes, and verified with the equivalent in Matlab (The MathWorks, Inc., 2005), and incorporated in the `NXT`: set of routines.

C.5 VERIFICATION

C.5.1 Tracking

Tracking verification was performed for a unit cell containing a multi-layered sphere. Selecting a few lines (5 or 6) lines out of the whole set, both track lengths and intersection points with the elements of the geometry, for various positions and radii of the spherical elements, were analyzed, looking for errors greater than the numerical precision. A secondary visual verification was also performed, using module `TLM`:, able to generate a Matlab file that can illustrate either the whole set of tracking lines (Figure C.4), or the intersections of the tracking lines with specified planes (Figure C.5), the former option producing figures that become rapidly crowded with an increasing number of lines being represented, but being however very useful for detecting segments out of their respective regions, or regions not being tracked.

C.5.2 Volumes

The first verification was performed by observing that error induced in analytically calculated volumes by `NXT`: were all within the numerical precision of the machine used. Another step regarding verification has been performed, using the numerical volumes calculated with the weighted segments presented in section C.3.2, to ensure both internal code consistency, and that the refinement of tracking parameters (track density and angular quadrature) leads to a better numerical volume evaluation. An exemple can be seen in Table C.1, for a simple $(3 \times 3 \times 3)$ Cartesian geometry containing a three-layered sphere.

It should be observed that the track density is the main factor in determining the precision of the numerical volumes calculation, that being caused by the very weak angular dependance of equation C.8, whereas when CPs need to be calculated, a large enough number of angular directions is needed to represent neutrons directions.

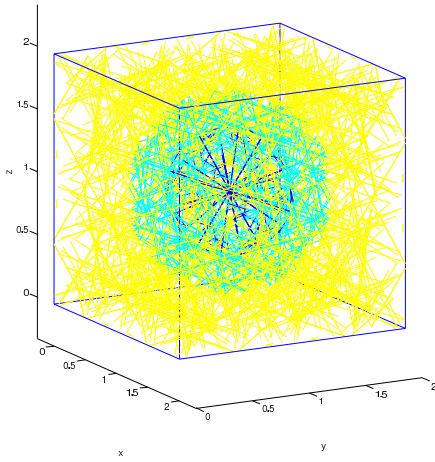


Figure C.4 Tracking visualization with TLM: module

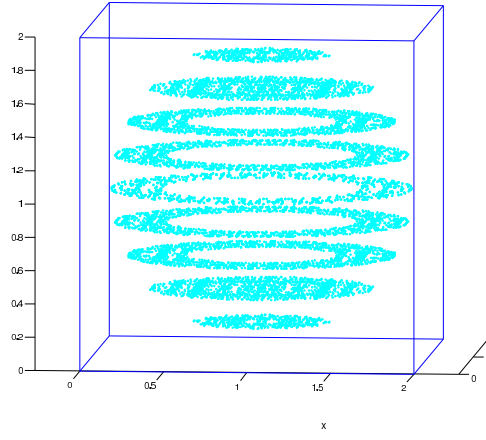


Figure C.5 Intersection of tracking lines with regularly spaced planes for one region

Table C.1 Comparison between analytical and numerical volumes using an equal weight quadrature

Density (cm^{-2})	Order	RMS Error (%)	Max. Error (%)	Average Error (%)
10.0	2	18.24550	90.06638	1.48001
	8	16.64095	90.06638	-2.55817
	16	16.85245	90.06638	-1.58190
20.0	2	2.48498	10.06485	0.47423
	8	3.53168	10.06484	-1.35797
	16	3.72187	10.06482	1.65092
50.0	2	1.92751	7.78483	-0.04890
	8	1.79627	7.78483	-0.52046
	16	1.78751	7.78484	-0.64180
100.0	2	0.75645	2.73456	0.29326
	8	0.74120	2.86485	-0.00515
	16	0.53025	2.73458	0.01692

C.6 INFINITE GRAIN LATTICE CALCULATION

We present in this section a simple infinite grain benchmark calculation, using the specifications given in Ref. (DeHart et Ulses, 2009). The first part of this proposed study consists in a benchmark modeling a single TRISO (Nabielek *et al.*, 1990) fuel particle in its graphite matrix. A cubic lattice with a spherical fuel particle inside is assumed, corresponding to the same packing fraction found in a pebble bed reactor. The simplicity of this geometry will allow us to make a direct code-to-code comparison. For the time being, no fuel depletion has been performed and only calculations carried with fresh fuel will be discussed. 3D spherical collision probability results will be confronted to pointwise, Monte Carlo results using the code SERPENT, and to those obtained using two double-heterogeneity (DH) treatments already available in the lattice code DRAGON, the Hébert model (Hébert, 2008), and the Sanchez-Pomraning model (Sanchez et Pomraning, 1991).

To summarize, three different methods will be challenged:

1. SERPENT 3D pointwise Monte Carlo calculation, which will be considered as reference solution.
2. DRAGON 3D lattice calculation with explicit collision probability method (CP).
 - A full 3D tracking of the cube with its embedded sphere is generated using the new capacity of the `NXT`: module described above.
3. DRAGON 3D lattice calculation with double-heterogeneity treatment (Hébert and Sanchez-Pomraning).
 - An equivalent 3D double heterogeneity problem is modeled, assuming a 9.04 % packing fraction of a single TRISO particle in a cubic lattice.

C.6.1 Validation Against Monte-Carlo Calculations

The DRAGON calculations have been performed with ENDBF7r0-based input libraries, in Draglib format (Hébert et Saygin, 1992) with SHEM 361-g structure (Hébert et Santamarina, 2008). For the SERPENT code, a ENDFB7r0-based library in the ACE format is used for the sake of consistency. $S(\alpha, \beta)$ data for the thermal scattering cross-sections of graphite are taken into account in both libraries. Also note that CP calculations require a self-shielding stage, performed by a subgroup method, described in Ref (Hébert, 2008). This advanced model has been retained for its ability to model spatially dependent self-shielding

Table C.2 Influence of the phase-space discretization for a full 3D spherical CP method

	k_{inf} SERPENT: 1.35035 ± 49 pcm	
$S_i\{N_{\text{angles}} \times \text{density (tracks/cm}^2)\}$	k_{inf} CP	diff. vs. Serpent (pcm)
$S_1\{44 \times 5000\}$	1.36195	1161
$S_2\{44 \times 10000\}$	1.34307	-728
$S_3\{44 \times 20000\}$	1.34683	-352
$S_4\{44 \times 25000\}$	1.35469	435
$S_5\{44 \times 30000\}$	1.34407	-628
$S_6\{44 \times 35000\}$	1.34646	-388
$S_7\{44 \times 40000\}$	1.35068	33

cross-sections.

A preliminary study was intended to find the minimum discretization parameters to be used in the 3D tracking operator. Results displayed in Tab. C.2 reveal that a very large track density, combined with a high order angular quadrature, is mandatory for reaching a good level of accuracy. The order for the angular Legendre-Chebyshev (P_n - T_n) quadrature has to be set to its maximum value, and more importantly, the density of integration lines should be at least 40,000 tracks per cm^2 . This computational burden was expected, due to the tiny volumes involved in this benchmark. As a matter of fact, spatial convergence of the collision probability method requires that all optical paths are lower than the minimum mean free path of a neutron in the geometry considered.

The best discretization, denoted as S_7 , is then used to generate two-group reaction rates, which are compared to those generated using Serpent. The limit between the fast (index 1) and the thermal (index 2) domains is set to 0.625 eV. Results, exposed in Tab. C.3, demonstrate the accuracy of the 3D CP method to model the neutronic behaviour of a single TRISO particle, when a sufficient track density is used. Further validation tests will eventually need to be undertaken, for instance on problems involving burnup calculations, or with a more complex geometry containing a stochastic distribution of a larger number of TRISO particles.

C.6.2 Comparison With the Double-Heterogeneity Model

A subsequent comparison has been made by comparing results obtained using the DH models already available in DRAGON, with the ones produced through the new CP capacities. Both double-heterogeneity models available in the DRAGON code (Sanchez-Pomraning model and

Table C.3 Group constant comparison

Parameters		CP	Serpent	diff. vs. Serpent (%)	DH	diff. vs. DH (%)
Integrated flux	1	2.27418E+02	2.27396E+02	9.7E-03	2.27419E+02	-4E-04
	2	5.54760E+01	5.55959E+01	-2.157E-01	5.54912E+01	-2.74E-02
$\nu \times$ Fission rate	1	2.24770E-01	2.24964E-01	-8.62E-02	2.24710E-01	2.67E-02
	2	1.12591E+00	1.12425E+00	1.477E-01	1.12595E+00	-3.6E-03
Absorption rate	1	4.18215E-01	4.19202E-01	-2.354E-01	4.18196E-01	4.5E-03
	2	5.81785E-01	5.81055E-01	1.256E-01	5.81804E-01	-3.3E-03

Hébert model) gave the same results on this benchmark, for k_{inf} and the group constants (see Tab. C.3), within 10^{-5} . This result was somehow expected, due to the fact that this benchmark is considering only one spherical element in an infinite homogeneous lattice (reflective boundary conditions), falling well within the domain of validity of these DH models. The k_{inf} found using both DH methods is 1.35066, for a difference with the S_7 discretization scheme of only 3 pcm. Also note that the greater consistency between CP and DH can be emphasized, since the same input library was used for both calculations.

C.7 CONCLUSIONS

The goal of this project, which was to apply full 3D collision probabilities to spherical geometries has been successfully achieved. A thorough verification has been made, and preliminary validation has been performed, leading to results consistent with those obtained through Monte-Carlo simulations and previously implemented double-heterogeneity treatment in lattice code DRAGON.

The 3D CP method with explicit spherical representation does not eliminate the need for innovative double-heterogeneity formalism, and can possibly be viewed as an additional validation tool for developers. Results can also be compared more consistently with an any explicit geometrical description of a double-heterogeneity problem in a Monte Carlo algorithm. Further validation still needs to be performed, mainly focusing on results obtained with random generation of spherical particles in a uniform matrix using a specified density, that has already been implemented, or with burnup progression calculations. The next development objectives should eventually focus on implementation of capacities for the treatment of full 3D hexagonal cells, and integration of combined spherical-cylindrical geometries, permitting full treatment of prismatic VHTR geometries.

ACKNOWLEDGEMENTS

The authors would like to thank the National Science and Engineering Research Council of Canada for providing the financial support for this research.