

Titre: Développement d'une méthode de Monte Carlo dépendante du temps et application au réacteur de type CANDU-6
Title:

Auteur: Mehdi Mahjoub
Author:

Date: 2016

Type: Mémoire ou thèse / Dissertation or Thesis

Référence: Mahjoub, M. (2016). Développement d'une méthode de Monte Carlo dépendante du temps et application au réacteur de type CANDU-6 [Thèse de doctorat, École Polytechnique de Montréal]. PolyPublie. <https://publications.polymtl.ca/2410/>
Citation:

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

URL de PolyPublie: <https://publications.polymtl.ca/2410/>
PolyPublie URL:

Directeurs de recherche: Jean Koclas
Advisors:

Programme: Génie nucléaire
Program:

UNIVERSITÉ DE MONTRÉAL

DÉVELOPPEMENT D'UNE MÉTHODE DE MONTE CARLO DÉPENDANTE DU
TEMPS ET APPLICATION AU RÉACTEUR DE TYPE CANDU-6

MEHDI MAHJOUB
DÉPARTEMENT DE GÉNIE PHYSIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE NUCLÉAIRE)
DÉCEMBRE 2016

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

DÉVELOPPEMENT D'UNE MÉTHODE DE MONTE CARLO DÉPENDANTE DU
TEMPS ET APPLICATION AU RÉACTEUR DE TYPE CANDU-6

présentée par : MAHJOUB Mehdi

en vue de l'obtention du diplôme de : Philosophiæ Doctor

a été dûment acceptée par le jury d'examen constitué de :

M. MARLEAU Guy, Ph. D., président

M. KOCLAS Jean, Ph. D., membre et directeur de recherche

M. HÉBERT Alain, Doctorat, membre

M. FORGET Benoit, Ph. D., membre externe

REMERCIEMENTS

*À mes parents qui ont toujours cru en moi,
À mes frères et ma sœur qui ont toujours été solidaires,
À ma femme qui m'a énormément soutenu dans les moments les plus difficiles,
À mon adorable fils Adam,
À mon directeur de recherche Jean Koclas,
À tous les professeurs de l'EPM qui ont contribué de près ou de loin dans ma formation,
À tous mes collègues du labo, des associations étudiantes, et d'enseignement,
À M. Benoit Forget et tous les membres du groupe CRPG au MIT pour leur accueil et
hospitalité,
À tous mes amis,
Merci...*

RÉSUMÉ

La résolution de l'équation de Boltzmann demeure une étape importante dans la prédiction du comportement d'un réacteur nucléaire. Malheureusement, la résolution de cette équation présente toujours un défi pour une géométrie complexe (réacteur) tout comme pour une géométrie simple (cellule). Ainsi, pour prédire le comportement d'un réacteur nucléaire, un schéma de calcul à deux étapes est nécessaire. La première étape consiste à obtenir les paramètres nucléaires d'une cellule du réacteur après une étape d'homogénéisation et condensation. La deuxième étape consiste en un calcul de diffusion pour tout le réacteur en utilisant les résultats de la première étape tout en simplifiant la géométrie du réacteur à un ensemble de cellules homogènes le tout entouré de réflecteur.

Lors des transitoires (accident), ces deux étapes sont insuffisantes pour pouvoir prédire le comportement du réacteur. Comme la résolution de l'équation de Boltzmann dans sa forme dépendante du temps présente toujours un défi de taille pour tous types de géométries, un autre schéma de calcul est nécessaire. Afin de contourner cette difficulté, l'hypothèse adiabatique est utilisée. Elle se concrétise en un schéma de calcul à quatre étapes. La première et deuxième étapes demeurent les mêmes pour des conditions nominales du réacteur. La troisième étape se résume à obtenir les nouvelles propriétés nucléaires de la cellule à la suite de la perturbation pour les utiliser, au niveau de la quatrième étape, dans un nouveau calcul de réacteur et obtenir l'effet de la perturbation sur le réacteur.

Ce projet vise à vérifier cette hypothèse. Ainsi, un nouveau schéma de calcul a été défini. La première étape de ce projet a été de créer un nouveau logiciel capable de résoudre l'équation de Boltzmann dépendante du temps par la méthode stochastique Monte Carlo dans le but d'obtenir des sections efficaces qui évoluent dans le temps. Ce code a été utilisé pour simuler un accident LOCA dans un réacteur nucléaire de type CANDU-6. Les sections efficaces dépendantes du temps ont été par la suite utilisées dans un calcul de diffusion espace-temps pour un réacteur CANDU-6 subissant un accident de type LOCA affectant la moitié du cœur afin d'observer son comportement durant toutes les phases de la perturbation. Dans la phase de développement, nous avons choisi de démarrer avec le code OpenMC, développé au MIT, comme plateforme initiale de développement. L'introduction et le traitement des neutrons retardés durant la simulation ont présenté un grand défi à surmonter.

Il est important de noter que le code développé utilisant la méthode Monte Carlo peut être utilisé à grande échelle pour la simulation de tous les types des réacteurs nucléaires si les supports informatiques sont disponibles.

Des techniques numériques, comme l'application des filtres et les corrections des sections

efficaces, ont été utilisées afin d'améliorer l'analyse des résultats.

En conclusion, les résultats obtenus montrent un comportement des sections efficaces différent de celui initialement prédit au début du projet. Ce dernier est complètement ignoré par l'hypothèse adiabatique. Lors des perturbations, nous avons observé une variation très importante des différentes sections efficaces qui ensuite convergent aux valeurs statiques calculées en utilisant l'approche adiabatique. À l'échelle du réacteur, la puissance et la réactivité du réacteur ont été affectées par ce comportement. Ce projet remet en question l'utilisation de l'hypothèse adiabatique durant les simulations de sûreté nucléaire, et met en évidence que la méthode Monte Carlo dépendante du temps est une alternative viable pour ce type d'application.

ABSTRACT

The solution of the time dependent Boltzmann equation remains quite a challenge. For transient problems, the two-steps approach is still used, with the nuclear reactor data obtained using an adiabatic assumption, whereby perturbed cross sections are calculated from static computations.

We developed a new method using the stochastic Monte Carlo approach to solve the time dependent transport equation. There are two reasons for this choice. First, at the cell level, we are able to obtain time dependent homogenized cross sections for use in full core diffusion calculations. Second, the Monte Carlo methods are scalable to perform full core calculations if and when appropriate computer resources become available.

The OpenMC code developed at MIT was used as the initial platform for our work, to which we added time dependent modules. Particular attention was given to the treatment of delayed neutrons in our implementation.

As a general observation, time dependent cross sections show an important variation when perturbations are introduced before finally converging to their static cross sections.

Using the time dependent cross sections, a full core time dependent diffusion simulation is performed for CANDU-6 reactor with a LOCA accident affecting half of the 380 reactor fuel channels. Some numerical techniques, like filters and corrections of cross sections, are implemented to improve the quality of results.

Our results show that the adiabatic assumption for cross sections in the transport step should be reviewed and that the time dependent Monte Carlo method is a viable alternative for the solution of time dependent neutron transport problems.

TABLE DES MATIÈRES

REMERCIEMENTS	iii
RÉSUMÉ	iv
ABSTRACT	vi
TABLE DES MATIÈRES	vii
LISTE DES TABLEAUX	x
LISTE DES FIGURES	xi
LISTE DES SIGLES ET ABRÉVIATIONS	xiv
LISTE DES ANNEXES	xv
CHAPITRE 1 INTRODUCTION	1
1.1 Problématique	3
1.2 Organisation du document	4
CHAPITRE 2 TRANSPORT ET DIFFUSION NEUTRONIQUE	5
2.1 Transport neutronique	5
2.1.1 Méthodes de résolutions déterministes	10
2.1.2 Méthode de résolution Monte-Carlo	11
2.1.3 Condensation et homogénéisation	15
2.1.4 Calcul de réactivité statique	16
2.1.5 Modèles numériques en transport	17
2.2 Diffusion neutronique	21
2.2.1 Résolution de l'équation de diffusion	22
CHAPITRE 3 APPROCHE MONTE CARLO POUR LE TRANSPORT DÉPENDANT DU TEMPS	24
3.1 Principe général et prototype développé	25
3.2 Traitement temporel des neutrons	25
3.3 Principales modifications par rapport au modèle Monte Carlo statique	29
3.4 Traitement des neutrons retardés	32

3.4.1	Traitement des neutrons retardés durant la simulation	33
3.4.2	Introduction des neutrons retardés au début de la simulation	37
3.5	Calcul des paramètres nucléaires dépendants du temps pour la cellule homogène	43
3.5.1	Calcul des sections efficaces condensées et homogénéisées	43
3.5.2	Calcul de la vitesse moyenne des neutrons	45
CHAPITRE 4	MODIFICATIONS APPORTÉES AU CODE OPENMC	47
4.1	Modifications apportées aux fichiers d'entrées XML	47
4.2	Modifications apportées aux codes sources	50
4.2.1	Modifications apportées au module d'initialisation	50
4.2.2	Modifications apportées au module de création de la géométrie	51
4.2.3	Modifications apportées au module de transport	52
4.3	Modifications apportées à la procédure de décompte des résultats	54
4.3.1	Modification dans le code source de la procédure de décompte	55
4.3.2	Modification dans les fichiers d'entrées XML	56
4.3.3	Création du fichier de tracking	58
CHAPITRE 5	MODÈLE DE DIFFUSION AVEC SECTIONS EFFICACES DÉPENDANTES DU TEMPS	61
5.1	Schéma de calcul espace-temps pour un réacteur CANDU-6 en utilisant le code DONJON	61
5.2	Modifications apportées au code DONJON	63
5.2.1	Modifications apportées au module INIRES:	63
5.2.2	Module OMCDRG:	64
5.2.3	Module UPTDXS:	68
5.3	Techniques numériques utilisées pour l'analyse des résultats	69
5.3.1	Élimination de la correction des sections efficaces de fission par le k_{eff}	69
5.3.2	Application des filtres numériques	69
5.3.3	Correction des sections efficaces	70
CHAPITRE 6	VALIDATION DU MODÈLE STATIQUE	72
6.1	Résultats statiques déterministes	72
6.2	Résultats statiques Monte Carlo	73
6.2.1	Simulations SERPENT pour une cellule CANDU-6	73
6.2.2	Simulations OpenMC pour une cellule CANDU-6	74
6.3	Comparaison des sections efficaces condensées et homogénéisées	74
6.4	Analyse des résultats statiques	74

CHAPITRE 7	RÉSULTATS DES SIMULATIONS DYNAMIQUES	78
7.1	Validation du modèle dynamique par des simulations en cinétique ponctuelle	78
7.2	Résultats en transport neutronique	80
7.2.1	Simulation dépendante du temps d'une cellule homogène avec plusieurs itérations	80
7.2.2	Résultats pour une cellule CANDU-6	84
7.2.3	Résultats pour une cellule CANDU-6 lors d'un accident de type LOCA	84
7.3	Résultats en diffusion neutronique espace-temps	94
7.3.1	Résultats de la simulation adiabatique	94
7.3.2	Résultats d'un calcul dynamique du réacteur en utilisant les sections efficaces statiques obtenues avec OpenMC	95
7.3.3	Résultats d'un calcul dynamique du réacteur en utilisant les sections efficaces dynamiques obtenues avec OpenMC-TD	97
7.3.4	Amélioration des résultats obtenus	102
CHAPITRE 8	CONCLUSION	108
8.1	Synthèse des travaux	108
8.2	Limitations de la solution proposée	108
8.3	Améliorations futures	108
RÉFÉRENCES	110
ANNEXES	114

LISTE DES TABLEAUX

Tableau 2.1	Valeurs nominales des paramètres de la cellule du réacteur CANDU-6	19
Tableau 3.1	Liste des variables relatives aux groupes de neutrons retardés	34
Tableau 4.1	Exemple du contenu du fichier de tracking	60
Tableau 5.1	Liste des modules créés et modifiés	63
Tableau 5.2	Technique des <i>deltas sigmas</i>	71
Tableau 6.1	k_{eff} et calcul de réactivité pour une cellule CANDU-6 en utilisant le module EXCELT: pour le tracking.	72
Tableau 6.2	k_{eff} et calcul de réactivité pour une cellule CANDU-6 en utilisant le module NXT: pour le tracking.	73
Tableau 6.3	k_{eff} et calcul de réactivité pour une cellule CANDU-6 en utilisant le code Serpent	73
Tableau 6.4	k_{eff} et calcul de réactivité pour une cellule CANDU-6 en utilisant le code OpenMC	74
Tableau 6.5	Différences entre les sections efficaces homogénéisées et condensées obtenues par Serpent et OpenMC pour une cellule CANDU-6 refroidie en utilisant la librairie ENDF-B7.0	75
Tableau 6.6	Différences entre les sections efficaces homogénéisées et condensées obtenues par Serpent et OpenMC pour une cellule CANDU-6 refroidie en utilisant la librairie Jeff3.1.1	75
Tableau 6.7	Différences entre les sections efficaces homogénéisées et condensées obtenues par Serpent et OpenMC pour une cellule CANDU-6 vidangée en utilisant la librairie ENDF-B7.0	76
Tableau 6.8	Différences entre les sections efficaces homogénéisées et condensées obtenues par Serpent et OpenMC pour une cellule CANDU-6 vidangée en utilisant la librairie Jeff3.1.1	76
Tableau 7.1	Résultats d'une simulation adiabatique pour un réacteur CANDU-6 initialement sous-critique subissant un accident LOCA	95
Table B.1	Main records and sub-directories in /CPOTD/	124
Table B.2	Example of homogeneous /CPOTD/ mixture sub-directories	124
Table B.3	Contents of a mixture sub-directory in /CPOTD/	125
Table B.4	Example of homogeneous mixture directories	125

LISTE DES FIGURES

Figure 2.1	Définition des groupes d'énergie	7
Figure 2.2	Parcours aléatoire d'un neutron par la méthode de Monte Carlo . . .	15
Figure 2.3	Cellule du réacteur CANDU-6 refroidie et vidée de son caloporteur .	18
Figure 2.4	Cellules affectées par la vidange du caloporteur dans un accident LOCA du réacteur CANDU-6	23
Figure 3.1	Évolution de la population de neutrons dans un milieu multiplicatif en subissant deux perturbations à T_1 et T_2	26
Figure 3.2	Vie d'un neutron	28
Figure 3.3	Libre parcours du neutron par intervalle de temps	29
Figure 3.4	Les différentes boucles d'un algorithme Monte Carlo	30
Figure 3.5	Processus simplifié pour la génération des neutrons retardés	35
Figure 3.6	Spectres des neutrons retardés	38
Figure 3.7	Gestion des neutrons retardés au début de la simulation	39
Figure 3.8	Évolution de la concentration des précurseurs à la suite d'un arrêt du réacteur à $t = 0s$	41
Figure 3.9	Évolution de la concentration des neutrons retardés produits à la suite d'un arrêt du réacteur à $t = 0s$	41
Figure 3.10	Distribution spatiale des neutrons retardés introduits par crayon d'Ura- nium	42
Figure 3.11	Distribution spatiale des neutrons retardés introduits par crayon d'Ura- nium représentée sur une grappe CANDU-6	43
Figure 3.12	Décomposition du temps de simulation en intervalles de récoltes . . .	44
Figure 4.1	Boucles de transport neutronique dans OpenMC-TD	53
Figure 5.1	Modélisation du réacteur CANDU-6 avec le logiciel DONJON-NDF .	62
Figure 5.2	Identification des zones en utilisant ACC-ZONES du module INIRES: .	64
Figure 7.1	Puissance thermique d'une configuration de cellules homogènes infinies	79
Figure 7.2	Puissance thermique d'une configuration de cellules homogènes infinies entre 0.03s et 0.05s.	79
Figure 7.3	Flux neutronique (groupe rapide)	81
Figure 7.4	Flux neutronique (groupe thermique)	81
Figure 7.5	Comparaison du facteur $\frac{1}{v}$ (groupe rapide)	82
Figure 7.6	Comparaison du facteur $\frac{1}{v}$ (groupe thermique)	82
Figure 7.7	Comparaison de la section efficace totale (groupe rapide)	83

Figure 7.8	Comparaison de la section efficace totale (groupe thermique)	83
Figure 7.9	Densité neutronique totale (relative) pour une cellule CANDU-6 initialement sous-critique	84
Figure 7.10	Densités neutroniques (groupes rapide et thermique)	85
Figure 7.11	Flux neutroniques totaux de la cellule pour les groupes rapide et thermique	85
Figure 7.12	Taux de réaction total pour les groupes rapide et thermique	86
Figure 7.13	Taux de réaction d'absorption pour les groupes rapide et thermique	86
Figure 7.14	Taux de réaction de fission pour les groupes rapide et thermique	87
Figure 7.15	Taux de réaction de transport pour les groupes rapide et thermique	87
Figure 7.16	Sections efficaces totales condensées à deux groupes et homogénéisées sur toute la cellule	88
Figure 7.17	Sections efficaces de fission condensées à deux groupes et homogénéisées sur toute la cellule	88
Figure 7.18	Sections efficaces d'absorption condensées à deux groupes et homogénéisées sur toute la cellule	89
Figure 7.19	Sections efficaces de transport condensées à deux groupes et homogénéisées sur toute la cellule	89
Figure 7.20	Le rapport entre les sections efficaces totales statiques et dynamiques	90
Figure 7.21	Le rapport entre les sections efficaces de fission statiques et dynamiques	90
Figure 7.22	Vitesses moyennes des neutrons du groupe rapide.	91
Figure 7.23	Vitesses moyennes des neutrons du groupe thermique.	91
Figure 7.24	$1/v$ des neutrons du groupe rapide.	92
Figure 7.25	$1/v$ des neutrons du groupe thermique	92
Figure 7.26	Variation de la puissance thermique du réacteur CANDU-6 sans aucune perturbation	95
Figure 7.27	Variation de la réactivité du réacteur CANDU-6 sous critique	96
Figure 7.28	Variation de la puissance thermique du réacteur CANDU-6 avec l'introduction d'un LOCA en utilisant des sections efficaces statiques	96
Figure 7.29	Variation de la réactivité du réacteur CANDU-6 avec l'introduction d'un LOCA en utilisant des sections efficaces statiques	97
Figure 7.30	Variation de la puissance thermique avec l'introduction d'un LOCA en utilisant des sections efficaces dynamiques et en présence de la correction par le k_{eff}	98
Figure 7.31	Variation de la réactivité avec l'introduction d'un LOCA en utilisant des sections efficaces dynamiques k_{eff} en présence de la correction par le k_{eff}	98

Figure 7.32	Variation de la puissance thermique avec l'introduction d'un LOCA en utilisant des sections efficaces dynamiques	99
Figure 7.33	Variation de la réactivité avec l'introduction d'un LOCA en utilisant des sections efficaces dynamiques	99
Figure 7.34	Différence entre la puissance thermique en utilisant des sections efficaces dynamiques et la puissance thermique en en utilisant les sections efficaces statiques	100
Figure 7.35	Regression exponentielle sur la puissance thermique après l'accident en utilisant des sections efficaces d'OpenMC-TD	101
Figure 7.36	Regression exponentielle sur la puissance thermique après l'accident en utilisant des sections efficaces statiques	101
Figure 7.37	Variation de la puissance thermique avec l'introduction d'un LOCA en utilisant un filtre numérique pour les sections efficaces	102
Figure 7.38	Variation de la réactivité avec l'introduction d'un LOCA en utilisant un filtre numérique pour les sections efficaces	103
Figure 7.39	Variation de la puissance thermique en lui appliquant un filtre numérique avec l'introduction d'un LOCA	104
Figure 7.40	Variation de la puissance thermique avec l'introduction d'un LOCA en utilisant la technique des delta-sigmas pour la correction des sections efficaces	105
Figure 7.41	Variation de la réactivité avec l'introduction d'un LOCA en utilisant la technique des delta-sigmas pour la correction des sections efficaces	105
Figure 7.42	Variation de la puissance thermique avec l'introduction d'un LOCA en utilisant un filtre numérique et la technique des delta-sigmas pour la correction des sections efficaces	106
Figure 7.43	Variation de la réactivité avec l'introduction d'un LOCA en utilisant un filtre numérique et la technique des delta-sigmas pour la correction des sections efficaces	106
Figure 7.44	Regression exponentielle sur la puissance thermique après l'accident en utilisant des sections efficaces d'OpenMC-TD et en présence des filtres numériques	107
Figure A.1	Aperçu des communications entre les processus	116
Figure A.2	Diagramme d'état simplifié présentant les communications entre les processus <i>MANAGER</i> et <i>Evolution</i>	120
Figure A.3	Diagramme d'état simplifié présentant les communications entre les différents types de processus	121

LISTE DES SIGLES ET ABRÉVIATIONS

IGN	Institut de génie nucléaire
ENDF	Evaluated Nuclear Data File
CP	Méthode des probabilités de collision
Pij	Méthode des probabilités de collision
MoC	Méthode des caractéristiques
CANDU	CANada Deutorium Uranium : Réacteur à eau lourde Canadien
CANDU-6	Réacteur à eau lourde Canadien à 665 MWe
PWR,REP,REL	Réacteur à eau sous pression
CDF	Cumulative Distribution Function
k_{eff}	Facteur de multiplication effectif
k_{∞}	Facteur de multiplication infini
$\Sigma_{t,s,a}$	Section efficace macroscopique totale, de diffusion ou d'absorption
$\sigma_{t,s,a}$	Section efficace microscopique totale, de diffusion ou d'absorption
ν	Nombre total de neutrons libérés suite à une absorption d'un neutron
$\chi_{p,d}$	Spectre d'énergie des neutrons instantanés, retardés
λ	Constante de décroissance
C_k	Concentration des précurseurs du groupe k
LOCA	Loss Of Coolant Accident
FIFO	First In First Out
XS	Section efficace
RUC	Système de refroidissement d'urgence du cœur

LISTE DES ANNEXES

Annexe A	PROGRAMMATION PARALLÈLE DU CODE OPENMC-TD . . .	114
Annexe B	STRUCTURE DE DONNÉES /CPOTD/	124
Annexe C	GÉOMÉTRIE DE CELLULE CANDU-6 AVEC OPENMC-TD : GEO- METRY.XML	126
Annexe D	PARAMÈTRES DE LA SIMULATION DYNAMIQUE DE CELLULE CANDU-6 AVEC OPENMC-TD : SETTINGS.XML	134
Annexe E	FICHIER XML DES PARAMÈTRES POUR LES TALLIES POUR OPENMC-TD : TALLIES.XML	135
Annexe F	SCHÉMA DE CALCUL DYNAMIQUE POUR LE RÉACTEUR CANDU- 6 EN UTILISANT DONJON ET NDF	136

CHAPITRE 1 INTRODUCTION

L'histoire de l'énergie nucléaire remonte au 20^{ème} siècle. À cette époque, on a découvert que l'énergie encapsulée dans le noyau d'un atome est des millions de fois plus importante que celle des électrons orbitaux. Au départ, cette découverte a été exploitée dans le domaine militaire et, durant la deuxième guerre mondiale, elle a conduit à l'apparition des bombes nucléaires. Peu après, l'attention s'est tournée vers l'utilisation de ce type d'énergie dans le but de produire de l'électricité. Les recherches ont montré qu'on pouvait extraire cette énergie par deux types de réaction : la fission ou la fusion.

Le premier réacteur nucléaire de puissance, se basant sur la fission du noyau, a vu le jour aux États-Unis durant les années cinquante du 20^{ème} siècle (Glasstone et Sesonske, 1991). Depuis, l'industrie nucléaire a connu une véritable révolution et différents modèles de réacteurs ont été construits un peu partout dans le monde.

Afin de prédire les comportements des centrales nucléaires, plusieurs simulations s'avèrent nécessaires. Dans ce cadre, de nombreux logiciels de calculs dans divers domaines, comme la neutronique, la thermohydraulique, la dispersion atmosphérique etc., ont été développés. Cette thèse se concentre sur l'aspect neutronique des réacteurs nucléaires de type CANDU-6.

Une centrale nucléaire du type CANDU-6, comme plusieurs centrales d'ailleurs, se compose principalement de :

- un cœur de réacteur qui contient le combustible nucléaire,
- des pompes de caloporteur,
- un pressuriseur pour garder le caloporteur sous pression (≈ 10 MPa),
- des générateurs de vapeur pour produire de la vapeur et
- une turbine pour produire de l'électricité.

Le cœur du réacteur CANDU-6 comprend 380 canaux (tubes de force) contenant chacun 12 grappes de combustible. Chaque grappe est constituée de 37 crayons d'uranium entourés d'une gaine et retenue par une structure en Zircaloy.

L'un des principaux défis de la neutronique consiste à déterminer la distribution de neutrons dans le réacteur. Plusieurs logiciels visent à résoudre les équations de la neutronique et ainsi prédire le comportement des neutrons et la puissance produite dans les réacteurs. L'équation mathématique qui décrit le comportement des neutrons dans le réacteur est l'équation de transport. Cette équation est très complexe à résoudre et les solutions analytiques ne sont connues que pour des problèmes assez simples. Par contre, une simplification par des approximations et méthodes numériques permet d'approcher les solutions, par exemple la méthode des différences finies, la méthode des éléments finis et la méthode Monte Carlo (Bell

et Glastone, 1970),(Hébert, 2009)). Malheureusement, la technologie informatique accessible n'est toujours pas adéquate pour permettre d'effectuer un calcul de cœur complet avec une discrétisation fine des assemblages en prenant en considération tous les détails du cœur sans aucune autre approximation.

Afin de contourner ce problème, un schéma de calcul par étapes s'avère nécessaire ; d'abord par un calcul de cellule, en domaine spatial restreint, afin de résoudre l'équation de transport. Cette étape permet d'obtenir les données nucléaires de la cellule homogénéisée telles que les sections efficaces macroscopiques. Il convient ensuite de les utiliser dans un calcul de réacteur pour résoudre l'équation de diffusion qui est une approximation de l'équation de transport. Durant les transitoires, les mêmes étapes sont appliquées en se basant sur l'hypothèse stipulant que les sections efficaces utilisées afin de réaliser les calculs de réacteur (diffusion) en transitoire sont toujours calculées pour des conditions statiques d'un calcul de cellule.

On distingue deux types de codes neutroniques capables de résoudre l'équation de transport : les codes déterministes et les codes stochastiques. Les codes déterministes utilisent des méthodes approximatives de résolutions dites analytiques : MoC, Pij, etc... (Hébert, 2009). On peut citer comme exemple les codes de réseau DRAGON (Marleau *et al.*, 2014) et APPOLO (Hoffman *et al.*, 1973), etc . . . Le second type de codes se base sur une approche statistique. Le code SERPENT (Leppänen, 2009), le code MCNP (Briesmeister, 2000), le code MORSE (Emmett, 1975), le code TRIPOLI (Petit *et al.*, 2008), le code VIM (Blomquist, 2009), le code TART (Cullen, 2005), en sont des exemples.

La deuxième étape de simulation pour un réacteur nucléaire se concrétise par la résolution de l'équation de diffusion pour un réacteur au complet. Durant cette étape, il faut utiliser des logiciels capables de résoudre cette équation. DONJON (Varin *et al.*, 2005), PARCS (Downar *et al.*, 2004), SKETCH-N (Zimin *et al.*, 2001) et RFSP (Rouben, 2002) sont des exemples.

Les algorithmes stochastiques ont servi dans divers champs de recherche en nucléaire, par exemple : l'application des algorithmes *Tabou* et *Génétiq*ue dans le calcul des perturbations locales et l'amélioration de la composition du réflecteur (Mahjoub, 2011), dans la radio protection et le calcul des doses absorbées (Trindade et Campos, 2011), dans la gestion de combustible (Chambon, 2006), dans l'analyse des calandres (A.Haghighat et Petrovic, 1996), pour la résolution de l'équation de transport neutronique (Sharp, 1999), etc.

Les méthodes stochastiques appliquées au domaine nucléaire portent le nom de méthodes Monte Carlo (Metropolis et Ulam, 1949). Par le passé, ces dernières ont été peu utilisées, en comparaison avec les logiciels déterministes résolvant l'équation de transport plus rapidement. Ces méthodes stochastiques voyaient leur importance principale dans la vérification des différents calculs neutroniques. Grâce à l'évolution de l'informatique, les simulations Monte

Carlo sont devenues de plus en plus abordables. L'*hypothèse* définie précédemment pourrait alors être examinée en détail. Nous pouvons, aujourd'hui, vérifier cette dernière en effectuant des calculs évoluant dans le temps et en comparant leurs résultats à ceux effectués avec la méthode traditionnelle. Il convient ainsi de faire en premier lieu un calcul de cellule évoluant dans le temps d'une manière continue. Ce calcul peut être effectué soit par les méthodes déterministes ; la méthode des caractéristiques (MoC) (Taylor, 2007) et la méthode des probabilités de collision (Pij) (Keller et Lee, 1998) qui semblent être très complexes et coûteuses en temps et en effort, soit par des calculs Monte Carlo comme le suggère Shayesteh avec le code TDMC (Shayesteh et Shahriari, 2009) ou Russell avec le code GEANT4 (Russell *et al.*, 2012).

Le code SERPENT vient d'être mis à jour. En 2013, un module dépendant du temps ne prenant pas en considération les neutrons retardés a été introduit (Leppänen, 2013) puis ensuite il a été mis à jours afin de les inclure (Valtavirta *et al.*, 2016).

Malheureusement, les algorithmes Monte Carlo précédents ne peuvent pas aider dans le cadre de ce projet et ceci est dû soit à leurs manques de documentation, soit au manque d'outils nécessaires pour nos simulations, ou encore la non disponibilité des fichiers sources du logiciel en plus de leurs arrivées tardives dans le cadre de ce projet.

La résolution de l'équation de transport dépendante du temps nécessite un traitement particulier des neutrons retardés. Une approche se basant sur des distributions Maxwelliennes des spectres d'énergies des neutrons retardés a été définie dans le code TART (Cullen, 2004). Cette approche a déjà été implémentée et un processus se basant sur la formule C64 défini par le laboratoire national de *Los Alamos* dans le document 'Monte Carlo Sampler' (Everett et Cashwell, 1983) .

Une fois obtenues, les données dépendantes du temps de la cellule pourront être utilisées dans un calcul de réacteur en transitoire. Plusieurs types de transitoires peuvent être simulées pour un réacteur nucléaire, par exemple ; un accident super critique comme celui de Tchernobyl, un accident lié à la chaleur résiduelle à la suite de l'arrêt du réacteur comme celui de Fukushima, une perte de caloporteur (Loss Of Coolant Accident (LOCA)) (Muzumdar et Meneley, 2009) etc. Ces simulations peuvent être réalisées avec le code DONJON.

1.1 Problématique

Dans le cadre de cette thèse, nous voulions vérifier la validité de l'hypothèse adiabatique des sections efficaces appliquées dans les calculs de sûreté. Cette hypothèse permet l'utilisation des données nucléaires d'une cellule de réacteur obtenues grâce à des calculs statiques, en se basant sur la résolution de l'équation de transport stationnaire, dans des calculs de

transitoires très rapides (accidents).

Cet objectif nous mène à étudier les problèmes suivants :

- Comment obtenir les données nucléaires évolutives dans le temps d’une manière continue pour une cellule de réacteur ?
- À quel moment et sous quelles conditions devrions-nous effectuer des calculs dépendants du temps et quelles sont les conditions nécessaires et suffisantes pour appliquer la méthode traditionnelle ?
- Quel type de transitoire doit être considéré afin de vérifier cette hypothèse ?
- Quelle approche numérique devons nous prendre afin de développer un modèle de vérification ?

La démarche entreprise pour vérifier cette thèse commence par la production de sections efficaces dépendantes du temps grâce à des modifications apportées au code OpenMC afin de le rendre dynamique et capable de suivre l’évolution des neutrons dans le temps. Par la suite, de nouveaux modules ont été développés et d’autres ont été modifiés dans le logiciel DONJON afin de pouvoir effectuer des simulations espace-temps du réacteur CANDU-6 en utilisant les résultats obtenus par le code OpenMC Dynamique (OpenMC-TD). Finalement, les résultats obtenus ont été comparés à ceux obtenus par un schéma de calcul faisant appel à l’hypothèse adiabatique.

1.2 Organisation du document

Ce document comporte cinq chapitres principaux. La théorie générale de la neutronique est présentée au niveau du chapitre 2, l’équation de transport ainsi que l’équation de diffusion y sont définies et discutées. La notion du transport dépendant du temps, l’algorithme développé et les techniques utilisées seront introduites au chapitre 3. Le chapitre 4 inclut les modifications apportées au code OpenMC. La programmation parallèle développée est discutée dans le chapitre 5. Le chapitre 6 présente le modèle de diffusion et la transitoire qui sera simulée en utilisant DONJON. Après avoir validé nos modèles au chapitre 6, les résultats obtenus pour la simulation d’un accident de type LOCA dans un réacteur CANDU-6 seront présentés au chapitre 7. Une conclusion présentera certains commentaires ainsi que les travaux qui pourraient être menés par la suite.

CHAPITRE 2 TRANSPORT ET DIFFUSION NEUTRONIQUE

2.1 Transport neutronique

Avant d'aborder l'équation de transport, une reformulation rapide des hypothèses déployées dans le transport neutronique s'impose. Ces hypothèses ont été considérées dans le développement de l'équation de transport et dans sa résolution. Les principales hypothèses sont (Hébert, 2009) :

- les neutrons sont des objets ponctuels,
- les matériaux sont isotropes dans l'espace,
- les neutrons voyagent en ligne droite entre deux collisions,
- les interactions neutron-neutron peuvent être ignorées,
- les collisions sont considérées instantanées, et
- dans les collisions neutron-noyau, l'énergie totale, la quantité de mouvement et le moment cinétique sont conservés.

L'objectif des calculs neutroniques est d'obtenir la distribution des neutrons dans le réacteur de même que les taux de réactions. Chaque neutron est caractérisé par les variables suivantes :

1. Sa position \vec{r} : $\vec{r} = x \vec{i} + y \vec{j} + z \vec{k}$
2. Sa vitesse \vec{v} , de norme $v = \|\vec{v}\|$ et de direction $\vec{\Omega}$
3. Le temps t

La population de neutrons est représentée par une distribution. La densité neutronique $n(\vec{r}, v, \vec{\Omega}, t)$ est définie de telle sorte que $n(\vec{r}, v, \vec{\Omega}, t) d^3r dv d^2\Omega$ représente le nombre de neutrons au temps t , dans le volume d^3r au tour du point \vec{r} , avec une vitesse appartenant à l'intervalle dv au tour de v et se propageant dans l'élément d'angle solide $d^2\Omega$ autour de $\vec{\Omega}$.

En neutronique, le flux, plutôt que la densité, est la variable principale. Le flux angulaire est défini comme le produit de la densité par la vitesse :

$$\phi(\vec{r}, v, \vec{\Omega}, t) = vn(\vec{r}, v, \vec{\Omega}, t). \quad (2.1)$$

On utilise aussi l'énergie plutôt que la vitesse comme variable. Le flux est obtenu en résolvant l'équation de transport neutronique qui est une équation de Boltzmann (Hébert, 2009)

$$\begin{aligned}
\frac{1}{v} \frac{\partial}{\partial t} \phi(\vec{r}, E, \vec{\Omega}, t) &= - \Sigma_t(\vec{r}, E, t) \phi(\vec{r}, E, \vec{\Omega}, t) \\
&\quad - \vec{\Omega} \vec{\nabla} \phi(\vec{r}, E, \vec{\Omega}, t) \\
&\quad + \int_0^\infty dE' \int_0^{4\pi} \Sigma_s(\vec{r}, E' \rightarrow E, \vec{\Omega}' \rightarrow \vec{\Omega}, t) \phi(\vec{r}, E', \vec{\Omega}', t) d^2\Omega' \\
&\quad + (1 - \beta) \chi_p(E) \int_0^\infty dE' \int_0^{4\pi} \nu(E') \Sigma_f(\vec{r}, E', t) \phi(\vec{r}, E', \vec{\Omega}', t) d^2\Omega' \\
&\quad + \sum_{k=1}^K \lambda_k C_k(\vec{r}, t) \chi_{dk}(E)
\end{aligned} \tag{2.2}$$

Les termes sont définis comme suit :

1. Taux de variations du flux neutronique dans le temps divisé par la vitesse :

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

2. Taux de perte des neutrons par collision :

$$\Sigma_t(\vec{r}, E, t) \phi(\vec{r}, E, \vec{\Omega}, t)$$

3. Taux de perte des neutrons par fuite :

$$\vec{\Omega} \vec{\nabla} \phi(\vec{r}, E, \vec{\Omega}, t)$$

4. Taux de neutrons acquis par collision :

$$\int_0^\infty dE' \int_0^{4\pi} \Sigma_s(\vec{r}, E' \rightarrow E, \vec{\Omega}' \rightarrow \vec{\Omega}, t) \phi(\vec{r}, E', \vec{\Omega}', t) d^2\Omega'$$

5. Taux de neutrons instantanés acquis par fission :

$$(1 - \beta) \chi_p(E) \int_0^\infty dE' \int_0^{4\pi} \nu(E') \Sigma_f(\vec{r}, E', t) \phi(\vec{r}, E', \vec{\Omega}', t) d^2\Omega'$$

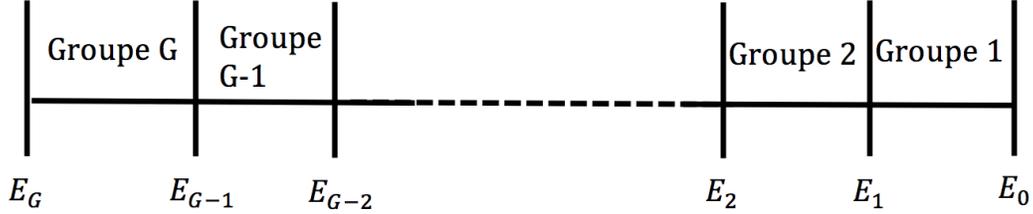
6. Taux de production des neutrons retardés :

$$\sum_{k=1}^K \lambda_k C_k(\vec{r}, t) \chi_{dk}(E)$$

L'équation de Boltzmann présentée ci-dessus représente le bilan des neutrons dans l'hyper-volume $d^3r d^2\Omega dE$ au temps t . Un terme Q_{ex} pourrait être rajouté aux termes à droite afin de représenter une source externe de neutrons. Dans notre cas, on ignore les sources externes. Dans cette équation, nous remarquons la présence du terme $C_k(\vec{r}, t)$ qui représente la concentration des précurseurs de neutrons retardés du groupe k . Pour fermer le système, il faut établir les équations d'évolutions des précurseurs pour évaluer ces concentrations. Ces quantités sont définies comme suit :

$$\begin{aligned}
\frac{\partial}{\partial t} C_k(\vec{r}, t) &= - \lambda_k C_k(\vec{r}, t) \\
&\quad + \beta_k \int_0^\infty dE' \int_0^{4\pi} d^2\Omega \nu(E') \Sigma_f(\vec{r}, E', t) \phi(\vec{r}, E', \vec{\Omega}, t) \quad k = 1, 2, \dots, K
\end{aligned} \tag{2.3}$$

En pratique, la résolution de l'équation de transport selon la méthode déterministe nécessite une discrétisation en énergie. Pour ce faire, il faut remplacer les fonctions continues en énergie par d'autres fonctions continues seulement par intervalle. Les intervalles d'énergie portent le nom de *groupes d'énergies*. Ces derniers sont définis de tels sorte que le groupe d'indice le plus petit a la plus grande valeur d'énergie, comme le montre la figure 2.1.



E_0 : Énergie maximale

E_G : Énergie minimale

Figure 2.1 Définition des groupes d'énergie

Les constantes des groupes d'énergie sont définies comme suit :

$$\phi_g(\vec{r}, \vec{\Omega}, t) = \int_{E_g}^{E_{g-1}} \phi(\vec{r}, E, \vec{\Omega}, t) dE \quad (2.4)$$

$$\begin{aligned} \frac{1}{v_g} \frac{\partial}{\partial t} \phi_g(\vec{r}, \vec{\Omega}, t) &= \int_{E_g}^{E_{g-1}} \frac{1}{v} \frac{\partial}{\partial t} \phi(\vec{r}, E, \vec{\Omega}, t) dE \\ &= \frac{\partial}{\partial t} \int_{E_g}^{E_{g-1}} \frac{1}{v} \phi(\vec{r}, E, \vec{\Omega}, t) dE \end{aligned} \quad (2.5)$$

$$\begin{aligned} \Sigma_{s,g' \rightarrow g}(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, t) \phi_{g'}(\vec{r}, \vec{\Omega}', t) &= \\ \int_{E_g}^{E_{g-1}} \int_{E_{g'}}^{E_{g'-1}} \Sigma_{s,g}(\vec{r}, E' \rightarrow E, \vec{\Omega}' \rightarrow \vec{\Omega}, t) \phi(\vec{r}, E', \vec{\Omega}', t) dE' dE \end{aligned} \quad (2.6)$$

$$\Sigma_{t,g}(\vec{r}, t) \phi_g(\vec{r}, \vec{\Omega}, t) = \int_{E_g}^{E_{g-1}} \Sigma_t(\vec{r}, E, t) \phi(\vec{r}, E, \vec{\Omega}, t) dE \quad (2.7)$$

$$\nu \Sigma_{f,g}(\vec{r}, t) \phi_g(\vec{r}, \vec{\Omega}, t) = \int_{E_g}^{E_{g-1}} \nu(E) \Sigma_f(\vec{r}, E, t) \phi(\vec{r}, E, \vec{\Omega}, t) dE \quad (2.8)$$

$$\chi_{p,g} = \int_{E_g}^{E_{g-1}} \chi_p(E) dE \quad (2.9)$$

$$\chi_{dk,g} = \int_{E_g}^{E_{g-1}} \chi_{dk}(E) dE \quad (2.10)$$

$$\nu_g = \int_{E_g}^{E_{g-1}} \nu(E) dE \quad (2.11)$$

Avec ces définitions, l'on obtient l'équation de transport des neutrons multi groupes suivante :

$$\begin{aligned} \frac{1}{v_g} \frac{\partial}{\partial t} \phi_g(\vec{r}, \vec{\Omega}, t) = & - \Sigma_{t,g}(\vec{r}, t) \phi_g(\vec{r}, \vec{\Omega}, t) \\ & - \vec{\Omega} \vec{\nabla} \phi_g(\vec{r}, \vec{\Omega}, t) \\ & + \sum_{g'=1}^G \int_0^{4\pi} \Sigma_{s,g \leftarrow g'}(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, t) \phi_{g'}(\vec{r}, \vec{\Omega}', t) d^2\Omega \\ & + (1 - \beta) \chi_{p,g} \sum_{g'=1}^G \int_0^{4\pi} \nu_{g'} \Sigma_{f,g'}(\vec{r}, t) \phi_{g'}(\vec{r}, \vec{\Omega}', t) d^2\Omega' \\ & + \sum_{k=1}^K \lambda_k C_k(\vec{r}, t) \chi_{dk,g} \end{aligned} \quad (2.12)$$

et les équations d'évolutions des précurseurs multi groupes suivantes :

$$\frac{\partial}{\partial t} C_k(\vec{r}, t) = -\lambda_k C_k(\vec{r}, t) + \beta_k \sum_{g'=1}^G \int_0^{4\pi} \nu_{g'} \Sigma_{f,g'}(\vec{r}, t) \phi_{g'}(\vec{r}, t) \quad k = 1, 2, \dots, K \quad (2.13)$$

L'équation de transport peut être intégrée sur tous les angles solides, comme suit :

$$\phi_g(\vec{r}, t) = \int_0^{4\pi} \phi_g(\vec{r}, \vec{\Omega}, t) d^2\Omega \quad (2.14)$$

$$\begin{aligned} \frac{1}{v_g} \frac{\partial}{\partial t} \phi_g(\vec{r}, t) &= \int_{4\pi} \frac{1}{v_g} \frac{\partial}{\partial t} \phi_g(\vec{r}, \vec{\Omega}, t) d^2\Omega \\ &= \frac{\partial}{\partial t} \int_{4\pi} \frac{1}{v_g} \phi(\vec{r}, \vec{\Omega}, t) d^2\Omega \end{aligned} \quad (2.15)$$

$$\Sigma_{t,g}(\vec{r}, t) \phi_g(\vec{r}, t) = \int_{4\pi} \Sigma_{t,g}(\vec{r}, t) \phi_g(\vec{r}, \vec{\Omega}, t) d^2\Omega \quad (2.16)$$

$$\Sigma_{f,g}(\vec{r}, t) \phi_g(\vec{r}, t) = \int_{4\pi} \Sigma_{f,g}(\vec{r}, t) \phi_g(\vec{r}, \vec{\Omega}, t) d^2\Omega \quad (2.17)$$

$$\vec{\nabla} \cdot \vec{J}_g(\vec{r}, t) = \int_0^{4\pi} \vec{\Omega} \cdot \vec{\nabla} \phi_g(\vec{r}, \vec{\Omega}, t) d^2\Omega = \int_0^{4\pi} \vec{\nabla} \cdot \vec{\Omega} \phi_g(\vec{r}, \vec{\Omega}, t) d^2\Omega \quad (2.18)$$

Pour la section efficace de diffusion isotrope, en intégrant sur tous les angles d'incidence $\vec{\Omega}'$ et d'émission du neutron $\vec{\Omega}$, on obtient :

$$\Sigma_{s,g \leftarrow g'}(\vec{r}, t) \phi_g(\vec{r}, t) = \int_0^{4\pi} \int_0^{4\pi} \Sigma_{s,g \leftarrow g'}(\vec{r}, \vec{\Omega}' \rightarrow \vec{\Omega}, t) \phi_g(\vec{r}, \vec{\Omega}', t) d^2\Omega' d^2\Omega \quad (2.19)$$

Ainsi, on obtient l'équation de transport suivante :

$$\begin{aligned} \frac{1}{v_g} \frac{\partial}{\partial t} \phi_g(\vec{r}, t) &= -\Sigma_{t,g}(\vec{r}, t) \phi_g(\vec{r}, t) \\ &\quad - \vec{\nabla} \cdot \vec{J}_g(\vec{r}, t) \\ &\quad + \sum_{g'=1}^G \Sigma_{s,g \leftarrow g'}(\vec{r}, t) \phi_{g'}(\vec{r}, t) \\ &\quad + (1 - \beta) \chi_{p,g} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'}(\vec{r}, t) \phi_{g'}(\vec{r}, t) \\ &\quad + \sum_{k=1}^K \lambda_k C_k(\vec{r}, t) \chi_{dk,g} \end{aligned} \quad (2.20)$$

et les équations des précurseurs suivantes :

$$\begin{aligned} \frac{\partial}{\partial t} C_k(\vec{r}, t) &= -\lambda_k C_k(\vec{r}, t) \\ &\quad + \beta_k \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'}(\vec{r}, t) \phi_{g'}(\vec{r}, t) \quad k = 1, 2, \dots, K \end{aligned} \quad (2.21)$$

La résolution de l'équation de transport dépendante du temps est assez complexe. Afin de déterminer l'état du cœur du réacteur à un instant précis, il faut utiliser des équations où la variable temps disparaît et la variation de la densité des neutrons par rapport au temps est nulle. Dans ce cas, tous les neutrons obtenus grâce à la fission sont considérés instantanés. L'équation de transport stationnaire est la suivante :

$$\begin{aligned} \Sigma_g(\vec{r}) \phi_g(\vec{r}) + \vec{\nabla} \cdot \vec{J}_g(\vec{r}) &= \sum_{g'=1}^G \Sigma_{s,g \leftarrow g'}(\vec{r}) \phi_{g'}(\vec{r}) \\ &\quad + \frac{1}{k_{eff}} \chi_g \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'}(\vec{r}) \phi_{g'}(\vec{r}) \end{aligned} \quad (2.22)$$

où le facteur de multiplication effectif k_{eff} est introduit afin de balancer exactement les pertes et les productions de neutrons. On note aussi que le spectre total des neutrons en statique est une combinaison du spectre des neutrons instantanés et des spectres des neutrons retardés :

$$\chi_g = (1 - \beta)\chi_{p,g} + \sum_{k=1}^K \beta_k \chi_{kd,g} \quad (2.23)$$

2.1.1 Méthodes de résolutions déterministes

Méthode des probabilités de collision P_{ij}

Cette méthode de résolution se base sur la forme intégrale de l'équation de transport. Elle peut être utilisée pour un réseau infini de cellules tout comme pour un réseau fini en spécifiant les conditions frontières. Les sources de neutrons de diffusion et de fission utilisées dans cette approche doivent être isotropes. La procédure de résolution passe par une intégration numérique basée sur les coordonnées curvilignes et nécessite une discrétisation spatiale de la cellule considérée. La discrétisation spatiale est réalisée par l'étape de *tracking*. En deux dimensions, Le *tracking* dépend de plusieurs paramètres et génère un ensemble de lignes d'intégration dans la surface sur laquelle les sources sont intégrées. La résolution de l'équation de transport par la méthode des P_{ij} génère une matrice $I \times I$ pour chaque groupe d'énergie, où I est le nombre de régions. Cette méthode est préférable pour des simulations avec un nombre de régions modeste (Hébert, 2009).

Méthode des caractéristiques MoC

La méthode des caractéristiques repose sur la discrétisation de la forme caractéristique de l'équation de transport stationnaire. Cette méthode résout un ensemble d'équations différentielles ordinaires le long de segments qui définissent les chemins des neutrons. Cette technique se base sur la procédure de *tracking* tout comme la méthode des P_{ij} . Généralement, la MoC est appliquée pour la résolution de système comportant un très grand nombre de régions. La résolution du flux est effectuée en calculant le flux moyen sur chaque segment. Pour intégrer l'équation caractéristique de transport, il faut utiliser des hypothèses. La plus répandue est celle de « *Step Characteristic (SC)* ». Elle suppose que les termes de sources dans l'équation de transport sont constants sur chaque segment. Pour un flux approximé à une dimension, la MOC requiert la résolution d'un système à $(4n + l)$ inconnues, où n est le nombre de régions et l est le nombre de surfaces. Des méthodes d'accélération de convergence peuvent être utilisées. Cette méthode a été utilisée dans la résolution de l'équation de transport dépendante du temps (2.2) (Taylor, 2007).

Méthode des ordonnées discrètes S_n

Cette méthode se base aussi sur une discrétisation de la forme différentielle de l'équation de transport statique. Dans cette approche le flux est discrétisé et résolu pour des angles spécifiques. Des prés-conditionneurs et des approches d'accélération de convergences sont nécessaires pour obtenir la solution (Mattis et Haghghat, 1992).

Méthode des harmoniques sphérique P_n

La méthode des harmoniques sphérique P_n est une discrétisation angulaire de la forme différentielle de l'équation de transport statique. Cette méthode est la plus vieille approche de résolution de l'équation de transport. Elle se base sur une expansion du flux et des termes de sources en harmoniques sphériques. Cette approche a des limites qui dépendent des conditions frontières de la cellule. Pour chaque ordre l de l'expansion, une équation de transport est définie. La résolution numérique de ces équations nécessite l'utilisation d'une méthode de différences finies avec une procédure itérative. (Fletcher, 1983)

2.1.2 Méthode de résolution Monte-Carlo

La méthode Monte Carlo est une simulation directe d'une population de neutrons provenant d'un suivi du comportement aléatoire d'un grand nombre de neutrons. Grâce à un processus aléatoire, on définit les interactions et le comportement des particules. En encaissant certains types d'événements, on obtient un échantillon assez représentatif du comportement des particules vis à vis de ces interactions. En faisant appel à des approches statistiques, le comportement moyen des particules est déduit. Les résultats d'une simulation Monte Carlo sont des variables aléatoires accompagnées chacune par son erreur statistique.

Sachant que la solution de l'équation de transport par une méthode déterministe nécessite une discrétisation énergétique, spatiale et angulaire des variables, les simulations Monte Carlo ne requièrent aucune discrétisation. Ces dernières permettent une description détaillée pour de nombreuses configurations géométriques complexes. Les algorithmes de cette méthode se basent sur le libre parcours d'un neutron depuis sa naissance (par fission ou comme source fixe) jusqu'à sa disparition (par fuite ou absorption). Un grand nombre de neutrons est nécessaire pour approcher la solution de l'équation de transport et obtenir une description précise du comportement des neutrons dans le réacteur. En effet, la convergence de ces méthodes suit la loi des grands nombres.

Durant son parcours, le neutron subit un ensemble d'interactions avec la matière. Ces interactions sont définies par un processus aléatoire se basant sur les sections efficaces du milieu (dans ce contexte, elles retrouvent leurs sens mathématique qui sont : les probabilités

d'interactions du neutron avec la matière). Dans ces calculs de nature stochastique, différents types de sections efficaces peuvent être utilisées :

1. Des sections efficaces microscopiques ponctuelles correspondant aux données fournies dans les évaluations ENDF et traduite en format PENDF par NJOY (McLane, 2001).
2. Des sections efficaces macroscopiques par groupe d'énergie.
3. Des sections efficaces sous forme de tables de probabilités (Martin, 2011).

Plusieurs techniques de rejet et de réduction des variances ont été développées afin d'améliorer le temps de calcul et la performance des algorithmes Monte-Carlo. Parmi ces techniques, on peut nommer :

- L'algorithme d'inversion. Cet algorithme voit son importance dans le calcul de la fonction inverse d'une CDF (Cumulative Distribution Function (CDF)) en utilisant un générateur de variables aléatoires uniformément distribuées. Cet algorithme a été utilisé dans l'évaluation de l'énergie d'un neutron à sa naissance tout en faisant appel à la fonction inverse de la CDF de la fonction du spectre des neutrons (retardés ou instantanés).
- L'approche de *Woodcock* (Woodcock *et al.*, 1965). Elle permet de simplifier l'algorithme du libre parcours du neutron lorsqu'il s'agit d'un milieu hétérogène. Le seul aspect négatif de cette simplification se limite à l'ajout d'une interaction virtuelle à la liste des interactions possibles que le neutron pourrait subir. Le code Serpent se base sur cette technique (Leppänen, 2007).

Algorithme Monte Carlo statique

Ils existe plusieurs algorithmes Monte Carlo. En général, ces derniers contiennent les étapes suivantes :

1. On définit une population initiale distribuée uniformément dans la cellule, l'énergie de tous les neutrons est distribuée aléatoirement en respectant le spectre énergétique des neutrons. La population possède un poids initial prédéfini. Un nombre de cycles total est fixé par l'utilisateur.
2. On échantillonne aléatoirement la direction du parcours pour chaque neutron.
3. Un cycle est défini par le traitement de tous les neutrons de sa population. On traite neutron par neutron, chaque neutron étant suivi individuellement depuis sa naissance jusqu'à sa disparition.
4. On obtient le parcours L que le neutron effectuera entre deux collisions en utilisant la formule :

$$L = -\ln(r)/\Sigma_t \quad (2.24)$$

où r est une variable réelle aléatoire appartenant à l'intervalle $[0,1]$ et uniformément distribuée dans celui-ci.

5. Si le neutron sort de la cellule, on fait appel aux conditions frontières.
6. À la fin de chaque parcours, un tirage aléatoire permettra de définir le type d'interactions que le neutron doit effectuer.
 - (a) Dans le cas d'une réaction réelle (non virtuelle), on passe à l'étape suivante.
 - (b) Dans le cas d'une réaction virtuelle (approche *Woodcock*), un retour à l'étape 4 est effectué en utilisant le même neutron. (Il faut noter que cette technique est utilisée seulement dans quelques codes Monte Carlo comme *Serpent*)
7. Dans le cas d'une interaction réelle avec la matière, la nature de l'interaction doit être définie aléatoirement en utilisant les différentes valeurs de sections efficaces :
 - (a) Réaction de capture, on passe à l'étape 8.
 - (b) Réaction de fission, on passe à l'étape 9.
 - (c) Réaction de diffusion ou (n, Xn) , on passe à l'étape 10 (X est un nombre entier).
8. Lors d'une réaction de capture, le neutron disparaît et on revient à l'étape 3 avec un nouveau neutron.
9. Lors d'une réaction de fission, le neutron disparaît et ν nouveaux neutrons apparaîtront dans l'itération suivante.

En cas d'utilisation de la technique *Survival biasing*, À chaque collision, le poids du neutron est mis à jour comme suit :

$$w' = w * \left(1 - \frac{\Sigma_a}{\Sigma_t}\right) \quad (2.25)$$

Ainsi, les réactions de fissions sont échantillonnées seulement pour créer des sites de fissions pour l'itération suivante. Σ_a et Σ_t dépendent de l'isotope fissile échantillonné (s'il y en a plus que 1). Autrement,

10. Lors d'une réaction de diffusion, un des deux choix suivant est effectué :
 - (a) Si la réaction est une collision, le neutron change de direction et d'énergie. (retour à l'étape 4).
 - (b) Si la réaction est de type (n, Xn) , le neutron voit son poids multiplié par X . (retour à l'étape 4).
11. Les étapes de 2 à 10 sont répétées pour toute la population de neutrons dans un cycle.

12. À la fin de chaque cycle, la valeur du facteur de multiplication effectif du cycle (k_{eff}) est obtenue comme suit :

$$k_{eff,i} = \frac{\text{Poids total des neutrons à l'itération } i}{\text{Poids total des neutrons à l'itération } i-1} \quad (2.26)$$

13. On applique la roulette russe (élimination des neutrons dont le poids est inférieur à un certain seuil).
14. Une normalisation de la population est effectuée afin de retrouver le poids initial défini au départ de la simulation. La normalisation se fait par rapport au $k_{eff,i}$.
15. On répète les étapes de 2 à 14 autant de fois que le nombre de cycles défini par l'utilisateur.
16. On calcule le coefficient de multiplication effectif moyen de toute la simulation ainsi que sa variance (un ensemble de valeurs de k_{eff} des cycles du début de la simulation est ignoré). On calcule aussi la valeur moyenne de tous les paramètres nucléaires demandés par l'utilisateur dans les fichiers d'entrées.

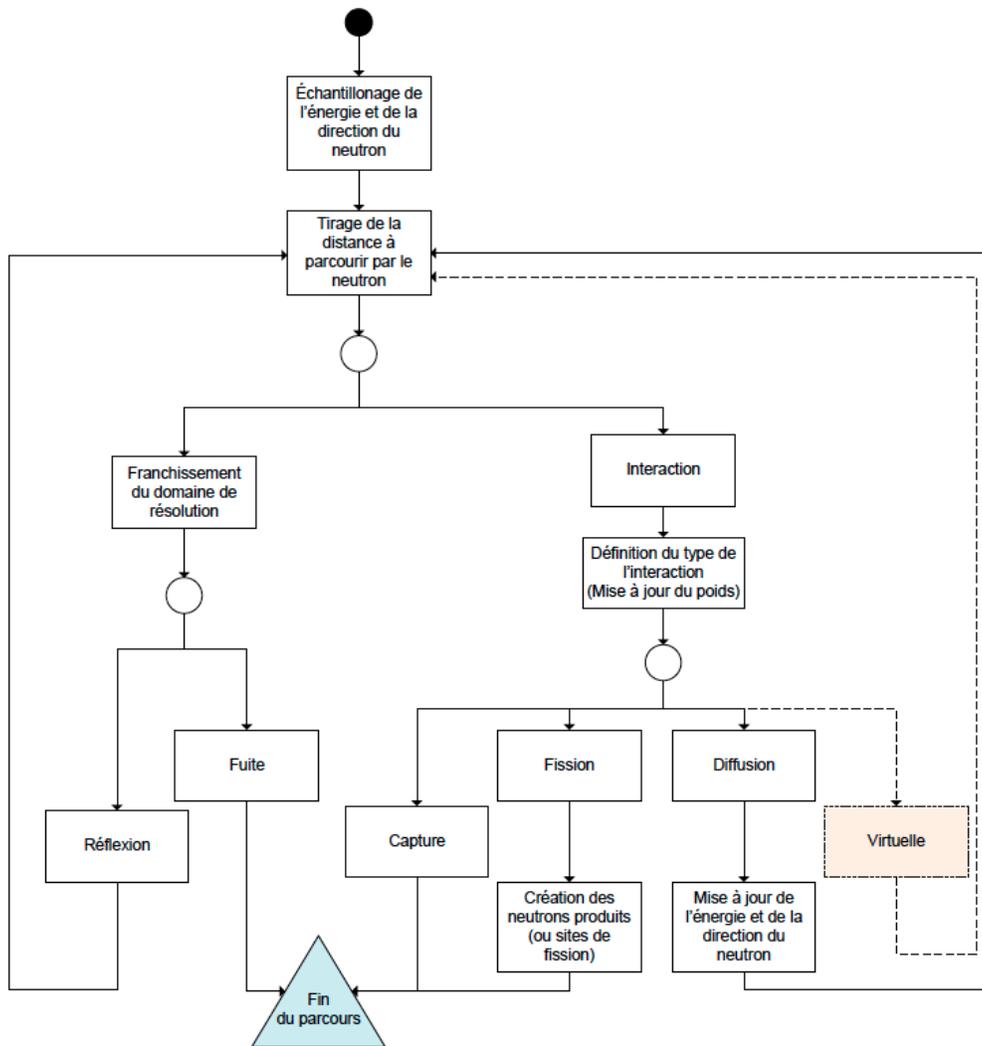


Figure 2.2 Parcours aléatoire d'un neutron par la méthode de Monte Carlo

2.1.3 Condensation et homogénéisation

À l'issue d'un calcul de transport, les résultats obtenus sont :

- la valeur du flux neutronique par groupe d'énergie et par région,
- les valeurs des taux de réactions par groupe d'énergie et par région, et
- les valeurs des sections efficaces pour chaque région.

Afin de passer à l'étape suivante qui se résume dans la résolution de l'équation de diffusion pour tout le réacteur, la procédure de condensation des groupes d'énergies (micros-groupes) en un nombre très petit de nouveaux groupes plus larges (macros-groupes) et la procédure d'homogénéisation des régions de la cellule en quelques macros régions doivent être effectuées.

Ces procédures sont définies comme suit :

1. Un macro volume m ($m \in M$, M est le nombre de macro régions) est tout simplement la somme des micros volumes i ($i \in M \cap N$, où N est le nombre de région initial) :

$$V_m = \sum_{i \in (N \cap M)} V_i \quad (2.27)$$

2. Le flux est une somme sur les régions et sur les groupes d'énergie, H est le nombre de macro-groupes d'énergie, G est le nombre de micro groupes d'énergie et h l'indice du macro groupe en question :

$$\phi_{m,h} = \frac{1}{V_m} \sum_{g \in (G \cap H)} \sum_{i \in (N \cap M)} V_i \phi_{i,g} \quad (2.28)$$

3. Les sections efficaces macroscopiques doivent elles aussi être condensées en énergie et homogénéisées en espace :

$$\Sigma_{x,m,h} = \frac{1}{V_m \phi_{m,h}} \sum_{g \in (G \cap H)} \sum_{i \in (N \cap M)} V_i \phi_{i,g} \Sigma_{x,i,g} \quad (2.29)$$

où x désigne le type de la section efficace.

2.1.4 Calcul de réactivité statique

Après avoir résolu l'équation de transport statique 2.22 et effectué la condensation et l'homogénéisation des sections efficaces, il est important de pouvoir quantifier la réactivité statique introduite dans la cellule suite à la suppression du caloporteur. Cette étape nécessite la résolution de l'équation de transport deux fois. La première fois avec une cellule standard sans aucune perturbation. La deuxième résolution est effectuée pour une cellule perturbée. La réactivité introduite est calculée comme suit :

$$\Delta\rho = \rho_{perturbé} - \rho_{initial} = \frac{k_{eff}(perturbé) - k_{eff}(initial)}{k_{eff}(perturbé) \times k_{eff}(initial)} \quad (2.30)$$

Dans cette thèse, nous calculerons la réactivité introduite dans une cellule CANDU-6 à la suite d'un accident de perte de caloporteur (voir figure 2.3). Par contre, nous ne tarderons pas à expliquer les raisons pour lesquelles la réactivité du vide est positive. Cette constatation a déjà fait l'objet de différentes publications et discussions (dont (Whitlock *et al.*, 1995))

2.1.5 Modèles numériques en transport

Dans cette section, les environnements de simulations choisis comme support dans ce projet ainsi que les modèles à simuler en transport neutronique à l'échelle d'une cellule seront présentés.

Description d'une cellule CANDU-6

Une cellule CANDU-6 (figure 2.3) est constituée des matériaux suivants :

1. Une grappe contenant 37 crayons de combustible sous forme de céramique d'uranium naturel entourée d'une structure (gaine) en Zircaloy, Zr_4 , avec un mince revêtement interne en graphite,
2. De l'eau lourde à une température autour de 300 degrés Celsius et à plus de $10MPa$ de pression qui s'écoule entre les crayons d'uranium jouant le rôle du caloporteur,
3. Un tube de force en Zr-Nb pour contenir la grappe et le caloporteur,
4. Un tube de calandre contenant le tube de force séparé par un espace contenant du CO_2 gazeux,
5. De l'eau lourde à environ 70 degrés Celsius qui joue le rôle de modérateur.

La cellule unitaire mesure $28.575cm$ de largeur. Une grappe de combustible mesure environ $50cm$ de longueur axiale. Il en faut 12 pour remplir un canal de réacteur CANDU-6 (sa longueur active étant de $5.94m$).

Pour des fins de simulation et dans le but de garder les conditions de simulations similaires entre les outils choisis pour effectuer les calculs, les conditions suivantes ont été appliquées :

1. on a considéré une cellule neuve avec un taux de combustion (*burnup*) nul,
2. on a utilisé des sections efficaces à des températures disponibles dans les bibliothèques nucléaires afin d'éviter toute interpolation,
3. les mêmes bibliothèques nucléaires, dans leurs formats multigroupe et continue, ont été utilisées dans toutes les simulations (ENDF-b7.0 et Jeff3.1.1).
4. les conditions frontières sont les mêmes (réflexion spéculaire),

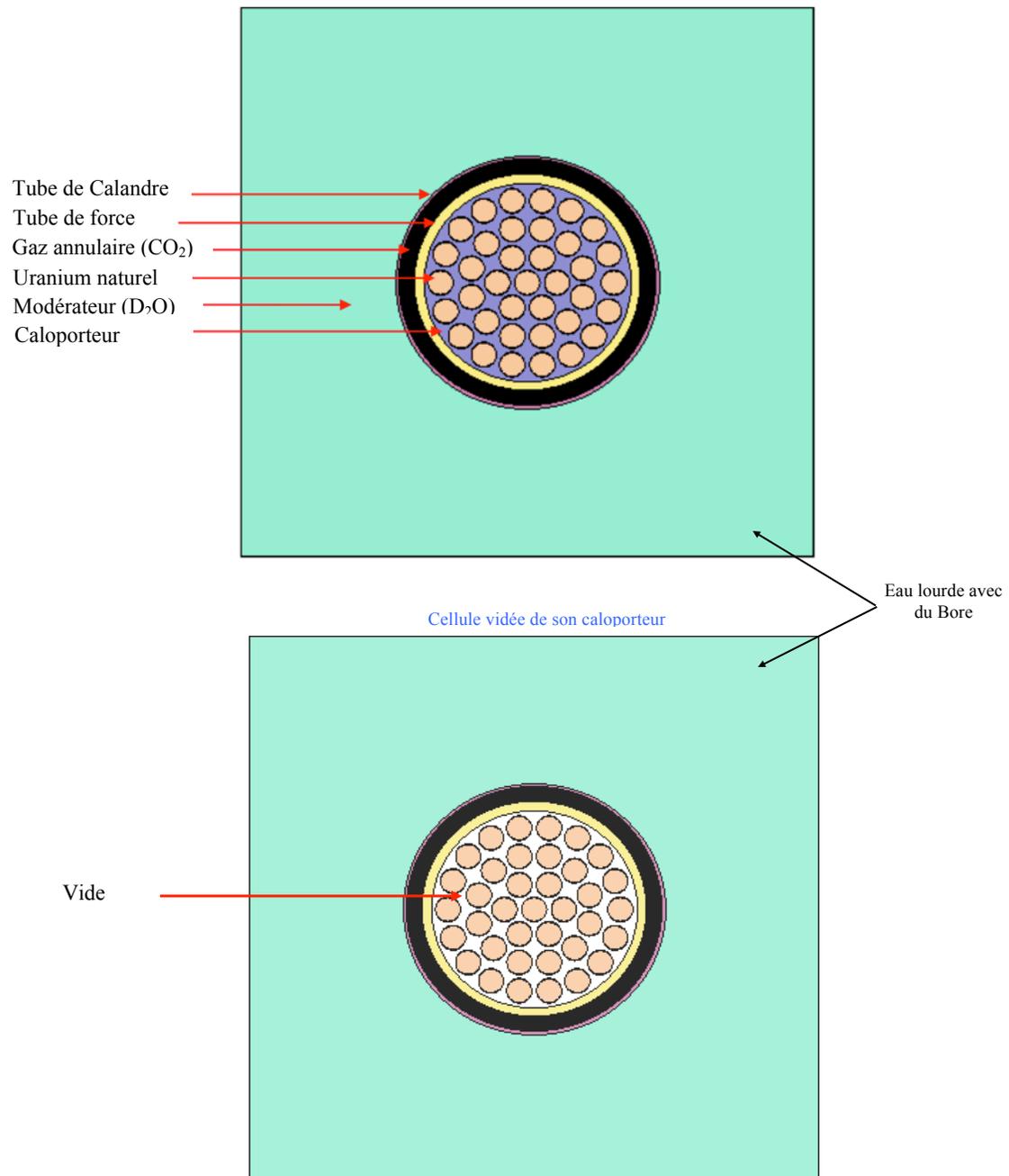


Figure 2.3 Cellule du réacteur CANDU-6 refroidie et vidée de son caloporteur

Le tableau 2.1 présente les paramètres qui sont utilisés dans les différentes simulations pour une cellule CANDU-6. Ce sont les conditions nominales d'une cellule CANDU-6.

Tableau 2.1 Valeurs nominales des paramètres de la cellule du réacteur CANDU-6

Paramètres de modélisation	Nominales
Masse UO ₂ d'une grappe (kg)	19.236
Enrichissement du combustible (%)	0.7114
Pas du réseau : cellule unitaire (cm)	28.575
Température du combustible (°K)	941.29
Température du caloporteur (°K)	560.66
Température du modérateur (°K)	345.66
Densité d'UO ₂ (g/cm ³)	10.4371
Densité du caloporteur (g/cm ³)	0.81212
Densité du modérateur (g/cm ³)	1.0829
Pureté du caloporteur (%D ₂ O)	99.30
Pureté du modérateur (% D ₂ O)	99.92

Modélisation avec le code DRAGON

Le logiciel DRAGON (Marleau *et al.*, 2014) est un code de transport neutronique développé à l'École Polytechnique de Montréal. Il inclut les techniques de résolutions suivantes :

- La méthode des probabilités de collisions P_{ij} .
- La méthode des caractéristiques MoC .
- La méthode des ordonnées discrètes Sn .

Le logiciel permet également la modélisation de différents types de cellules (CANDU, PWR, etc.). Ce dernier est basé sur une approche modulaire qui respecte la grammaire définie par le langage de programmation CLE-2000 (Roy, 1999). Nous avons utilisé la version 5 du code Dragon dans le cadre de cette thèse.

Le fichier d'entrée décrit les différentes composantes d'une cellule du réacteur CANDU-6 avec le code DRAGON. Ce fichier inclut aussi différents paramètres de simulation ainsi que les méthodes de résolution de l'équation de transport statique choisies. Il est à noter que la méthode des P_{ij} a été retenue comme approche de résolution. Une condensation énergétique et une homogénéisation spatiale de toute la cellule permet d'obtenir le flux neutronique et les sections efficaces macroscopiques à deux groupes d'énergie (ainsi que tout autre paramètre nucléaire de la cellule). Une homogénéisation spatiale pour la région de l'eau lourde du modérateur permettra d'obtenir les paramètres de l'eau lourde qui seront utilisés comme

réflecteur dans les simulations de tout le réacteur.

Modélisation avec le code Serpent

Serpent est un logiciel de transport Monte Carlo. L'une de ses principales applications est la génération de constantes de groupes homogénéisées qui pourront être introduites dans les simulations de réacteurs complets. Serpent est capable d'être exécuté en mode parallèle sur des ordinateurs qui possèdent plusieurs unités de calculs. Le parcours des neutrons est basé sur la méthode de Woodcock modifiée (*extended delta tracking method*, (Leppänen, 2007)). Cette technique a pour but d'éviter la vérification des frontières entre les différents milieux de la géométrie lors du passage du neutron d'une zone à une autre. Le mot *extended* est ajouté pour caractériser l'ajout apporté à cette méthode dans le code Serpent. Le code Serpent se base sur deux valeurs de majorants dans le cas d'une simulation de cellule en présence d'un absorbant. Un traitement particulier est appliqué pour le parcours du neutron dans ce cas.

La modélisation de la cellule CANDU-6 avec le logiciel Serpent doit avoir les paramètres les plus proches de ceux utilisés lors des simulations réalisées avec DRAGON afin de pouvoir comparer leurs résultats. Les paramètres ont été détaillés dans le tableau 2.1.

Modélisation avec le code OpenMC

OpenMC est un code de simulation libre développé par le groupe *CRPG* au *Massachusetts Institute of Technology* (MIT) (Romano et Forget, 2013). Ce logiciel utilise le principe de base de la simulation Monte Carlo où les parcours des neutrons sont interrompus à chaque interface séparant deux milieux homogènes différents. En d'autres mots, la technique de Woodcock n'est pas utilisée dans ce logiciel. Afin d'accélérer les calculs et optimiser le temps de simulation, OpenMC utilise des algorithmes de parallélisations des processus assez avancés (Romano *et al.*, 2016). Une procédure de lecture des fichiers d'entrées, se basant sur le principe des balises des fichiers XML, est utilisée afin de simplifier la lecture du modèle à simuler. Dans le cadre de ce projet, des comparaisons du code OpenMC seront effectuées avec le code Serpent et le code DRAGON pour une cellule CANDU-6.

Tout comme pour le code Serpent, la modélisation de la cellule CANDU-6 avec le logiciel OpenMC doit avoir les paramètres les plus proches de ceux utilisés dans les simulations réalisées avec le logiciel DRAGON afin de pouvoir comparer leurs résultats. Par contre, ces paramètres doivent être exactement les mêmes que ceux utilisés dans les simulations réalisées avec le code Serpent. La géométrie de la cellule CANDU-6 décrite avec le code OpenMC est fournie en annexe C.

2.2 Diffusion neutronique

Afin d'exprimer le bilan des neutrons uniquement au moyen du flux scalaire, un ensemble d'approximations doit être appliqué à l'équation de transport pour pouvoir la résoudre à l'échelle du réacteur. Par la suite, l'équation de diffusion suivante est obtenue (Rozon, 1992) :

$$\begin{aligned}
\frac{1}{v_g} \frac{\partial}{\partial t} \phi_g(\vec{r}, t) &= -\Sigma_{t,g}(\vec{r}, t) \phi_g(\vec{r}, t) \\
&+ \vec{\nabla} \cdot D_g(\vec{r}) \vec{\nabla} \phi_g(\vec{r}, t) \\
&+ \sum_{g'=1}^G \Sigma_{s,g \leftarrow g'}(\vec{r}, t) \phi_{g'}(\vec{r}, t) \\
&+ (1 - \beta) \chi_{p,g} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'}(\vec{r}, t) \phi_{g'}(\vec{r}, t) \\
&+ \sum_{k=1}^K \lambda_k C_k(\vec{r}, t) \chi_{dk,g}
\end{aligned} \tag{2.31}$$

ainsi que les équations des précurseurs suivantes :

$$\begin{aligned}
\frac{\partial}{\partial t} C_k(\vec{r}, t) &= -\lambda_k C_k(\vec{r}, t) \\
&+ \beta_k \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'}(\vec{r}, t) \phi_{g'}(\vec{r}, t) \quad k = 1, 2..K
\end{aligned} \tag{2.32}$$

où le coefficient de diffusion D est défini comme suit :

$$D(\vec{r}) = \frac{1}{3(\Sigma_t(\vec{r}) - \bar{\mu}_0 \Sigma_s(\vec{r}))} = \frac{1}{3\Sigma_{tr}(\bar{\mu}_0)} \tag{2.33}$$

$\bar{\mu}_0$ est la moyenne du cosinus de l'angle de diffusion dans le laboratoire.

Il est à noter que la loi de *Fick* a été utilisée afin de se débarrasser du courant neutronique et garder seulement le flux comme seule inconnue.

L'équation de diffusion statique est la suivante :

$$\begin{aligned}
\Sigma_{t,g}(\vec{r}) \phi_g(\vec{r}) - \vec{\nabla} \cdot D_g(\vec{r}) \vec{\nabla} \phi_g(\vec{r}) &= \sum_{g'=1}^G \Sigma_{s,g \leftarrow g'}(\vec{r}) \phi_{g'}(\vec{r}) \\
&+ \frac{1}{k_{eff}} \chi_{p,g} \sum_{g'=1}^G \nu_{g'} \Sigma_{f,g'}(\vec{r}) \phi_{g'}(\vec{r})
\end{aligned} \tag{2.34}$$

L'équation de diffusion statique, tout comme l'équation de transport statique, sont des équations à valeurs propres et vecteurs propres. La solution triviale $\phi(r) = 0, \forall r$ vérifie ces équations mais ne présente aucun intérêt. Différentes valeurs propres existent pour chacune d'elles. La plus grande valeur propre correspond à la solution fondamentale du problème.

Elle est égale au facteur de multiplication k_{eff} . Le vecteur propre correspondant à la solution fondamentale représente le flux neutronique $\phi(r)$. Seule la solution fondamentale fournit un vecteur de flux positif pour toute les régions et tous les groupes d'énergie.

2.2.1 Résolution de l'équation de diffusion

Plusieurs techniques de résolution de l'équation de diffusion existent. Pour des problèmes d'ordre important en nombre de groupe d'énergie ou bien en géométrie complexe, des méthodes numériques, comme les méthodes nodales, la méthode des différences finies ou la méthode des éléments finis, peuvent être utilisées (Hébert, 2009). Dans la majorité des techniques utilisées pour résoudre l'équation de diffusion, la géométrie est décomposée en régions homogènes afin de supprimer la dépendance spatiale fine des sections efficaces et du coefficient de diffusion (raison pour laquelle la procédure d'homogénéisation et de condensation est effectuée au préalable).

Afin d'obtenir la distribution du flux neutronique dans le réacteur en fonction du temps, en cinétique espace-temps, il est nécessaire de résoudre l'équation de diffusion dépendante du temps (2.31). Pour y arriver, une discrétisation du flux neutronique et des concentrations des précurseurs en utilisant une relation de différences finies dans le temps est nécessaire. Dans ce cadre, plusieurs méthodes de calculs existent : le modèle *implicite*, *explicite*, la méthode *Crank-Nicolson*, etc. Dans le cadre de ce projet, la méthode implicite a été choisie pour déterminer le comportement du réacteur après l'introduction d'un accident de type (LOCA).

Rappel pour la méthode implicite :

$$\begin{aligned} \frac{d\phi(t)}{dt} &= R\phi(t) \\ \frac{\phi^{n+1} - \phi^n}{t^{n+1} - t^n} &= R\phi^{n+1} \\ \phi^{n+1} &= (I - \Delta t R)^{-1} \phi^n \end{aligned} \tag{2.35}$$

où $\Delta t = t^{n+1} - t^n$

Il est important de mentionner que le modèle de cinétique espace-temps exige une condition initiale (statique) pour la distribution du flux neutronique dans le réacteur. La distribution du flux obtenue en résolvant l'équation statique de diffusion (2.34) pour des conditions initiales nominales non perturbées a été choisie.

Un accident de type LOCA dans un réacteur de type CANDU-6 se résume dans la rupture d'un des huit collecteurs du caloporteur du circuit primaire qui sont reliés aux générateurs de vapeur. Cette rupture a pour conséquence la perte très rapide du caloporteur. Comme il existe huit collecteurs dont quatre pour chaque moitié du cœur, la rupture de ce dernier

affecte seulement la moitié du réacteur (figure 2.4). Le système de refroidissement d'urgence du cœur (RUC) est par la suite enclenché. Les simulations effectuées dans le cadre de ce projet sont de courtes durées et ne prennent pas en considération les actions entreprises par le RUC.

	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z		
1											2	2	2	2	2	2												
2								2	2	2	2	2	2	2	2	2	2	2	2	2								
3						2	2	2	2	2	2	1	1	1	1	1	1	1	2	2	2	2	2					
4					2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2					
5				2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2				
6			2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2			
7			2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2			
8		2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2		
9		2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2			
1	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2		
11	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
12	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
13	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
14	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
15	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
16	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
17	2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
18		2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
19		2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
2			2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
21			2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2		
22				2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2		
23					2	2	2	1	1	1	1	1	1	1	1	1	1	1	1	1	1	2	2	2				
24						2	2	2	2	2	2	1	1	1	1	1	1	2	2	2	2	2	2					
26								2	2	2	2	2	2	2	2	2	2	2	2	2								
26											2	2	2	2	2	2												
1	Cellules affectées par la perte du caloporteur																											

Figure 2.4 Cellules affectées par la vidange du caloporteur dans un accident LOCA du réacteur CANDU-6

CHAPITRE 3 APPROCHE MONTE CARLO POUR LE TRANSPORT DÉPENDANT DU TEMPS

Afin d'obtenir des sections efficaces macroscopiques homogénéisées dépendantes du temps pour une cellule de réacteur CANDU-6, 2 principales options existent :

- Développer une approche déterministe dépendante du temps, ou
- Développer une approche Monte Carlo dépendante du temps.

Les principales considérations dans l'approche d'aide à la décision sont les suivantes :

1. la disponibilité des codes sources des logiciels de transport afin de pouvoir leurs apporter les modifications nécessaires,
2. le langage de programmation utilisé pour ces logiciels,
3. l'ampleur des modifications à effectuer, et
4. la disponibilité de la documentation nécessaire.

Les logiciels de transport neutronique, dont les codes sources sont disponibles, sont uniquement DRAGON, OpenMC et SERPENT.

Du côté langage de programmation, les logiciels libres DRAGON (version 5) et OpenMC sont écrits en Fortran (2003), alors que Serpent est écrit avec le langage C. Ce point, tout comme le premier, ne constitue pas un critère important pour orienter le projet.

Le troisième point est décisif. D'une part, la résolution déterministe des équations de transport dépendantes du temps est assez complexe. D'autre part, le travail nécessaire afin de programmer et d'intégrer cette approche dans le code DRAGON exigerait une réécriture importante du logiciel. Par contre, développer et intégrer un nouveau module dépendant du temps dans le code Serpent ou OpenMC semble être faisable et plus réaliste.

Ainsi, une approche Monte Carlo a été choisie. Il demeure par contre important de faire le choix entre Serpent et OpenMC pour développer le nouveau code.

Serpent est plus ancien qu'OpenMC (2007 vs 2013) et une approche dépendante du temps fut initiée par son développeur (Leppänen, 2013). D'autre part, sa documentation manque de rigueur. Par contre, OpenMC est un logiciel très récent qui ne possède pas d'approches temporelles et sa documentation est assez complète. Ce dernier est aussi un logiciel libre (Open-Source), contrairement à Serpent.

Finalement, le projet s'est orienté vers une approche Monte Carlo tout en utilisant le logiciel OpenMC.

3.1 Principe général et prototype développé

Afin d'évaluer la faisabilité et prévoir les difficultés du projet, un prototype a été développé. Le développement a été réalisé en deux principales étapes.

Lors de la première étape, un modèle d'algorithme Monte Carlo multigroupe statique a été mis en place. Ce dernier comporte trois sections principales :

1. la gestion de la géométrie en développant un modèle orienté objets. Les géométries disponibles dans le prototype sont : cercle, carré et rectangle. D'autres géométries peuvent être développées et intégrées au prototype assez facilement.
2. un algorithme du libre parcours du neutron : cet algorithme comprend principalement les interactions suivantes : (n,n) , $(n,2n)$, (n,γ) , (n,f) .
3. un algorithme simplifié de la roulette russe.

L'objectif pour lequel ce prototype statique a été réalisé est de valider l'approche utilisée dans le traitement du parcours des neutrons.

Cette vérification a été effectuée en simulant un modèle simple et en comparant ses résultats avec DRAGON (Marleau *et al.*, 2014). Le calcul d'autoprotection des résonances a été réalisé par la méthode de Stamm'ler (Hébert et Marleau, 1991). Il s'agit d'une cellule annulaire avec des sections efficaces à deux groupes d'énergie (similaire au TCM01 du manuel de DRAGON (Marleau *et al.*, 2014)). Avec ce prototype et en utilisant les mêmes paramètres que ceux utilisés dans DRAGON, les valeurs de coefficients de multiplication effectif, k_{eff} , obtenues sont très similaires ($k_{eff} = 0.387$ pour le prototype, alors que pour DRAGON $k_{eff} = 0.390$).

La deuxième étape consistait à rendre le prototype dynamique et dépendant du temps. Durant cette étape, seul l'algorithme du libre parcours des neutrons a été retenu en lui apportant quelques modifications qui seront détaillées plus tard dans ce chapitre. Un traitement particulier des neutrons retardés ainsi que les paramètres temporels de tous les neutrons font parti des ajouts à ce prototype. Ainsi, la faisabilité du projet a été confirmée.

3.2 Traitement temporel des neutrons

Le traitement temporel des neutrons se base principalement sur deux paramètres, le libre parcours et l'énergie du neutron. Parmi les hypothèses qui ont été utilisées lors de l'établissement de l'équation de Boltzmann, nous avons celle qui stipulent qu'un neutron ne peut voyager qu'en ligne droite et que son énergie est toujours constante entre deux collisions. Ainsi, en déterminant la distance x que ce dernier doit parcourir (2.24), il est facile d'obtenir le temps nécessaire pour son déplacement :

$$E_i = \frac{1}{2}m_n v_i^2 \iff t_i = \frac{L_i}{\sqrt{\frac{2E}{m_n}}} \quad (3.1)$$

où $m_n = 1.67492735110^{-27} Kg$ est la masse du neutron et t_i la durée, en secondes, du parcours i .

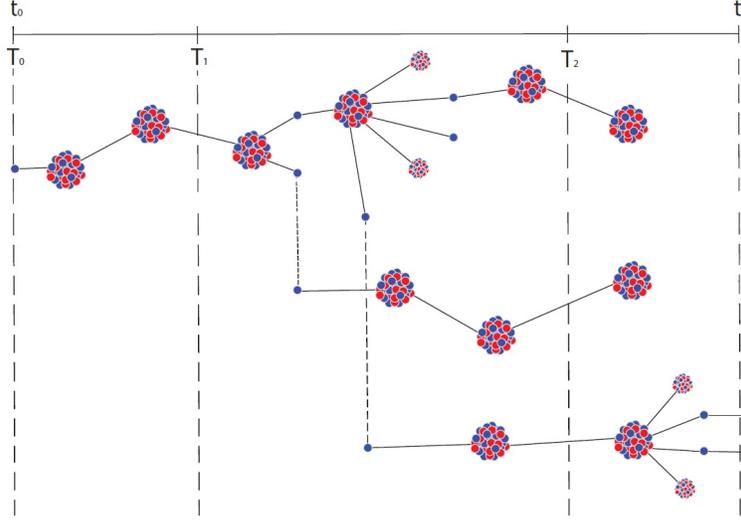


Figure 3.1 Évolution de la population de neutrons dans un milieu multiplicatif en subissant deux perturbations à T_1 et T_2

Lors de sa libération du noyau atomique, le neutron possède une grande énergie cinétique lui permettant de parcourir rapidement des trajectoires au cours desquelles il subit des collisions.

Le principe général du traitement temporel des neutrons se fonde sur l'enregistrement du moment d'apparition du neutron T_0 , le moment de sa disparition T_f , la durée de chaque parcours t_i et finalement, le moment de chaque collision. La durée de vie totale du neutron L_T est calculée comme suit :

$$L_T = \sum_i^I t_i = \sum_i^I \frac{L_i}{\sqrt{\frac{2E}{m_n}}} \quad (3.2)$$

où I est le nombre total de parcours effectués par le neutron.

Le temps de disparition de chaque neutron n'est autre que le temps d'apparition additionné à la durée de vie totale du neutron. Pour les neutrons instantanés et les neutrons retardés respectivement, les temps de disparitions sont :

$$\begin{aligned}
T_{0,p}^{n+1} &= T_0^n + L_T^n \\
T_{0,d}^{n+1} &= T_0^n + L_T^n + D_t^{n+1}
\end{aligned}
\tag{3.3}$$

où n est l'index du neutron incident et p et d sont les symboles respectifs pour les neutrons instantanés et retardés. Le paramètre D_t , qui représente le délai d'apparition d'un neutron retardé, sera introduit dans la section 3.4.

Ces enregistrements sont nécessaires afin de déterminer les différentes valeurs des taux de réactions à chaque intervalle de temps. D'autre part, une procédure est appelée afin d'enregistrer le temps d'apparition pour chaque neutron. Selon le type du neutron, instantané ou retardé, un temps d'apparition est calculé en conséquence. La figure 3.2 explicite le schéma général des différents calculs des variables temporelles.

En programmation, chaque neutron est représenté par un objet qui possède un vecteur de données. Chaque case de ce vecteur contient un ensemble de données qui enregistre principalement le moment et le lieu de chaque collision. Dans le cadre du traitement temporel, des modifications doivent être apportées à l'algorithme du libre parcours du neutron. Deux approches se présentent :

- La première approche se base sur l'enregistrement de toutes les données relatives au neutron et principalement sur le suivi du neutron depuis sa naissance jusqu'à sa disparition. Cette approche a été utilisée dans le prototype. Elle répond partiellement aux besoins du projet. Par contre, en utilisant cette méthode, la configuration géométrique et matérielle devra être conservée tout au long de la simulation. Cette approche nécessite des modifications mineures.
- La deuxième approche se base sur le traitement des neutrons par intervalle de temps. Au départ de chaque itération temporelle, la durée de l'intervalle est définie. Ensuite, chaque neutron est suivi en enregistrant ses données relatives à cet intervalle de temps seulement. Pour chaque neutron qui demeure dans le système à la fin de l'itération temporelle, une procédure enregistre sa position, sa direction et la distance restante de son parcours. Cette procédure de traitement par intervalle est réitérée jusqu'à la fin de la simulation. La figure 3.3 montre le parcours d'un neutron durant deux intervalles de temps. Cette approche offre une liberté très importante, car en délimitant les intervalles de temps, la géométrie et/ou les densités des différents matériaux peuvent être modifiés avant le début de chaque intervalle.

Ainsi, il serait possible de simuler plus efficacement les différentes transitoires où les densités des matériaux varient. Cette approche a été retenue pour la deuxième étape du projet et sera développée dans un nouveau module dynamique d'OpenMC.

Finalement, il a été décidé d'implémenter une approche combinant les deux précédentes : autrement dit, suivre neutron par neutron tout en mettant à jour les sections efficaces de la cellule lors des perturbations durant le parcours du neutron en question.

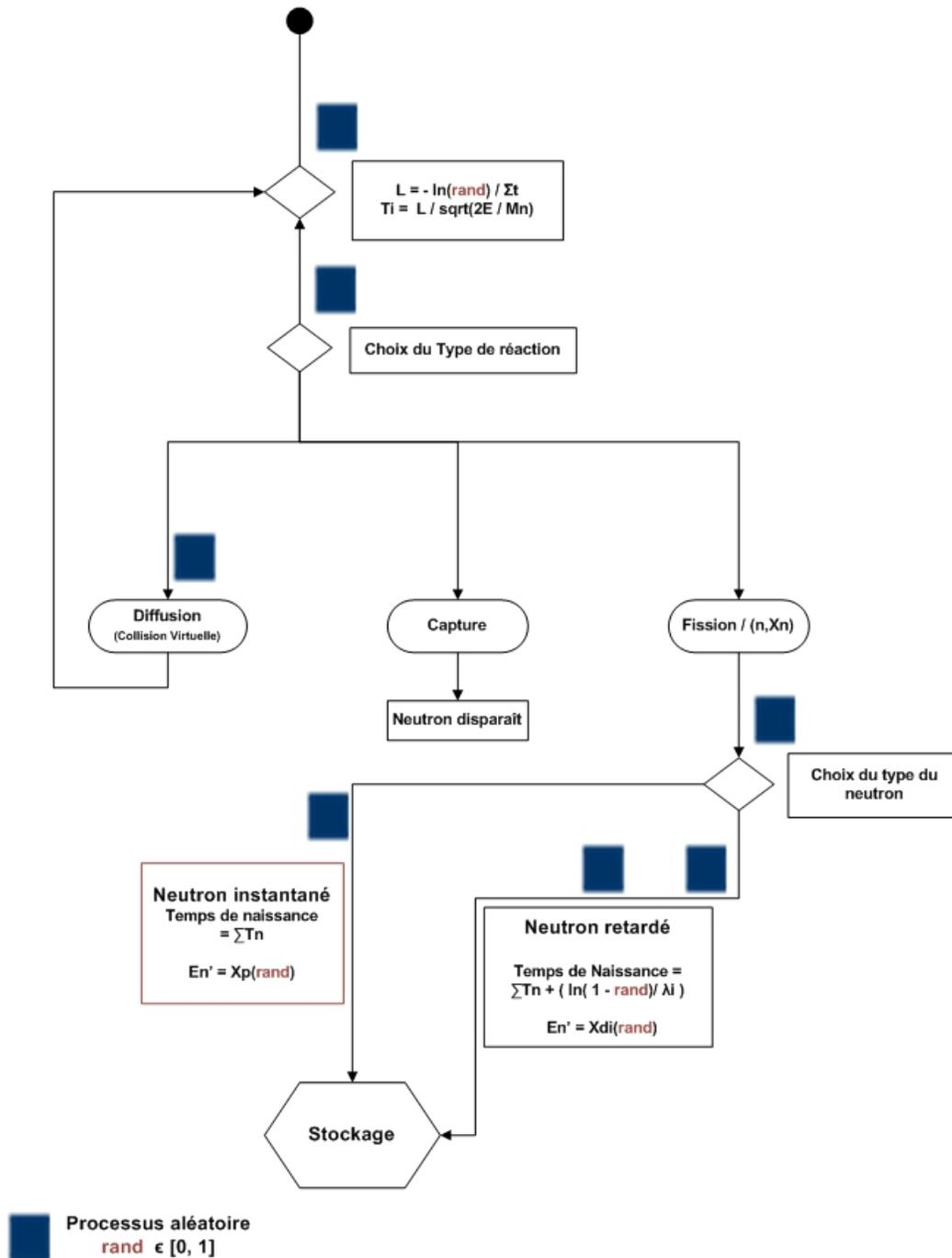


Figure 3.2 Vie d'un neutron

Les simulations qui seront effectuées dans le cadre de ce projet respecteront les mêmes

conditions et approches que celles utilisées dans les calculs standard. Autrement dit, une vidange complète du caloporteur sera effectuée à un instant t_1 pour ensuite observer le comportement des neutrons d'une manière continue dans le temps.

Il est à noter que le traitement temporel des neutrons par intervalle de temps n'a aucun lien avec les intervalles de récolte et traitement des données. Autrement dit, la procédure décrite ci-dessus pour le traitement par intervalle est développée afin de permettre la modification de la géométrie et des propriétés nucléaires des matériaux. Les intervalles seront choisis de telle sorte que leur nombre sera minimisé et que chaque intervalle est délimité par deux perturbations. Pour le calcul des sections efficaces macroscopiques homogénéisées et condensées, un pas de temps Δt constant sera défini durant le traitement des résultats (voir section 3.5.1).

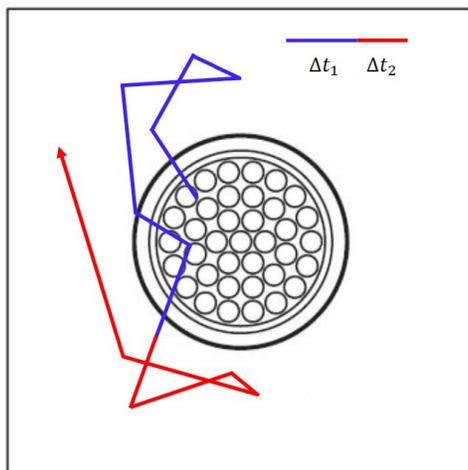


Figure 3.3 Libre parcours du neutron par intervalle de temps

3.3 Principales modifications par rapport au modèle Monte Carlo statique

La figure 3.4 présente un aperçu des boucles imbriquées présentes dans l'algorithme Monte Carlo statique. Ces mêmes boucles existent aussi dans le modèle dynamique. Les modifications qui ont été apportées à OpenMC seront présentées en procédant par boucle. Avant d'aborder ces modifications, une brève présentation des rôles principaux de chaque boucle dans l'algorithme statique doit être élucidé :

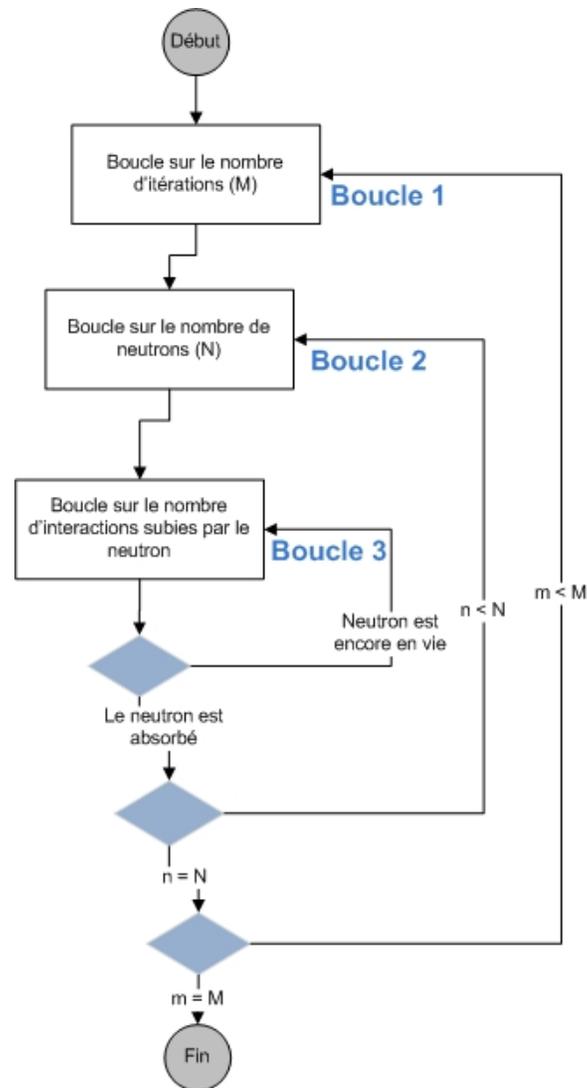


Figure 3.4 Les différentes boucles d'un algorithme Monte Carlo

- la Boucle 1 : cette boucle traite le nombre d'itérations général, M , de l'application. À chaque itération, tous les neutrons d'une population sont suivis. Par la suite, un appel au procédé de la roulette russe permet d'éliminer les neutrons de faible poids. À la fin de chaque itération, une normalisation du poids de la population de neutrons a lieu. Pour chaque itération, la valeur du k_{eff} est calculé.
- la Boucle 2 : dans cette boucle, un traitement neutron par neutron est réalisé. Chaque génération de neutrons doit avoir un poids total égale à N , chaque neutron possède un poids initial presque égal à 1.
- la Boucle 3 : le parcours des neutrons : chaque neutron subit un ensemble de collisions différentes. Des règles de probabilités sont appelées afin de déterminer le type

d'interaction.

La complexité de l'algorithme statique peut être déterminée en analysant la figure 3.4. Si on considère que le poids de chaque neutron est égal à 1, et que le nombre moyen de collisions que chaque neutron subit est C , la complexité de l'algorithme est $O(M \times N \times C)$.

Les modifications nécessaires afin de transformer cet algorithme d'un modèle statique à un modèle dynamique sont :

- Au niveau de la première boucle, le traitement itératif demeurera inchangé. Par contre chaque itération aura un nouveau contexte ; elle représente le traitement d'une génération de neutrons qui évolue dans le temps. L'itération commencera avec un nombre N de neutrons au temps $t = 0$ et finira lorsque tous les neutrons créés auront un temps de naissance plus grand que la durée de la simulation. La modification majeure au niveau de cette boucle est la disparition de la roulette russe et du calcul de k_{eff} . Ainsi, le nombre de neutrons n'est pas constant.
- Au niveau de la deuxième boucle, le principe général d'une génération est modifié. Dans le modèle statique, une génération de neutrons est un ensemble de neutrons dont le poids est $\approx N$, empilé dans une file de mémoire du type "First In First Out" (FIFO). La boucle prend fin après avoir traité tous les neutrons de la génération. Le poids de chaque neutron peut varier en fonction des interactions qu'il subit. Dans le modèle dynamique, le poids de chaque neutron est toujours égal à 1. Le nombre total de neutrons au début de l'itération est égal à N . Par contre, la fin de la boucle n'est pas liée au nombre N de neutrons, elle est liée aux neutrons qu'ils restent à traiter dans la file de mémoire. Cette modification est nécessaire car le nombre de neutrons de la génération n'est pas constant. Chaque neutron qui est né à la suite d'une interaction, dont le temps de naissance appartient à l'intervalle de la simulation, est ajouté à la file de stockage des neutrons. L'itération prend fin quand tous les neutrons de la file sont traités.
- La troisième boucle gère l'ensemble des interactions qu'un neutron subit durant son évolution. La gestion des interactions et les lois de probabilités ne subiront aucune modification. Chaque neutron possède une variable qui définit son temps de naissance et une autre qui définit son temps de disparation. Un tableau est utilisé pour enregistrer les durées des parcours de chaque neutron ainsi que les types d'interactions qu'il subit à la fin de chaque parcours.

Lors de la définition de la longueur du parcours que le neutron doit franchir, une procédure calcule la durée de ce dernier. Ce temps est additionné à la durée de vie totale du neutron. Dans cette boucle, les modifications sont liées principalement aux résultats des interactions.

Lors d'une réaction de capture ou de fission, le neutron incident disparaît et la variable qui définit le temps de disparition du neutron est mise à jour. Lors d'une collision, le même traitement que celui du modèle statique est appliqué. Par contre, si le neutron subit une fission ou une réaction (n, Xn) , une nouvelle procédure de traitement est appliquée. Dans le cas où la réaction est du type (n, Xn) , le neutron disparaît et donne lieu à X nouveaux neutrons de poids égal à 1 (X est un entier). Ces neutrons auront comme temps de naissance (apparition) le même que celui de la disparition de leur prédécesseur (*hypothèse : les interactions sont considérées instantanées*). Ils seront ajoutés à la file de stockage des neutrons de l'itération en cours.

De même, lors d'une fission, un nouveau processus est appliqué. En premier lieu, il faut définir le nombre de neutrons instantanés et le nombre de neutrons retardés qui apparaîtront à la suite de cette fission. Le processus de définition de ces paramètres sera détaillé plus tard dans ce chapitre. Si les nouveaux neutrons sont instantanés, la même procédure que celle de la réaction (n, Xn) est appliquée. Si les neutrons sont retardés, la première étape est de déterminer le groupe de neutrons retardés auquel appartient ce dernier (en utilisant les variables β_i de chaque groupe, voir section 3.4). Une fois le groupe du nouveau neutron retenu, son temps d'apparition est calculé en utilisant la constante de décroissance λ_i du groupe en question. Si ce dernier ne dépasse pas la durée totale de la simulation, le neutron est ajouté à la file des neutrons de l'itération.

Il est important de noter qu'une procédure de vérification de la durée de vie du neutron durant son parcours est appelée afin de stopper le neutron lors de la perturbation.

À la fin de chaque itération, un calcul des différents paramètres nucléaires de la cellule homogénéisée est effectué tout en décomposant la durée de la simulation en petits intervalles de temps respectant certaines contraintes. Une fois toutes les itérations effectuées, les moyennes des différentes variables nucléaires (comme sections efficaces macroscopiques, coefficients de diffusion, etc. . .) sont calculées pour chaque intervalle de temps. Ces valeurs moyennes seront utilisées dans la simulation du réacteur au complet.

Même si le nombre d'itérations M et le nombre d'interactions C demeurent constants dans l'algorithme dynamique, la complexité de ce dernier est beaucoup plus importante que celle de l'algorithme statique à cause du nombre variable de neutrons dans la deuxième boucle.

3.4 Traitement des neutrons retardés

Le code OpenMC est un code Monte Carlo qui a pour objectif de résoudre l'équation de transport statique. La résolution de l'équation de transport statique ne requiert aucun

traitement spécial pour les neutrons retardés. Pour cette raison, les données et méthodes disponibles dans le code OpenMC ne sont en aucun cas dédiés au traitement de ce paramètre. Dans le cadre de ce projet, une importance accrue est accordée aux neutrons retardés.

Le traitement des neutrons retardés dans une étude de cinétique espace-temps pour un réacteur nucléaire est très important. Grâce à la petite proportion de neutrons retardés et à leurs longues demi-vies, la période du réacteur à la suite d'une perturbation, est assez grande. Quelques travaux ont été menés dans le but d'introduire les neutrons retardés dans des simulations Monte Carlo (Sjenitzer et Hoogenboom, 2011) et la plus récente a été réalisé par l'équipe de développement de Serpent 2 (Valtavirta *et al.*, 2016). L'introduction des neutrons retardés est beaucoup plus simple en considérant les concentrations des précurseurs et leurs évolutions dans le temps. Cette approche a été utilisée dans le code Serpent 2. Malheureusement, le code OpenMC ne contient toujours pas un module de gestion de la concentration des précurseurs. Pour ne pas tomber dans une impasse et devoir attendre auprès des développeurs, Il a fallu inventer une nouvelle méthode pour contourner cette lacune.

Il est important de faire la différence entre deux ensembles de neutrons retardés qui apparaîtront durant la simulation. Lors des fissions qui auront lieu dans le combustible, une proportion des neutrons apparaîtra avec délai variable, ces neutrons seront gérés selon une procédure assez standard (expliquée dans la sous section suivante). Par contre, il est important de mentionner qu'au départ de la simulation une proportion des neutrons retardés manquera à la population initiale de neutrons.

Pour comprendre la provenance de ces neutrons, il faut imaginer que le réacteur était en fonction depuis un temps très grand. Durant cette période, qui précède le temps T_0 auquel démarre la simulation, des fissions ont eu lieu et ont donné naissance à des neutrons instantanés et d'autres retardés. Certaines de ces fissions font apparaître des neutrons retardés après le temps T_0 . Si ces derniers ne sont pas pris en considérations, une quantité importante de neutrons retardés manquera au démarrage du réacteur. La section 3.4.2 présente l'approche définie pour contourner cette difficulté.

3.4.1 Traitement des neutrons retardés durant la simulation

Avant d'aborder l'aspect technique du traitement des neutrons retardés, il est nécessaire de clarifier la procédure utilisée et, surtout, expliquer l'abus de langage qui sera utilisé dans le reste du chapitre. Lors de chaque fission, un ensemble de neutrons instantanés ainsi que des fragments de fissions sont produits. En d'autres mots, les produits de fissions sont très divers et varient d'une fission à une autre pour un même isotope fissile. La majorité des produits de fission auront des états excités et devront effectuer un ensemble de décroissances

radioactives pour atteindre un état stable. Parmi les produits de fission, certains auront des énergies d'excitation qui dépassent l'énergie de séparation d'un neutron. Ces derniers, après un certain temps, émettront un neutron et atteindront la stabilité. Ce neutron est appelé un *neutron retardé*. Chaque isotope, émettant un neutron retardé, respecte une loi de décroissance, bien précise, dictant le temps de libération du neutron. Cette loi de décroissance est caractérisée par une variable λ propre à l'isotope. Des études menées par Keepin (Keepin, 1965) ont permis de regrouper tous les isotopes libérant des neutrons retardés en six groupes (d'autres études ont mené à 8, JEFF3.1.1). Chaque groupe est caractérisé par sa propre valeur λ qui est l'inverse du temps moyen d'attente avant l'émission du neutron.

En résumant cette chaîne, une fission d'un atome produit des produits de fissions, quelques uns de ces produits de fissions émettront un neutron appartenant à un des groupes de neutrons retardés. Ces produits de fission attendront un certain temps avant d'émettre le neutron. Dans le reste du document, l'expression « les neutrons retardés produits à chaque fission » résume le processus décrit ci-dessus.

Dans le cadre de ce projet, les concentrations des précurseurs ne sont pas gérées. Lors d'une fission, le type du produit de fission généré n'est pas défini. Seulement, en appliquant une approche stochastique, le groupe auquel appartient le produit de fission est défini. Le neutron retardé qui sera émis appartiendra à ce groupe. Ensuite, la constante de décroissance du groupe, λ_i , sera utilisée pour obtenir le moment d'émission du neutron. La figure 3.5 montre, à gauche, la gestion de la concentration des précurseurs et à droite, un schéma équivalent, en nombre de neutrons émis à chaque instant en ignorant la concentration des précurseurs. Le schéma à droite traite seulement la production des neutrons retardés et leurs moments d'apparition dans la simulation.

Afin d'inclure le traitement des neutrons retardés, il faut en premier lieu retrouver leurs paramètres. Le tableau 3.1 liste les différentes variables relatives aux neutrons retardés.

Tableau 3.1 Liste des variables relatives aux groupes de neutrons retardés

Variables	Descriptions
β_k	Proportion des neutrons retardés du groupe k
λ_k	Variable de décroissance des neutrons retardés du groupe k
$\chi_{d,k}$	Spectre d'énergie des neutrons retardés du groupe k

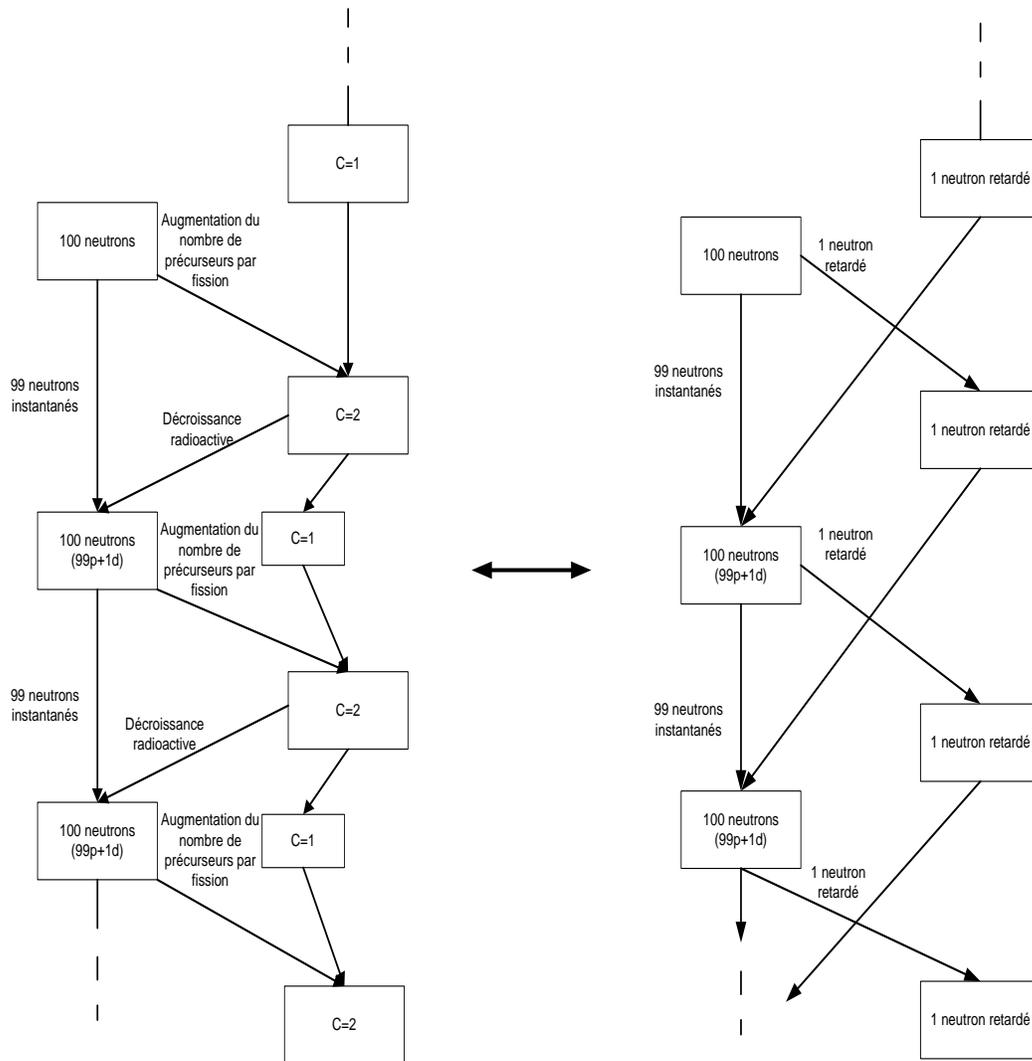


Figure 3.5 Processus simplifié pour la génération des neutrons retardés

Comme première étape, il est nécessaire de localiser ces données. Ces données relatives aux variables β_k et λ_k sont déjà traitées et existantes dans les fichiers des sections efficaces microscopique "ACE" produites par NJOY. Seulement les données relatives aux six groupes de neutrons retardés seront utilisées, les groupes des photos-neutrons ne seront pas pris en considération.

Afin de générer le bon nombre de neutrons retardés, à chaque fission, il faut déterminer le rapport entre le nombre de neutrons retardés et le nombre de neutrons instantanés.

Comme décrit auparavant, la variable β_k est utilisée pour définir le groupe de neutrons retardés auquel appartient le neutron. Les neutrons retardés sont émis avec des énergies gé-

néralement moins importantes que les neutrons instantanés. Pour cette raison, les énergies de ces neutrons atteignent le groupe thermique rapidement. Ces derniers engendreront d'autres fissions plus rapidement que les neutrons instantanés.

Lors d'une fission, un nombre ν de neutrons est émis. Ce dernier a fait le sujet de plusieurs études et recherches. Un modèle, basé sur une distribution gaussienne, a été introduit par Terrel (Terrel, 1957) afin de décrire la distribution des neutrons de fissions. Une implémentation de ce modèle a été bien détaillée dans (Cullen, 2006) et utilisée dans TART et MCNP. Il se présente sous la forme :

$$P(z)dz = \exp\left(\frac{-z^2}{2}\right)dz \quad (3.4)$$

$$z = \nu - \frac{\bar{\nu}}{W}$$

où $\bar{\nu}$ est le nombre moyen de neutrons émis par fission et W est l'écart type de la distribution. Les mêmes bibliothèques ("ACE") contiennent déjà les informations nécessaires pour retrouver la valeur ν pour chaque isotope fissile.

Afin de déterminer le groupe du neutron émis, en sachant que la somme de toutes les proportions de neutrons est normalisée à 1, on tire un nombre aléatoire réel r appartenant à $[0, 1]$ et selon l'intervalle auquel appartient ce tirage, le groupe auquel appartient le neutron est déterminé.

$$\begin{aligned} X5 &= 1 - \beta_6 \\ X4 &= 1 - (\beta_6 + \beta_5) \\ X3 &= 1 - (\beta_6 + \beta_5 + \beta_4) \\ X2 &= 1 - (\beta_6 + \beta_5 + \beta_4 + \beta_3) \\ X1 &= 1 - (\beta_6 + \beta_5 + \beta_4 + \beta_3 + \beta_2) \\ X0 &= 1 - (\beta_6 + \beta_5 + \beta_4 + \beta_3 + \beta_2 + \beta_1) \end{aligned} \quad (3.5)$$

$$r \in \{[0, X0] \cup [X0, X1] \cup [X1, X2] \cup [X2, X3] \cup [X3, X4] \cup [X4, X5] \cup [X5, 1]\}$$

La probabilité d'émettre un neutron retardé du groupe k est défini comme suit :

$$P_p = \beta_k = \frac{\bar{\nu}_{dk}}{\bar{\nu}_T} \quad (3.6)$$

Il est important maintenant de définir le délai d'apparition du neutron en question. La variable λ_k est la constante de décroissance du groupe du neutron. On sait que $N(t) = N_0 \exp(-\lambda_k t)$ détermine le nombre de neutrons qui apparaît à l'instant t appartenant au

groupe k . En normalisant l'aire sous la courbe de cette distribution à 1, le résultat exprimera la fonction de probabilité qui donne la probabilité d'émettre un neutron entre le temps t et le temps $t + dt$. En tirant un nombre aléatoire, r , uniformément distribué entre 0 et 1 et en le considérant égal à la fonction cumulative de cette distribution, la technique d'inversion permet d'obtenir le temps t_1 pour lequel la CDF est égale au nombre aléatoire tiré. Ce temps sera considéré comme le temps d'apparition du neutron :

$$\begin{aligned} r &= \lambda_i e^{-\lambda_i t} \\ D_T &= -\frac{1}{\lambda} \ln\left(\frac{r}{\lambda_i}\right) \end{aligned} \quad (3.7)$$

où λ_i est la constante de décroissance du groupe des neutrons retardés i .

Le code OpenMC utilise des variables continues en énergie. En conséquence, les spectres de neutrons retardés doivent être continus en énergie. Les bibliothèques de données nucléaires de type ENDF fournissent des spectres d'énergie non continus. Selon (Cullen, 2004), une bonne approximation des spectres de neutrons retardés peut être obtenue en utilisant des fonctions Maxwelliennes. Les équations des spectres de neutrons retardés pour l'uranium $U235$ pour chacun des groupes de neutrons retardés sont les suivants (en MeV), illustrés dans la figure 3.6 et tirés de (Cullen, 2004) :

$$\chi_1(E) = 7.34531 \times 10^{-9} \times e^{-3.48634 \times 10^{-6} \times E} \times \sqrt{E} \quad (3.8)$$

$$\chi_2(E) = 6.60177 \times 10^{-9} \times e^{-3.24691 \times 10^{-6} \times E} \times \sqrt{E} \quad (3.9)$$

$$\chi_3(E) = 7.17814 \times 10^{-9} \times e^{-3.43324 \times 10^{-6} \times E} \times \sqrt{E} \quad (3.10)$$

$$\chi_4(E) = 5.08362 \times 10^{-9} \times e^{-2.72780 \times 10^{-6} \times E} \times \sqrt{E} \quad (3.11)$$

$$\chi_5(E) = 5.22314 \times 10^{-9} \times e^{-2.77748 \times 10^{-6} \times E} \times \sqrt{E} \quad (3.12)$$

$$\chi_6(E) = 5.42016 \times 10^{-9} \times e^{-2.84689 \times 10^{-6} \times E} \times \sqrt{E} \quad (3.13)$$

Les distributions des spectres seront utilisées afin de déterminer l'énergie du neutron retardé. Les énergies des neutrons sont obtenues par la technique d'inversion appliquée aux distributions des spectres. Par contre, les bibliothèques ACE ont aussi introduit ces données en se basant sur la règle C64 (Everett et Cashwell, 1983) au lieu de la technique d'inversion.

3.4.2 Introduction des neutrons retardés au début de la simulation

Au début d'une simulation, la population de neutrons doit contenir une grande partie de neutrons instantanés et une petite partie de neutrons retardés. Les neutrons instantanés

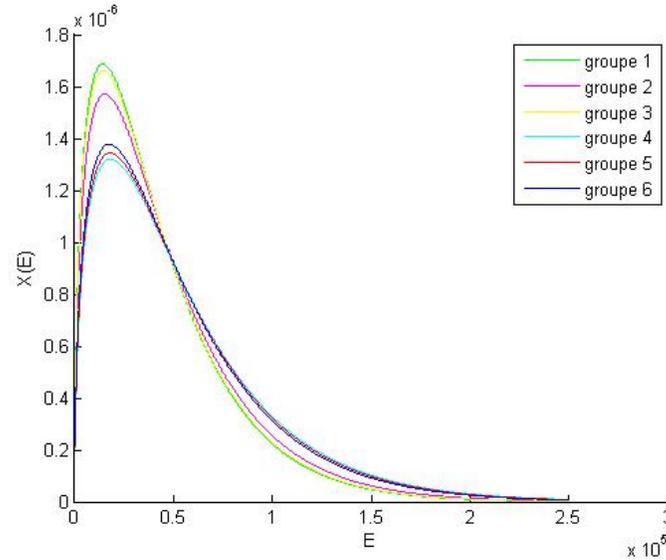


Figure 3.6 Spectres des neutrons retardés

apparaissent immédiatement (hypothèse énoncé lors de l'introduction de l'équation de Boltzmann 2.1), donc, les inclure dès le début de la simulation, à $t = 0s$, est assez simple et résout leur problème.

Par contre les neutrons retardés ont une toute autre logique. Imaginons que la simulation doit démarrer à un certain temps T_0 et que le réacteur était en activité depuis une très longue période avant T_0 . Pour un intervalle de temps Δt se trouvant avant le temps T_0 , les fissions qui auront lieu doivent produire une grande portion de neutrons instantanés ; par contre elle vont produire aussi une portion de neutrons retardés dont certains apparaîtront quelques millièmes de secondes plus tard et d'autres apparaîtront après de longs délais pouvant même dépasser le temps T_0 auquel démarre la simulation. Ainsi, il y aura des neutrons qui sont produits à la suite des fissions qui ont eu lieu à des temps précédant T_0 et qui apparaîtront à des moments ultérieurs à T_0 . Si on considère tous les intervalles de temps précédant la simulation, on doit avoir une quantité de neutrons retardés qui doit apparaître à des moments différents de la simulation et qui sont produits par des fissions qui ont eu lieu durant tous les moments précédant le début de la simulation (voir figure 3.7).

La question est comment quantifier et introduire cette portion de neutrons dans la simulation. En présence des concentrations des précurseurs, cette problématique disparaît vu que chaque précurseur qui disparaît produit un neutron retardé. Malheureusement, les concentrations des précurseurs ne sont pas encore traitées dans le code OpenMC.

En analysant cette situation, il a été remarqué que la population de neutrons retardés qui manque à la simulation est exactement la même que celle qui serait produite à la suite

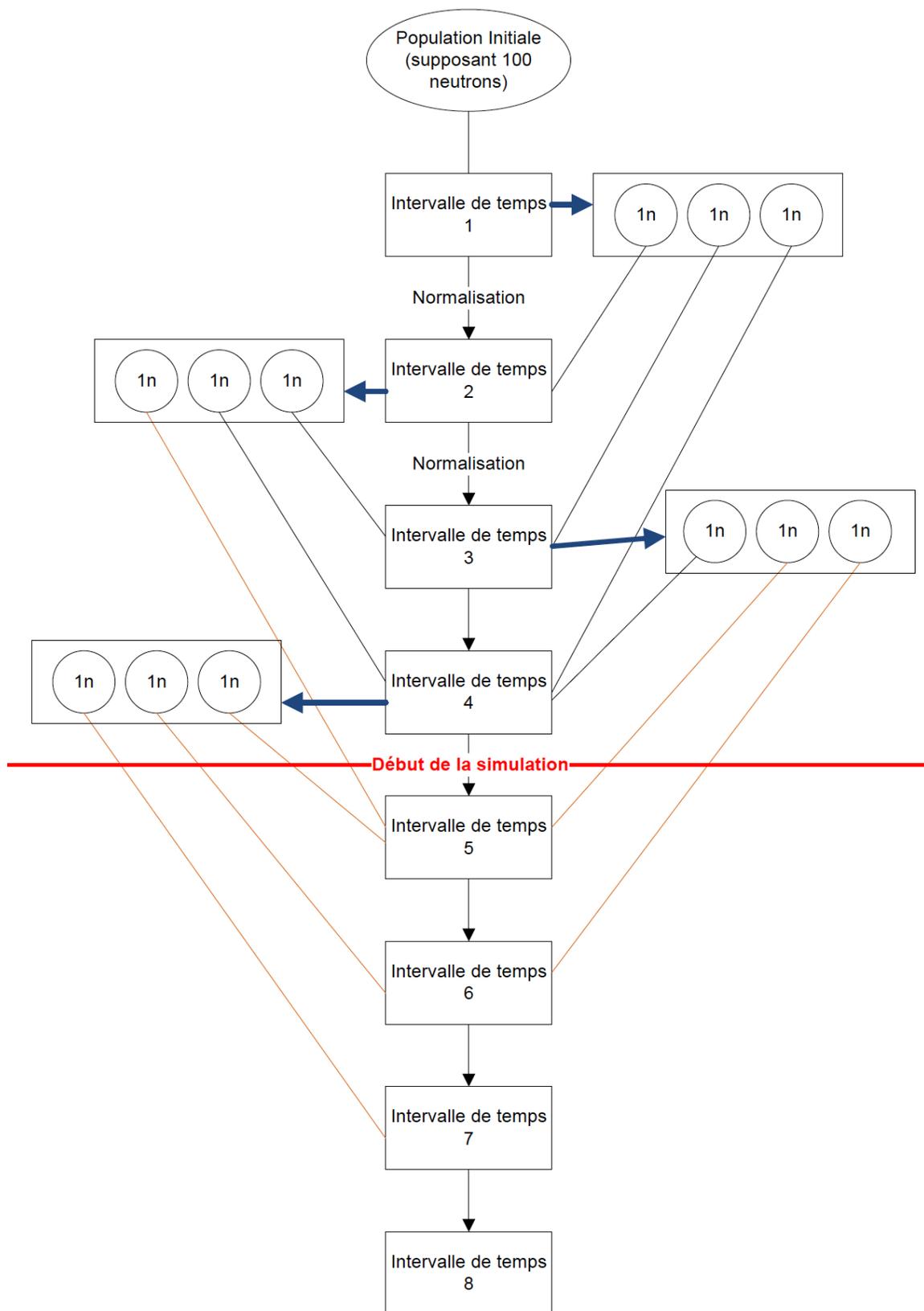


Figure 3.7 Gestion des neutrons retardés au début de la simulation

d'un arrêt du réacteur au temps T_0 . En utilisant cette logique, il a été décidé d'ajouter à la population de neutrons instantanés du début de simulation, une portion de neutrons retardés qui sera quantifiée exactement de la même manière que celle pour prédire la population de neutrons retardés qui sera produite à la suite d'un arrêt brusque du réacteur.

Lors d'un arrêt du réacteur, vu qu'il ne doit plus y avoir de fissions qui produisent des précurseurs, l'équation des précurseurs 2.32 devient :

$$\frac{\partial}{\partial t} C_k(t) = -\lambda_k C_k(t) \quad (3.14)$$

La solution de cette équation donne la concentration des précurseurs :

$$C_k(t) = C_0 e^{-\lambda_k t} \quad (3.15)$$

dont l'évolution temporelle est présentée dans la figure (3.8).

Sachant que pour chaque précurseur qui effectue une décroissance radioactive, un neutron retardé est produit, la concentration des neutrons retardés qui sont produits à la suite d'un arrêt du réacteur est alors :

$$n_{d,k}(t) = -\frac{\partial}{\partial t} C_k(t) = \lambda_k C_k(t)$$

$$\begin{aligned} n_{d,k}(t) &= \lambda_k C_k(t) \\ n_{d,k}(t) &= \lambda_k C_0 e^{-\lambda_k t} \\ n_{d,k}(t) &= n_{d0,k} e^{-\lambda_k t} \end{aligned} \quad (3.16)$$

dont l'évolution est représentée dans la figure (3.9).

La dernière difficulté à traiter est la détermination de la concentration initiale des neutrons retardés, n_{d0} à T_0 . En considérant que le réacteur était en état stationnaire depuis toujours, la proportion des neutrons retardés est alors égale à $n_{d0,k} = \beta_k N_0$ où N_0 est la population de neutrons totale.

$$n_{d,k}(t) = \beta_k N_0 e^{-\lambda_k t} \quad (3.17)$$

Ainsi, il faut rajouter $\int_{\Delta t} \beta_k N_0 e^{-\lambda_k t} dt$ neutrons, pour chaque groupe de neutrons retardés, pour remédier à l'absence des neutrons retardés produits par toutes les fissions qui auront dû avoir lieu avant le début de la simulation. Il est important de noter que cette population de neutrons retardés est ajoutée à la population initiale sauf que ces neutrons apparaîtront

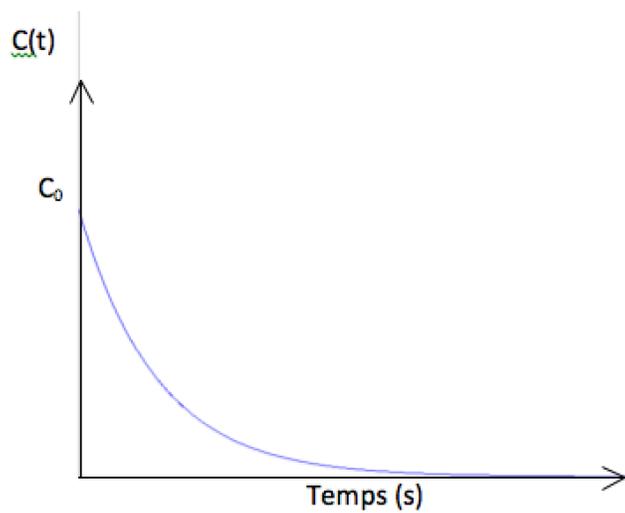


Figure 3.8 Évolution de la concentration des précurseurs à la suite d'un arrêt du réacteur à $t = 0s$

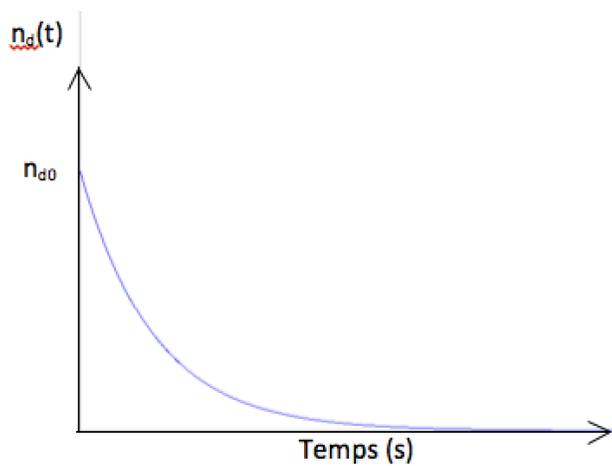


Figure 3.9 Évolution de la concentration des neutrons retardés produits à la suite d'un arrêt du réacteur à $t = 0s$

à des instants ultérieurs à t_0 tout en respectant la distribution décrite par l'équation (3.17).

La distribution spatiale utilisée pour ces neutrons retardés est la même que celle obtenue à la suite d'un calcul de transport Monte Carlo statique. Il est à noter que les neutrons ne peuvent être produits qu'en présence de matière fissile. Ainsi, les figures 3.10 et 3.11 présentent la distribution spatiale de ces neutrons dans les crayons d'Uranium pour une grappe du réacteur CANDU6.

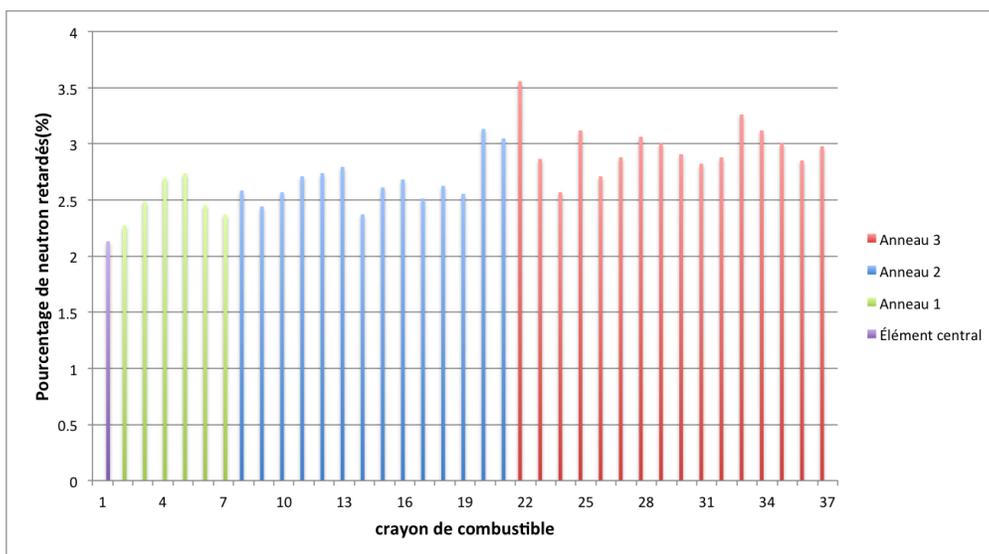


Figure 3.10 Distribution spatiale des neutrons retardés introduits par crayon d'Uranium

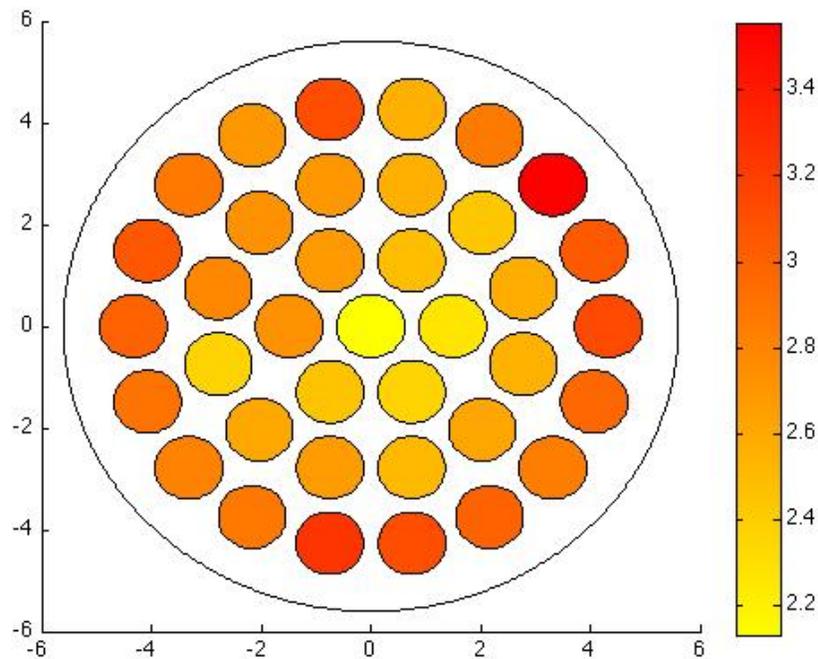


Figure 3.11 Distribution spatiale des neutrons retardés introduits par crayon d'Uranium représentée sur une grappe CANDU-6

3.5 Calcul des paramètres nucléaires dépendants du temps pour la cellule homogène

Les résultats dépendant du temps ne doivent pas être des fonctions continues en temps vu qu'ils seront, par la suite, utilisés dans un calcul de coeur de réacteur dépendant du temps où les méthodes d'Euler sont généralement utilisées pour résoudre l'équation de diffusion pour le temps. L'approche utilisée dans OpenMC-TD respecte l'aspect continu des parcours des particules et aucune approximation n'est introduite dans ce sens. Par contre, il est important d'introduire la notion d'*intervalle de récolte*. Le temps de simulation est divisé en un ensemble de petits intervalles de temps qu'on appelle **intervalle de récolte de résultats** (figure 3.12). L'idée est d'obtenir les variables nucléaires de la cellule homogénéisée et condensée relative à chaque intervalle de récolte.

3.5.1 Calcul des sections efficaces condensées et homogénéisées

Les calculs Monte Carlo sont basés sur une approche stochastique. Les résultats de ces simulations sont des valeurs moyennes pour un ensemble de résultats obtenues pour toutes

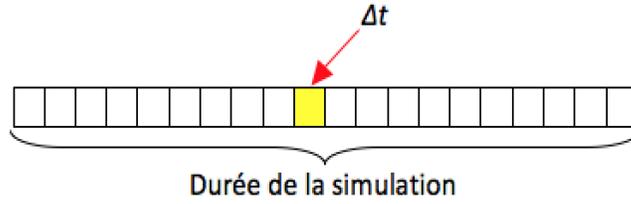


Figure 3.12 Décomposition du temps de simulation en intervalles de récoltes

les itérations. Ces résultats sont obtenus de deux manières différentes dans le code OpenMC : soit par la méthode directe, dite *analogue*, soit d'une manière indirecte, dite *track-length*. Les calculs de cellules en transport ont pour objectifs de récolter des sections efficaces macroscopiques homogénéisées et condensées pour toute la cellule afin de les introduire dans un calcul de diffusion. La technique d'homogénéisation, comme présentée dans la section (2.1.3), se base sur le principe de conservation du taux de réaction :

$$R = \int_t \int_V \int_E F(\vec{r}, E, t) \phi(\vec{r}, E, t) dE dr dt = F_{\vec{r}, E, T} \phi_{\vec{r}, E, t} \quad (3.18)$$

où $F(\vec{r}, E, T)$ représente la réaction visée par les calculs.

Dans la méthode analogue, cette technique se base sur un compteur, C , qui comptera le nombre de collisions (ou réactions) pour chaque région m , dans chaque macro groupe d'énergie h et à chaque intervalle de récolte δt . Ce compteur est équivalent au taux de réaction R .

$$R = C = F_{\vec{r}, g, t} \phi_{\vec{r}, g, t} \quad (3.19)$$

$$\begin{aligned} R_{\Delta t_i} &= \sum_{c \in \Delta t_i} 1 \\ R_{\Delta t_i} = \Sigma_{t, \Delta t_i} \phi_{\Delta t_i} &\iff \phi_{\Delta t_i} = \frac{R_{\Delta t_i}}{\Sigma_{t, \Delta t_i}} \\ \phi_{\Delta t_i} = \sum_{c \in \Delta t_i} \frac{1}{\Sigma_t} &\iff R_{x, \Delta t_i} = \sum_{i \in \Delta t_i} \frac{\Sigma_{x, \Delta t_i}}{\Sigma_{t, \Delta t_i}} \end{aligned} \quad (3.20)$$

Cette technique est dite implicite du fait qu'on doit calculer le taux de réaction en premier lieu pour obtenir ensuite les sections efficaces ainsi que le flux neutronique. Le calcul des sections efficaces en utilisant l'équation (3.20) nécessite la procédure de pré-calcul du flux neutronique intégré $\phi(\vec{r}, E, t)$. Le flux neutronique intégré n'est autre que la distance parcourue par l'ensemble des neutrons appartenant à une unité de volume dans l'intervalle d'une

unité de temps. De ce fait, le flux peut être estimé d'une manière analogue en enregistrant les parcours de tous les neutrons. Il s'agit de l'approche *track-length*. Il est aussi important de prendre soin de bien identifier les intervalles de temps auxquels appartient chaque portion de parcours afin de bien calculer le flux qui lui correspond.

$$\Delta t_i V \phi = \int dr \int dE \int d\Omega \int_{t_i}^{t_{i+1}} dt \Psi(r, \vec{\Omega}, E, t)$$

$$\Delta t_i V \phi = \int dr \int dE \int d\Omega \int_{t_i}^{t_{i+1}} dt v n(r, \vec{\Omega}, E, t)$$

$$\Delta t_i V \phi = \int dr \int dE \int_{l \in \Delta t_i} dl N(r, E, t)$$

$$\phi_{\Delta t_i} = \sum_{i \in \Delta t_i} l_i \quad (3.21)$$

$$R_{x, \Delta t_i} = \sum_{i \in \Delta t_i} l_i \Sigma_{x, \Delta t_i} \quad (3.22)$$

Il est primordial de noter que les taux de réaction définis par l'équation (3.19) dépendent principalement de la densité de la population de neutrons initiale définie pour chaque itération. Plusieurs procédures de normalisation sont définies pour contourner cette dépendance (Leppänen, 2007).

Afin d'obtenir la valeur du coefficient de diffusion, il faut calculer la section efficace de transport pour toute la cellule,

$$\Sigma_{tr, g} = \Sigma_{t, g} - \Sigma_{s, g} \bar{\mu}_g \quad (3.23)$$

$$\Sigma_{tr, g} = \frac{1}{3D_g} \quad (3.24)$$

$$\bar{\mu}_g = \frac{\Sigma_{t, g} - \Sigma_{tr, g}}{\Sigma_{s, g}} \quad (3.25)$$

3.5.2 Calcul de la vitesse moyenne des neutrons

Il est important de noter que l'équation de diffusion neutronique dépendante du temps (2.31) comprend le facteur $\frac{1}{v}$. Ce dernier est absent dans l'équation statique, et donc les codes de transport stochastique ne le calculent pas. Dans ce projet, il est important de trouver une solution afin de calculer ce paramètre pour chaque groupe d'énergie et homogénéisée sur toute la cellule.

Sachant que le flux neutronique est défini par :

$$\phi_g(r, \Omega, t) = n_g(r, \vec{\Omega}, t) \bar{v}(t) \quad (3.26)$$

Il est possible d'obtenir la vitesse moyenne des neutrons dans chaque groupe d'énergie à condition d'être en mesure de calculer la densité neutronique $n_g(r, \vec{\Omega}, t)$. Ainsi :

$$\bar{v}(t) = \frac{\phi_g(r, \Omega, t)}{n_g(r, \vec{\Omega}, t)} \quad (3.27)$$

La densité neutronique condensée par groupe et homogénéisée sur toute la cellule pour chaque intervalle de récolte est calculée comme suit :

$$\begin{aligned} N_{g, \Delta t_i} &= \int dr \int dE \int d\Omega \int_{t_i}^{t_{i+1}} dt n(r, \vec{\Omega}, E, t) \\ N_{g, \Delta t_i} &= \int dr \int dE \int_{l \in \Delta t_i} n(r, E, t) \end{aligned} \quad (3.28)$$

Le terme obtenu, N_g , représente le temps total vécu par tous les neutrons qui ont appartenu à l'intervalle de récolte i et au volume V . Il suffit donc d'additionner pour chaque intervalle de récolte les temps de vies de tous les neutrons dans cet intervalle ayant vécu dans le volume m .

Afin de calculer la densité neutronique moyenne, il suffit de diviser le temps de vie total des neutrons par la durée d'un intervalle de récolte Δt_i :

$$\begin{aligned} \bar{n}_{g, \Delta t_i} &= \frac{1}{\Delta t_i} \int dr \int dE \int d\Omega \int_{t_i}^{t_{i+1}} dt n(r, \vec{\Omega}, E, t) \\ \bar{n}_{g, \Delta t_i} &= \frac{1}{\Delta t_i} N_g \end{aligned} \quad (3.29)$$

et donc :

$$\bar{v}_{\Delta t_i} = \frac{\phi_g, \Delta t_i}{\bar{n}_g, \Delta t_i} \quad (3.30)$$

CHAPITRE 4 MODIFICATIONS APPORTÉES AU CODE OPENMC

Afin d'introduire le nouveau mode de simulation, de nombreuses modifications ont été apportées aux fonctions responsables de suivre l'évolution des particules dans la cellule. Désormais, chaque particule évolue non seulement en géométrie (évolution de la position) et en vitesse (variation en énergie), mais aussi en temps. Nous pouvons classer les modifications apportées à OpenMC en trois principales catégories :

- modifications apportées aux fichiers d'entrées XML ;
- modifications apportées au code source (Fortran) ;
- modifications apportées aux procédures de récoltes des données.

4.1 Modifications apportées aux fichiers d'entrées XML

Afin d'effectuer un calcul neutronique avec OpenMC, il est impératif de fournir principalement quatre fichiers d'entrées du type XML et portant les noms suivants : *settings.xml*, *tallies.xml*, *materials.xml* et *geometry.xml*.

Deux autres fichiers complémentaires peuvent être fournis et portent les noms : *plots.xml* et *cmfd.xml*.

Afin de parvenir à la réalisation d'une simulation dépendante du temps, des modifications aux fichiers d'entrée sont nécessaires. Ces modifications touchent principalement trois des quatre principaux fichiers d'entrées.

La première modification a été apportée au fichier *settings.xml*. Ce dernier encapsule les informations générales relatives à la simulation (comme l'adresse de la base de données des sections efficaces, le type de source des neutrons utilisée, la définition des intervalles d'énergie, le type de la simulation, etc...).

Dans les types de simulations proposés, nous avons ajouté deux nouvelles options, la première s'intitule : *time_dependent* dans laquelle l'utilisateur devra entrer les données suivantes afin d'initier un calcul dépendant du temps :

```
<time_dependent>
  <batches>5</batches>
  <delayed>0</delayed>
  <tally>1</tally>
  <particles>2</particles>
  <time_duration>0.5</time_duration>
  <time_dt>0.01</time_dt>
```

```
</time_dependent>
```

La deuxième option s'intitule *analysing_time_dependent* où l'utilisateur devra entrer les données suivantes afin d'initier une analyse des résultats :

```
<analysing_time_dependent>
  <path>/pincell-td/tracking_file.binary</path>
  <time_duration>1</time_duration>
  <time_dt>0.01</time_dt>
</analysing_time_dependent >
```

Ce mode d'analyse des données recueillies sera expliqué plus en détail dans la section 4.3.3.

La deuxième modification apportée aux fichiers d'entrées concerne l'ajout d'une nouvelle option pour les géométries. Cette dernière permet la modification des sections efficaces des matériaux constituant la géométrie. Cette modification touche la partie qui associe chaque composante de la géométrie aux propriétés nucléaires du matériau qui la constitue. L'objectif de cette modification est de simuler un accident dans un réacteur nucléaire. Lors d'un accident de type «perte du caloporteur» dans un réacteur de type CANDU-6, ce dernier garde la même géométrie physique sauf que la composition du matériel qui le définit passe de l'eau lourde au vide.

Les modifications qui ont été effectuées au niveau du logiciel permettent de définir des cellules avec de multiples matériaux. Chaque matériau constitue un élément de base de la cellule pendant un intervalle de temps durant la simulation. Pour ce faire, nous avons modifié la définition d'une cellule dans le logiciel OpenMC. Auparavant, une cellule était définie par les paramètres suivants :

- "id" de type entier.
- "universe" de type entier avec une valeur par default "0"
- "material" de type chaîne de caractère de longueur maximale "12"
- "fill" de type entier avec une valeur par default "0"
- "surfaces" de type vecteur d'entiers
- "rotation" de type vecteur de réels
- "translation" de type vecteur de réels

Afin de lui donner la possibilité d'alterner entre deux matériaux (ou plus), il était nécessaire de modifier le paramètre *material* pour le rendre capable de lire plusieurs matériaux et ce en le déclarant comme un tableau (que l'on note M ici). La deuxième modification à cette structure est l'ajout d'un nouveau paramètre. Ce dernier encapsule les moments auxquels les modifications des matériaux pour la cellule s'effectuent. Il doit aussi avoir la forme d'un

tableau (que l'on note T) afin de permettre d'enregistrer tous les moments des modifications des matériaux. Il est à noter que la longueur du tableau T doit correspondre à celle de M . Durant la simulation, le matériau numéro n du tableau des matériaux, $M(n)$, sera utilisé pour la cellule durant l'intervalle $[T(n-1), T(n)]$. Le premier matériau du tableau, $M(1)$, sera appliqué durant l'intervalle de temps $[0, T(1)]$. Il est à noter que le dernier élément du tableau des moments doit être égal à la durée totale de la simulation définie dans la structure *time_dependent* du fichier *settings.xml*.

La nouvelle structure prend la forme suivante :

- "id" de type entier.
- "universe" de type entier avec une valeur par default "0"
- "times" de type un vecteur de réels
- "materials" de type vecteur de chaînes de caractères dont chacune de longueur maximale "12"
- "fill" de type entier avec une valeur par default "0"
- "surfaces" de type vecteur d'entiers
- "rotation" de type vecteur de réels
- "translation" de type vecteur de réels

Note : cette même structure sera utilisée dans les autres modes de simulations. Dans le mode *Eigen_value*, l'utilisateur introduit une seule valeur pour *materials* et ignore le paramètre temps.

Ci-dessous un exemple pour une cellule qui évolue dans le temps (1 seconde) en utilisant deux matériaux différents (3 et 4). Le matériau identifié par le chiffre 3 est utilisé durant l'intervalle $[0.0s, 0.5s]$ et le matériau identifié par le chiffre 4 est utilisé durant l'intervalle de temps $]0.5s, 1.0s]$.

```
<cell id="4" times="0.5 1.0" materials="4 3" surfaces="3 4 -5 6 -7" />
<!-- commentaire: 4 = Eau , 3 = Vide -->
```

Pour la procédure de récolte de résultats, les modifications apportées aux différentes structures de données définies dans le fichier *tallies.xml* seront présentées dans la section (4.3).

4.2 Modifications apportées aux codes sources

Les modifications apportées au code source sont nombreuses et touchent à une grande partie du logiciel. OpenMC se compose de six principales parties :

1. saisie des données à partir des fichiers XML,
2. initialisation de la simulation,
3. création de la géométrie,
4. lecture des bibliothèques,
5. création des compteurs, et finalement
6. module de transport (qui effectue le tracking).

La lecture des bibliothèques des sections efficaces a été modifiée afin de prendre en considération les différents paramètres des neutrons retardés tout en se basant sur la documentation fournie pour les fichiers ACE (module 3 du guide d'utilisation de MCNP).

Afin de pouvoir lire les nouvelles balises XML présentées dans la section précédente, des modifications au niveau de la saisie des fichiers XML ont été effectuées et de nouvelles structures et variables ont été rajoutées. La principale modification est l'ajout de deux nouvelles structures pour définir les modes `TIME_DEPENDENT` et `ANALYSING_DATA_TD`.

4.2.1 Modifications apportées au module d'initialisation

Au niveau de l'initialisation de la simulation, plusieurs modifications ont été apportées. D'abord, une nouvelle structure de données définissant le type des particules en mode «time_dependent» a été créée. Comme le nombre de particules initiales, défini par l'utilisateur dans les fichiers *settings.xml*, n'est autre que celui du début de chaque itération (à $t=0$ seconde), ces particules doivent évoluer pendant une certaine durée. En se référant au facteur de multiplication effectif, k_{eff} , le nombre total de particules subit une augmentation exponentielle si ce dernier est supérieur à un, alors que s'il est inférieur à un, le nombre de neutrons chutera de même manière.

Ce phénomène implique que le nombre total des particules ne peut pas être défini. Dans ce cadre, la structure de stockage des données relative aux particules ne peut pas être un tableau (à cause de sa limite de taille invariable et les coûts engendrés durant la procédure de copie des éléments dans un nouveau tableau de taille plus grande). La solution trouvée est l'utilisation d'une **liste chaînée** qui, elle, n'a pas de restriction de taille. Pour y arriver, nous avons défini un nouveau type de données s'appelant «Bank_TD». Ce dernier se différencie du type BANK utilisé dans les simulations à valeur propre par :

- la suppression de la variable *wgt* qui définit le poids de chaque particule,

- l’ajout d’une variable *time_appear* qui dicte le moment d’apparition de la particule,
- l’ajout de deux pointeurs qui pointent vers l’élément précédent et l’élément suivant de la liste chaînée, et
- la conservation des éléments qui décrivent la position et l’énergie de la particule.

4.2.2 Modifications apportées au module de création de la géométrie

Comme expliqué auparavant, afin d’effectuer des simulations dépendantes du temps, nous avons besoin de définir plusieurs matériaux pour une même partie de la géométrie de la cellule. La première modification liée à cette exigence était de remplacer l’attribut *material* par un tableau de matériaux au niveau de la structure *cell* définissant la cellule. L’ajout d’un nouveau tableau de réels constitue la seconde modification pour la structure. Ce dernier enregistre les différents temps auxquels s’appliquent les modifications des propriétés nucléaires de la cellule. Ainsi, il était nécessaire que la fonction d’initialisation de la géométrie dans le module *input_xml* soit modifiée.

Durant la simulation d’un accident, il est important d’utiliser les bonnes sections efficaces aux instants correspondant de la simulation. Afin d’y parvenir, d’autres modifications importantes ont été réalisées. Chaque particule effectue un ensemble de parcours et de collisions avant de disparaître. Après chaque parcours, les paramètres de la particule sont mis à jour (sa position, direction, énergie, etc.). Les propriétés du milieu dans lequel se trouve la particule, eux aussi varient constamment après chaque parcours et/ou collision. D’où, un ensemble d’étapes sont réalisées, dans l’ordre, afin de déterminer correctement les propriétés du milieu après chaque collision.

Premièrement, la distance que la particule doit parcourir dans une direction déterminée est échantillonnée. Ensuite, la nouvelle position de la particule est calculée. Par la suite, une fonction est appelée afin de fournir l’identificateur de la cellule dans laquelle se trouve la particule. L’étape suivante est d’obtenir les nouvelles propriétés nucléaires du milieu en fonction de l’énergie de la particule. Ces données sont enregistrées dans une structure appelée «*micro_xs*». Finalement, il ne reste qu’à définir le type de la réaction que la particule doit subir. Cette dernière étape est effectuée en faisant appel au module *transport*.

Ce cheminement élaboré définit toutes les étapes nécessaires dans les simulations de type Monte Carlo (à valeur propre ou dépendante du temps). Une nouvelle sous-étape doit être prise en considération afin de simuler un accident. Il est clair que la modification qui doit être apportée se situe dans l’étape de définition des propriétés de la cellule. Ainsi, un test *y* a été rajouté. Ce dernier situe le moment de la collision de la particule avec l’atome dans l’intervalle de temps du tableau temporel de la cellule.

Chaque cellule possède un tableau contenant les moments des « accidents » et donc les

moments de changement des propriétés de la cellule. Comme chaque intervalle est associé à un matériau différent pour remplir chaque cellule de la géométrie, le matériau approprié est utilisé et ainsi les propriétés nucléaires adéquates sont enregistrées dans la variable *micro_xs*.

Cette étape seule n'est pas suffisante, une autre étape est nécessaire afin de définir les moments juste avant et juste après les accidents.

Au départ, nous avons une position de la particule, ensuite on fait appel à l'échantillonnage pour définir la distance que le neutron peut parcourir avant sa prochaine collision. Par la suite, on vérifie si le neutron devra passer à travers une surface durant son parcours. Si c'est le cas, la distance qu'il peut parcourir est diminuée et devient la distance qui le sépare de la surface. Ainsi la nouvelle position d'arrivée du neutron devient la position de l'intersection de la ligne de la trajectoire avec la surface. L'étape suivante est de vérifier si la durée de temps nécessaire pour atteindre la nouvelle position est inférieure à la durée de temps restante avant le déclenchement du prochain accident. Si cette observation s'avère vraie, une nouvelle position d'arrivée est obtenue en considérant seulement la distance que le neutron peut parcourir avant l'accident.

Il est important de noter que l'implémentation effectuée ne prend pas en considération la fonction «*fill*» et la fonction «*Lattice*».

4.2.3 Modifications apportées au module de transport

Afin de rendre le code OpenMC capable d'effectuer des calculs dépendants du temps, les principales modifications se situent au niveau du module de transport. Ces modifications couvrent les points suivants :

- la modification de la boucle de traitement des particules,
- la modification du module «*physics*» qui gère les collisions des neutrons et toutes les réactions qu'un neutron effectue durant son parcours,
- la modification du système de sauvegarde des résultats,
- la modification des procédures de création des produits de fission,
- l'ajout de la prise en compte des neutrons retardés et leurs temps d'apparitions, et
- l'ajout de la gestion de la durée de vie des neutrons.

Les deux principales boucles d'évolution des particules dans ce module sont représentées dans la figure (4.1).

Il est important de noter que ces deux boucles se situent à l'intérieur de la boucle des *itérations*. Chaque itération représente une évolution dépendante du temps de la simulation en question durant toute la durée prédéfinie. À la fin de chacune, les résultats sont enregistrés dans un tableau. La moyenne ainsi que l'écart type de tous les résultats sont calculés vers la fin de la simulation.

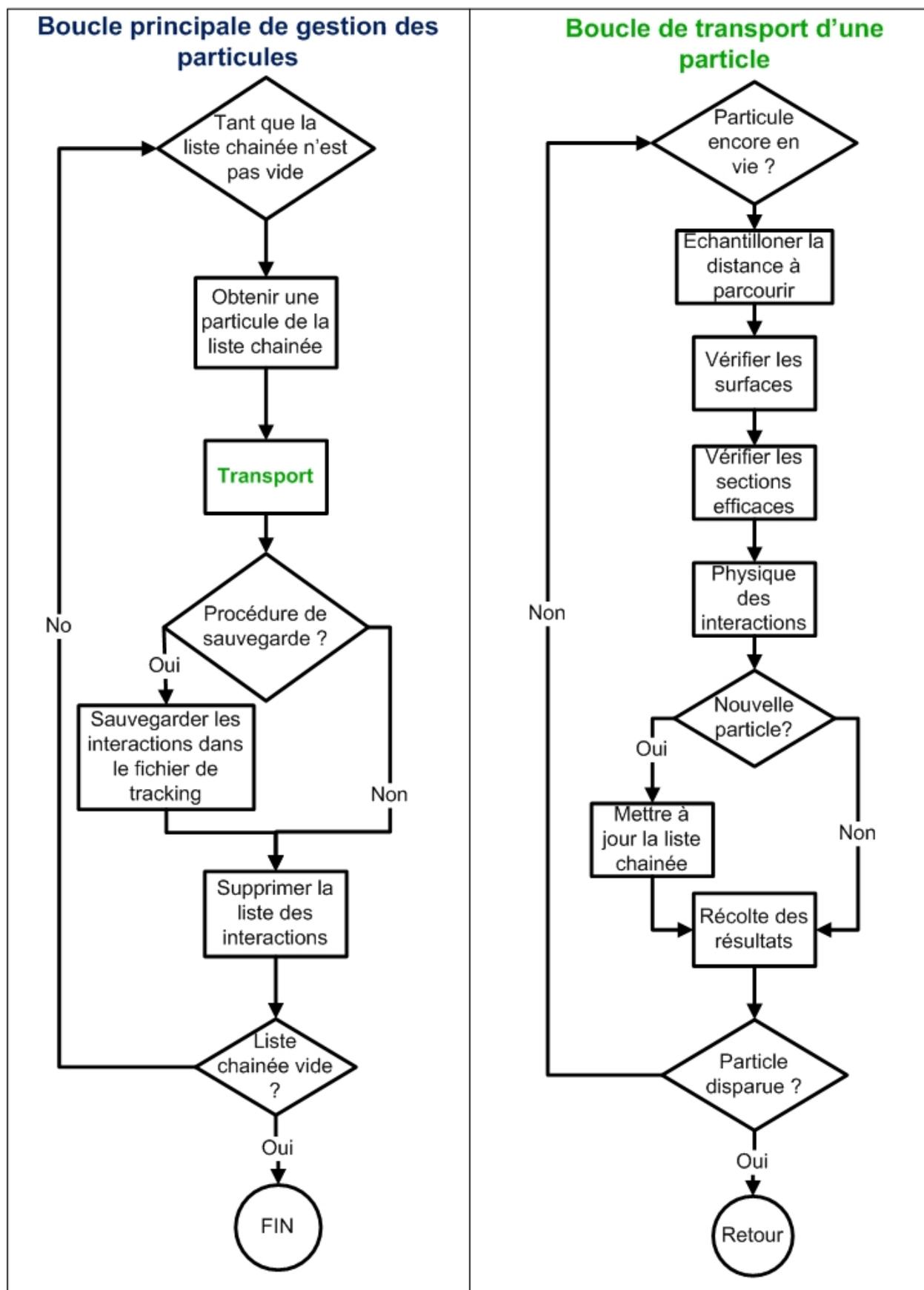


Figure 4.1 Boucles de transport neutronique dans OpenMC-TD

4.3 Modifications apportées à la procédure de décompte des résultats

Avant tout, l'implémentation réalisée pour effectuer la procédure de décompte des résultats dans une simulation dépendante du temps n'est pas opérationnelle avec les fonctions optionnelles suivantes :

- «mesh»
- «CMFD»
- «Survival_biaising»

Toutes ces fonctions ne sont pas importantes pour la réalisation d'une simulation dépendante du temps et le tri de ses résultats. Il est à noter que le fait que certaines fonctions ne soient pas utilisées est un choix fondamental d'implémentation et de projet (comme le «Survival biaising»). D'autres fonctions n'ont pas été considérées à cause de leurs ordres de priorités dans le cadre de ce projet. Il est clair que si on désire que le logiciel devienne un outil incontournable pour les simulations dépendantes du temps, toutes ces fonctions devraient être opérationnelles.

La procédure de décompte des résultats a été développée de telle sorte que l'utilisateur aura le choix entre deux options, dépendamment de la capacité de son support matériel. Dans OpenMC-TD, on trouve :

- La première option, dite *Tally_Inside*, effectue le décompte des résultats au fur et à mesure que la simulation roule. Autrement dit, la simulation commence par l'initialisation des vecteurs de résultats et à chaque parcours de chaque neutron, la fonction d'enregistrement des résultats est appelée. Cette option permet de fournir les résultats de la simulation instantanément et ne requiert aucune sauvegarde de données. Pour effectuer le *Tally_Inside*, le fichier «*tallies.xml*» doit exister et fournir les structures de «tally» et les fonctions résultats souhaitées.
- La deuxième option, dite *Tally_Later*, enregistre tous les parcours des neutrons ainsi que leurs paramètres dans un fichier binaire. Ce fichier est utilisé ensuite par la simulation pour effectuer la procédure de décompte des résultats.

Chaque méthode possède ses avantages ainsi que ses inconvénients :

Tally_inside :

- Avantages :
 - Affichage des résultats immédiatement après la simulation.
 - Utilisation d'un espace **mémoire disque** minimal pour enregistrer seulement le fichier des résultats (quelques Kb).
 - Fonctionne en mode MPI.
- Inconvénient :
 - Nécessite une **mémoire vive** très grande pour la liste chaînée des particules, en plus des vecteurs des résultats.
 - Simulation légèrement plus lente à cause des étapes des décomptes.
 - Obligation de redémarrer la simulation à chaque modification des fonctions résultats souhaitées.

Tally_Later :

- Avantages :
 - Nécessite une **mémoire vive** moins importante que celle pour la méthode *Tally_Inside* vu que les vecteurs des résultats ne sont pas alloués durant la simulation.
 - Fonctionne en mode MPI.
 - Simulation plus rapide grâce à l'absence de l'étape de décompte.
 - Lors de la modification des fonctions résultats souhaitées, il suffit de relancer la procédure de décompte avec le même fichier de tracking.
 - Flexibilité des données à enregistrer dans le fichier de sortie.
- Inconvénient :
 - Affichage des résultats seulement après le lancement de la procédure de décompte.
 - Utilisation d'un espace mémoire disque très grand pour enregistrer les interactions de chaque particule.
 - Obligation de relancer la simulation dans le cas d'une erreur de la part de l'utilisateur au niveau du choix des données à enregistrer pour le fichier de tracking.

4.3.1 Modification dans le code source de la procédure de décompte

Le développement de cette partie du code a été effectué en trois phases. La première phase consiste à décortiquer et analyser la procédure de décompte initialement implémentée dans le code OpenMC pour le mode statique.

La deuxième est l'adaptation de la procédure de décompte existante pour fonctionner durant un mode dépendant du temps. Comme présenté à la section (3.5), les résultats obtenus sont différents pour chaque intervalle de temps de décompte. Ainsi, des modifications dans la

création des vecteurs des résultats sont nécessaires pour prendre en considération la nouvelle dimension temporelle. Le nouveau vecteur des résultats est défini comme suit :

Nouveau vecteur: `tally % results (time_index, score_index, filter_index)`

L'ajout de cette nouvelle dimension a été accompagné de l'ajout de nouvelles fonctions ayant pour objectif d'identifier l'intervalle de temps auquel appartient chaque évènement afin de bien comptabiliser les différents taux de réactions correctement.

Il est important, à cette étape, de mettre l'accent sur la difficulté qui s'est présentée en utilisant la méthode de décompte *Track-Length*. En utilisant cette méthode, la somme des longueurs des parcours des neutrons constitue le flux neutronique. Les différents taux de réaction qui sont comptabilisés dans le vecteur des résultats sont obtenus en multipliant la longueur des distances parcourues par les sections microscopiques du milieu en question. Il est judicieux de mentionner que chaque parcours de neutron ne peut appartenir qu'à un seul milieu (vu que la méthode de *Woodcock* n'est pas utilisée). Par contre, cette même distance parcourue dans le temps peut appartenir à deux (ou plus) intervalles de temps différents. Il était alors nécessaire de créer une nouvelle fonction qui a pour rôle de découper ces distances en fonction de leurs appartenances aux intervalles de décompte.

Finalement, la troisième phase consiste à implémenter toute la procédure de décompte en se basant sur le fichier de tracking des neutrons. Cette étape a été très laborieuse, car il fallait définir tout un nouveau mode de fonctionnement dans l'application, créer les vecteurs des résultats, lire les interactions enregistrées pour chaque neutron séparément, créer une nouvelle boucle d'évolution de particules selon les données lues dans le fichier, et finalement effectuer la procédure du décompte en utilisant toutes les fonctions d'indentification des intervalles de décompte présentées auparavant dans cette section.

4.3.2 Modification dans les fichiers d'entrées XML

Avant d'expliquer les nouvelles structures d'entrées, il est important de revoir la structure *time_dependent* définie plutôt et expliquer le rôle de chacun de ses paramètres.

- Le premier attribut s'appelle `<batches>`. Il joue le même rôle que celui dans la structure *eigenvalue*. Il définit le nombre de fois que toute la simulation sera répétée afin d'optimiser les résultats et de diminuer l'écart type.
- Le deuxième attribut de cette structure est appelé `<delayed>`. Il précise si l'application devrait prendre en considération les neutrons retardés (`delayed = 1`), ou bien les ignorer.
- Le troisième attribut s'appelle `<tally>`. Ce dernier indique que la procédure de décompte sera effectuée selon le processus *tally_inside* si sa valeur est égale à 1, sinon

la procédure *tally_later* sera utilisée et ainsi un fichier de tracking sera produit dans le même répertoire que celui des fichiers d'entrées XML.

- Le paramètre suivant, *<particles>*, définit le nombre de particules initial avec lequel débute la simulation.
- Le paramètre *<time_duration>* définit la durée totale de la simulation.
- Finalement, le dernier attribut de cette structure s'appelle *<time_dt>*, il indique le pas de temps qu'il faut utiliser dans la procédure de décompte si la méthode *tally_inside* est appelée.

Une nouvelle structure dans le fichier "tallies.xml" a été définie. Cette dernière a pour objectif de définir les attributs des particules qui seront enregistrés dans le fichier de tracking. Elle possède la forme suivante :

```
<tally_file_td>
  <tally_energy> 1 </tally_energy>
  <tally_x> 1 </tally_x>
  <tally_y> 1 </tally_y>
  <tally_z> 1 </tally_z >
  <tally_time> 1 </tally_time>
  <tally_mu> 1 </tally_mu>
  <tally_event_mt> 1 </tally_event_mt>
  <tally_event_nuclide> 1 </tally_event_nuclide>
</tally_file_td>
```

Chaque élément de cette structure peut avoir soit la valeur un, soit zéro. Ces valeurs indiquent si la variable doit être enregistrée ou non, dans le fichier de tracking. En l'absence de cette structure, seulement les cinq premiers éléments seront enregistrés (l'énergie de la particule, la position de la particule (x,y,z), et le moment de la collision). La liberté de choisir les éléments qui doivent être enregistrés permet à l'utilisateur de limiter la quantité de données à stocker dans le cas où il cherche à obtenir quelques résultats seulement.

La modification suivante au niveau des fichiers d'entrée est l'ajout d'un nouveau mode de simulation dans le fichier *settings.xml*. Cette structure sert à démarrer le processus de décompte des résultats à partir d'un fichier de tracking. Elle contient l'adresse du fichier de tracking, la durée totale de la simulation, et le pas de temps à utiliser pour la procédure de décompte.

```
<analysing_time_dependent>
  <path>./tracking_file.binary</path>
  <time_duration>1</time_duration>
```

```
<time_dt>0.01</time_dt>
</analysing_time_dependent >
```

Des tests de vérifications ont été effectués afin de valider la nouvelle procédure. Une population de neutrons, dont les interactions sont définies préalablement, a été utilisée dans les deux types de décompte afin de s'assurer de la validité des résultats.

4.3.3 Création du fichier de tracking

Le fichier de tracking est un fichier binaire. Le choix d'utiliser un fichier binaire au lieu d'un fichier texte ou bien un fichier XML est basé sur la rapidité d'écriture/lecture et l'espace disque limité que ce type occupe.

Le fichier de tracking contient les informations relatives à tous les neutrons qui ont été suivis durant la simulation. Pour chaque neutron et après chaque parcours effectué, une procédure enregistre toutes informations relatives à ce dernier dans une liste chaînée appelée *Interaction*. Ces structures sont ensuite transcrites dans le fichier binaire.

Dans le but de minimiser la quantité d'informations enregistrée dans le fichier, le choix a été d'écrire les données dans un ordre bien déterminé et éviter les indicateurs. Cet ordre est le même que celui de la structure *interaction* présenté auparavant. Autrement dit, à chaque parcours que le neutron effectue, un ensemble de données constituant la structure *interaction* est écrit dans le fichier binaire.

Afin d'identifier la lecture d'un nouveau neutron dans le fichier binaire, une première interaction est enregistrée dont tous ses éléments sont (-1). Une autre interaction dont tous ses éléments sont (-9) est utilisée pour spécifier la fin des interactions d'un neutron.

Toujours dans le but d'optimiser l'utilisation de la mémoire, il a été décidé de ne pas rajouter une variable dans la structure *Interaction* pour spécifier le type de l'interaction, mais plutôt de rajouter une interaction complète lors des cas particuliers qui sont : absorption, fission ou (n, Xn) .

Les interactions spéciales contiendront les informations suivantes :

- $E = 0$
- $x =$
 - 0 \Rightarrow capture radiative,
 - 1 \Rightarrow fission,
 - -1 $\Rightarrow (n, Xn)$.
- $y =$
 - 0 si absorption,
 - ν si fission,
 - X si (n, Xn) .

— Le reste des paramètres (z , time, ..) sont tous zéros.

Lorsqu'une réaction mène à l'apparition de nouveaux neutrons, telle que la fission ou la réaction (n, Xn) , une interaction pour chaque nouveau neutron est inscrite afin de spécifier son énergie, sa position, et le moment auquel il apparaît.

Pour résumer, les interactions sont enregistrées lorsque :

- un neutron est né (interaction spéciale initialisée à -1),
- le neutron franchit une surface spatiale ou temporelle,
- le neutron effectue une collision (diffusion, fission, absorption ou (n, Xn)),
- une fission a lieu (interaction spéciale),
- une capture radiative a lieu,
- une collision du type (n, Xn) a lieu,
- un nouveau neutron apparaît, et finalement
- un neutron disparaît (interaction spéciale initialisée à -9).

Dans le tableau 4.1, un exemple du contenu du fichier de tracking est représenté.

Tableau 4.1 Exemple du contenu du fichier de tracking

	Energie (Mev)	X (cm)	Y (cm)	Z (cm)	Temps (s)	Commentaires
Premier	-1	-1	-1	-1	-1	
neutron	9.5	0.3	0.4	-1.5	0	
	4.48	3.5	18.5	10	1.3E-4	
	4.48	8.3	12.25	1.6	2.53E-4	
		...				
	0.48E-3	6.5	11.45	1.18	9.95E-4	
	0	0	0	0	0	Capture radiative
	18	1.3	12.4	-3.2	1.73E-3	Neutron incident
	-9	-9	-9	-9	-9	
Deuxième	-1	-1	-1	-1	-1	
neutron			...			
	0	-1	2	0	0	Fission
	1.83	2.35	15.5	5.2	1.8E-1	PF neutron instantané
	5.3	2.35	15.5	5.2	5.8	PF neutron retardé
	0	2.35	15.5	5.2	1.8E-1	Neutron incident
	-9	-9	-9	-9	-9	
... Jusqu'à la dernière particule						
Dernier	-1	-1	-1	-1	-1	
neutron			...			
	-9	-9	-9	-9	-9	

CHAPITRE 5 MODÈLE DE DIFFUSION AVEC SECTIONS EFFICACES DÉPENDANTES DU TEMPS

5.1 Schéma de calcul espace-temps pour un réacteur CANDU-6 en utilisant le code DONJON

En se basant sur les résultats recueillis à la suite de la simulation Monte Carlo dépendante du temps (des sections efficaces homogénéisées et condensées), un schéma de calcul se composant de trois principales parties a été défini afin de simuler le comportement du réacteur CANDU-6 durant un accident du type LOCA en utilisant les modules additionnels NDF pour le code DONJON.

La première partie sert à définir la géométrie du réacteur (module `GEOM:`), lire les sections efficaces pour chaque région (module `CPO:` et le nouveau module `OMCDRG:`), identifier les canaux qui doivent être vidés du caloporteur durant la transitoire (le module `INIRES:` modifié) et finalement identifier et positionner les différents éléments de contrôle du réacteur comme les capteurs, les barres de contrôles et les zones liquides en faisant appel à leurs modules respectifs.

La deuxième partie du schéma de calcul sert à déterminer le flux neutronique initial pour le réacteur. Comme nous considérons que le réacteur a été stable et à l'équilibre pour une longue période de temps avant le début de la transitoire, le flux neutronique, qui est une solution de l'équation de diffusion statique, peut être utilisé comme condition initiale du réacteur au temps zéro. Pour ce faire, nous faisons appel à un ensemble de modules qui définissent le tracking et ensuite résolvent l'équation de diffusion statique (modules `CPLING:`, `FLXNRMNDF:`, `STATICNDF:`, `OUTER:` et `INTER:`).

Finalement, la troisième partie présente la boucle itérative du calcul de diffusion en cinétique espace-temps. À chaque pas de temps, les paramètres nucléaires de la zone du réacteur vidée sont mis à jour à partir de la bibliothèque de sections efficaces construite par le module `OMCDRG:`. Nous associons de nouvelles valeurs de sections efficaces aux différents canaux affectés en utilisant le nouveau module `UPTDXS:`. Pour chaque itération, la méthode implicite est utilisée pour résoudre l'équation de diffusion dépendante du temps afin d'obtenir la nouvelle distribution du flux neutronique. Un appel au module `POWERKIN:` nous permet d'obtenir la distribution de puissance spatiale du réacteur.

5.2 Modifications apportées au code DONJON

La seule modification au niveau des modules existants de DONJON-NDF a été apportée au module INIRES:. Par contre, deux nouveaux modules ont été ajoutés : OMCDRG: et UPTDXS:.

Tableau 5.1 Liste des modules créés et modifiés

Acronyme du module :	Nom complet du module & description
OMCDRG :	OpenMC-TD to DRAGON Ce module génère la table des sections efficaces qui évoluent dans le temps
INIRES :	Initialiser le réseau Ce module a été mis à jour afin de définir les zones de la géométrie qui sont affectées par la perturbation (accident)
UPTDXS :	Update Time Dependent Cross Section Ce module mettra à jour les sections efficaces en utilisant la carte et la table de sections efficaces produites par les modules précédents

5.2.1 Modifications apportées au module INIRES:

Le module INIRES: a été modifié afin de prendre en considération un nouveau mot clé : «ACC-ZONES». Il sera suivi d'un vecteur d'entiers (dont la longueur est égale au nombre de canaux de combustible). Chaque entier spécifie la zone de modification à laquelle appartient chaque canal de combustible.

Dans la figure 5.2, les 380 canaux du réacteur CANDU-6 sont divisés en deux groupes en associant l'indice 1 à la moitié gauche et l'indice 2 à la moitié droite du réacteur.

```

ACC-ZONES
      1 1 1 2 2 2
    1 1 1 1 1 1 2 2 2 2 2 2
  1 1 1 1 1 1 1 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
  1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
    1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
      1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2

1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2

1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2 2
  1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2
  1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2 2
    1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2 2
    1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2 2
      1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
      1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2 2
        1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
        1 1 1 1 1 1 1 1 1 1 1 2 2 2 2 2
          1 1 1 1 1 1 1 1 1 1 1 2 2 2 2
          1 1 1 1 1 1 1 1 1 1 1 2 2 2 2
            1 1 1 1 1 1 1 1 1 1 1 2 2 2
            1 1 1 1 1 1 1 1 1 1 1 2 2 2

```

Figure 5.2 Identification des zones en utilisant ACC-ZONES du module INIRES:

5.2.2 Module OMCDRG:

Le nouveau module, OMCDRG: (OpenMC to DRAGON), effectue la lecture des sections efficaces générées par OpenMC-TD. Il est important de noter que ce module effectue la lecture de ces données lorsqu'elles sont présentées dans un ordre bien défini. Autrement, la lecture et les résultats peuvent être corrompus.

Les résultats des taux de réaction doivent être présentés dans l'ordre suivant :

```

TIME INTERVAL 1:]0.0 ->1.00000E-04 ]
-----

```

```

Neutron density : 1.17859E+06

```

```

=====> TALLY 1 <=====

```

```

Incoming Energy [0.0, 4.00000E-06)

```

```

  Total Material

```

Total Reaction Rate	1.94676E+07
Flux	5.10257E+07
Fission Rate	46610.2
Nu-Fission Rate	1.13575E+05
Absorption Rate	1.20318E+05
Kappa-Fission Rate	9.01830E+06
Density	8.68294E+05

Incoming Energy [4.00000E-06, 20.0000)

Total Material

Total Reaction Rate	3.14495E+07
Flux	9.84888E+07
Fission Rate	17989.6
Nu-Fission Rate	46976.7
Absorption Rate	1.47501E+05
Kappa-Fission Rate	3.52075E+06
Density	48277.5

=====> TALLY 2 <=====

Incoming Energy [0.0, 4.00000E-06)

Total Material

Transport Rate	1.57902E+07
(n,2n)	0.0
(n,3n)	0.0
(n,4n)	0.0
Scattering Rate	1.86348E+07

Incoming Energy [4.00000E-06, 20.0000)

Total Material

Transport Rate	2.52297E+07
(n,2n)	4811.00
(n,3n)	3.00000
(n,4n)	0.0
Scattering Rate	3.12974E+07

=====> TALLY 3 <=====

```

Incoming Energy [0.0, 4.00000E-06)
  Outgoing Energy [0.0, 4.00000E-06)
    Total Material
      Scattering Rate          1.86342E+07
  Outgoing Energy [4.00000E-06, 20.0000)
    Total Material
      Scattering Rate          614.000
Incoming Energy [4.00000E-06, 20.0000)
  Outgoing Energy [0.0, 4.00000E-06)
    Total Material
      Scattering Rate          1.02595E+06
  Outgoing Energy [4.00000E-06, 20.0000)
    Total Material
      Scattering Rate          3.02715E+07

```

```

===== > TALLY 4 <=====

```

```

Outgoing Energy [0.0, 4.00000E-06)
  Total Material
    Nu-Fission Rate          0.0
  Outgoing Energy [4.00000E-06, 20.0000)
    Total Material
      Nu-Fission Rate          1.75197E+05

```

```

TIME INTERVAL 2:]1.00000E-04 ->2.00000E-04 ]

```

```

...

```

Afin d'obtenir cet affichage des résultats, un fichier d'entrée d'OpenMC-TD nommé «Tallies.xml» doit être fourni. Ce dernier doit contenir la séquence de données suivante :

```

<tallies>
  <!-- whole cell -->
  <tally id="1">
    <filter type="energy" bins="0.0 0.4e-5 20.0" />
    <estimator>tracklength</estimator>
    <scores>total flux fission nu-fission absorption kappa-fission density</scores>
  </tally>
  <tally id="2">

```

```

    <filter type="energy" bins="0.0 0.4e-5 20.0" />
    <scores>transport N2N N3N N4N scatter</scores>
</tally>
<tally id="3">
    <filter type="energy" bins="0.0 0.4e-5 20.0" />
    <filter type="energyout" bins="0.0 0.4e-5 20.0" />
    <scores>scatter</scores>
</tally>
<tally id="4">
    <filter type="energyout" bins="0.0 0.4e-5 20.0" />
    <scores>nu-fission</scores>
</tally>
</tallies>

```

Le fichier de sortie du module `OMCDRG` : est un fichier contenant toutes les sections efficaces écrites en respectant un format et un ordre très similaire à celui d'une `L_CPO` de `DRAGON`. Il s'agit d'une bibliothèque de sections efficaces dont le paramètre d'évolution est le temps.

L'appel à `OMCDRG` : est assez simple. Seulement deux paramètres obligatoires sont à fournir :

- Le premier paramètre est fourni en spécifiant le mot clé «`INPUT_F`» suivi du nom du fichier d'entrée qui doit être lu (il faut que le nom soit accompagné du chemin du fichier s'il n'est pas dans le même répertoire). Il s'agit ici du fichier résultat d'`OpenMC-TD`. Dans l'exemple qui va suivre, le fichier s'appelle *tallies-TD.out* et se trouve dans le dossier «`DEV_cpo`».
- Le deuxième paramètre à fournir est le nom du fichier de sortie. En utilisant le mot clé «`OUTPUT_F`» suivi du nom qui doit être donné au fichier dans lequel toutes les sections efficaces seront écrites sous le format des `/CPO/` légèrement modifié et appelé `/CPOTD/` (annexe B).

Dans l'exemple ci-dessous, le fichier de sortie s'appelle «`TDCPO`».

Exemple d'utilisation du module «`OMCDRG` : » :

```
OMCDRG: ::
```

```
EDIT 0
```

```
! Fichier d'entree: fichier obtenue a la fin de la simulation
```

```
! d'OpenMc-TD
```

```
INPUT_F DEV_cpo/tallies-TD.out
```

```
! Fichier de sortie: fichier qui contiendra les sections efficaces ecrites
```

```
! dans le meme format que celui d'une L_CPO
```

```

OUTPUT_F TDCPO ;
! XS-CORRECT 1 technique des pourcentages (*)
! XS-CORRECT 2 technique des deltas sigmas (-)
XS-CORRECT 1 REF-FILE DEV_cpo/tallies-static.out
! Application d'un filtre numerique
FILTER 1 TAU 0.015 DT 0.0001

```

«XS-CORRECT» et «FILTER» sont des mots clés définis pour l'application de la technique de correction des sections efficaces ou bien pour l'application des filtres sur les sections efficaces. Ces deux techniques sont expliquées aux sections 5.3.3 et 5.3.2 .

5.2.3 Module UPTDXS:

Le module UPTDXS: a pour rôle de mettre à jour la Macrolib des sections efficaces du réacteur. Le module commence par lire la géométrie, la carte des zones d'accidents, la librairie des sections efficaces fournie par OMCDRG: ainsi que la Macrolib initiale.

Ce module reçoit en entrée deux paramètres :

1. Le premier paramètre représente l'identifiant de la zone qui va subir les perturbations. Il faut utiliser le mot clé «ZONE» suivit d'un entier qui représente une zone d'accident. Cette dernière a déjà été définie en faisant appel à INIRES:.
2. Le deuxième paramètre est le temps. Cette variable sert à identifier le dossier des sections efficaces qui sera utilisé pour remplacer les anciennes valeurs. Il suffit d'utiliser le mot clé «TIME» suivi d'un réel.

Dans l'exemple qui va suivre, c'est la zone 2 qui sera mise à jour et le temps utilisé est 0.05000E-04 seconde.

En utilisant une boucle qui parcourt toute la matrice de la géométrie, le module remplace, pour les régions identifiées par le même indice de zone, les anciennes sections efficaces de la Macrolib par les nouvelles sections efficaces lues à partir de la bibliothèque.

Exemple d'utilisation du module UPTDXS: :

```

MACRO1 := UPTDXS: MACRO1 RESEAU MACTD GEO ::
EDIT 0
! On lit, dans la L_CPO, les sections efficaces identifiées par le temps
! TIME (en seconde)
TIME 0.05000E-04
! on met a jour les zones identifiées par le chiffre 2
ZONE 2 ;

```

5.3 Techniques numériques utilisées pour l'analyse des résultats

5.3.1 Élimination de la correction des sections efficaces de fission par le k_{eff}

Comme les simulations en espace-temps commencent toujours par résoudre l'équation de Boltzmann statique pour obtenir la distribution du flux neutronique initiale, les sections efficaces de fission sont divisées par la valeur estimée du facteur de multiplication effective k_{eff} . Cette étape a pour objectif de balancer les termes de production avec les termes d'élimination dans l'équation de diffusion statique (2.34).

Dans la deuxième phase de la simulation, c'est à dire lors de la résolution de l'équation dépendante du temps (3.20), il est important de noter qu'il n'y pas de correction des sections efficaces de fission par le facteur de multiplication k_{eff} . Nous avons dû revoir le code afin de supprimer toutes les corrections des sections efficaces de fission par le coefficient de multiplication effectif, k_{eff} , durant toute la phase de la résolution implicite.

5.3.2 Application des filtres numériques

Dû à la durée globale des simulations d'OpenMC-TD, il nous a été impossible d'utiliser la loi des grands nombres pour le calcul des sections efficaces dépendantes du temps dans chaque intervalle de récolte dt . Afin d'obtenir des sections efficaces moyennes par intervalle de temps, il est nécessaire répéter la simulation dépendante du temps un grand nombre de fois pour ensuite calculer les valeurs moyennes des résultats.

Il est important de noter que chaque simulation dure en moyenne une semaine (Processeur 2.7 GHz Intel Core i7 avec une mémoire de 16 GB 1600 MHz DDR3). De ce fait, les sections efficaces ont été calculées à la suite d'une seule simulation dépendante du temps pour une cellule CANDU-6. Les résultats obtenus montrent une légère oscillation des résultats que la loi des grands nombres aurait pu éliminer.

Le comportement des résultats est similaire aux résultats obtenus par les détecteurs des systèmes de régulations installés dans les réacteurs nucléaires. La solution utilisée dans ce genre de situation est celle des filtres numériques. Leurs rôles est d'atténuer les fluctuations dues aux bruits.

La théorie des filtres numériques est connue et utilisée dans plusieurs domaines. Elle a été abordée par plusieurs ouvrages. Les équations qui seront présentées se basent sur la théorie présentée dans (Boukas, 1995). En pratique, l'équation décrivant les filtres numériques utilisés dans ce projet est :

$$Y(t) = \exp\left(-\frac{t-t_0}{\tau}\right)Y(t_0) + \left(1 - \exp\left(-\frac{t-t_0}{\tau}\right)\right)X(t_0) \quad (5.1)$$

tel que :

- $Y(t)$ représente la variable filtrée au temps t ,
- $Y(t_0)$ représente la valeur de la variable filtrée au début de l'intervalle de temps, et
- $X(t_0)$ représente la valeur courante de la variable à filtrer.

Sachant que $T_s = t_{n+1} - t_n$, les valeurs filtrées sont calculées comme suit :

$$Y_{n+1} = \exp\left(-\frac{T_s}{\tau}\right) Y_n + \left(1 - \exp\left(-\frac{T_s}{\tau}\right)\right) X_{n+1} \quad (5.2)$$

Nous avons deux possibilités d'utilisation de ces filtres :

- La première est d'appliquer ces filtres sur les sections efficaces obtenues par OpenMC. Cette démarche introduit une quasi-stabilité dans les valeurs de sections efficaces avant l'accident. Par contre, elle atténue l'impact de l'accident sur les sections efficaces et donc affecte les résultats obtenus pour l'accident en question.
- La deuxième démarche est d'appliquer les filtres sur la courbe de la puissance du réacteur. Cette démarche a pour but de rendre plus lisible et observable le comportement temporel de la puissance du réacteur.

Les résultats, avec et sans les filtres, sont présentés dans le chapitre 7.

5.3.3 Correction des sections efficaces

Technique des deltas sigmas

La technique des *deltas sigmas* consiste à utiliser les variations des sections efficaces à la place d'utiliser les valeurs calculées. Cette technique permet d'utiliser comme valeurs initiales des sections efficaces obtenues par des calculs statiques pour les mêmes conditions de cellules que celles du début de la simulation dépendante du temps. À chaque pas de temps, nous corrigerons les sections efficaces initiales (obtenues en calcul statiques) par la différence entre les sections efficaces relatives à un intervalle de temps n et les sections efficaces dépendantes du temps obtenues à l'instant zéro.

Tableau 5.2 Technique des *deltas sigmas*

Intervalle de temps	Δt_0	Δt_1	Δt_2	Δt_3	Δt_4	Δt_5	Δt_{n-1}	Δt_n
TDXS	Σ_{t0}	Σ_{t1}	Σ_{t2}	Σ_{t3}	Σ_{t4}	Σ_{t5}	Σ_{tn-1}	Σ_{tn}
$\Delta\Sigma$	0	$\Sigma_{t1} - \Sigma_0$	$\Sigma_{t2} - \Sigma_0$	$\Sigma_{t3} - \Sigma_0$	$\Sigma_{t4} - \Sigma_0$	$\Sigma_{t5} - \Sigma_0$	$\Sigma_{tn-1} - \Sigma_0$	$\Sigma_{tn} - \Sigma_0$
		$\Delta\Sigma_1$	$\Delta\Sigma_2$	$\Delta\Sigma_3$	$\Delta\Sigma_4$	$\Delta\Sigma_5$	$\Delta\Sigma_{n-1}$	$\Delta\Sigma_n$
Nouvelles sections efficaces	$\Sigma_{statique}$	$\Sigma_{statique} + \Delta\Sigma_1$	$\Sigma_{statique} + \Delta\Sigma_2$	$\Sigma_{statique} + \Delta\Sigma_3$	$\Sigma_{statique} + \Delta\Sigma_4$	$\Sigma_{statique} + \Delta\Sigma_5$	$\Sigma_{statique} + \Delta\Sigma_{n-1}$	$\Sigma_{statique} + \Delta\Sigma_n$
	$\Sigma_{statique}$	Σ_1^*	Σ_2^*	Σ_3^*	Σ_4^*	Σ_5^*	Σ_7^*	Σ_8^*

CHAPITRE 6 VALIDATION DU MODÈLE STATIQUE

Avant de s’attaquer à la problématique et commencer les modifications dans le code OpenMC, il fallait tout d’abord mettre en place un modèle OpenMC pour une cellule CANDU-6 et s’assurer de la qualité des résultats obtenus pour ce type de réacteur. Dans ce cadre, nous avons décidé de mener une série de simulations pour une cellule CANDU-6 avec différentes bibliothèques de sections efficaces afin de comparer ces résultats avec ceux obtenus par d’autres codes de transports déterministes et stochastiques. Les comparaisons ont été réalisées en faisant appel aux codes DRAGON et Serpent. Les bibliothèques utilisées sont ENDF-B7.0 et Jeff 3.1.1. Les résultats de cette étude ont fait l’objet d’un article (voir annexe A).

6.1 Résultats statiques déterministes

Le code DRAGON contient deux modules différents de tracking s’appelant NXT: et EXCELT: (MARLEAU, 2005). EXCELT: traite seulement les géométries simples et fournit ses résultats en simple précision alors que NXT: est le module le plus récent qui permet de traiter des géométries complexes en double précision. Afin d’effectuer une analyse initiale complète de l’accident LOCA et son influence sur le réacteur, nous avons décidé de lancer un ensemble de simulations avec deux bibliothèques de sections efficaces microscopiques différentes (ENDFB7 and JEFF3.1.1) avec différents schémas de découpage des groupes d’énergies (XMAS-172, SHEM-281, etc) et finalement avec les deux modules de tracking de DRAGON.

Tableau 6.1 k_{eff} et calcul de réactivité pour une cellule CANDU-6 en utilisant le module EXCELT: pour le tracking.

DRAGON		XMAS-172	SHEM-281	SHEM-295	SHEM-315	SHEM-361
Cellule CANDU-6 refroidie	endfb7	1.120501	1.121469	1.123489	1.123617	1.123449
	Jeff 3.1.1	1.118341	1.119294	1.121263	1.121389	1.121220
Cellule CANDU-6 vidangée	endfb7	1.143286	1.144173	1.146300	1.146444	1.146248
	Jeff 3.1.1	1.141095	1.141966	1.144044	1.144188	1.143988
Δk_{eff}	endfb7	0.022785	0.022704	0.022811	0.022827	0.022799
	Jeff 3.1.1	0.022754	0.022672	0.022781	0.022799	0.022768
ρ perturbation	endfb7	0.017786	0.017694	0.017713	0.017721	0.017705
	Jeff 3.1.1	0.017830	0.017738	0.017759	0.017769	0.017750

Tableau 6.2 k_{eff} et calcul de réactivité pour une cellule CANDU-6 en utilisant le module NXT: pour le tracking.

DRAGON		XMAS- 172	SHEM- 281	SHEM- 295	SHEM- 315	SHEM- 361
Cellule CANDU-6 refroidie	endfb7	1.120499	1.121465	1.123484	1.123611	1.123444
	Jeff 3.1.1	1.118344	1.119288	1.121258	1.121383	1.121215
Cellule CANDU-6 vidangée	endfb7	1.143290	1.144183	1.146311	1.146455	1.146260
	Jeff 3.1.1	1.141101	1.141976	1.144055	1.144199	1.144001
Δk_{eff}	endfb7	0.022791	0.022718	0.022827	0.022844	0.022816
	Jeff 3.1.1	0.022757	0.022688	0.022797	0.022816	0.022786
ρ perturbation	endfb7	0.017791	0.017705	0.017725	0.017734	0.017718
	Jeff 3.1.1	0.017833	0.017750	0.017772	0.017782	0.017764

6.2 Résultats statiques Monte Carlo

Dans les simulations Monte Carlo, il a été décidé d'utiliser 3000 itérations dont chacune simule 100,000 particules. Les résultats des 150 premières itérations ont été ignorés dans le calcul des résultats (flux neutronique et sections efficaces homogénéisées). Afin de pouvoir comparer les résultats obtenus avec ceux des simulations déterministes, les mêmes bibliothèques de sections efficaces ont été utilisées dans leurs formes continues en énergie. Il est à noter que les mêmes fichiers des bibliothèques ont été utilisés pour les codes Serpent et OpenMC.

6.2.1 Simulations SERPENT pour une cellule CANDU-6

Les résultats des simulations effectuées avec Serpent pour une cellule CANDU-6 sont présentés dans le tableau (6.3).

Tableau 6.3 k_{eff} et calcul de réactivité pour une cellule CANDU-6 en utilisant le code Serpent

SERPENT		k_{eff} analog	+/-	k_{eff} implicit	+/-
Cellule CANDU-6 refroidie	endfb7	1.12393	0.00040	1.12347	0.00010
	Jeff 3.1.1	1.12221	0.00040	1.12173	0.00011
Cellule CANDU-6 vidangée	endfb7	1.14065	0.00041	1.14063	0.00010
	Jeff 3.1.1	1.13894	0.00040	1.13856	0.00010
Δk_{eff}	endfb7	0.01672		0.01716	
	Jeff 3.1.1	0.01673		0.01683	
ρ perturbation	endfb7	0.013042		0.013391	
	Jeff 3.1.1	0.013089		0.013178	

6.2.2 Simulations OpenMC pour une cellule CANDU-6

OpenMC est un code assez récent qui ne permet pas l'évolution isotopique des matériaux. Ainsi, seulement des résultats pour une cellule neuve sont présentés dans ce chapitre. Les résultats des simulations effectuées avec OpenMC pour une cellule CANDU-6 sont présentés dans le tableau 6.4

Tableau 6.4 k_{eff} et calcul de réactivité pour une cellule CANDU-6 en utilisant le code OpenMC

OpenMC		k_{eff} Collision	k_{eff} Track-length	k_{eff} Absorption	Combined k_{eff}
Cellule CANDU-6 refroidie	endfb7	1.12330	1.12345	1.12355	1.12345
	+/-	0.00037	0.00041	0.00035	0.00028
	Jeff 3.1.1	1.12154	1.12149	1.12231	1.12196
	+/-	0.00038	0.00042	0.00035	0.00028
Cellule CANDU-6 vidangée	endfb7	1.14114	1.14133	1.14025	1.14064
	+/-	0.00038	0.00041	0.00033	0.00026
	Jeff 3.1.1	1.13776	1.13789	1.13876	1.13831
	+/-	0.00037	0.00040	0.00034	0.00027
Δk_{eff}	endfb7	0.01784	0.01788	0.0167	0.01719
	Jeff 3.1.1	0.01622	0.0164	0.01645	0.01635
ρ perturbation	endfb7	0.013917	0.013944	0.013035	0.013414
	Jeff 3.1.1	0.012711	0.012851	0.012871	0.0128021

6.3 Comparaison des sections efficaces condensées et homogénéisées

Il est important de comparer les paramètres nucléaires autre que le k_{eff} et la réactivité. Dans ce cadre, une comparaison des différentes valeurs de sections efficaces condensées et homogénéisées pour la cellule CANDU-6 obtenues par les codes OpenMC et Serpent est effectuée. Les tableaux 6.5 à 6.8 présentent les différences obtenues dans les sections efficaces totale, d'absorption, de collision, et de fission pour une cellule refroidie et une cellule vidée de son caloporteur. Ces différences ne sont pas significatives.

6.4 Analyse des résultats statiques

Quatre principales raisons peuvent expliquer les différences observées (MAHJOUR et KOCLAS, 2015) :

- L'utilisation de différentes bases de données est la différence la plus importante entre les codes. Le code de transport déterministe DRAGON utilise des sections efficaces

Tableau 6.5 Différences entre les sections efficaces homogénéisées et condensées obtenues par Serpent et OpenMC pour [une cellule CANDU-6 refroidie](#) en utilisant la librairie ENDF-B7.0

			Serpent	OpenMC	
			ENDF-B7	ENDF-B7	$\Delta\Sigma$
Homogenized Cross Sections for the cell	Fast Gr	total XS	3.16E-01	3.16E-01	0.00E+00
		absorption XS	1.77E-03	1.77E-03	0.00E+00
		scattering XS	3.15E-01	3.15E-01	0.00E+00
		fission XS	3.38E-04	3.38E-04	0.00E+00
	Thermal Gr	total XS	4.19E-01	4.19E-01	0.00E+00
		absorption XS	3.19E-03	3.19E-03	0.00E+00
		scattering XS	4.16E-01	4.16E-01	0.00E+00
		fission XS	1.61E-03	1.61E-03	0.00E+00
Homogenized Cross Sections for the moderator	Fast Gr	total XS	3.33E-01	3.33E-01	0.00E+00
		absorption XS	2.94E-05	2.94E-05	0.00E+00
		scattering XS	3.33E-01	3.33E-01	0.00E+00
		fission XS	0.00E+00	0.00E+00	0.00E+00
	Thermal Gr	total XS	4.37E-01	4.37E-01	0.00E+00
		absorption XS	3.60E-05	3.60E-05	0.00E+00
		scattering XS	4.37E-01	4.37E-01	0.00E+00
		fission XS	0.00E+00	0.00E+00	0.00E+00

Tableau 6.6 Différences entre les sections efficaces homogénéisées et condensées obtenues par Serpent et OpenMC pour [une cellule CANDU-6 refroidie](#) en utilisant la librairie Jeff3.1.1

			Serpent	OpenMC	
			Jeff3.1.1	Jeff3.1.1	$\Delta\Sigma$
Homogenized Cross Sections for the cell	Fast Gr	total XS	3.19E-01	3.16E-01	-3.00E-03
		absorption XS	1.67E-03	1.77E-03	1.00E-04
		scattering XS	3.14E-01	3.14E-01	0.00E+00
		fission XS	3.30E-04	3.36E-04	6.00E-06
	Thermal Gr	total XS	4.22E-01	4.18E-01	-4.00E-03
		absorption XS	3.29E-03	3.20E-03	-9.00E-05
		scattering XS	4.19E-01	4.15E-01	-4.00E-03
		fission XS	1.67E-03	1.61E-03	-6.00E-05
Homogenized Cross Sections for the moderator	Fast Gr	total XS	3.36E-01	3.34E-01	-2.00E-03
		absorption XS	2.89E-05	3.20E-05	3.10E-06
		scattering XS	3.36E-01	3.33E-01	-3.00E-03
		fission XS	0.00E+00	0.00E+00	0.00E+00
	Thermal Gr	total XS	4.41E-01	4.37E-01	4.00E-03
		absorption XS	3.71E-05	3.60E-05	1.10E-06
		scattering XS	4.41E-01	4.37E-01	-4.00E-03
		fission XS	0.00E+00	0.00E+00	0.00E+00

Tableau 6.7 Différences entre les sections efficaces homogénéisées et condensées obtenues par Serpent et OpenMC pour [une cellule CANDU-6 vidangée](#) en utilisant la librairie ENDF-B7.0

			Serpent	OpenMC			
			ENDF-B7	ENDF-B7	$\Delta\Sigma$		
Homogenized Cross Sections for the cell	Fast Gr	Total XS	3.01E-01	2.98E-01	-3.00E-03		
		Absorption XS	1.52E-03	1.59E-03	7.00E-05		
		Scattering XS	2.97E-01	2.96E-01	-1.00E-03		
		Fission XS	3.38E-04	3.43E-04	5.00E-06		
	Thermal Gr	Total XS	4.13E-01	4.09E-01	-4.00E-03		
		Absorption XS	3.35E-03	3.25E-03	-1.00E-04		
		Scattering XS	4.10E-01	4.06E-01	-4.00E-03		
		Fission XS	1.70E-03	1.64E-03	-6.00E-05		
		Homogenized Cross Sections for the moderator	Fast Gr	Total XS	3.35E-01	3.32E-01	-3.00E-03
				Absorption XS	2.81E-05	3.11E-05	3.00E-06
Scattering XS	3.35E-01			3.32E-01	-3.00E-03		
Fission XS	0.00E+00			0.00E+00	0.00E+00		
Thermal Gr	Total XS		4.41E-01	4.37E-01	-4.00E-03		
	Absorption XS		3.71E-05	3.58E-05	-1.30E-06		
	Scattering XS		4.41E-01	4.37E-01	-4.00E-03		
	Fission XS		0.00E+00	0.00E+00	0.00E+00		

Tableau 6.8 Différences entre les sections efficaces homogénéisées et condensées obtenues par Serpent et OpenMC pour [une cellule CANDU-6 vidangée](#) en utilisant la librairie Jeff3.1.1

			Serpent	OpenMC			
			Jeff3.1.1	Jeff3.1.1	$\Delta\Sigma$		
Homogenized Cross Sections for the cell	Fast Gr	Total XS	3.01E-01	2.98E-01	-3.00E-03		
		Absorption XS	1.52E-03	1.59E-03	7.00E-05		
		Scattering XS	2.97E-01	2.96E-01	-1.00E-03		
		Fission XS	3.35E-04	3.40E-04	5.00E-06		
	Thermal Gr	Total XS	4.13E-01	4.09E-01	-4.00E-03		
		Absorption XS	3.36E-03	3.26E-03	-1.00E-04		
		Scattering XS	4.09E-01	4.06E-01	-3.00E-03		
		Fission XS	1.71E-03	1.64E-03	-7.00E-05		
		Homogenized Cross Sections for the moderator	Fast Gr	Total XS	3.35E-01	3.32E-01	-3.00E-03
				Absorption XS	3.06E-05	3.39E-05	3.30E-06
Scattering XS	3.35E-01			3.32E-01	-3.00E-03		
Fission XS	0.00E+00			0.00E+00	0.00E+00		
Thermal Gr	Total XS		4.41E-01	4.37E-01	-4.00E-03		
	Absorption XS		3.71E-05	3.58E-05	-1.30E-06		
	Scattering XS		4.41E-01	4.37E-01	-4.00E-03		
	Fission XS		0.00E+00	0.00E+00	0.00E+00		

multi-groupes obtenues par NJOY. Dans ce cas, seule l'équation de transport multi-groupe peut être résolue pour obtenir des résultats précis. Dans les simulations de type Monte Carlo, la bibliothèque des sections efficaces utilisée est continue en énergie et ne nécessite aucun prétraitement et aucune autoprotection.

- La deuxième différence est basée sur le processus de résolution de l'équation de transport. L'équation de transport sous forme multi groupes est résolue par DRAGON en utilisant la méthode de la probabilité de collision. Cependant, dans les codes de Monte Carlo, un nombre important de particules est suivi, une par une, pour prédire leur comportement général.
- L'écart entre les simulations Monte Carlo et les simulations déterministes est accentué par les différences de température utilisée dans les matériaux de la géométrie. Les codes Monte Carlo utilisent des sections efficaces continues pour chaque nuclide mais seulement avec six températures différentes (300 K, 600 K, 900 K, 1200 K, 1500 K et 1800 K). Cependant, DRAGON utilise des sections efficaces multi-groupes pour différentes températures et procède avec un processus d'interpolation pour obtenir les sections pour la température exacte.
- Enfin, les différences entre OpenMC et Serpent peuvent s'expliquer par l'approche de suivi des neutrons utilisée par chaque code. OpenMC suit les particules entre deux collisions et les parcours sont échantillonnés en utilisant la section efficace totale du matériau traversé par la particule. Aux surfaces séparant deux matériaux, les particules sont arrêtées et de nouvelles pistes sont échantillonnées en utilisant les nouvelles sections transversales. Cependant, Serpent utilise l'approche *delta-tracking* où les particules sont arrêtées seulement aux limites de la géométrie. Les parcours sont ensuite échantillonnés en utilisant la plus grande valeur de section efficace totale de tous les matériaux présents dans la géométrie.

Il est à noter que les différences observées, dues à toutes ces raisons, sont minimes. Ainsi, ces travaux démontrent la fiabilité du code OpenMC et confirment qu'il peut être utilisé pour générer les différentes propriétés nucléaires pour une cellule CANDU-6. Ces dernières peuvent être utilisées dans des simulations en diffusion neutronique à l'échelle du réacteur.

CHAPITRE 7 RÉSULTATS DES SIMULATIONS DYNAMIQUES

7.1 Validation du modèle dynamique par des simulations en cinétique ponctuelle

Rappel : Théorie de la cinétique ponctuelle

La cinétique ponctuelle permet d'étudier le fonctionnement et le contrôle des réacteurs nucléaires afin de s'assurer des conditions limites de fonctionnement. Ainsi, on cherche à évaluer le comportement global du réacteur et non pas à définir les comportements des différents composants de ce dernier comme nous le faisons en utilisant la cinétique espace temps. De ce fait, les équations de la cinétique ponctuelle comprennent seulement des variables globales qui décrivent tout le réacteur.

Afin de pouvoir comparer la cinétique espace-temps avec la cinétique ponctuelle d'une manière efficace, il est important d'utiliser un réacteur homogène uniforme et infini dans le but de supprimer les effets spatiaux affectant les résultats dans la cinétique espace temps.

À partir de l'équation de Boltzmann (2.2) et les équations des précurseurs (2.3), les équations de la cinétique ponctuelle sont (Rozon, 1992) :

$$\frac{dn(t)}{dt} = \frac{\rho - \beta}{\Lambda} n(t) + \sum_{k=1}^K \lambda_k C_k(t) \quad (7.1)$$

$$\frac{dC_k(t)}{dt} = \frac{\beta_k}{\Lambda} n(t) + \lambda_k C_k(t) \quad k = 1..K \quad (7.2)$$

où :

- $n(t)$ est la population de neutrons
- Λ est le temps de génération tel que $\Lambda = \frac{\text{population neuronique}}{\text{taux de production}}$
- ρ est la réactivité tel que $\rho = \frac{\text{taux de production} - \text{taux de perte}}{\text{taux de production}}$
- K est le nombre des groupes de neutrons retardés.

La solution du système 7.1-7.2 est :

$$n(t) = \sum_{i=1}^{K+1} A_i e^{\omega_i t} \quad (7.3)$$

où les valeurs des w_i , racines du polynôme caractéristique, sont données par :

$$\rho = \Lambda \omega + \beta - \sum_{k=1}^{K_{max}} \lambda_k \frac{\beta_k}{(\omega_i + \lambda_k)} \quad i = 1, \dots, K + 1 \quad (7.4)$$

Vérification des résultats pour une cellule homogène

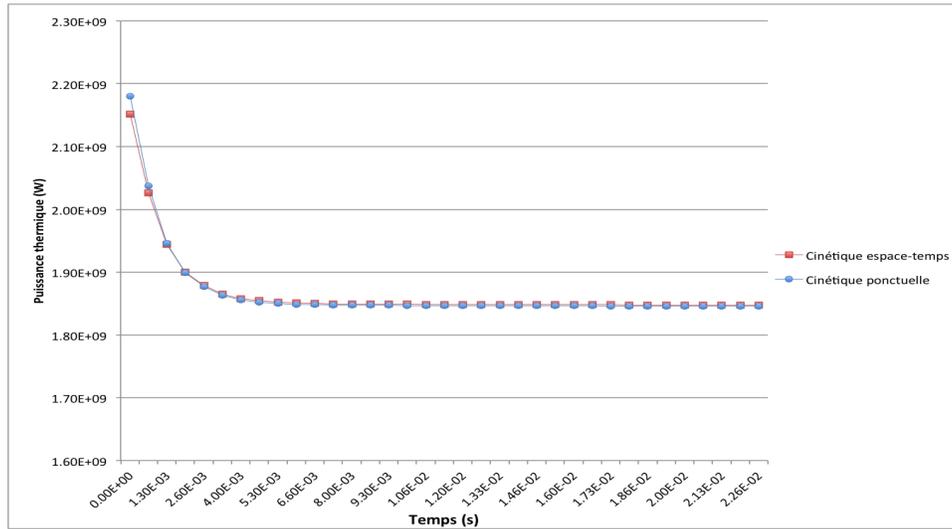


Figure 7.1 Puissance thermique d'une configuration de cellules homogènes infinies

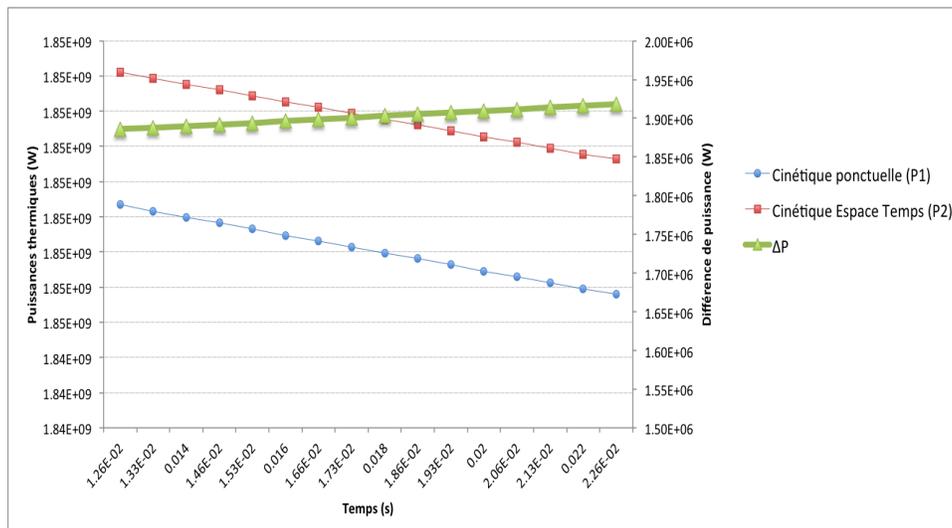


Figure 7.2 Puissance thermique d'une configuration de cellules homogènes infinies entre 0.03s et 0.05s.

Les courbes de la figure 7.1 sont obtenues en utilisant les paramètres nucléaires dépendants du temps obtenus par OpenMC-TD pour un réseau de cellules sous-critique ($k_{eff} = 0.99924$) en utilisant les deux méthodes : la cinétique ponctuelle et la cinétique espace temps. L'observation immédiate confirme que les résultats obtenus par la cinétique espace-temps sont

conformes avec ceux obtenus par la simulation en cinétique ponctuelle. La différence observée entre les deux courbes de la figure 7.2 ne dépasse pas les $1.08E-3\%$. Ainsi, une telle différence est suffisamment petite pour conclure la conformité des résultats entre les deux approches.

Cette constatation est assez importante car elle permet de valider les résultats qui seront présentés par la suite.

7.2 Résultats en transport neutronique

7.2.1 Simulation dépendante du temps d'une cellule homogène avec plusieurs itérations

Il est important de rappeler que tous les résultats obtenus à la suite d'une simulation Monte Carlo ne sont que des valeurs moyennes pour des calculs de statistiques accompagnées par leurs intervalles de confiance. La loi des grands nombres affirme que plus l'échantillon est grand plus la valeur moyenne obtenue est proche de la valeur réelle. En outre, plus l'échantillon est grand plus l'intervalle de confiance se rétrécit. Ainsi, en augmentant le nombre de neutrons suivis, les différentes valeurs des sections efficaces ainsi que tous les autres paramètres à calculer se rapprochent de leurs valeurs réelles respectives.

Dans les simulations avec OpenMC-TD, il existe deux manières différentes d'augmenter la taille de l'échantillon :

- soit en augmentant le nombre de particules (neutrons).
- soit en gardant le même nombre de particules mais en reprenant la simulation à plusieurs reprises (plusieurs itérations) pour calculer, par la suite, la valeur moyenne des résultats des différentes itérations.

Puisque dans les simulations dépendantes du temps, avec OpenMC-TD, le nombre de particules est variable, la première option n'est pas recommandée. Ainsi, nous avons décidé d'utiliser la deuxième méthode pour vérifier les résultats obtenus pour la cellule homogène à une seule itération («batch») avec ceux pour 10 itérations («batchs»). Les figures (7.3 à 7.8) présentent l'évolution des densités neutroniques, des facteurs $1/v$, et des sections efficaces totales par groupe pour les deux simulations.

Il est important de noter que la cellule choisie pour les simulations dépendante du temps est sous-critique. Ce choix est lié principalement au type d'accident à observer. Sachant qu'un accident de type LOCA dans un réacteur CANDU-6 introduit une réactivité positive, une cellule initialement critique devient super-critique après l'accident. Afin d'éviter que le nombre de neutrons à suivre devienne très grand et que la durée de la simulation devienne trop grande, il a été décidé de simuler une cellule initialement sous-critique qui devient légèrement sur-critique après l'accident.

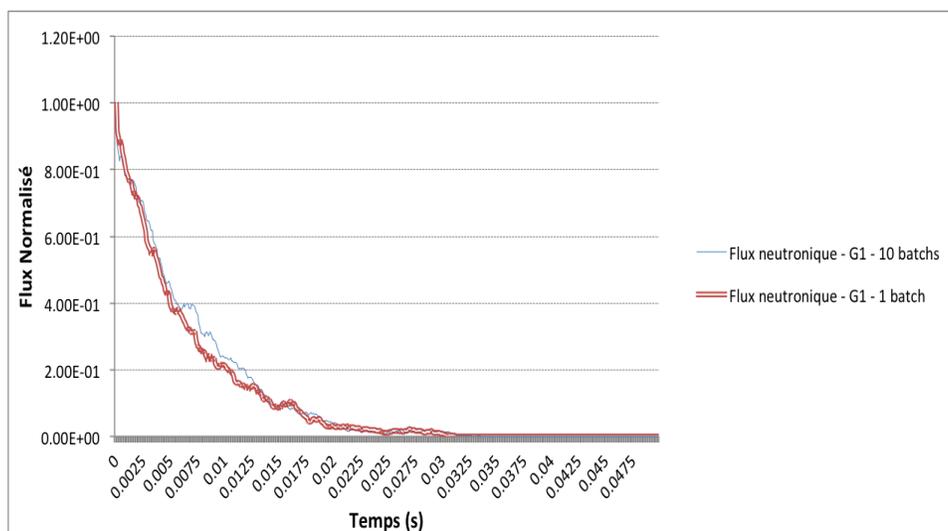


Figure 7.3 Flux neutronique (groupe rapide)

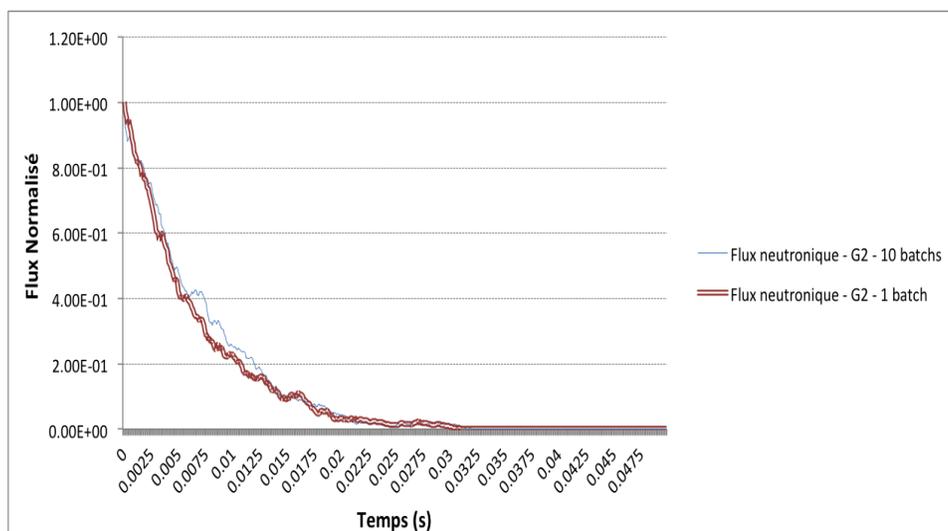


Figure 7.4 Flux neutronique (groupe thermique)

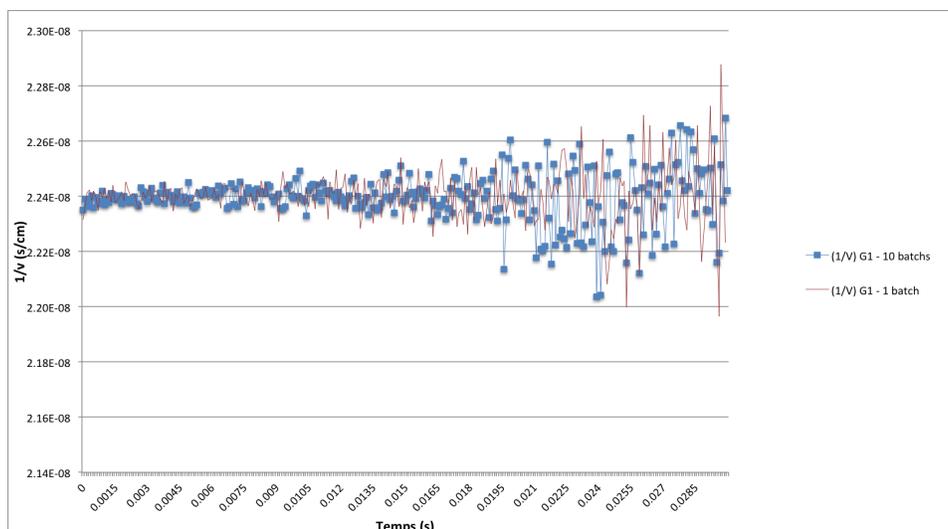


Figure 7.5 Comparaison du facteur $\frac{1}{v}$ (groupe rapide)

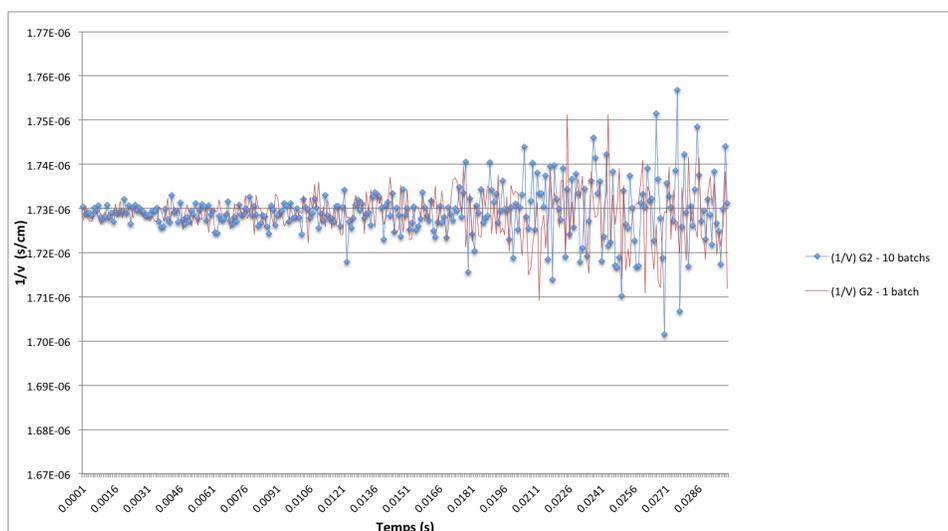


Figure 7.6 Comparaison du facteur $\frac{1}{v}$ (groupe thermique)

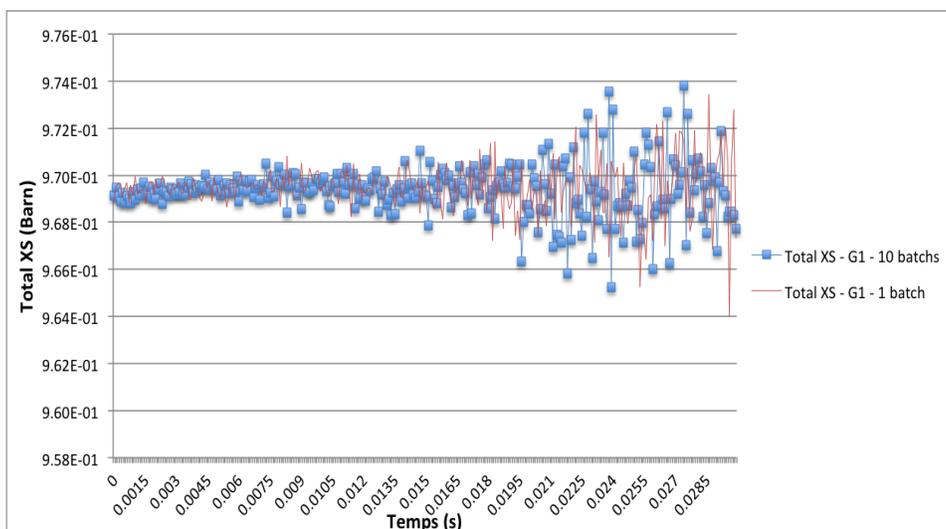


Figure 7.7 Comparaison de la section efficace totale (groupe rapide)

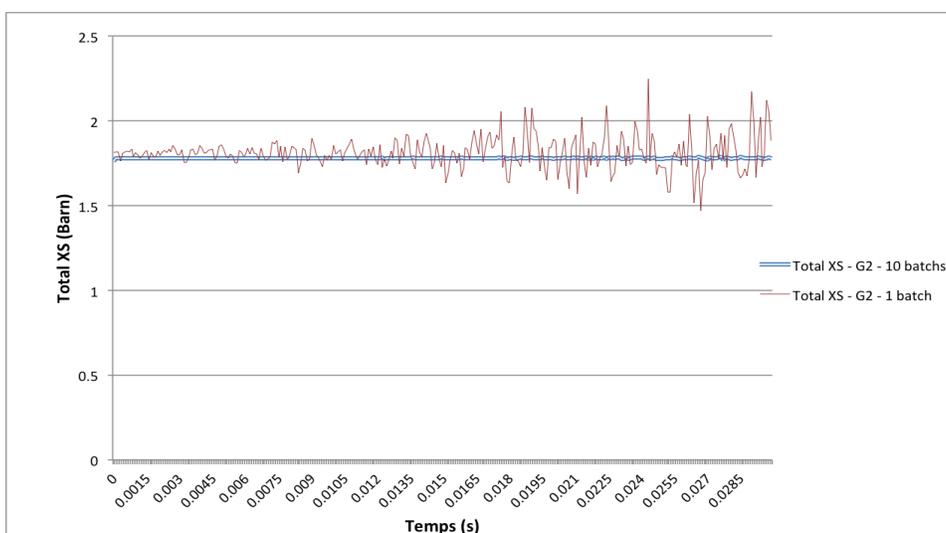


Figure 7.8 Comparaison de la section efficace totale (groupe thermique)

Ainsi, on observe, d'une manière plus claire à partir de la figure 7.8, que la loi des grands nombres s'applique dans cette situation et que les résultats obtenus démontrent une certaine amélioration au niveau de la stabilité. Il est aussi nécessaire de noter que les fluctuations observées dans l'extrémité droite des courbes sont principalement dues à la diminution importante du nombre de neutrons avec le temps, car la cellule est sous-critique.

7.2.2 Résultats pour une cellule CANDU-6

Avant d'aborder les simulations d'une cellule CANDU-6 subissant un accident de type LOCA, il est important d'observer le comportement de cette cellule en l'absence de toute perturbation. Ainsi, une cellule sous critique est définie et évoluée dans le temps grâce à l'outil OpenMC-TD. Le comportement observé (figure 7.9) est tout à fait normal et prévisible. La population de neutrons décroît rapidement selon une courbe exponentielle.

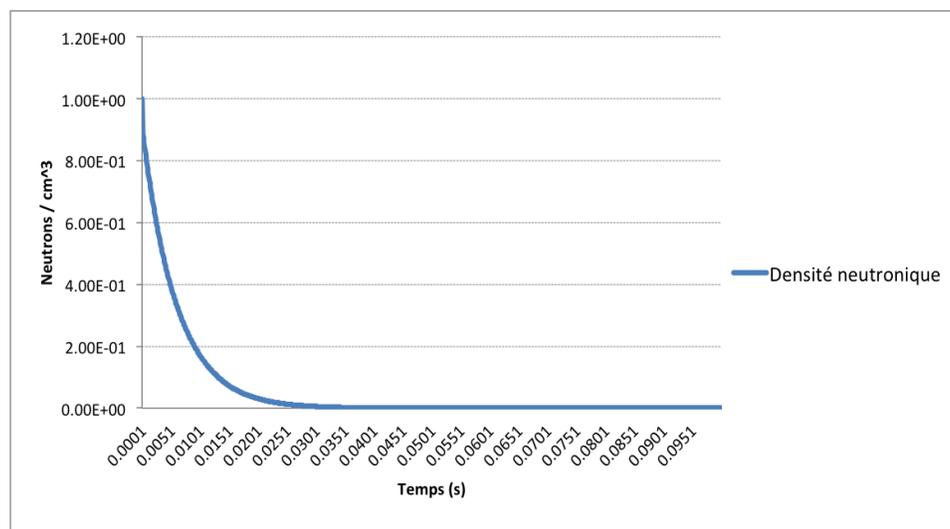


Figure 7.9 Densité neutronique totale (relative) pour une cellule CANDU-6 initialement sous-critique

7.2.3 Résultats pour une cellule CANDU-6 lors d'un accident de type LOCA

Après avoir vérifié et validé les résultats obtenus par le nouveau code pour une cellule homogène ainsi que pour une cellule CANDU-6 sous critique non perturbée, la phase des simulations pour une cellule CANDU-6 subissant un accident de perte du caloporteur a été déclenchée.

Les conditions de la simulation sont les suivantes :

- Le k_{eff} de la cellule pour la même géométrie obtenu par un calcul de valeur propre dans OpenMC est : $= 0.995$.
- La durée totale simulée est : $0.2s$.
- Le vide a été introduit pour remplacer tout le caloporteur à $t = 0.1s$.
- Les intervalles de récoltes sont égales à $10^{-4}s$.

Les résultats obtenus sont présentés dans les figures 7.10 à 7.15.

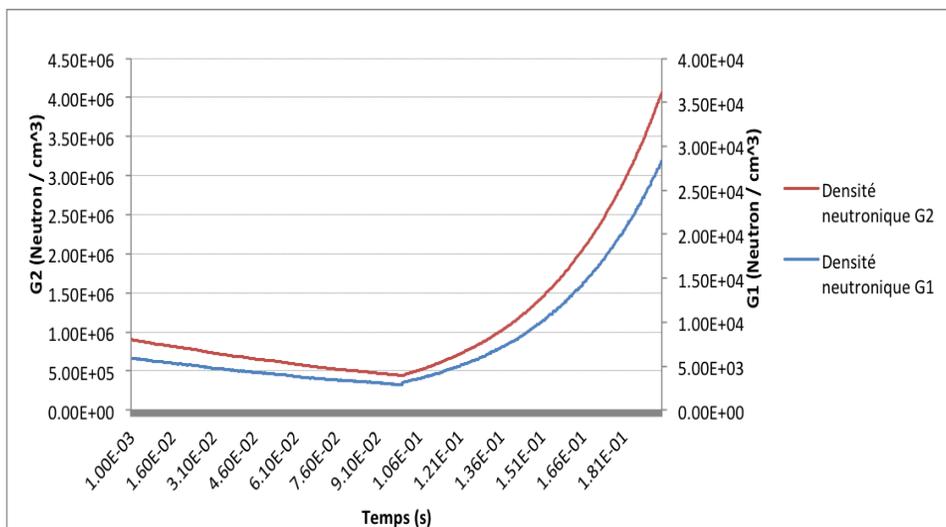


Figure 7.10 Densités neutroniques (groupes rapide et thermique)

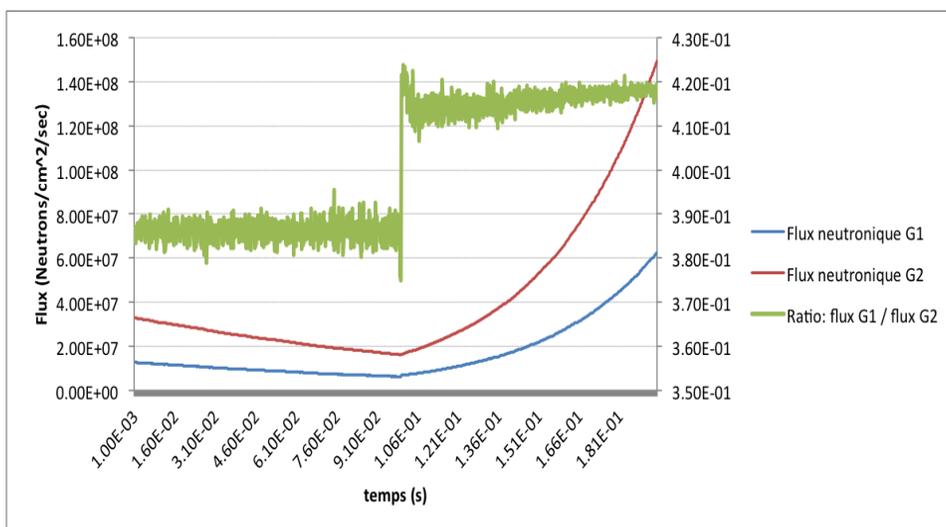


Figure 7.11 Flux neutroniques totaux de la cellule pour les groupes rapide et thermique

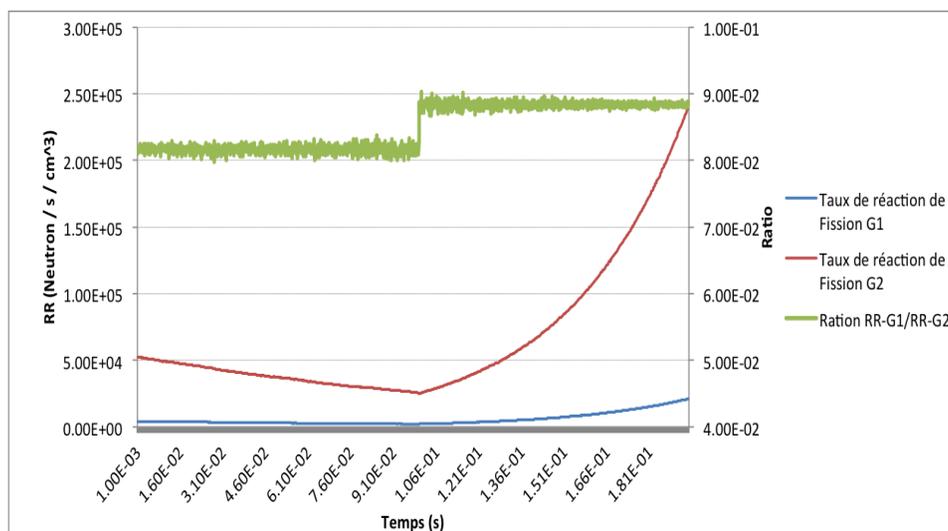


Figure 7.12 Taux de réaction total pour les groupes rapide et thermique

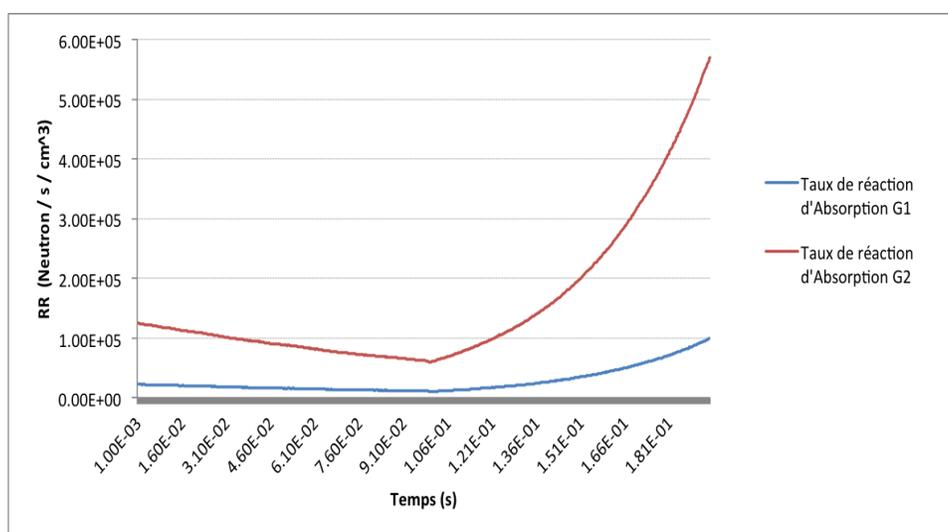


Figure 7.13 Taux de réaction d'absorption pour les groupes rapide et thermique

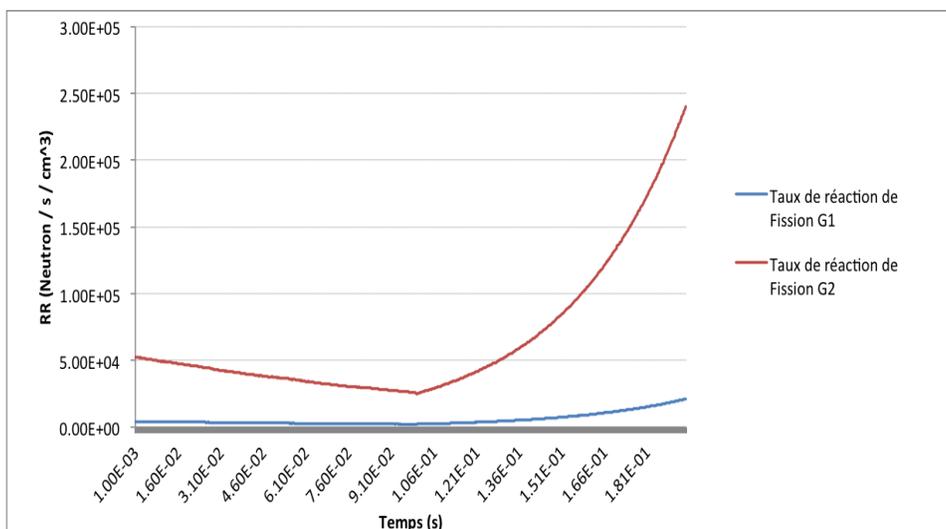


Figure 7.14 Taux de réaction de fission pour les groupes rapide et thermique

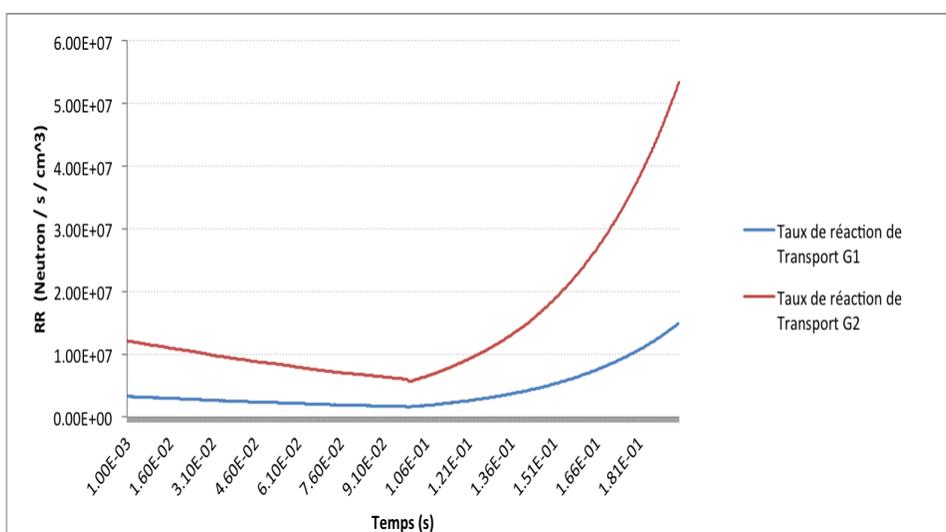


Figure 7.15 Taux de réaction de transport pour les groupes rapide et thermique

En divisant les différents taux de réactions par le flux neutronique pour le groupe d'énergie correspondant, nous obtenons les sections efficaces relatives aux différents taux de réactions en question. Les figures 7.16 à 7.25 présentent les différents paramètres nucléaires de cette cellule.

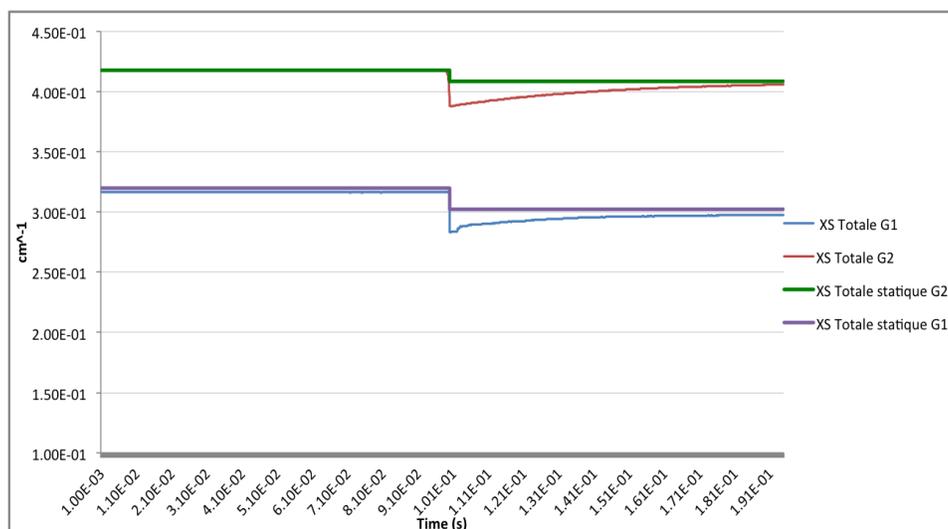


Figure 7.16 Sections efficaces totales condensées à deux groupes et homogénéisées sur toute la cellule

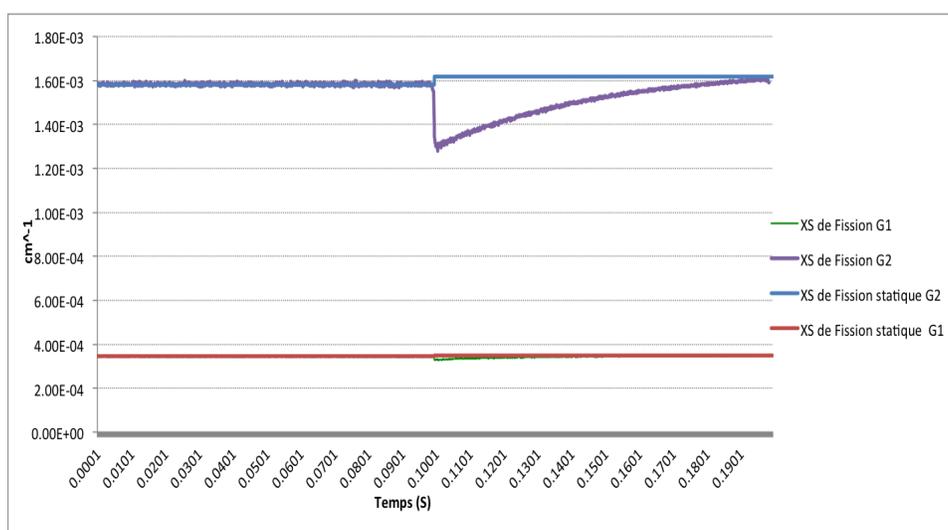


Figure 7.17 Sections efficaces de fission condensées à deux groupes et homogénéisées sur toute la cellule

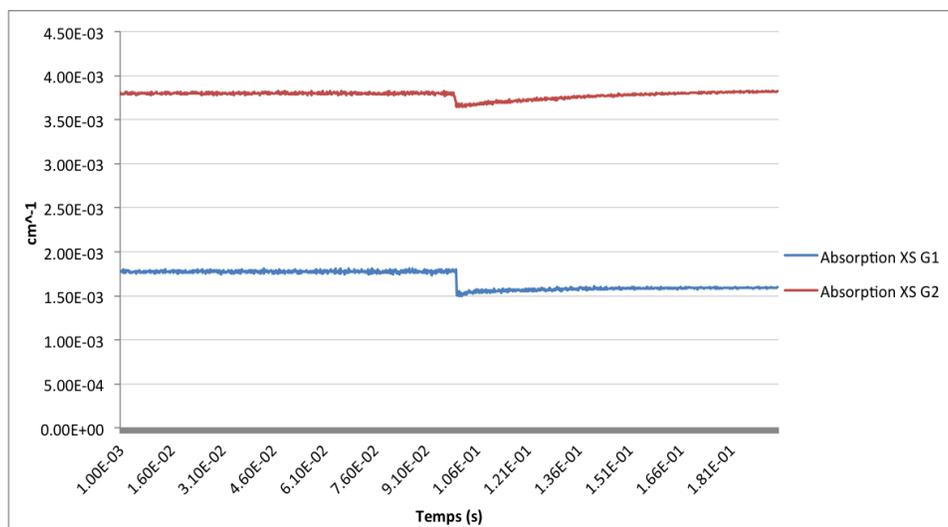


Figure 7.18 Sections efficaces d'absorption condensées à deux groupes et homogénéisées sur toute la cellule

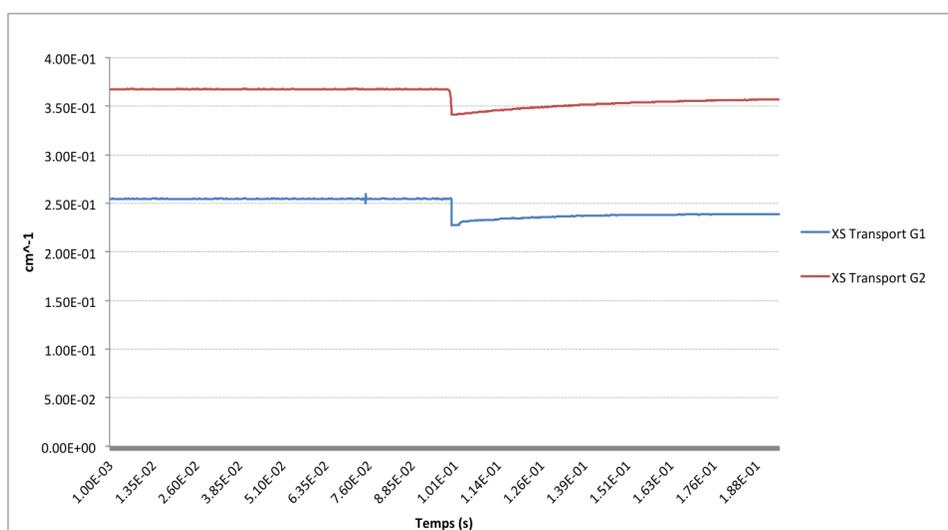


Figure 7.19 Sections efficaces de transport condensées à deux groupes et homogénéisées sur toute la cellule

Le ratio entre les sections efficaces statiques et dynamique sont présentées dans les courbes 7.20 et 7.21.

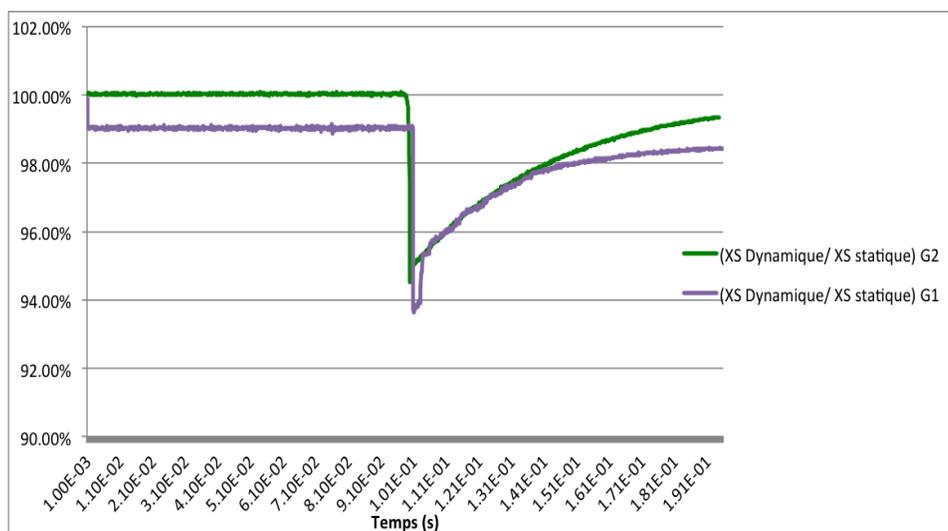


Figure 7.20 Le rapport entre les sections efficaces totales statiques et dynamiques

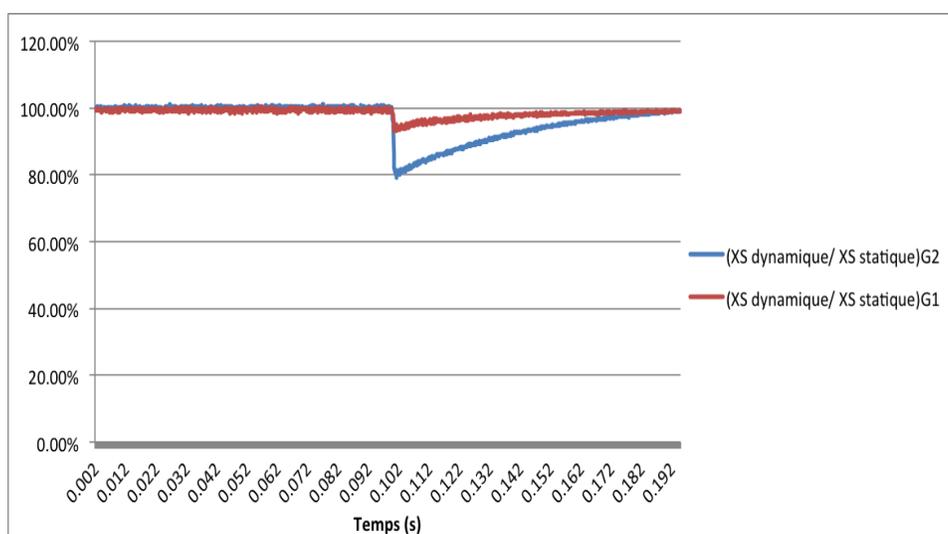


Figure 7.21 Le rapport entre les sections efficaces de fission statiques et dynamiques

Pour obtenir la vitesse moyenne des neutrons pour chaque intervalle de temps, il suffit de faire appel à l'équation (3.27).

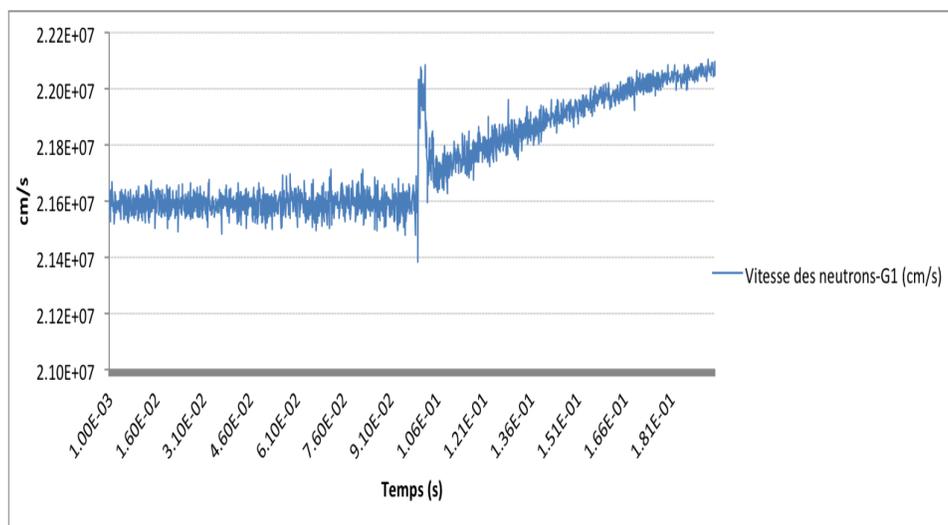


Figure 7.22 Vitesses moyennes des neutrons du groupe rapide.

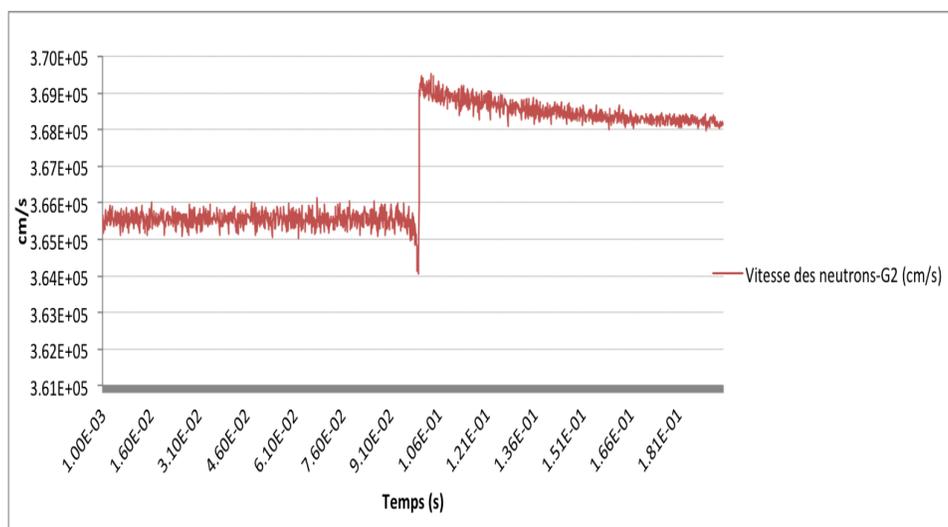


Figure 7.23 Vitesses moyennes des neutrons du groupe thermique.

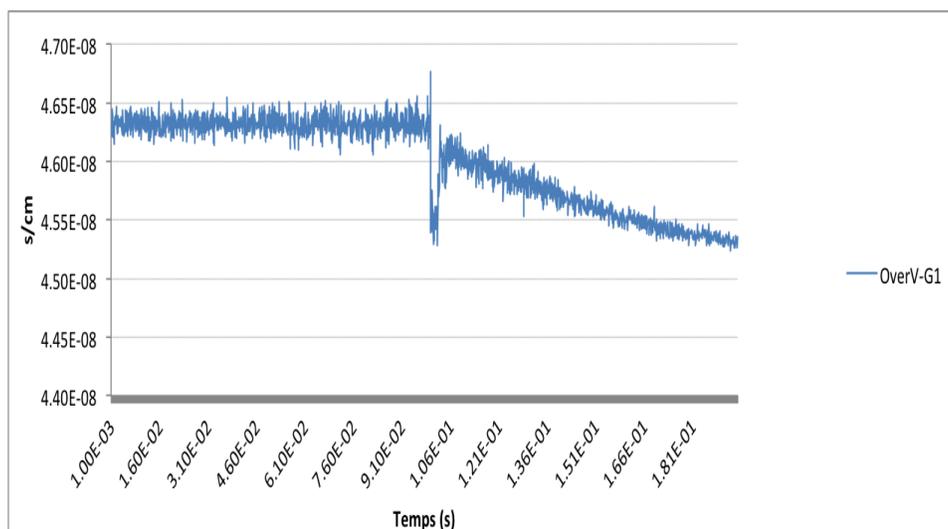


Figure 7.24 $1/v$ des neutrons du groupe rapide.

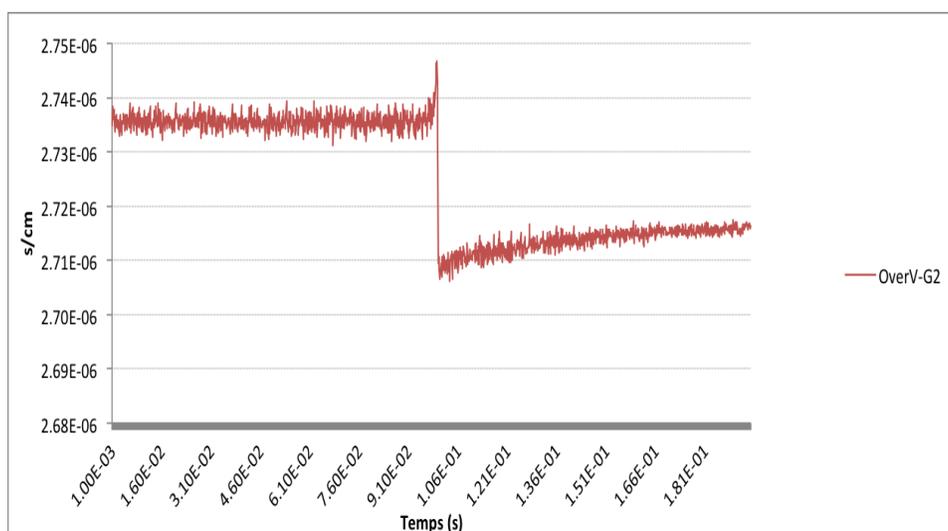


Figure 7.25 $1/v$ des neutrons du groupe thermique

Les figures précédentes montrent des résultats très intéressants. Il est à rappeler que lors des simulations des accidents dans des réacteurs nucléaires, l'hypothèse adiabatique est utilisée. Autrement dit, la simulation d'un réacteur subissant une perturbation majeure, comme la vidange du caloporteur, se fait en deux étapes. Un calcul de transport initial de valeur propre en transport neutronique est effectué pour obtenir la distribution du flux et des taux des réactions dans la cellule à son état nominal. Par la suite, un deuxième calcul à valeur propre dans lequel les propriétés de la cellule sont modifiés afin d'introduire la perturbation

en question.

Les sections efficaces condensées et homogénéisées résultantes de ces deux calculs en transport sont utilisées dans un calcul de diffusion à l'échelle du réacteur. La variation des valeurs de sections efficaces dans la simulation en diffusion est une variation en échelle (2 valeurs constantes). Par contre, les sections efficaces dépendantes du temps obtenues par OpenMC-TD présentent une évolution graduelle continue dans le temps. La majorité de ces dernières présentent une diminution très importante au moment de la perturbation avant de subir une augmentation progressive pour atteindre la valeur obtenue par les simulations adiabatiques. Par contre, d'autres sections efficaces, comme la fission, devraient subir une augmentation de leurs valeurs à la suite de l'accident alors qu'on observe, initialement, une chute de leurs valeurs avant d'augmenter progressivement et converger vers la valeur calculée en simulation adiabatique.

Enfin, les calculs dépendants du temps montrent que les sections efficaces convergent vers les valeurs calculées dans les simulations statiques sauf que cette convergence nécessite un certain délai assez important qui risque d'être plus long que prévu lors des accidents. Ainsi, la méthode adiabatique cache un intervalle de temps durant lequel des situations inattendues peuvent se produire. La section suivante a pour objectif de comparer le comportement du réacteur en utilisant les résultats obtenus en simulations dépendantes du temps en les comparant à des simulations utilisant les résultats en transport statique.

Il est important de noter les évolutions particulières des paramètres $1/v$ pour les deux groupes. La perte du caloporteur implique que les nouveaux neutrons se rendent plus rapidement dans la zone du modérateur où ils subissent plusieurs collisions pour atteindre des énergies du groupe thermique. Il est évident que les neutrons parcourent des distances plus grandes avec des vitesses élevées avant que leurs énergies n'atteignent le groupe thermique. Ainsi, la vitesse moyenne des neutrons du groupe rapide augmente. Pour le groupe thermique, la vitesse moyenne des neutrons augmente en escalier après l'accident. Ceci est dû aux parcours plus long des neutrons lors de leurs retours à la zone du combustible. L'augmentation observée pour la vitesse des neutrons du groupe rapide est graduelle. Cet aspect est complètement ignoré par les calculs statiques. Les deux variations sont intéressantes, par contre celle du groupe rapide aura un effet plus important sur la résolution de l'équation de diffusion dépendante du temps ainsi que sur le calculs des paramètres de la cinétique ponctuelle.

Les résultats précédents ont fait l'objet d'un article de conférence (Mahjoub et Koclas, 2015).

7.3 Résultats en diffusion neutronique espace-temps

Il est important de noter une modification importante dans les procédures de simulations comparativement aux méthodes de simulations espace-temps adiabatiques.

Le réacteur simulé est de type CANDU-6 avec une puissance totale de 2065 MW thermique. Les barres de contrôle sont insérées à moitié. Les barres de contrôle liquide sont aussi remplies à moitié par de l'eau légère. Durant les simulation, aucun mécanisme de contrôle de bouge et le niveau d'eau dans les contrôleurs liquide ne bouge pas.

Dans ce projet, les résultats obtenus présentant une évolution temporelle des sections efficaces sont pour une cellule initialement sous-critique dont le k_{eff} est de $0,995$. En utilisant les sections efficaces homogénéisées et condensées de cette cellule et en les appliquant pour tout le réacteur, nous obtenons un réacteur sous critique dont le k_{eff} est de $0,921$. Durant la simulation temporelle, nous modifions les valeurs des sections efficaces pour la moitié du cœur seulement afin d'introduire l'accident en question.

Donc, le réacteur est initialement sous-critique et durant toute la simulation il le demeure. Même après l'accident, la réactivité du vide introduite n'est pas assez importante pour atteindre et dépasser la criticité.

7.3.1 Résultats de la simulation adiabatique

Les résultats d'un calcul de diffusion en faisant appel à la méthode adiabatique sont présentés en premier lieu. Dans cette méthode, la forme du flux utilisée est celle obtenue en résolvant l'équation de diffusion statique avec les paramètres qui reflètent les conditions instantanées du réacteur. La simulation consiste à résoudre, initialement, l'équation de diffusion avec des sections efficaces condensées et homogénéisées des cellules avant la vidange. Ensuite, il faut résoudre la même équation mais cette fois en utilisant des sections efficaces de la cellule vidée de son caloporteur pour la moitié du réacteur.

Les sections efficaces utilisées dans les deux simulations sont obtenues à la suite d'un calcul de transport statique. Les résultats de cette simulation sont deux distributions de flux neutronique et deux valeurs du coefficient de multiplication effectif k_{eff} .

La réactivité introduite dans le réacteur à la suite de la vidange du cœur est donnée par l'équation suivante :

$$\Delta\rho = \rho_{voided} - \rho_{cooled} = \frac{k_{eff}(voided) - k_{eff}(cooled)}{k_{eff}(voided) \times k_{eff}(cooled)}$$

Tableau 7.1 Résultats d'une simulation adiabatique pour un réacteur CANDU-6 initialement sous-critique subissant un accident LOCA

k_{eff} initial	0.921951
k_{eff} pour une cellule vidée	0.941105
Réactivité statique ρ	2.2075E-02

7.3.2 Résultats d'un calcul dynamique du réacteur en utilisant les sections efficaces statiques obtenues avec OpenMC

Il est important d'amorcer la présentation des résultats dynamiques par les résultats d'un calcul de diffusion espace-temps tout en utilisant les sections efficaces obtenues à la suite d'un calcul de transport statique. Afin d'effectuer cette simulation, nous avons utilisé le même schéma de calcul qui sera utilisé pour les sections efficaces dépendantes du temps. Les sections efficaces utilisées dans ce calcul sont obtenues en utilisant le mode statique (à valeur propre) du logiciel OpenMC. Il faut noter que même les simulations adiabatiques effectuées ne comprennent pas de corrections des sections efficaces de fission par la valeur du facteur de multiplication effectif calculée initialement.

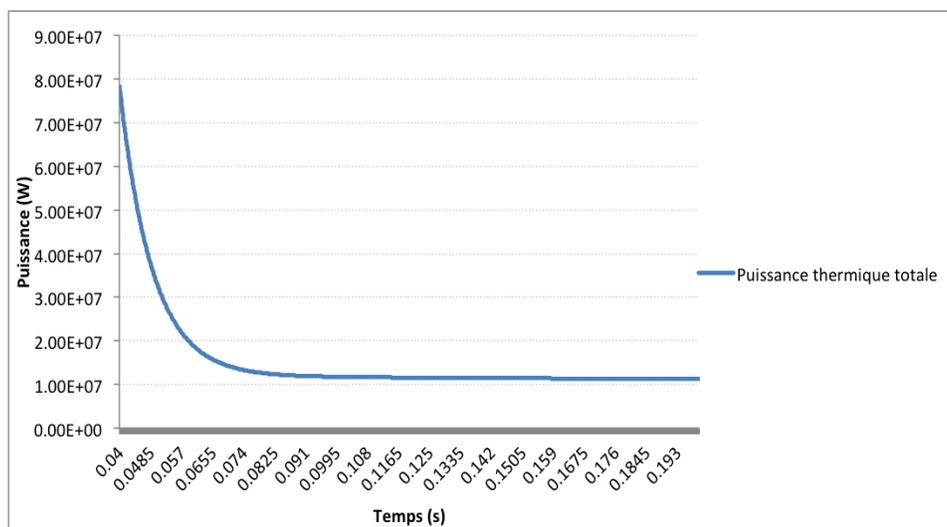


Figure 7.26 Variation de la puissance thermique du réacteur CANDU-6 sans aucune perturbation

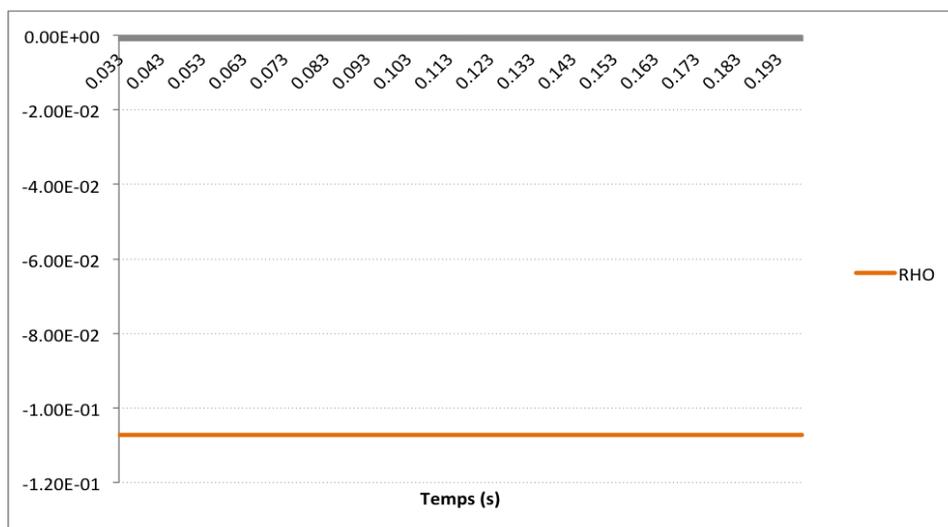


Figure 7.27 Variation de la réactivité du réacteur CANDU-6 sous critique

En introduisant l'accident pour la moitié des canaux du réacteur, et en utilisant toujours des sections efficaces obtenues à la suite de simulations statiques, nous obtenons les variations de puissance et de réactivité suivantes :

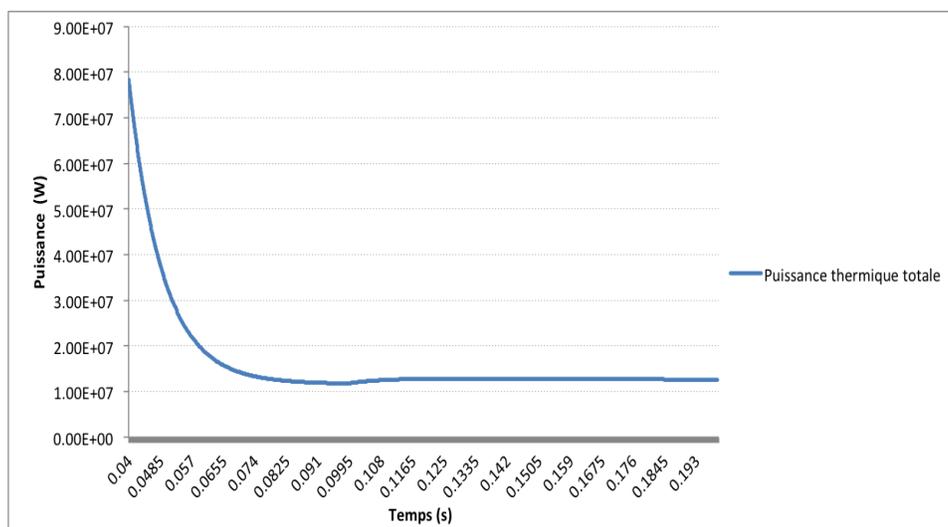


Figure 7.28 Variation de la puissance thermique du réacteur CANDU-6 avec l'introduction d'un LOCA en utilisant des sections efficaces statiques

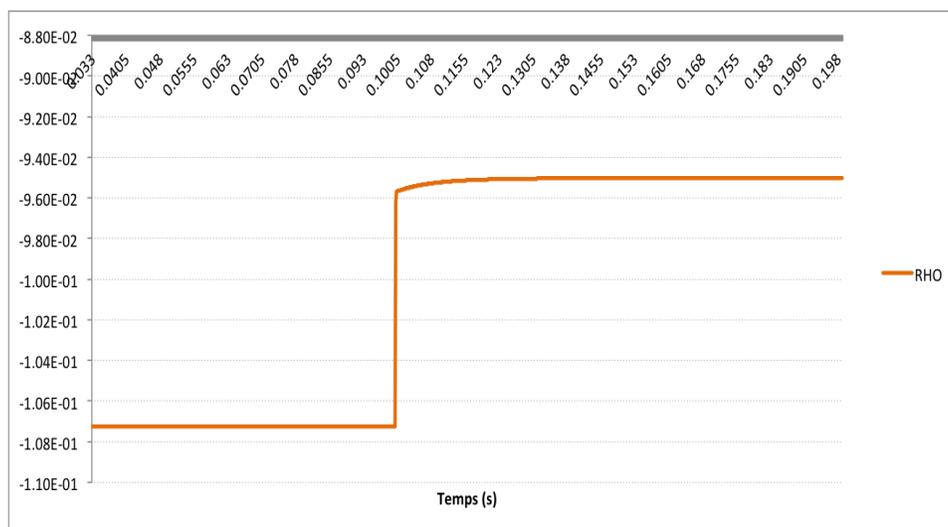


Figure 7.29 Variation de la réactivité du réacteur CANDU-6 avec l'introduction d'un LOCA en utilisant des sections efficaces statiques

Dans la figure (7.28), l'accident introduit montre une légère augmentation de la puissance à $t = 0.1s$. La perturbation introduit une réactivité positive. Comme expliqué auparavant, cette dernière n'est pas suffisante pour rendre le réacteur critique. Ainsi, la courbe présente une légère augmentation de courte durée pour la puissance suivie d'une diminution continue pour le reste de la simulation.

Les résultats présentés dans cette section forment notre base de données de référence. Tous les résultats qui seront présentés par la suite seront comparés à ces derniers.

7.3.3 Résultats d'un calcul dynamique du réacteur en utilisant les sections efficaces dynamiques obtenues avec OpenMC-TD

L'étape la plus importante se résume à observer le comportement du réacteur durant l'accident en utilisant les résultats obtenus par OpenMC-TD. Malheureusement, initialement, les résultats étaient difficiles à analyser (figures 7.30 et 7.31). La division des sections efficaces de fission par le k_{eff} a pour but de démarrer avec un réacteur initialement critique dans lequel on introduit la perturbation. Les figures montrent que la réactivité était initialement presque nulle pour ensuite atteindre une valeur autour de $60mk$ après le début de l'accident. La traduction de ces résultats en puissance thermique donne initialement une puissance stable qui augmente exponentiellement après l'accident.

Par la suite, nous avons constaté que le code NDF contient une correction des sections efficaces de fission en les divisant par le facteur de multiplication effectif (k_{eff}). Cette démarche était présente afin de satisfaire l'hypothèse qui stipule que les réacteurs ont toujours

été stables pendant de très longues durées de temps avant toute perturbation. Contrairement à ce que stipule cette hypothèse, dans ce projet le réacteur est initialement sous critique.

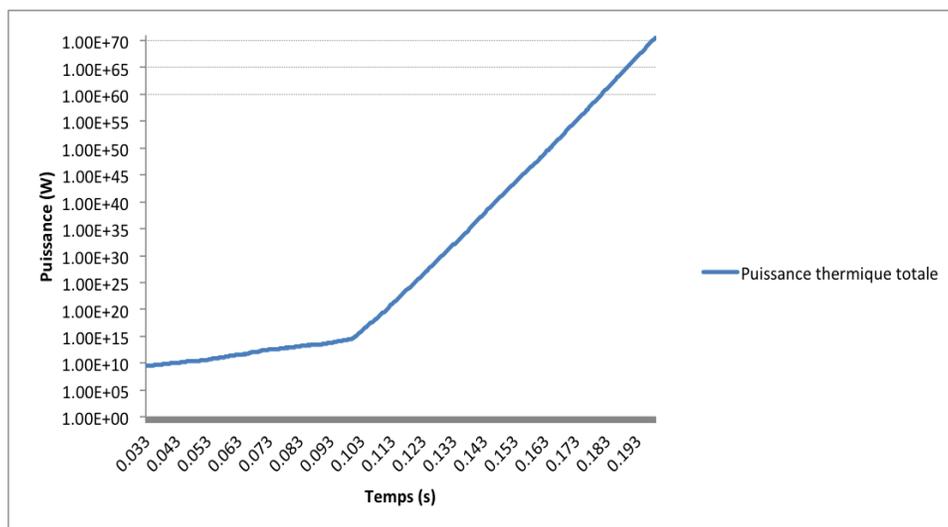


Figure 7.30 Variation de la puissance thermique avec l'introduction d'un LOCA en utilisant des sections efficaces dynamiques et en présence de la correction par le k_{eff}

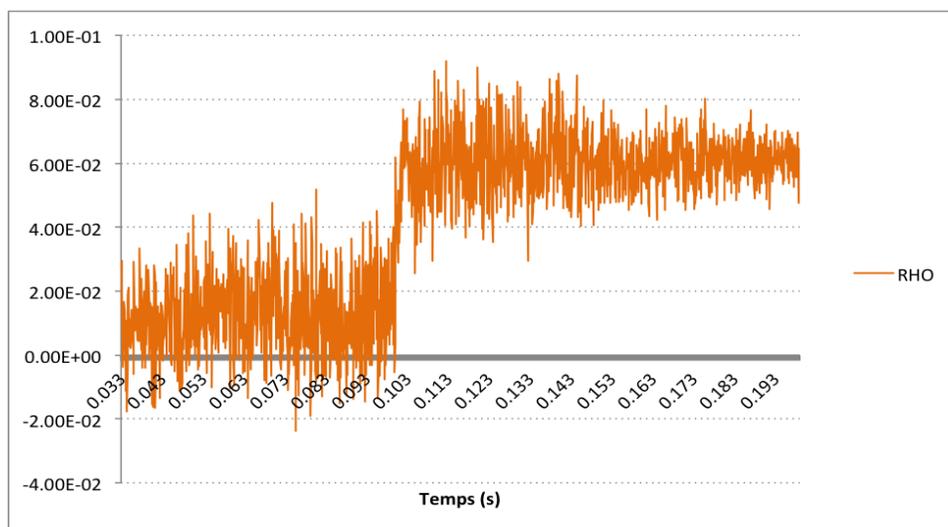


Figure 7.31 Variation de la réactivité avec l'introduction d'un LOCA en utilisant des sections efficaces dynamiques en présence de la correction par le k_{eff}

Après la suppression de la correction par le k_{eff} des sections efficaces de fission, nous avons obtenu des résultats cohérents qui montrent que le réacteur subit initialement une

chute de puissance avant d'observer une croissance temporaire lors de l'accident pour ensuite continuer à diminuer.

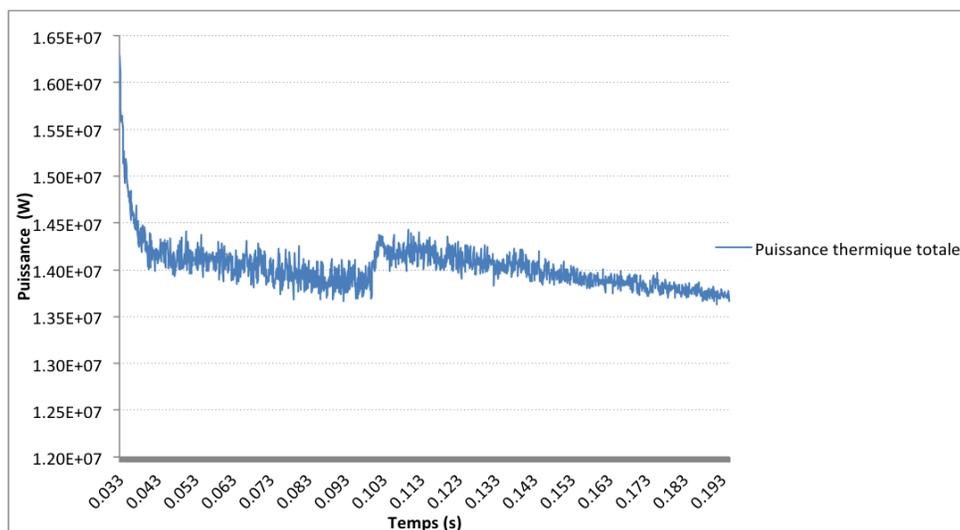


Figure 7.32 Variation de la puissance thermique avec l'introduction d'un LOCA en utilisant des sections efficaces dynamiques

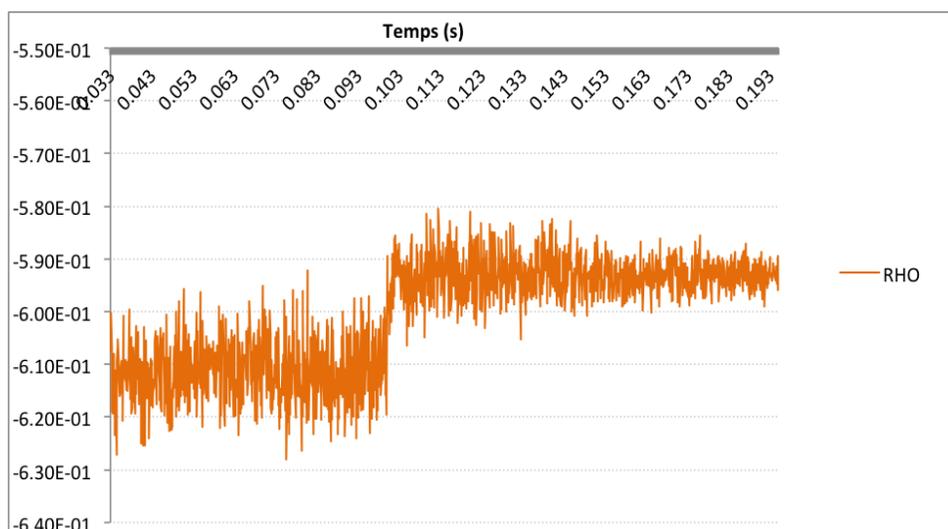


Figure 7.33 Variation de la réactivité avec l'introduction d'un LOCA en utilisant des sections efficaces dynamiques

L'allure de la courbe de la puissance est très similaire à celle obtenue par le calcul de référence (7.28). Les fluctuations observées sont dues à la nature des résultats observés dans les sections efficaces. La différence entre les deux courbes de puissance est présentée dans

la figure 7.34. Il est clair que la différence est toujours inférieure à 1% et qu'elle s'atténue rapidement au départ pour ensuite réaugmenter après l'accident pour atteindre une valeur maximale de 0.2% et se stabilise au tour de 0.1%. Il est important d'observer que cette différence diminue très lentement après l'introduction de l'accident LOCA et ceci est principalement dû à la lente convergence des sections efficaces dépendantes du temps aux valeurs statiques calculées.

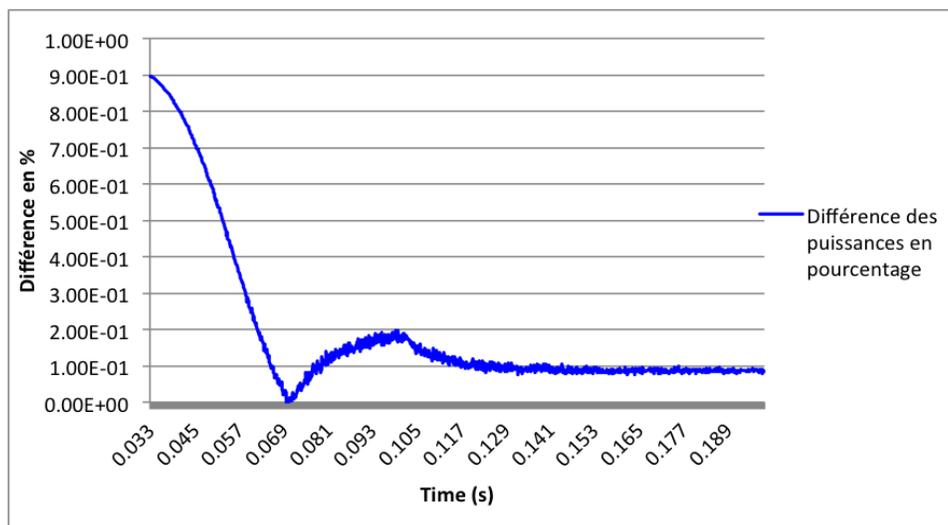


Figure 7.34 Différence entre la puissance thermique en utilisant des sections efficaces dynamiques et la puissance thermique en en utilisant les sections efficaces statiques

Après l'accident, nous comparons les pentes des courbes obtenues dans les figures (7.28) et 7.32. Le challenge consiste alors à trouver l'équation de la fonction qui synthétise une progression en fonction de l'ensemble des points obtenus principalement dans la simulation utilisant les sections efficaces dépendantes du temps. En se basant sur la cinétique ponctuelle, il est évident que la courbe de puissance devrait respecter une évolution exponentielle. Ainsi, une régression simple sur une tendance exponentielle a été utilisée. Les figures (7.35) et (7.36) présentent les portions des courbes utilisées ainsi que les régressions obtenues.

Il est important de remarquer que les deux régressions aboutissent au même facteur dans l'exponentielle. Cette constatation est prévue car l'évolution de la puissance à des instants relativement éloignés des perturbations est dictée par le groupe des neutrons retardés qui a la demi-vie la plus grande. Ainsi, il est évident que l'utilisation des sections efficaces dépendantes du temps a une influence importante seulement lors des perturbations et des instants qui les suivent immédiatement après et que leur influence à long terme est négligeable. La différence entre les deux régressions se situe au niveau du facteur de corrélation qui est beaucoup plus proche de 1 pour la courbe de référence alors qu'il est 0.763 pour la courbes de puissance

obtenue en utilisant les sections efficaces dépendantes du temps.

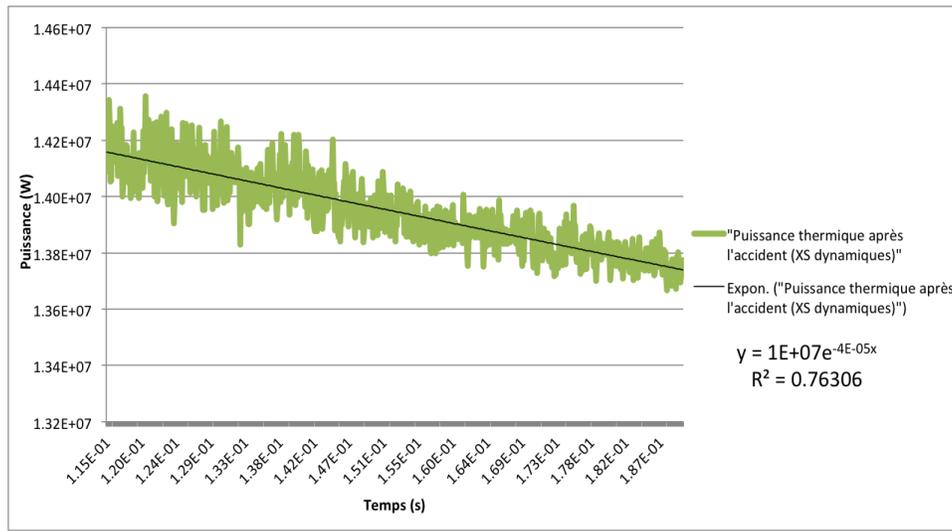


Figure 7.35 Regression exponentielle sur la puissance thermique après l'accident en utilisant des sections efficaces d'OpenMC-TD

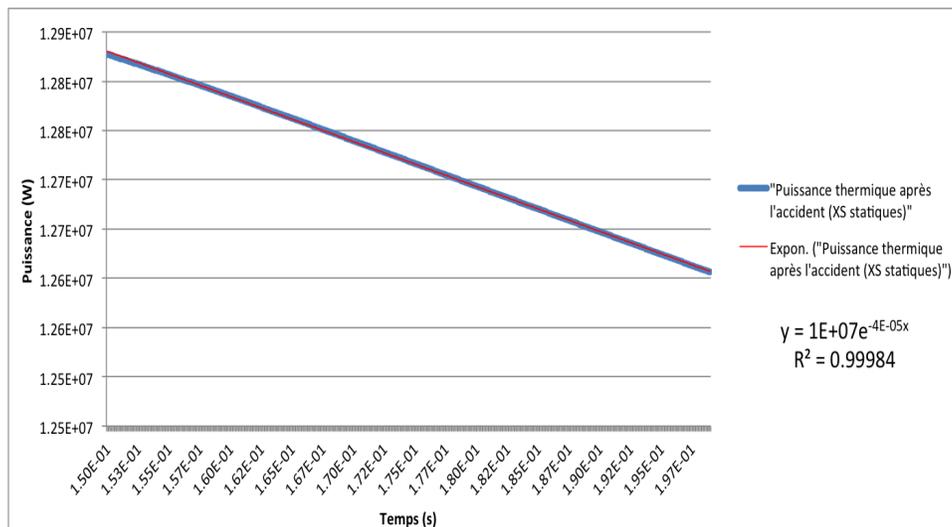


Figure 7.36 Regression exponentielle sur la puissance thermique après l'accident en utilisant des sections efficaces statiques

Pour la réactivité, le comportement est très similaire à celui de référence et les résultats sont cohérents. La présence des fluctuations peut être corrigée par l'application des filtres numériques dont la théorie a été présentée plus tôt dans cette thèse.

7.3.4 Amélioration des résultats obtenus

Afin d'améliorer les résultats obtenus, des techniques de corrections de sections efficaces et des filtres numériques ont été appliqués séparément. Par la suite, ils ont été appliqués ensemble.

Résultats sans la correction des sections efficaces tout en appliquant les filtres numériques

Nous avons repris la simulation du réacteur CANDU-6 en utilisant les sections efficaces dépendantes du temps tout en leur appliquant les filtres numériques. Dans cette simulation, nous n'avons pas fait appel aux méthodes de corrections présentées à la section (5.3.3). Les résultats sont présentés dans les figures suivantes :

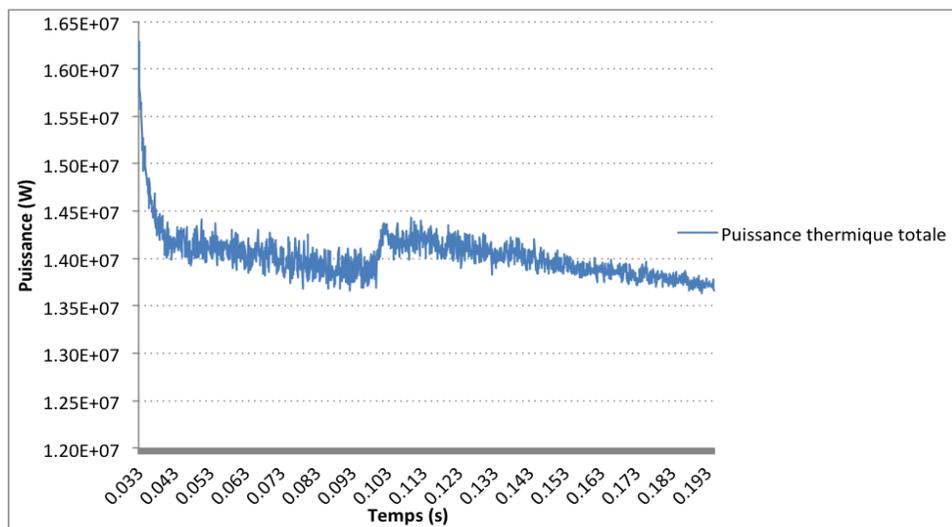


Figure 7.37 Variation de la puissance thermique avec l'introduction d'un LOCA en utilisant un filtre numérique pour les sections efficaces

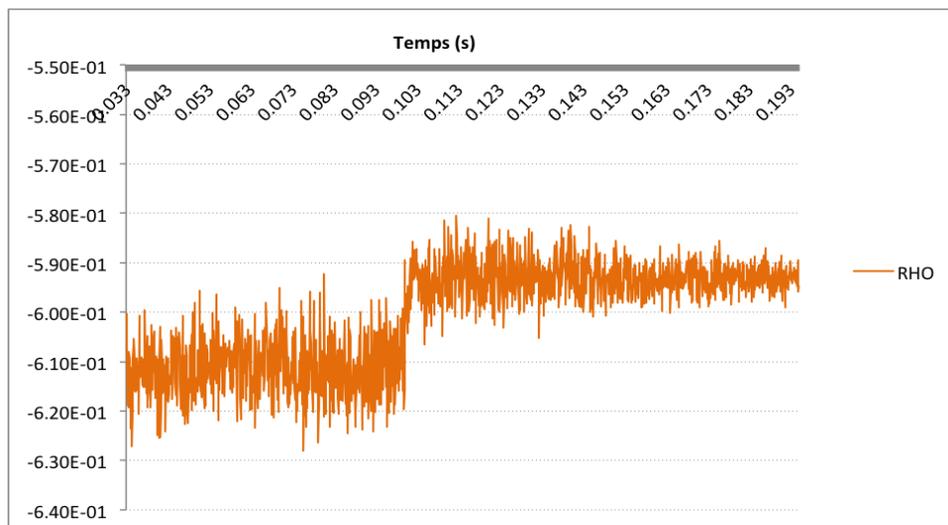


Figure 7.38 Variation de la réactivité avec l'introduction d'un LOCA en utilisant un filtre numérique pour les sections efficaces

Les résultats obtenus montrent que les paramètres choisis pour le filtre ont permis de diminuer les fluctuations observées pour la puissance tout comme pour la réactivité. Néanmoins, le filtre n'a pas éliminé ce phénomène.

Il est à noter que l'application des filtres numériques peut être effectuée sur les résultats obtenus au lieu de les appliquer sur les données utilisées en entrée (i.e. sections efficaces). Le but serait d'améliorer la lecture et l'analyse des résultats. En appliquant un filtre numérique à la courbe de la puissance présentée dans la figure (7.32), on obtient les résultats présentés à la figure 7.39.

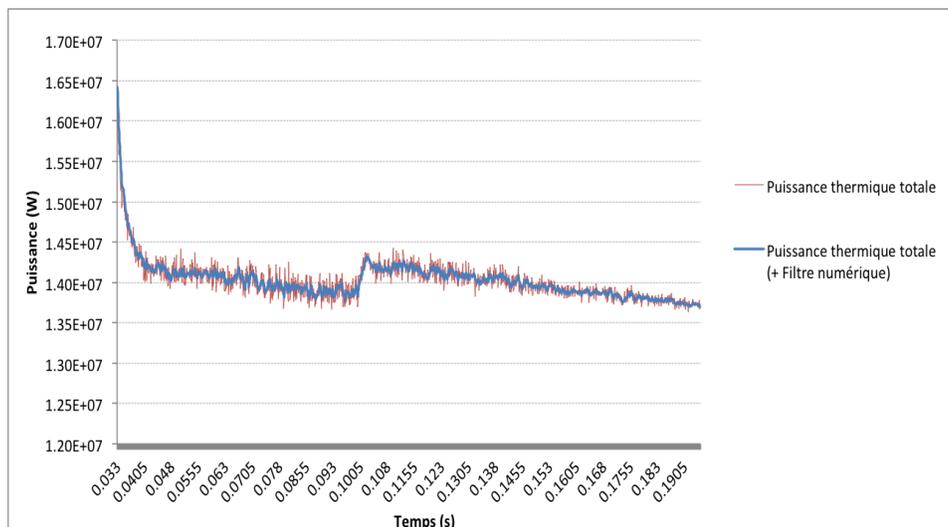


Figure 7.39 Variation de la puissance thermique en lui appliquant un filtre numérique avec l'introduction d'un LOCA

Une nette amélioration des résultats est observée. La courbe de puissance obtenue se rapproche de plus en plus de celle utilisée comme référence (7.28). Ainsi l'application des filtres numériques à la dernière étape semble être une solution intéressante pour améliorer la qualité des résultats en l'absence de la possibilité d'utiliser la loi des grands nombres dans le calcul OpenMC-TD.

Résultats en appliquant seulement la correction des sections efficaces (sans application des filtres numériques)

Dans cette section, les résultats obtenus (voir figures 7.40 et 7.41) en faisant appel à la méthode de correction des sections efficaces, **deltas sigmas**, sont présentés. Il est important de noter qu'aucun filtre n'est appliqué.

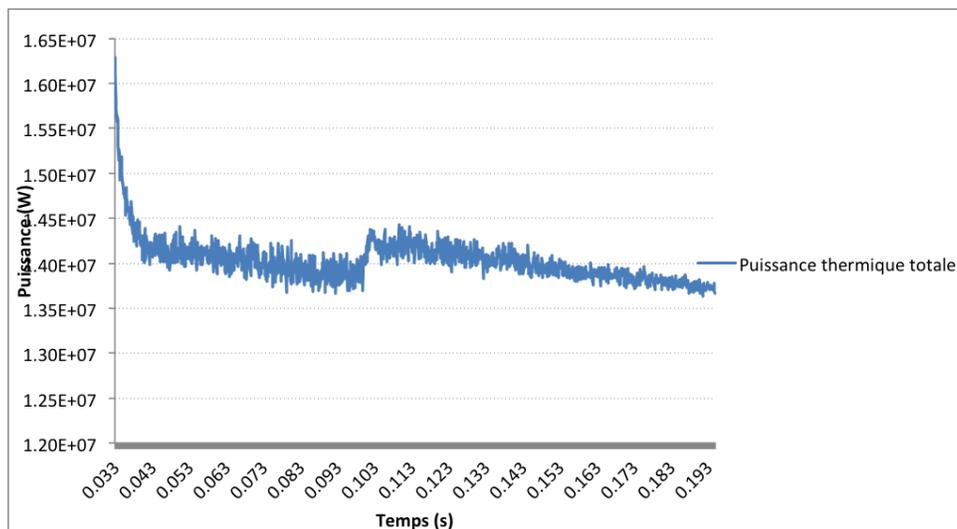


Figure 7.40 Variation de la puissance thermique avec l'introduction d'un LOCA en utilisant la technique des delta-sigmas pour la correction des sections efficaces

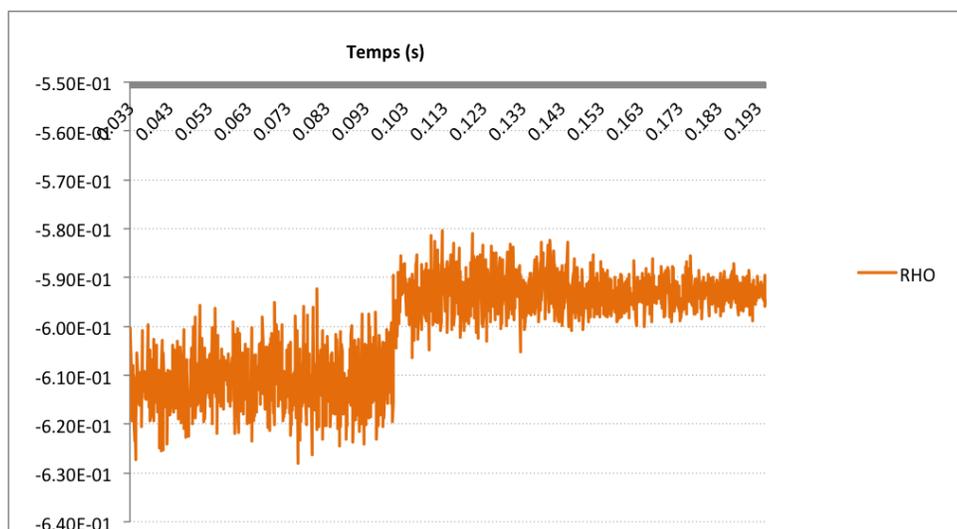


Figure 7.41 Variation de la réactivité avec l'introduction d'un LOCA en utilisant la technique des delta-sigmas pour la correction des sections efficaces

Il est clair que la *technique des deltas sigmas* montre des résultats très similaires à ceux obtenus dans la section 7.3.3 avec une faible amélioration en diminuant légèrement les fluctuations.

Résultats en appliquant la correction des sections efficaces et les filtres numériques

Dans cette section, le choix est d'appliquer les filtres numériques en même temps que la correction des sections efficaces. Avec la technique des **deltas sigmas** et en appliquant les filtres, nous obtenons les résultats illustrés dans les figures 7.42 et 7.43 .

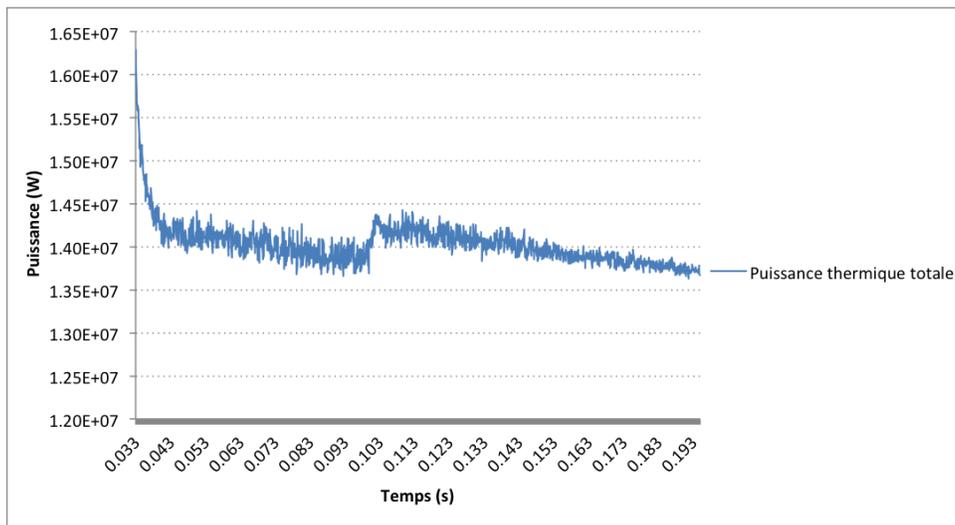


Figure 7.42 Variation de la puissance thermique avec l'introduction d'un LOCA en utilisant un filtre numérique et la technique des delta-sigmas pour la correction des sections efficaces

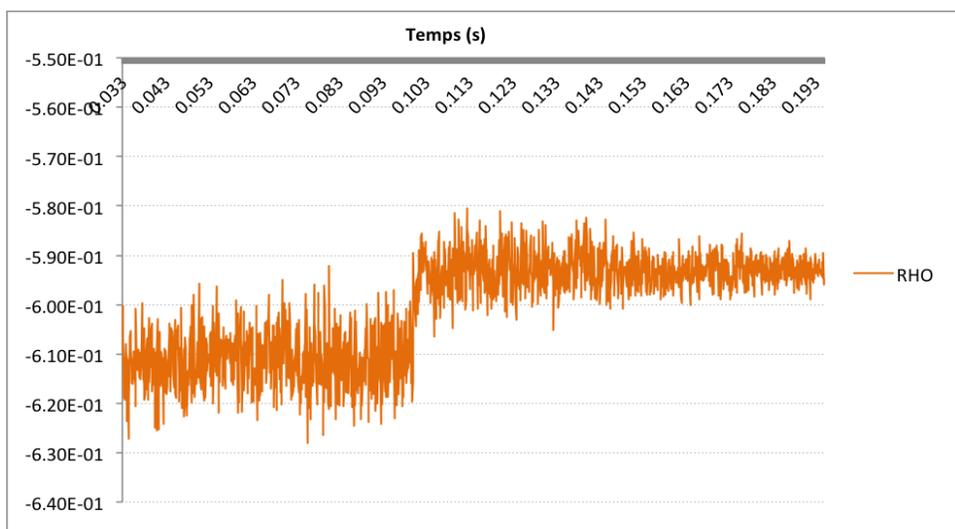


Figure 7.43 Variation de la réactivité avec l'introduction d'un LOCA en utilisant un filtre numérique et la technique des delta-sigmas pour la correction des sections efficaces

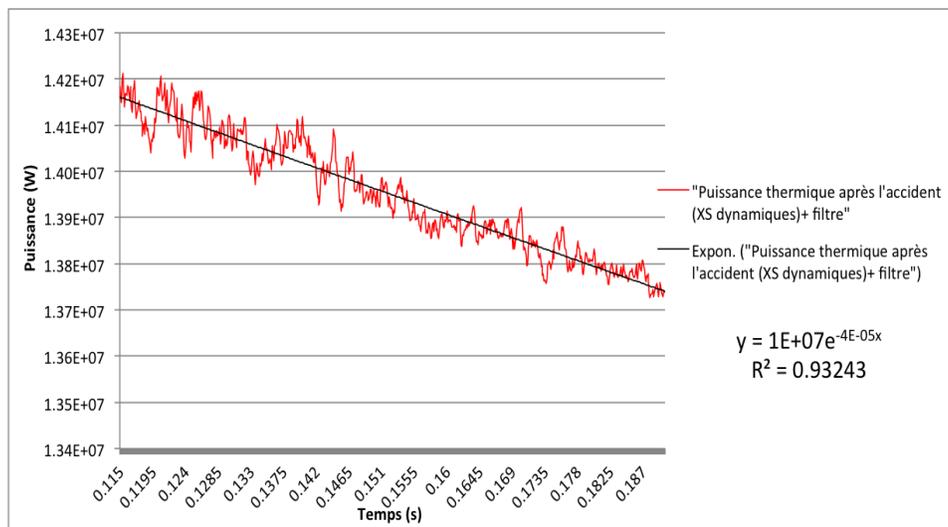


Figure 7.44 Regression exponentielle sur la puissance thermique après l'accident en utilisant des sections efficaces d'OpenMC-TD et en présence des filtres numériques

Les résultats observés dans la section précédente demeurent valides. La technique des delta-sigmas en présence d'un filtre numérique appliqué sur les sections efficaces en entrée a montré un progrès dans les résultats. Finalement nous pouvons conclure que la présence des filtres numériques accompagnés de la technique des deltas sigmas améliore les résultats en absence de la loi des grands nombres.

Il est important de noter que le facteur de corrélation en appliquant la regression exponentielle pour les résultats obtenus de la puissance thermique après l'accident et à la suite à l'application des filtres numériques sont améliorés et ont atteint une valeur de 0.932 alors qu'il était seulement de 0.763 sans l'application des techniques d'amélioration des résultats.

*Ainsi, au niveau de la figure (7.42), on observe une importante variation de la puissance à la suite de l'accident. Cette même variation a été observée dans la courbe (7.28) mais son amplitude est beaucoup moins importante. Nous pouvons alors conclure que les sections efficaces obtenues en utilisant la méthode **adiabatique** ne peuvent pas prédire correctement les différents comportements du réacteur durant des délais très courts à la suite d'une perturbation majeure dans le réacteur.*

CHAPITRE 8 CONCLUSION

8.1 Synthèse des travaux

Le projet a démarré par une analyse du code initial d'OpenMC et une évaluation de ses performances par la comparaison des résultats de simulations statiques à d'autres effectués par le code Serpent et le code DRAGON. L'analyse a permis de cerner les difficultés et planifier les modifications. La deuxième phase a été la programmation du nouveau code OpenMC-TD qui permet d'effectuer les simulations dépendantes du temps pour des cellules de réacteurs. Les résultats de ces simulations pourraient servir dans un code de diffusion adapté. Ainsi, la phase suivante a été démarrée par la programmation des nouveaux modules de DONJON.

Par la suite, la phase des simulations a été initiée pour des cas dynamiques avec OpenMC-TD pour une cellule CANDU-6 subissant la vidange de son caloporteur suivie par des simulations en cinétique espace-temps effectuées par le code DONJON et utilisant les résultats obtenus par OpenMC-TD. Finalement, une procédure d'amélioration des résultats a mené à la programmation de nouvelles techniques comme les filtres numériques et les corrections de sections efficaces.

Le projet a montré que les résultats obtenus sont importants du point de vue sûreté des centrales nucléaires car ils mettent en évidence un aspect ignoré, jusqu'à date, dans le comportement dynamique des réacteurs nucléaires.

8.2 Limitations de la solution proposée

Nous avons choisi de supprimer complètement le biaisage ; chaque neutron a un poids de un et lors des absorptions ou des fissions, il disparaît. Le "survival biasing" dans lequel les neutrons ont des poids différents durant la simulation serait une modification intéressante.

Le code OpenMC-TD fonctionne parfaitement sur un ordinateur monoprocesseur. L'implémentation parallèle a été réalisée, mais non testée en profondeur. Il serait judicieux de la compléter pour observer l'effet de cette parallélisation sur les temps de simulation.

8.3 Améliorations futures

Plusieurs améliorations à notre algorithme sont envisageables. Nous avons choisi de simuler un accident de type LOCA pour un réacteur CANDU-6 pour une courte durée et ceci dû à la contrainte de temps de simulation. Il est donc possible d'améliorer les performances

du code source et valider la programmation parallèle décrite dans l'annexe afin d'effectuer la même simulation pour un Δt beaucoup plus long. D'autre part, l'introduction du vide dans le réacteur a été immédiate dans les simulations effectuées. Une des applications futures serait de simuler le comportement du réacteur lors d'une introduction graduelle du vide.

Une autre catégorie de perturbations provoquées par le déplacement de mécanismes de réactivité demeure inexplorée, et devrait être analysée.

Le code OpenMC-TD est parfaitement fonctionnel pour tout type de réacteurs et tout type d'accident ou bien manipulation. Ainsi, parmi les applications futures, il serait judicieux d'analyser le comportement d'autres types de réacteurs (PWR, BWR, SCWR, etc.) durant leurs accidents ou perturbations.

Il serait intéressant aussi de pouvoir effectuer des comparaisons entre l'évolution la concentration des neutrons retardés dans OpenMC-TD avec les concentrations des précurseurs obtenues par d'autres codes.

RÉFÉRENCES

- A.HAGHIGHAT, J. W. et PETROVIC, B. (1996). Monte Carlo transport calculations and analysis for reactor pressure vessel neutron fluence. *Nuclear Technology*, 114, 373–380.
- BELL, G. et GLASTONE, S. (1970). *Nuclear reactor theory*. Van Nostrand Reinhold Company.
- BLOMQUIST, R. (2009). Status of the VIM Monte Carlo neutron/photon transport code. Argonne National Laboratory.
- BOUKAS, E. K. (1995). *Systèmes asservis*. Édition de l'École Polytechnique de Montréal.
- BRIESMEISTER, J. (2000). *MCNP5—A general Monte Carlo N—Particle transport code*. Los Alamos National Laboratory.
- CHAMBON, R. (2006). *Optimisation de la gestion du combustible dans les reacteurs CANDU refroidis à l'eau légère*. Thèse de doctorat, École Polytechnique de Montréal, Université de Montréal.
- CULLEN, D. (2004). A simple model of delayed neutron emission. Rapport technique UCRL-TR-204743, Lawrence Livermore National Laboratory.
- CULLEN, D. (2005). TART2005 - a coupled neutron-photon 3-D, combinatorial 31 geometry, time dependent monte carlo transport code. Rapport technique SM-218009, UCRL.
- CULLEN, D. (2006). Sampling the number of neutrons emitted per fission. Rapport technique, Lawrence Livermore National Laboratory.
- DOWNAR, T., LEE, D., XU, Y. et KOZLOWSKI, T. (2004). Parcs v2.6 u.s. nrc core neutronics simulator theory manual. *Purdue University/NRC*.
- EMMETT, M. B. (1975). The MORSE Monte Carlo radiation transport code system. Rapport technique, ORNL.
- EVERETT, C. et CASHWELL, E. (1983). Third Monte Carlo sampler. revision and extension of samplers i and ii. Rapport technique, Los Alamos National Lab., NM (USA), United States.
- FLETCHER, J. (1983). The solution of the multigroup neutron transport equation using spherical harmonics. *Nuclear science and engineering*, 84, 33–46.
- GLASTONE, S. et SESONSKE, A. (1991). *Nuclear reactor engineering*. Krieger Publishing Company, Malabar, Floride, troisième édition.
- HÉBERT, A. (2009). *Applied reactor physics*. Presses internationales Polytechnique.

- HÉBERT, A. et MARLEAU, G. (1991). Generalization of the Stamm'ler method for the self-shielding of resonant isotopes in arbitrary geometries. *Nuclear Science and Engineering*, vol. 108, 230–239.
- HÉBERT, A., MARLEAU, G. et ROY, R. (2013). A description of the data structures for DRAGON release 3.06l (2013/07/05). Rapport technique, Institut de génie nucléaire, Département de génie physique, École Polytechnique de Montréal.
- HOFFMAN, A., JEANPIERRE, F., KAVENOKY, A., LIVOLANT, M. et LORAIN, H. (1973). Code multigroupe de résolution de l'équation du transport pour les neutrons thermiques et rapides. Rapport technique, Commissariat à l'Énergie Atomique. CEA-N-1610.
- KEEPIN, G. (1965). *Reactor physics constants*. Addison-Wesley.
- KELLER, P. et LEE, J. (1998). A time-dependent collision probability method for one-dimensional space-time nuclear reactor kinetics. *Nuclear Science and Engineering*, 129, 124–148.
- LEPPÄNEN, J. (2007). *Development of a new Monte Carlo reactor physics code*. Thèse de doctorat, Helsinki University of Technology, Espoo, Finland.
- LEPPÄNEN, J. (2009). PSG2 / Serpent - a continuous-energy Monte Carlo reactor physics burnup calculation code. Rapport technique, Methodology - User's Manual - Validation Report.
- LEPPÄNEN, J. (2013). Development of a dynamic simulation mode in Serpent 2 Monte Carlo code. *M&C 2013, Sun Valley, Idaho*.
- MAHJOUB, M. (2011). *Application de la théorie des perturbations généralisées et des algorithmes stochastiques afin d'améliorer les réflecteurs des réacteurs CANDU-6*. Mémoire de maîtrise, École Polytechnique de Montréal.
- MAHJOUB, M. et KOCLAS, J. (2015). Comparative static simulations of a CANDU6 cell using different transport codes. *35th Annual Conference of the Canadian Nuclear Society*.
- MAHJOUB, M. et KOCLAS, J. (2015). OpenMC - TD, a new module for Monte Carlo time dependent simulations used to simulate a CANDU-6 cell LOCA accident. *7th International Conference on Modelling and Simulation in Nuclear Science and Engineering (7ICMSNSE)*. Ottawa Marriott Hotel, Ottawa, Ontario, Canada.
- MARLEAU, G. (2005). New geometries processing in DRAGON : the NXT module. Rapport technique, École Polytechnique de Montréal, Institut de génie nucléaire.
- MARLEAU, G., HÉBERT, A. et ROY, R. (2014). A user guide for DRAGON version5, Report IGE-335. Rapport technique, École Polytechnique de Montréal. Canada.

- MARTIN, N. (2011). *Application de la méthode des sous-groupes au calcul Monte-Carlo multigroupe*. Thèse de doctorat, École Polytechnique de Montréal.
- MATTIS, R. et HAGHIGHAT, A. (1992). Domain decomposition of a Two-Dimensional s_n method. *Nuclear Sci. Eng.*, 111, 180–196.
- MCLANE, V. (2001). Data formats and procedures for the evaluated nuclear data file ENDF-6. Rapport technique Brookhaven National Laboratory report BNL-NCS-44945 01/04-Rev. and ENDF-102, Brookhaven National Laboratory.
- METROPOLIS, N. et ULAM, S. (1949). The Monte Carlo method. *Journal of the American statistical association*, 44, 335–341.
- MUZUMDAR, A. et MENELEY, D. (2009). Large LOCA margins in CANDU reactors - an overview of the COG report. *Proceedings of CNS 30th Annual Conference*.
- PETIT, O., HUGOT, F., LEE, Y. et JOUANNE, C. (2008). Tripoli-4 version 4 user guide. Rapport technique CEA-R-6169, Commissariat à l'Énergie Atomique (CEA), France.
- ROMANO, P., TRAMM, J. et SIEGEL, A. (2016). Efficacy of hardware threading for Monte Carlo particle transport calculations on multi- and many-core systems. *PHYSOR 2016, Sun Valley, Idaho*.
- ROMANO, P. K. et FORGET, B. (2013). The OpenMC Monte Carlo particle transport code. *Ann. Nucl. Energy*, 51, 274–281.
- ROUBEN, B. (2002). The industry standard tool computer program for CANDU reactor core design and analysis. *Proceedings of 13th Pacific Basin Nuclear Conference, Shenzhen, China*.
- ROY, R. (1999). The CLE-2000 tool-box. Rapport technique, Institut de génie nucléaire, École Polytechnique de Montréal.
- ROZON, D. (1992). *Introduction à la cinétique des réacteurs nucléaires*. Édition de l'École Polytechnique de Montréal.
- RUSSELL, L., JONKMANS, G. et BUIJS, A. (2012). A method for simulating Real-Time neutron populations, materials and geometries using the GEANT4 Monte Carlo toolkit. *33rd Annual Conference of the Canadian Nuclear Society*. Saskatoon, Saskatchewan.
- SHARP, W. (1999). *Developpement and implementation of stochastic neutron transport equation and developpement and analysis of finite difference and Galerkin methods for approximate solution to Volterra's population equation with diffusion and noise*. Thèse de doctorat, Texas Tech University.
- SHAYESTEHE, M. et SHAHRIARI, M. (2009). Calculation of Time-Dependent neutronic parameters using Monte Carlo method. *Annals of Nuclear Energy*, 36, 901–909.

- SJENITZER, B. et HOOGENBOOM, J. (2011). A Monte Carlo method for calculation of the dynamic behaviour of nuclear reactors. *Progress in Nuclear Science and Technology*, 2, 716–721.
- TAYLOR, J. (2007). *The development of a three-dimensional nuclear reactor kinetics methodology based on the method of characteristics*. Thèse de doctorat, The Pennsylvania State University.
- TERREL, J. (1957). Distribution of fission neutron numbers. *Phys. Rev.*, 108, 783 – 788.
- TRINDADE, B. et CAMPOS, T. (2011). Stochastic method-based computational system for neutron/photon dosimetry applied to radiotherapy and radiology. *Radiol Bras*, 44, 109–116.
- VALTAVIRTA, V., HESSAN, M. et LEPPÄNEN, J. (2016). Delayed neutron emission model for Time Dependent simulations with the Serpent 2 Monte Carlo code - first results. *PHYSOR 2016, Sun Valley*.
- VARIN, E., HÉBERT, A. et ROY, R. (2005). A user guide for DONJON 3.01, report IGE-208 Rev 4. Rapport technique, École Polytechnique de Montréal, Canada.
- WHITLOCK, J. J., GARLAND, W. J. et MILGRAM, M. S. (1995). Effects contributing to positive coolant void reactivity in CANDU. *ANS Annual Meeting*. Philadelphia, PA.
- WOODCOCK, E., MURPHY, T., HEMMINGS, P. et LONGWORTH, T. (1965). Techniques used in the GEM code for Monte Carlo neutronics calculations in reactors and other systems of complex geometry. *Proc. Conference on the Application of Computing Methods to Reactor Problems, ANL-7050*, 557–579.
- ZIMIN, V., ASAKA, H. et ANODA, Y. (2001). SKETCH-N : a nodal neutron diffusion code for solving steady-state and kinetics problems. Model description and user's guide. Rapport technique, JAERI.

ANNEXE A PROGRAMMATION PARALLÈLE DU CODE OPENMC-TD

L'implémentation parallèle utilisée se base principalement sur une approche par processus et non pas par «thread». L'intérêt de faire appel à la programmation parallèle est de réduire le temps calcul et de pouvoir ainsi effectuer des simulations plus complexes et détaillées.

Nous avons choisi d'utiliser le standard MPI (Message Passing Interface). Une application MPI est constituée de plusieurs processus, chacun autonome, exécutant un code source identique ou complètement différent des autres processus. Ces processus communiquent entre eux via des messages en utilisant des appels à des sous-programmes de la bibliothèque MPI.

Contrairement à OpenMP, MPI utilise de la mémoire distribuée, c'est à dire, chaque processus alloue son propre espace mémoire indépendamment des autres processus. Dans notre application, l'ensemble des processus se compose d'un processus *MANAGER* et plusieurs processus nommés *Processus d'évolution*.

L'approche de programmation est très importante. Il est nécessaire d'éviter les communications inutiles pour créer un équilibre performant pour le code. Dans notre cas, nous avons fait appel à une combinaison d'approches :

- la première approche est le *Parallélisme de données* qui se base sur la distribution de données au sein des processus en y opérant les mêmes actions. Dans OpenMC-TD, tous les *Processus d'évolution* effectuent le même travail sur l'ensemble des particules se trouvant initialement dans leurs banques de particules
- la deuxième approche est appelée *MIMD* ; c'est une approche où on effectue des exécutions différentes sur des données différentes. Le processus principal, *MANAGER*, exécute un algorithme complètement différent de celui des *Processus d'évolution*.

Protocoles utilisés

Protocole du MANAGER

Le protocole "Manager" (ou superviseur) est basé sur la présence d'un ensemble de processus faisant une tâche similaire et qui sont supervisés par un processus hôte. Ce dernier fournit les directives et distribue les tâches selon un algorithme bien déterminé. La tâche du superviseur peut parfois se limiter à l'observation mais elle peut aussi être cruciale à l'évolution de la simulation.

Dans ce projet, le *MANAGER* possède un double rôle : le premier rôle est de recevoir et sauvegarder les résultats des parcours des neutrons et le second rôle est de contrôler les listes de banques de particules des processus et d'ordonner, selon les besoins, un transfert de

données d'un processus vers un autre.

Protocole du hand-shake

Le protocole du «*hand-shake*» est utilisé principalement dans le domaine de la télécommunication. Le «*hand-shaking*» est une approche automatisée de négociation qui établit les paramètres d'une communication avant de commencer le transfert des données.

Ce protocole est utilisé dans notre application seulement à la fin de la simulation. Comme chaque processus communique le nombre d'éléments de sa banque de particules, le manager peut constater que tous les *Processus d'évolution* ont fini de traiter toutes leurs particules et leurs listes sont alors vides (en mode veille). Dans cette situation, il leur envoie un signal pour leur permettre de sortir de leurs états de veille et quitter la simulation.

Une fois ce signal reçu par les *Processus d'évolution*, ils doivent ré-émettre au processus manager la confirmation de la fin de simulation. Le manager attendra la réception de ce signal avant de finaliser la simulation.

La raison d'un tel échange est d'éviter qu'un processus ait reçu des particules à traiter de la part d'un autre processus après avoir envoyé une information indiquant que sa liste est vide au processus manager. Si le «hand-shake» n'avait pas été utilisé, on aurait observé le *MANAGER* quitter la simulation avant que les processus d'évolution n'aient fini de traiter les nouvelles particules et perdre ainsi des données de la simulation.

Messages et communications échangés

Habituellement, les communications sont échangées soit horizontalement, du manager vers les processus d'évolution, soit verticalement, d'un processus d'évolution vers un autre. Dans l'implémentation effectuée, les deux directions ont été utilisées en même temps (voir figure A.1).

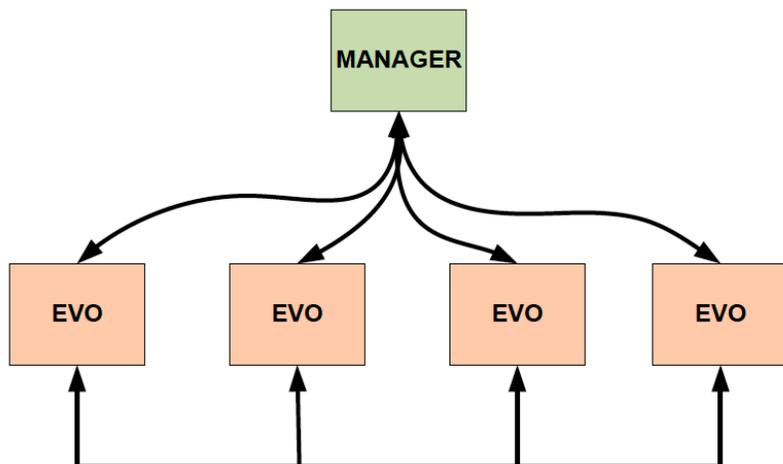


Figure A.1 Aperçu des communications entre les processus

Communications MANAGER-PROC_ EVO

Les communications entre le processus manager et les autres processus sont les suivantes :

- **Envoi des résultats du tracking d'une particule** : Après l'évolution d'une particule et sa disparition (par absorption), le processus d'évolution doit envoyer les résultats de son *Tracking* au processus MANAGER afin que ce dernier les enregistre dans le fichier des résultats.
- **Envoi du nombre d'éléments de la banque des particules** : Chaque *Processus d'évolution* doit communiquer au MANAGER le nombre de particules stockées dans sa banque. Ce dernier sauvegarde cette information afin de l'utiliser dans la vérification et ordonner, si requis, l'équilibrage des listes.
- **Envoi du signal de transfert de données** : Le manager peut constater un écart important entre les banques des particules des processus d'évolution et ordonne un équilibrage des listes en envoyant un signal aux processus concernés.
- **Envoi du signal de fin (MANAGER)** : Le MANAGER envoie le signal de fin au *Processus d'évolution* lorsqu'il constate que tous ces processus ont fini de traiter leurs particules.
- **Envoi du signal de fin (Processus d'évolution)** : Les *Processus d'évolutions* envoient au manager un signal pour annoncer la fin de leurs exécutions.

Communications PROC__ EVO-PROC__ EVO

Il existe une seule communication possible entre les *Processus d'évolutions*. Ces derniers communiquent entre eux après avoir reçu un signal de la part du manager les informant de la nécessité d'effectuer un transfert de données. Le message reçu de la part du manager contient les informations suivantes :

- Pour l'émetteur des données : l'identificateur du récepteur ainsi que la quantité de données à transférer.
- Pour le récepteur des données : l'identificateur de l'émetteur ainsi que la quantité de données à transférer.

Une fois le signal d'échange reçu, les deux processus concernés se mettent en mode bloquant afin d'envoyer/recevoir les données en utilisant chacun un tableau de données nommé «buffer_in» / «buffer_out ».

L'émetteur effectue une vérification du nombre d'éléments à transmettre avant de procéder au transfert. Cette vérification est due à l'aspect asynchrone de l'envoi initial du nombre d'éléments dans la banque de chaque processus.

Le récepteur procède à une vérification du nombre d'éléments à recevoir avant d'effectuer la réception de données en faisant appel à la fonction `MPI_get_count()`.

Types de données et TAGs utilisés

Dans les échanges de messages entre les processus (voir figure A.2 et A.3), il est important de spécifier le type de données à envoyer/recevoir et le tag (identifiant) de la communication.

Quatre types de données sont utilisés dans les communications entre les processus. Le premier type représente les résultats d'une évolution d'une particule. Des résultats sont transmis du *Processus d'évolution* au *MANAGER*. Les données transmises sont enregistrées dans un «buffer» de type «*Interaction__ MPI*» qui est dérivé du Type «*Interaction*» en supprimant les pointeurs de la liste chaînée.

```

type Interaction_MPI
  sequence
  real(8) :: energy
  real(8) :: x
  real(8) :: y
  real(8) :: z
  real(8) :: time
end type Interaction_MPI

```

Le tag associé à ce type d'échange est : TAG__ RESULT__ MPI.

Le deuxième type est utilisé aussi sous forme de tableau. Il s'agit du type «*Bank__ td__ MPI*» qui encapsule toutes les informations relatives aux particules stockées dans la banque pour être traitées.

```

type Bank_td_MPI
  sequence
  real(8) :: xyz(3)      ! location of bank particle
  real(8) :: uvw(3)     ! directional cosines
  real(8) :: E          ! energy
  real(8) :: born_time ! moment of creation of the particle
end type Bank_td_MPI

```

Le tag associé à ce type d'échange est : TAG__ TRANSFERT__ DATA__ NEIGH__ MPI.

Ces deux derniers types sont utilisés pour créer des buffers de tailles variables au moment de la transmission des données.

Le type utilisé lorsque le *MANAGER* ordonne un transfert de données est un tableau de deux entiers. Le tag associé à ce type d'échange est : TAG__ TRANSFERT__ DATA__ MPI.

Finalement, le quatrième type, «*logical*», est utilisé lors des échanges de messages concernant la fin des processus. Le tag associé à ce type d'échange est : TAG__ PROC__ ENDS__ MPI.

Algorithmes utilisés

Dans cette section, nous présentons les algorithmes utilisés dans les deux modèles de processus.

Algorithme processus d'évolution

```

DÉBUT ALGORITHME PROC_ EVO
TANT_ QUE (fini = faux) et (Element = null)
  SI ( Element != null)
    FONCTION Evoluer_ Particle ()
    FONCTION Envoyer_ resultat_ au_ MANAGER ()
    Element = Element_ suivant
    FONCTION verifier_ reception_ signal_ fin ()
    SI(recu)
      fini = vrai
    FIN SI
  SINON
    FONCTION Envoyer_NB_element_au_MANAGER ( nb_ element = 0 )
  Fin SI
  Protocole_ envoi_ du_ NB_ element()

FIN TANT_ QUE
FIN ALGORITHME PROC_ EVO

```

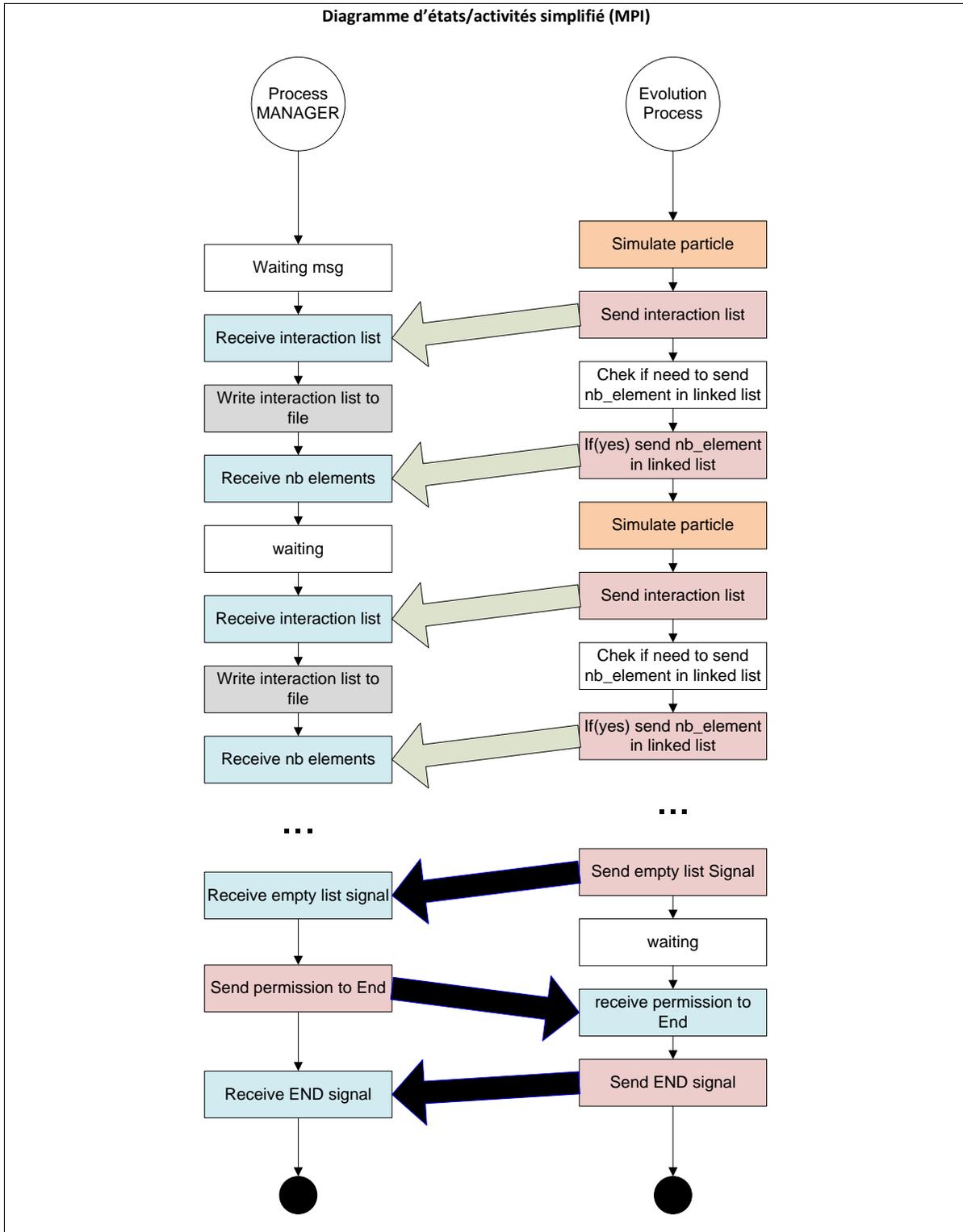


Figure A.2 Diagramme d'état simplifié présentant les communications entre les processus *MANAGER* et *Evolution*

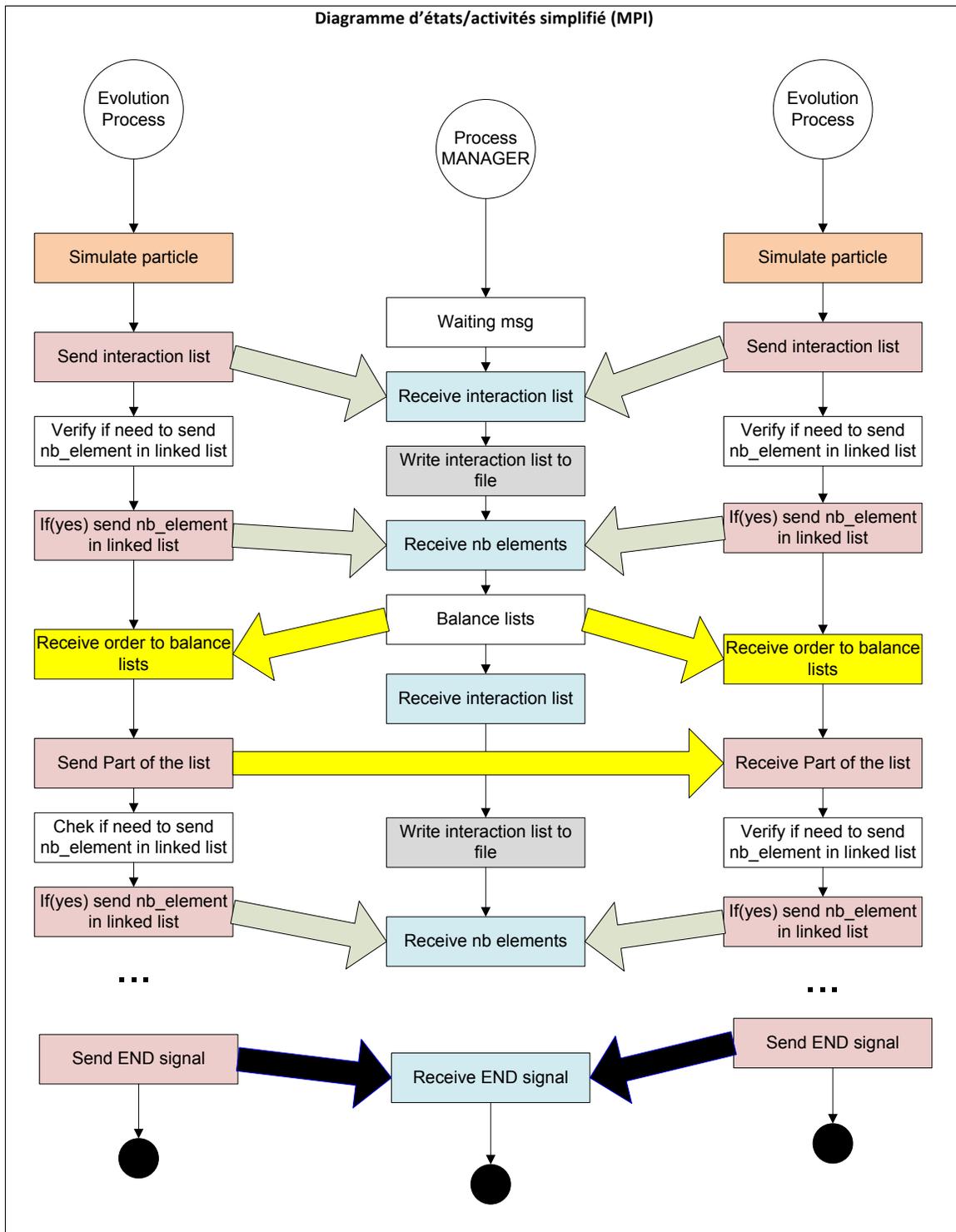


Figure A.3 Diagramme d'état simplifié présentant les communications entre les différents types de processus

Algorithme du manager

```

DÉBUT ALGORITHM MANAGER
TANT QUE (fini = faux) et (flag = vrai)
  FONCTION MPI_Iprob ( any__ source, any__ tag , flag )
  SI ( tag == Tag__ Résultat)

      FONCTION recevoir__ et__ enregistrer__ les__ résultats()
  SINON SI (tag = tag__ NB__ ELEMENT__ BANK)
      FONCTION enregistrer__ le__ NB__ element__ dans__ tableau()
      NB__ proc = NB__ proc + 1
      Si ( NB__ proc == NB__ proc__ total )

          FONCTION Vérifier__ Échange__ donnée ( )
          FONCTION réinitialiser__ tableau ( )
          FONCTION Vérifier__ fin__ simulation ( )
      FIN SI
  SINON SI (tag = Tag__ FIN__ PROC)

      NB__ proc__ end = NB__ proc__ end + 1

      SI (NB__ proc__ end == NB__ proc__ total)

          fini = vrai

      FIN SI
  FIN SI
FIN TANT QUE
FIN ALGORITHM MANAGER

```

Algorithme de transmission du nombre de particules

La création de cet algorithme répond aux objectifs suivants :

- diminuer le nombre de communications inutiles et augmenter la performance du traitement parallèle,
- éviter qu'un processus d'évolution se mette en mode veille alors que d'autres processus

possèdent des banques de particules remplies.

Toutes les communications, autres que la transmission des résultats et la fin des processus, dépendent principalement des nombres d'éléments se trouvant dans les banques des processus d'évolutions. Afin d'améliorer l'efficacité des communications, les diminuer ou les augmenter, il est impératif de contrôler les communications entre les *Processus d'évolution* et le *MANAGER* qui transmettent le nombre d'éléments de la banque. Nous avons opté pour une approche stochastique se basant sur un échantillonnage aléatoire décisionnel de l'envoi du message. La formule choisie est :

$$rand() > \frac{1}{nb_{proc}} + tampon \quad (A.1)$$

Lorsque le nombre de processus en exécution augmente, nous observons des écarts du nombre d'éléments dans les banques. Nous avons donc besoin d'équilibrer ces dernières plus souvent. La fonction choisie prend en considération cet aspect ; elle utilise une formule inversement proportionnelle au nombre de processus en exécution. Cette méthode nous permet d'améliorer l'efficacité des communications tout en évitant de retarder la procédure de vérification des banques de la part du *MANAGER* pendant une longue période.

La présence d'une variable «*tampon*» a pour objectif d'apporter une certaine précision à la formule. Cette variable dépendra du type du serveurs utilisé, du temps nécessaire pour la transmission des messages et surtout du nombre d'éléments se trouvant dans les banques de particules.

ANNEXE B STRUCTURE DE DONNÉES /CPOTD/

Le contenu de la structure de données /CPOTD/ est décrit dans cet annexe. La langue anglaise a été utilisée afin de pouvoir inclure cette partie dans le document décrivant les structures de donnée de DRAGON (HÉBERT *et al.*, 2013).

The signature for this data structure is SIGNA=L_TDXS_{uuuuuu}.

Table B.1 Main records and sub-directories in /CPOTD/

Name	Type	Units	Comment
SIGNATURE _{uuu}	C*12		parameter SIGNA containing the signature of the data structure.
STATE-VECTOR	I(40)		array S_i^c containing various parameters that are required to describe this data structure.
{/MIXDIR/}	Dir		list of sub-directories that contain homogeneous mixture information.

The array S_i^c contains the following information :

- $S_1^c = N_H$, total number of homogeneous mixtures saved.
- $S_2^c = M_G$, maximum number of groups considered.
- $S_3^c = M_I$, maximum number of isotopes.
- $S_4^c = M_L$, maximum order for the scattering anisotropy.
- $S_5^c = M_T$, maximum number of time steps +1.

Table B.2 Example of homogeneous /CPOTD/ mixture sub-directories

Name	Type	Units	Comment
MIXTH _{uuuuuu} 1	Dir		sub-directory that contains the information associated with mixture 1.
MIXTH _{uuuuuu} 2	Dir		sub-directory that contains the information associated with mixture 2.

Finalement, les informations sauvegardées dans les dossier "TIME" sont similaires à ceux sauvegardées dans les dossier "Burnup" qui sont décrites dans le document "DRAGON data structure"(HÉBERT *et al.*, 2013).

Table B.3 Contents of a mixture sub-directory in /CPOTD/

Name	Type	Units	Comment
PARAM _{uuuuuuu}	I(4)		array P_i^{cpo} containing the various parameters associated with this mixture.
TITLE _{uuuuuuu}	C*72		parameter TITLE containing the title stored in the EDITION data structure.
ENERGY _{uuuuuuu}	R(G+1)		array P_i^{cpo} containing the various parameters associated with this mixture.
TIMES _{uuuuuuu}	R(T+1)		array of times containing the time steps.
ISOTOPESNAME	$C(P_2^{cpo})*12$		array ISO_i containing the name of the various isotopes saved for this mixture
VOLUME _{uuuuuuu}	R(1)	cm^3	parameter V_i containing the volume of the region associated with this homogeneous mixture in the EDITION data structure
{/TIMEDIR/}	Dir		sub-directory that contains the time dependent multigroup cross sections associated with each interval of time .

Table B.4 Example of homogeneous mixture directories

Name	Type	Units	Comment
TIME _{uuuuuuu} 1	Dir		sub-directory that contains the cross sections associated with the time interval [TIMES(0), TIMES(1)].
TIME _{uuuuuuu} 2	Dir		sub-directory that contains the cross sections associated with the time interval [TIMES(1), TIMES(2)].

ANNEXE C GÉOMÉTRIE DE CELLULE CANDU-6 AVEC OPENMC-TD : GEOMETRY.XML

```

<?xml version="1.0"?>
<geometry>

  <!--pressure tube: inner cylinder -->
  <surface id="2" type="z-cylinder" coeffs="0.0 0.0 5.16890" />

  <!--pressure tube: outer cylinder , CO2 gaz: inner cylinder-->
  <surface id="3" type="z-cylinder" coeffs="-0.0 0.0 5.60320 " />

  <!--CO2 gaz: outer cylinder , calandria tube : inner cylinder-->
  <surface id="4" type="z-cylinder" coeffs="0.0 0.0 6.44780 " />

  <!--calandria tube: outer cylinder , moderator: inner cylinder-->
  <surface id="5" type="z-cylinder" coeffs="0.0 0.0 6.58750 " />

  <!--definition of surfaces of the cube -->
  <surface id="10" type="x-plane" coeffs="14.2875" boundary="reflective" />
  <surface id="11" type="x-plane" coeffs="-14.2875" boundary="reflective" />
  <surface id="12" type="y-plane" coeffs="14.2875" boundary="reflective" />
  <surface id="13" type="y-plane" coeffs="-14.2875" boundary="reflective" />
  <surface id="14" type="z-plane" coeffs="24.765" boundary="reflective" />
  <surface id="15" type="z-plane" coeffs="-24.765" boundary="reflective" />

  <!--definition of crayons surfaces -->
  <surface id= "500" type="z-cylinder" coeffs="0.0 0.0 0.6122"/>
  <surface id= "501" type="z-cylinder" coeffs="1.488500 0.000000 0.6122"/>
  <surface id= "502" type="z-cylinder" coeffs="0.744250 1.289079 0.6122"/>
  <surface id= "503" type="z-cylinder" coeffs="-0.744250 1.289079 0.6122"/>
  <surface id= "504" type="z-cylinder" coeffs="-1.488500 0.000000 0.6122"/>
  <surface id= "505" type="z-cylinder" coeffs="-0.744250 -1.289079 0.6122"/>
  <surface id= "506" type="z-cylinder" coeffs="0.744250 -1.289079 0.6122"/>

```

```
<surface id= "507" type="z-cylinder" coeffs="2.777520 0.744234 0.6122"/>
<surface id= "508" type="z-cylinder" coeffs="2.033286 2.033286 0.6122"/>
<surface id= "509" type="z-cylinder" coeffs="0.744234 2.777520 0.6122"/>
<surface id= "510" type="z-cylinder" coeffs="-0.744234 2.777520 0.6122"/>
<surface id= "511" type="z-cylinder" coeffs="-2.033286 2.033286 0.6122"/>
<surface id= "512" type="z-cylinder" coeffs="-2.777520 0.744234 0.6122"/>
<surface id= "513" type="z-cylinder" coeffs="-2.777520 -0.744234 0.6122"/>
<surface id= "514" type="z-cylinder" coeffs="-2.033286 -2.033286 0.6122"/>
<surface id= "515" type="z-cylinder" coeffs="-0.744234 -2.777520 0.6122"/>
<surface id= "516" type="z-cylinder" coeffs="0.744234 -2.777520 0.6122"/>
<surface id= "517" type="z-cylinder" coeffs="2.033286 -2.033286 0.6122"/>
<surface id= "518" type="z-cylinder" coeffs="2.777520 -0.744234 0.6122"/>
<surface id= "519" type="z-cylinder" coeffs="4.330500 0.000000 0.6122"/>
<surface id= "520" type="z-cylinder" coeffs="4.069339 1.481118 0.6122"/>
<surface id= "521" type="z-cylinder" coeffs="3.317355 2.783592 0.6122"/>
<surface id= "522" type="z-cylinder" coeffs="2.165250 3.750323 0.6122"/>
<surface id= "523" type="z-cylinder" coeffs="0.751983 4.264710 0.6122"/>
<surface id= "524" type="z-cylinder" coeffs="-0.751983 4.264710 0.6122"/>
<surface id= "525" type="z-cylinder" coeffs="-2.165250 3.750323 0.6122"/>
<surface id= "526" type="z-cylinder" coeffs="-3.317355 2.783592 0.6122"/>
<surface id= "527" type="z-cylinder" coeffs="-4.069339 1.481118 0.6122"/>
<surface id= "528" type="z-cylinder" coeffs="-4.330500 0.000000 0.6122"/>
<surface id= "529" type="z-cylinder" coeffs="-4.069339 -1.481118 0.6122"/>
<surface id= "530" type="z-cylinder" coeffs="-3.317355 -2.783592 0.6122"/>
<surface id= "531" type="z-cylinder" coeffs="-2.165250 -3.750323 0.6122"/>
<surface id= "532" type="z-cylinder" coeffs="-0.751983 -4.264710 0.6122"/>
<surface id= "533" type="z-cylinder" coeffs="0.751983 -4.264710 0.6122"/>
<surface id= "534" type="z-cylinder" coeffs="2.165250 -3.750323 0.6122"/>
<surface id= "535" type="z-cylinder" coeffs="3.317355 -2.783592 0.6122"/>
<surface id= "536" type="z-cylinder" coeffs="4.069339 -1.481118 0.6122"/>

<surface id= "100" type="z-cylinder" coeffs="0.0 0.0 0.6540 "/>
<surface id= "101" type="z-cylinder" coeffs="1.488500 0.000000 0.6540"/>
<surface id= "102" type="z-cylinder" coeffs="0.744250 1.289079 0.6540"/>
<surface id= "103" type="z-cylinder" coeffs="-0.744250 1.289079 0.6540"/>
<surface id= "104" type="z-cylinder" coeffs="-1.488500 0.000000 0.6540"/>
```

```

<surface id= "105" type="z-cylinder" coeffs="-0.744250 -1.289079 0.6540"/>
<surface id= "106" type="z-cylinder" coeffs="0.744250 -1.289079 0.6540"/>
<surface id= "107" type="z-cylinder" coeffs="2.777520 0.744234 0.6540"/>
<surface id= "108" type="z-cylinder" coeffs="2.033286 2.033286 0.6540"/>
<surface id= "109" type="z-cylinder" coeffs="0.744234 2.777520 0.6540"/>
<surface id= "110" type="z-cylinder" coeffs="-0.744234 2.777520 0.6540"/>
<surface id= "111" type="z-cylinder" coeffs="-2.033286 2.033286 0.6540"/>
<surface id= "112" type="z-cylinder" coeffs="-2.777520 0.744234 0.6540"/>
<surface id= "113" type="z-cylinder" coeffs="-2.777520 -0.744234 0.6540"/>
<surface id= "114" type="z-cylinder" coeffs="-2.033286 -2.033286 0.6540"/>
<surface id= "115" type="z-cylinder" coeffs="-0.744234 -2.777520 0.6540"/>
<surface id= "116" type="z-cylinder" coeffs="0.744234 -2.777520 0.6540"/>
<surface id= "117" type="z-cylinder" coeffs="2.033286 -2.033286 0.6540"/>
<surface id= "118" type="z-cylinder" coeffs="2.777520 -0.744234 0.6540"/>
<surface id= "119" type="z-cylinder" coeffs="4.330500 0.000000 0.6540"/>
<surface id= "120" type="z-cylinder" coeffs="4.069339 1.481118 0.6540"/>
<surface id= "121" type="z-cylinder" coeffs="3.317355 2.783592 0.6540"/>
<surface id= "122" type="z-cylinder" coeffs="2.165250 3.750323 0.6540"/>
<surface id= "123" type="z-cylinder" coeffs="0.751983 4.264710 0.6540"/>
<surface id= "124" type="z-cylinder" coeffs="-0.751983 4.264710 0.6540"/>
<surface id= "125" type="z-cylinder" coeffs="-2.165250 3.750323 0.6540"/>
<surface id= "126" type="z-cylinder" coeffs="-3.317355 2.783592 0.6540"/>
<surface id= "127" type="z-cylinder" coeffs="-4.069339 1.481118 0.6540"/>
<surface id= "128" type="z-cylinder" coeffs="-4.330500 0.000000 0.6540"/>
<surface id= "129" type="z-cylinder" coeffs="-4.069339 -1.481118 0.6540"/>
<surface id= "130" type="z-cylinder" coeffs="-3.317355 -2.783592 0.6540"/>
<surface id= "131" type="z-cylinder" coeffs="-2.165250 -3.750323 0.6540"/>
<surface id= "132" type="z-cylinder" coeffs="-0.751983 -4.264710 0.6540"/>
<surface id= "133" type="z-cylinder" coeffs="0.751983 -4.264710 0.6540"/>
<surface id= "134" type="z-cylinder" coeffs="2.165250 -3.750323 0.6540"/>
<surface id= "135" type="z-cylinder" coeffs="3.317355 -2.783592 0.6540"/>
<surface id= "136" type="z-cylinder" coeffs="4.069339 -1.481118 0.6540"/>

<!--definition of crayon cells -->
<cell id="5000" universe="0" times="0.20" material="2"
surfaces="-100 500 -14 15 " />

```

```
<cell id="5001" universe="0" times="0.20" material="2"
surfaces="-101 501 -14 15 " />
<cell id="5002" universe="0" times="0.20" material="2"
surfaces="-102 502 -14 15 " />
<cell id="5003" universe="0" times="0.20" material="2"
surfaces="-103 503 -14 15 " />
<cell id="5004" universe="0" times="0.20" material="2"
surfaces="-104 504 -14 15 " />
<cell id="5005" universe="0" times="0.20" material="2"
surfaces="-105 505 -14 15 " />
<cell id="5006" universe="0" times="0.20" material="2"
surfaces="-106 506 -14 15 " />
<cell id="5007" universe="0" times="0.20" material="2"
surfaces="-107 507 -14 15 " />
<cell id="5008" universe="0" times="0.20" material="2"
surfaces="-108 508 -14 15 " />
<cell id="5009" universe="0" times="0.20" material="2"
surfaces="-109 509 -14 15 " />
<cell id="5010" universe="0" times="0.20" material="2"
surfaces="-110 510 -14 15 " />
<cell id="5011" universe="0" times="0.20" material="2"
surfaces="-111 511 -14 15 " />
<cell id="5012" universe="0" times="0.20" material="2"
surfaces="-112 512 -14 15 " />
<cell id="5013" universe="0" times="0.20" material="2"
surfaces="-113 513 -14 15 " />
<cell id="5014" universe="0" times="0.20" material="2"
surfaces="-114 514 -14 15 " />
<cell id="5015" universe="0" times="0.20" material="2"
surfaces="-115 515 -14 15 " />
<cell id="5016" universe="0" times="0.20" material="2"
surfaces="-116 516 -14 15 " />
<cell id="5017" universe="0" times="0.20" material="2"
surfaces="-117 517 -14 15 " />
<cell id="5018" universe="0" times="0.20" material="2"
surfaces="-118 518 -14 15 " />
```

```
<cell id="5019" universe="0" times="0.20" material="2"
surfaces="-119 519 -14 15 " />
<cell id="5020" universe="0" times="0.20" material="2"
surfaces="-120 520 -14 15 " />
<cell id="5021" universe="0" times="0.20" material="2"
surfaces="-121 521 -14 15 " />
<cell id="5022" universe="0" times="0.20" material="2"
surfaces="-122 522 -14 15 " />
<cell id="5023" universe="0" times="0.20" material="2"
surfaces="-123 523 -14 15 " />
<cell id="5024" universe="0" times="0.20" material="2"
surfaces="-124 524 -14 15 " />
<cell id="5025" universe="0" times="0.20" material="2"
surfaces="-125 525 -14 15 " />
<cell id="5026" universe="0" times="0.20" material="2"
surfaces="-126 526 -14 15 " />
<cell id="5027" universe="0" times="0.20" material="2"
surfaces="-127 527 -14 15 " />
<cell id="5028" universe="0" times="0.20" material="2"
surfaces="-128 528 -14 15 " />
<cell id="5029" universe="0" times="0.20" material="2"
surfaces="-129 529 -14 15 " />
<cell id="5030" universe="0" times="0.20" material="2"
surfaces="-130 530 -14 15 " />
<cell id="5031" universe="0" times="0.20" material="2"
surfaces="-131 531 -14 15 " />
<cell id="5032" universe="0" times="0.20" material="2"
surfaces="-132 532 -14 15 " />
<cell id="5033" universe="0" times="0.20" material="2"
surfaces="-133 533 -14 15 " />
<cell id="5034" universe="0" times="0.20" material="2"
surfaces="-134 534 -14 15 " />
<cell id="5035" universe="0" times="0.20" material="2"
surfaces="-135 535 -14 15 " />
<cell id="5036" universe="0" times="0.20" material="2"
surfaces="-136 536 -14 15 " />
```

```
<cell id="1000" universe="0" times="0.20" material="1"
surfaces="-500 -14 15 " />
<cell id="1001" universe="0" times="0.20" material="1"
surfaces="-501 -14 15 " />
<cell id="1002" universe="0" times="0.20" material="1"
surfaces="-502 -14 15 " />
<cell id="1003" universe="0" times="0.20" material="1"
surfaces="-503 -14 15 " />
<cell id="1004" universe="0" times="0.20" material="1"
surfaces="-504 -14 15 " />
<cell id="1005" universe="0" times="0.20" material="1"
surfaces="-505 -14 15 " />
<cell id="1006" universe="0" times="0.20" material="1"
surfaces="-506 -14 15 " />
<cell id="1007" universe="0" times="0.20" material="1"
surfaces="-507 -14 15 " />
<cell id="1008" universe="0" times="0.20" material="1"
surfaces="-508 -14 15 " />
<cell id="1009" universe="0" times="0.20" material="1"
surfaces="-509 -14 15 " />
<cell id="1010" universe="0" times="0.20" material="1"
surfaces="-510 -14 15 " />
<cell id="1011" universe="0" times="0.20" material="1"
surfaces="-511 -14 15 " />
<cell id="1012" universe="0" times="0.20" material="1"
surfaces="-512 -14 15 " />
<cell id="1013" universe="0" times="0.20" material="1"
surfaces="-513 -14 15 " />
<cell id="1014" universe="0" times="0.20" material="1"
surfaces="-514 -14 15 " />
<cell id="1015" universe="0" times="0.20" material="1"
surfaces="-515 -14 15 " />
<cell id="1016" universe="0" times="0.20" material="1"
surfaces="-516 -14 15 " />
<cell id="1017" universe="0" times="0.20" material="1"
```

```
surfaces="-517 -14 15 " />
<cell id="1018" universe="0" times="0.20" material="1"
surfaces="-518 -14 15 " />
<cell id="1019" universe="0" times="0.20" material="1"
surfaces="-519 -14 15 " />
<cell id="1020" universe="0" times="0.20" material="1"
surfaces="-520 -14 15 " />
<cell id="1021" universe="0" times="0.20" material="1"
surfaces="-521 -14 15 " />
<cell id="1022" universe="0" times="0.20" material="1"
  surfaces="-522 -14 15 " />
<cell id="1023" universe="0" times="0.20" material="1"
  surfaces="-523 -14 15 " />
<cell id="1024" universe="0" times="0.20" material="1"
  surfaces="-524 -14 15 " />
<cell id="1025" universe="0" times="0.20" material="1"
  surfaces="-525 -14 15 " />
<cell id="1026" universe="0" times="0.20" material="1"
  surfaces="-526 -14 15 " />
<cell id="1027" universe="0" times="0.20" material="1"
  surfaces="-527 -14 15 " />
<cell id="1028" universe="0" times="0.20" material="1"
  surfaces="-528 -14 15 " />
<cell id="1029" universe="0" times="0.20" material="1"
  surfaces="-529 -14 15 " />
<cell id="1030" universe="0" times="0.20" material="1"
  surfaces="-530 -14 15 " />
<cell id="1031" universe="0" times="0.20" material="1"
  surfaces="-531 -14 15 " />
<cell id="1032" universe="0" times="0.20" material="1"
  surfaces="-532 -14 15 " />
<cell id="1033" universe="0" times="0.20" material="1"
  surfaces="-533 -14 15 " />
<cell id="1034" universe="0" times="0.20" material="1"
surfaces="-534 -14 15 " />
<cell id="1035" universe="0" times="0.20" material="1"
```

```
surfaces="-535 -14 15 " />
<cell id="1036" universe="0" times="0.20" material="1"
surfaces="-536 -14 15 " />

<!-- definition of the cube with heavy water -->
<cell id="1100" universe="0" times="0.20" material="6"
    surfaces="-10 11 -12 13 5 -14 15 " />
<!-- definition of the calandria cylinder -->
<cell id="1101" universe="0" times="0.20" material="4"
    surfaces="4 -5 -14 15 " />
<!-- definition of the cylinder with CO2 -->
<cell id="1102" universe="0" times="0.20" material="void"
    surfaces="3 -4 -14 15 " />
<!-- definition of the pressure tube cylinder -->
<cell id="1103" universe="0" times="0.20" material="3"
    surfaces="2 -3 -14 15 " />
<!-- definition of the coolant tube cylinder -->
<cell id="1104" universe="0" times="0.1 0.20" material="5 void"
    surfaces="-2 100 101 102 103 104 105 106 107 108 109 110 111
    112 113 114 115 116 117 118 119 120 121 122 123 124 125 126
    127 128 129 130 131 132 133 134 135 136 -14 15 " />

</geometry>
```

ANNEXE D PARAMÈTRES DE LA SIMULATION DYNAMIQUE DE CELLULE CANDU-6 AVEC OPENMC-TD : SETTINGS.XML

```
<?xml version="1.0"?>
<settings>

<cross_sections>
  /MyProg/openMc/data/cross_section-serpent-endfb7-mehdi.xml
</cross_sections>

  <time_dependent>
    <batches>1</batches>
    <delayed>1</delayed>
    <tally>1</tally>
    <!--tracking_file>tracking-file</tracking_file-->
    <particles>1000000</particles>
    <time_duration>0.20</time_duration>
    <time_dt>1.0e-4</time_dt>
  </time_dependent>

<output>summary cross_sections</output>

<seed>1</seed>

  <!-- Starting source -->
  <source>
    <space type="box">
      <parameters>-14.2875 -14.2875 -24.765 14.2875
        14.2875 24.765 </parameters>
    </space>
  </source>
  <value> 1 </value>

</settings>
```

ANNEXE E FICHER XML DES PARAMÈTRES POUR LES TALLIES
POUR OPENMC-TD : TALLIES.XML

```
<?xml version="1.0"?>
<tallies>

  <!-- hole cell -->
  <tally id="1">
    <filter type="energy" bins="0.0 0.4e-5 20.0" />
    <estimator>tracklength</estimator>
    <!--estimator>analog<estimator/-->
    <scores>total flux fission nu-fission
              absorption kappa-fission density </scores>
  </tally>
  <tally id="2">
    <filter type="energy" bins="0.0 0.4e-5 20.0" />
    <!--estimator>tracklength</estimator-->
    <!--estimator>analog<estimator/-->
    <scores>transport N2N N3N N4N scatter</scores>
  </tally>

  <tally id="3">
    <filter type="energy" bins="0.0 0.4e-5 20.0" />
    <filter type="energyout" bins="0.0 0.4e-5 20.0" />
    <scores>scatter</scores>
  </tally>

  <tally id="4">
    <filter type="energyout" bins="0.0 0.4e-5 20.0" />
    <scores>nu-fission</scores>
  </tally>

</tallies>
```

ANNEXE F SCHÉMA DE CALCUL DYNAMIQUE POUR LE RÉACTEUR CANDU-6 EN UTILISANT DONJON ET NDF

Procédure principale

```

!*****
!* PROCEDURES DE CALCUL NDF CINETIQUE ESPACE TEMPS
!* METHODE IMPLICITE
!* POUR UN CANDU-6
!* COEUR NEUF
!* AVEC SECTIONS EFFICACES
!* DEPENDANTES DU TEMPS
!*****
!* Jean Koclas
!* modifier: Mehdi Mahjoub
!*****
PROCEDURE Pgeo Pdevice Pdevproc Pmacro Pmap ;

MODULE END: GREP: DELETE: ;
MODULE INIMAC: LINKDS: DECLARE: UPTDXS: OMCDRG: ;

!*****
!* DECLARATION DES OBJETS *
!*****
LINKED_LIST GEO INDEX DEVICE PROCEDE MACTD
          MACRO0 MACRO1 MACRO2 RESEAU TDXS
          POWERFLUX ADJFLUX TRKNDF ;

!*****
!* DECLARATION DES VARIABLES *
!*****
!* Numero de melange des BC *
!*****
INTEGER mRai mRao mRbi mRci mRco mRdi

```

```

mNRai mNRao mNRbi mNRci mNRco mNRdi :=
  5 6 7 8 9 10
 11 12 13 14 15 16 ;

!*****
!* Numero de melange des BL *
!*****
INTEGER m23M m23E m23F m12M m12E m12F m01M m01E m01F :=
 17 18 19 20 21 22 23 24 25 ;

!*****
!* Niveau moyen et temps de remplissage *
!*****
REAL Fmean Timbc ;

!*****
!* Puissance et Keff pour le calcul statique *
!*****
REAL Ptot Keff ;

!*****
!* INITIALISATION *
!*****
EVALUATE Ptot := 2.18E+9 ; !*Puissance totale du coeur (watt)
EVALUATE Timbc Fmean := 60.0 0.5 ;

!*****
!* GEOMETRIE *
!*****
GEO INDEX := Pgeo ;

!*****
!* MAPP ACCIDENT *
!*****
ECHO " -----MAPP ACCIDENT----- " ;
RESEAU := Pmap GEO ;

```

```

!*****
!* PMACRO: CALCUL DES SECTIONS EFFICACES *
!* DU COMBUSTIBLE DU REFLECTEUR DES BC ET DES BL *
!*****
ECHO " -----PMACRO----- " ;
MACRO0 := Pmacro ;

!*****
!* CALCUL DE DEVICE ET MACRO CORRESPONDANT *
!*****
DEVICE INDEX := Pdevice INDEX GEO ::
    <<Fmean>> <<m23M>> <<m12M>> <<m01M>>
        <<mNRai>> <<mNRao>> <<mNRbi>>
        <<mNRci>> <<mNRco>> <<mNRdi>> ;

MACRO1 := INIMAC: MACRO0 INDEX ;
PROCEDE := Pdevproc DEVICE :: <<Timbc>> ;

!*****
!* OMCDRG: CREER LA BIBLIOTHEQUE DES SECTIONS EFFICACES *
!*****
!* Remarque: ce module fonctionne d'un maniere fige; ie, il faut
!* que dans le fichier de sortie d'OpenMC y avoir trois tally
!* avec des parametres bien specifiques:
!* les bornes des groupes d'energie sont variable ainsi
!* que l'ordre des fonctions recherchees
!*
!*<tally id="1">
!*<filter type="energy" bins="0.0 0.4e-6 20.0" />
!*<scores>total flux fission nu-fission absorption transport
!* kappa-fission</scores>
!*</tally>
!*
!*<tally id="2">
!*<filter type="energy" bins="0.0 0.4e-6 20.0" />

```

```

!*<filter type="energyout" bins="0.0 0.4e-6 20.0" />
!*<scores>scatter</scores>
!*</tally>
!*
!*<tally id="3">
!*<filter type="energyout" bins="0.0 0.4e-6 20.0" />
!*<scores>nu-fission</scores>
!*</tally>
!*****
ECHO " -----OMCDRG----- " ;

OMCDRG: ::
EDIT 0
INPUT_F DEV_cpo/tallies-Res.out
OUTPUT_F TDCPO
!*** XS-CORRECT 1 technique des pourcentages (*)
!*** XS-CORRECT 2 technique des deltas sigmas (-)
! XS-CORRECT 1 REF-FILE DEV_cpo/tallies-static.out
! FILTER 1 TAU 0.015 DT 0.0001
;

ECHO " ----- chargement du nouveau Fichier ----- " ;
SEQ_ASCII openmc_res
:: FILE 'TDCPO' ;
LINKED_LIST MACTD ;
MACTD := openmc_res ;

!*****
!* UPTDX: METTRE A JOUR LA MACROLIB DES SECTIONS EFFICACES *
!* EN UTILISANT LES SECTIONS EFFICACES DEPENDENDANTES DU *
!* TEMPS *
!*****
!* Remarque: UPTDX doit absolument Etre appele aprEs INIMAC et
!* avant NEWMAC apres INIMAC car UPTDX modifie les sections
!* efficaces par rEgion et avant Newmac afin que les sections
!* efficaces par region soient corrigEes en fonction des

```

```

!* devices
!*****
LINKED_LIST MACRO ;
ECHO " -----UPTDX----- " ;
! mettre a jour tout le reacteur avec les sections efficaces d'OpenMC

MACRO1 := UPTDXS: MACRO1 RESEAU MACTD GEO ::
EDIT 0
! On lit dans le tableau les sections efficaces identifiees
! par le temps TIME
TIME 0.03000
! on met a jour les zones identifiees par le chiffre 2
ZONE 2 ;

MACRO1 := UPTDXS: MACRO1 RESEAU MACTD GEO ::
EDIT 0
! On lit dans le tableau les sections efficaces identifiees
! par le temps TIME
TIME 0.03000
! on met a jour les zones identifiees par le chiffre 1
ZONE 1 ;

MACRO DELETE: MACRO ;
MACRO := MACRO1 ;
!*****
!* ASSIGNER LES POSITIONS DES DEVICE *
!*****
DEVICE PROCEDE := LINKDS: DEVICE PROCEDE GEO ::
    EDIT 0 DEVC TYPE CONTROLLED
    LZC LZC POS ALL 0.5
    ROD ADJUSTER POS ALL 1.0
    END ;

!*****
!* EVALUATION DES MACROS EN FCT. DE LA POSITION DES DEVICES *
!*****

```

```

MODULE NEWMAC: ;
MACRO2 := NEWMAC: DEVICE MACRO1 GEO ::
    TYPE CONTROLLED XFAC 2.0
    TYPE PERT XFAC 1.0 ;

!*****
!* CALCUL STATIQUE NDF *
!*****
!*****
!* PREPARATION POUR CALCUL STATIQUE *
!*****
POWERFLUX := DECLARE: :: L_POWER ;
ADJFLUX := DECLARE: :: L_ADJOINT ;
TRKNDF := DECLARE: :: L_TRKNDF ;

!*****
!* MISE AU POINT DE TRKNDF *
!*****
MODULE SETNDF: ;
TRKNDF := SETNDF: TRKNDF MACRO2 GEO ::
    NEDTX 3 LMESHX 3.15 239.975 525.725 762.55
    NEDTY 3 LMESHY 3.15 239.975 525.725 762.55
    NEDTZ 1 LMESHZ 0.0 598.305 ;

!*****
!* CALCUL STATIQUE *
!*****
INTEGER Standard := 1 ;
INTEGER Adjoint := 2 ;
PROCEDURE Pcalstat ;
POWERFLUX := Pcalstat POWERFLUX TRKNDF MACRO2
:: <<Ptot>> <<Standard>> ;
ADJFLUX := Pcalstat ADJFLUX TRKNDF MACRO2
:: <<Ptot>> <<Adjoint>> ;

```

```

!*****
!* IMPRESSION DE KEFF ET DE LA PUISSANCE *
!*****
  GREP: POWERFLUX :: GETVAL XKEFF 1 >>Keff<< ;
  ECHO "Keff : " Keff ;
  ECHO "Puissance totale : " Ptot ;

!*****
!* SAUVEGARDE DES RESULTATS STATIQUES *
!*****
  MODULE NDFXMLOUTPT: ;
  NDFXMLOUTPT: TRKNDF POWERFLUX :: OUTPUT_ON C6TDX.xml
  STATIC TIME 0. POWER <<Ptot>> POWERDISTRIBUTION_ON COUNT 0 ;

!*****
! INITIALISATION DES PARAMETRES DE CINETIQUE *
!*****
  MODULE INITRANS: PRECO: ;
  LINKED_LIST TRANS ; TRANS := DECLARE: :: L_TRANS ;
  REAL DeltaT Tmax ;
  EVALUATE DeltaT := 0.0001 ; !*Pas de temps
  EVALUATE Tmax := 0.1990 ; !* Temps max de la simulation

  TRANS TRKNDF := INITRANS: TRANS TRKNDF :: NDG 6
      BET 0.000189 LAMBDA 0.0129
      BET 0.001245 LAMBDA 0.0311
      BET 0.001157 LAMBDA 0.134
      BET 0.002088 LAMBDA 0.331
      BET 0.000658 LAMBDA 1.260
      BET 0.000239 LAMBDA 3.210
  CHID
      1.0 1.0 1.0 1.0 1.0 1.0 ! FAST
      0.0 0.0 0.0 0.0 0.0 0.0 ! THERMAL
  <<Tmax>>
  <<DeltaT>>
  ITERMETHOD

```

```

SOR 1.4
GROUPMETHOD SIMULTANEOUS
ORDERING REDBLACK
EPSPHI 1.E-5 ;

```

```

!*****
!* INITIALISATION DES PRECURSEURS DE NEUTRONS RETARDES *

```

```

!*****
LINKED_LIST POWERKIN ;
POWERKIN := POWERFLUX ;
POWERKIN := PRECO: POWERKIN MACRO2 TRANS TRKNDF :: ;

```

```

!*****
!* PUISSANCES INITIALES *

```

```

!*****
REAL pth pth0 pthr ;
GREG: POWERKIN :: GETVAL POWER 1 >>pth<< ;
GREG: POWERKIN :: GETVAL POWER 1 >>pth0<< ;
EVALUATE pthr := pth pth0 / ;
ECHO "Rapport initial de puissance : " pthr ;

```

```

!*****
!* CHOIX DE LA FONCTON DE POIDS POUR CALCUL *

```

```

!* DES PARAMETRES DE CINETIQUE PONCTUELLE *
!*****
STRING WeightFn := "ADJOINT" ;

```

```

!*****
!* PREMIERE EVALUATION DES PARAMETRES PK *

```

```

!* NB: PKPAR PARAMETRES DYNAMIQUES METHODE PK *
!* NB: PKPAROUT PARAMETRES APPROXIMATION PK *
!*****
MODULE PKPAR: PKPAROUT: DIRECTINTMG: DIRECTINT: ;
!MODULE ADDXENONXS: XENONDYN: ;
LINKED_LIST PKPARAM ;
PKPARAM := DECLARE: :: L_PKPAR ;

```

```
PKPAR: TRKNDF TRANS POWERKIN ADJFLUX MACRO2
      :: DISCON_OFF PRINT_FLAG 1 OUTPUT_OFF
      WEIGHT <<WeightFn>> ;
```

```
!*****
!* BOUCLE DE CINETIQUE *
!*****
!* DEVICES DOIVENT MODIFIER MACRO2 *
!*****
PROCEDURE PMoveDevices ;
INTEGER Icount := 0 ;
REAL IREAL := 0. ;
REAL Time ;
REAL TimeS ;
REAL TimeC ;

!*****
!* EVALUER LE TEMPS INITIALEMENT *
!*****
EVALUATE Icount := Icount 1 + ;
EVALUATE IREAL := Icount I_TO_R ;
EVALUATE Time := IREAL DeltaT * ;
EVALUATE TimeS := 0.03 ;
EVALUATE TimeC := 0.0 ;

REPEAT

!*****
!* UPTDX: METTRE A JOUR LA MACROLIB DES SECTIONS EFFICACEs *
!* EN UTILISANT LES SECTIONS EFFICACES DEPENDENDANTES DU TEMPS*
!*****

ECHO " -----UPTDX----- " ;
MACRO1 DELETE: MACRO1 ;
MACRO1 := MACRO ;
```

```

MACRO1 := UPTDXS: MACRO1 RESEAU MACTD GEO ::
EDIT 1
! On lit dans le tableau les section efficaces identifiees
! par le temps TIME
TIME <<Time>>
! on met a jour les zannes identifiees par le chiffre 2
ZONE 2 ;

!-----
!Appel a NEWMAC obligatoire apres UPTDX pour considerer les
!devices
!*****
!* EVALUATION DES MACROS EN FCT. DE LA POSITION DES DEVICES *
!*****
MACRO2 DELETE: MACRO2 ;
MACRO2 := NEWMAC: DEVICE MACRO1 GEO ::
TYPE CONTROLLED XFAC 2.0
TYPE PERT XFAC 1.0 ;

!*****
!* EVALUATION DES PARAMETRES DE CINETIQUE PONCTUELLE *
!* POUR INFORMATION SEULEMENT *
!*****
    PKPAR: TRKNDF TRANS POWERKIN ADJFLUX MACRO2
           :: DISCON_OFF PRINT_FLAG 1 OUTPUT_OFF
           WEIGHT <<WeightFn>> ;

!*****
!* INTEGRATION TEMPORELLE FLUX ET PRECURSEURS *
!*****
    TRANS POWERKIN := DIRECTINT: POWERKIN MACRO2 TRANS TRKNDF
                   :: PRINT_FLAG 0
                   DELTA_T <<DeltaT>> ;

!*****

```

```

!* IMPRESSION DE RESULTATS *
!*****
  GREP: POWERKIN :: GETVAL POWER 1 >>pth<< ;
  EVALUATE pthr := pth pth0 / ;
  ECHO "Resultats au temps : " Time " " pthr pth ;

!*****
!* SAUVEGARDE DE RESULTATS *
!*****
  NDFXMLOUTPT: TRKNDF POWERFLUX :: OUTPUT_ON C6TDX.xml
  DYNAMIC TIME <<Time>> POWER <<Ptot>>
  POWERDISTRIBUTION_ON COUNT <<Icount>> ;

!*****
!* EVALUER LE TEMPS *
!*****
EVALUATE Icount := Icount 1 + ;
EVALUATE IREAL := Icount I_TO_R ;
EVALUATE TimeC := IREAL DeltaT * ;
EVALUATE Time := TimeS TimeC + ;

UNTIL Time Tmax > ;

END: ;
QUIT .

```

Procédure PGEO

```

!*****
!* *
!* GEOMETRIE POUR CANDU6 *
!* *
!*****

PARAMETER GEO2 INDEX ::
      ::: LINKED_LIST GEO2 INDEX ;
      ;

```

```
LINKED_LIST GEOM ;
```

```
MODULE END: GEOD: USPLIT: ;
```

```
GEOM := GEOD: :: CAR3D 26 26 12
```

```
    EDIT 0
```

```
    X- VOID X+ VOID
```

```
    Y- VOID Y+ VOID
```

```
    Z- VOID Z+ VOID
```

```
MIX
```

```
PLANE 1
```

```

0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 2 2 2 1 1 1 1 1 1 1 2 2 2 0 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0 0 0 0 0 0 0 0
0 0 0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0 0 0 0 0 0
0 0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0 0 0 0 0
0 0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0 0 0 0
0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0 0 0
0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0 0
0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0
0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0
0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0
0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0
0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0
0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0
0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0
0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0
0 0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0
0 0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0
0 0 0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0 0
0 0 0 0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0 0
0 0 0 0 0 0 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 0 0 0 0 0

```

```

0 0 0 0 0 0 0 2 2 2 1 1 1 1 1 1 2 2 2 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0

```

PLANE 3

```

0 0 0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0
0 0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0
0 0 0 0 0 2 2 2 2 2 1 1 1 1 1 1 2 2 2 2 2 0 0 0 0 0
0 0 0 0 2 2 2 1 1 1 1 1 1 1 1 1 1 1 2 2 2 0 0 0 0
0 0 0 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 0 0 0
0 0 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 0 0
0 0 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 0 0
0 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 0
0 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 0
2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2
2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2
0 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 0
0 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 0
0 0 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 0 0
0 0 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 0 0
0 0 0 2 2 2 1 1 1 1 1 1 1 1 1 1 1 1 2 2 2 0 0 0
0 0 0 0 2 2 2 2 2 1 1 1 1 1 1 2 2 2 2 2 0 0 0 0 0
0 0 0 0 0 2 2 2 2 2 2 2 2 2 2 2 2 2 0 0 0 0 0 0
0 0 0 0 0 0 0 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0

```

PLANE 2 SAME 1

PLANE 4 SAME 3

PLANE 5 SAME 3

```

PLANE 6 SAME 3
PLANE 7 SAME 3
PLANE 8 SAME 3
PLANE 9 SAME 3
PLANE 10 SAME 3
PLANE 11 SAME 1
PLANE 12 SAME 1

```

```

MESHX 3.150 45.050 68.525 97.100 125.675 154.250 182.825 211.400
      239.975 268.550 297.125 325.700 354.275 382.850 411.425
      440.000 468.575 497.150 525.725 554.300 582.875 611.450
      640.025 668.600 697.175 720.650 762.550

```

```

MESHY 3.150 45.050 68.525 97.100 125.675 154.250 182.825 211.400
      239.975 268.550 297.125 325.700 354.275 382.850 411.425
      440.000 468.575 497.150 525.725 554.300 582.875 611.450
      640.025 668.600 697.175 720.650 762.550

```

```

MESHZ 3.940 53.470 103.000 152.530 202.060 251.590 301.120 350.650
      400.180 449.710 499.240 548.770 598.30

```

```
;
```

```

GEO2 INDEX := USPLIT: GEOM ::
      MAXR 2000000 ;

```

```
END: ;
```

```
QUIT .
```

Procédure PMAP

```
!*****
```

```
!* *
```

```
!* MAP ZONES POUR CANDU6 *
```

```
!* *
```

```
!*****
```

```
PARAMETER RESEAU GEO ::
```


QUIT .

Procédure PDEVICE

```

!* *****
!* *
!* DEVICES DU CANDU6 *
!* *
!* *****

PARAMETER DEVICE INDEX GEO2 ::
      ::: LINKED_LIST DEVICE INDEX GEO2 ;
      ;

MODULE END: INIDEV: ;

!* MEAN FILLING FOR ZCRS *)
REAL fmean ;
INTEGER m23M m12M m01M mNai mNao mNbi mNci mNco mNdi ;
!* FILLINGS FOR ALL ZCRS: ***CHANGE AT YOUR CONVENIENCE *** *)
REAL f01 f02 f03 f04 f05 f06 f07
      f08 f09 f10 f11 f12 f13 f14 ;

!* DEFINE MIX NUMBERS FOR ZCR BY TYPE: 32 21 10 FOR FULL/EMPTY *)
INTEGER m23E m23F m12E m12F m01E m01F :=
      18 19 21 22 24 25 ;
!* DEFINE MIX NUMBERS FOR BC BY TYPE *)
INTEGER mRai mRao mRbi mRci mRco mRdi :=
      5 6 7 8 9 10 ;

!* ENTREE DES PARAMETERS DE LA PROCEDURE: NUMERO DE MIXTURES *)
!* ***** *)
:: >>fmean<< >>m23M<< >>m12M<< >>m01M<< >>mNai<<
      >>mNao<< >>mNbi<< >>mNci<< >>mNco<< >>mNdi<< ;

* ***** !
!* FILLINGS FOR ALL ZCRS: ***CHANGE AT YOUR CONVENIENCE *** *)

```

```

EVALUATE f01 f02 f03 f04 f05 f06 f07
      f08 f09 f10 f11 f12 f13 f14 :=
      fmean fmean fmean fmean fmean fmean fmean
      fmean fmean fmean fmean fmean fmean fmean ;
!* DEFINE LATTICE PITCH *)
REAL lp := 28.5750 ;
!* Y REFERENCE LENGTHS FOR ZCR *)
REAL yzcL1 yzcL2 yzcL3 yzcL4 yzcL5 :=
      lp 8.504 * lp 8. * lp 7.489 * lp 7. * lp 7. * ;
!* Y+ REFERENCE COORDINATES FOR ZCR *)
REAL yzcP1 yzcP2 yzcP3 yzcP4 yzcP5 :=
      354.2750 582.8750 239.9750 440.0000 640.0250 ;

!* Y LENGHTS FOR ALL ZCRS IN G2 *)
REAL L01 L02 L03 L04 L05 L06 L07
      L08 L09 L10 L11 L12 L13 L14 :=
      yzcL1 yzcL2 yzcL3 yzcL4 yzcL5 yzcL1 yzcL2
      yzcL1 yzcL2 yzcL3 yzcL4 yzcL5 yzcL1 yzcL2 ;

!* Y+ COORDINATES FOR ALL ZCRS IN G2 *)
REAL yP01 yP02 yP03 yP04 yP05 yP06 yP07
      yP08 yP09 yP10 yP11 yP12 yP13 yP14 :=
      yzcP1 yzcP2 yzcP3 yzcP4 yzcP5 yzcP1 yzcP2
      yzcP1 yzcP2 yzcP3 yzcP4 yzcP5 yzcP1 yzcP2 ;

!* Y- COORDINATES FOR ALL ZCRS IN G2 *)
REAL yM01 := yP01 L01 - ;
REAL yM02 := yP02 L02 - ;
REAL yM03 := yP03 L03 - ;
REAL yM04 := yP04 L04 - ;
REAL yM05 := yP05 L05 - ;
REAL yM06 := yP06 L06 - ;
REAL yM07 := yP07 L07 - ;
REAL yM08 := yP08 L08 - ;
REAL yM09 := yP09 L09 - ;
REAL yM10 := yP10 L10 - ;

```

```

REAL yM11 := yP11 L11 - ;
REAL yM12 := yP12 L12 - ;
REAL yM13 := yP13 L13 - ;
REAL yM14 := yP14 L14 - ;

```

```

!* YO COORDINATES FOR ALL ZCRS IN G2: *)
!* TO SEPARATE FILL AND EMPTY ZONES *)

```

```

REAL y001 := yP01 f01 L01 * - ;
REAL y002 := yP02 f02 L02 * - ;
REAL y003 := yP03 f03 L03 * - ;
REAL y004 := yP04 f04 L04 * - ;
REAL y005 := yP05 f05 L05 * - ;
REAL y006 := yP06 f06 L06 * - ;
REAL y007 := yP07 f07 L07 * - ;
REAL y008 := yP08 f08 L08 * - ;
REAL y009 := yP09 f09 L09 * - ;
REAL y010 := yP10 f10 L10 * - ;
REAL y011 := yP11 f11 L11 * - ;
REAL y012 := yP12 f12 L12 * - ;
REAL y013 := yP13 f13 L13 * - ;
REAL y014 := yP14 f14 L14 * - ;

```

```

DEVICE INDEX := INIDEV: INDEX GEO2 ::

```

```

NDEV 35 DELTA NDESC 67

```

```

DEVICE!* DATA #1 *)
CONTROLLED ADJUSTER BC#1 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    197.1135 225.6885 325.700 440.000 196.3600 245.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
BC#1B !* DESCRIPTION #1 *)
!* POSITION *)
    197.1135 225.6885 325.700 440.000 196.3600 245.8900
<<mRdi>> <<mNdi>> !* MIX *)

```

ENDDEV

DEVICE!* DATA #2 *)

CONTROLLED ADJUSTER BC#2 !* TYPE FAMILY SEQUENCE NUMBER *)

!* POSITON MAX *)

254.2635 282.8385 211.400 554.300 196.3600 245.8900

Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)

BC#2C !* DESCRIPTION #1 *)

!* POSITION *)

254.2635 282.8385 211.400 297.125 196.3600 245.8900

<<mRco>> <<mNco>> !* MIX *)

BC#2B !* DESCRIPTION #2 *)

!* POSITION *)

254.2635 282.8385 297.125 468.575 196.3600 245.8900

<<mRci>> <<mNci>> !* MIX *)

BC#2A !* DESCRIPTION #3 *)

!* POSITION *)

254.2635 282.8385 468.575 554.300 196.3600 245.8900

<<mRco>> <<mNco>> !* MIX *)

ENDDEV

DEVICE!* DATA #3 *)

CONTROLLED ADJUSTER BC#3 !* TYPE FAMILY SEQUENCE NUMBER *)

!* POSITON MAX *)

311.4135 339.9885 211.400 554.300 196.3600 245.8900

Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)

BC#3C-B-A !* DESCRIPTION #1 *)

!* POSITION *)

311.4135 339.9885 211.400 554.300 196.3600 245.8900

<<mRbi>> <<mNbi>> !* MIX *)

ENDDEV

DEVICE!* DATA #4 *)

CONTROLLED ADJUSTER BC#4 !* TYPE FAMILY SEQUENCE NUMBER *)

!* POSITON MAX *)

368.5625 397.1375 211.400 554.300 196.3600 245.8900

Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)

BC#4C !* DESCRIPTION #1 *)

!* POSITION *)

```

368.5625 397.1375 211.400 297.125 196.3600 245.8900
<<mRao>> <<mNao>> !* MIX *)
BC#4B !* DESCRIPTION #2 *)
!* POSITION *)
368.5625 397.1375 297.125 468.575 196.3600 245.8900
<<mRai>> <<mNai>> !* MIX *)
BC#4A !* DESCRIPTION #3 *)
!* POSITION *)
368.5625 397.1375 468.575 554.300 196.3600 245.8900
<<mRao>> <<mNao>> !* MIX *)
ENDDEV
DEVICE!* DATA #5 *)
CONTROLLED ADJUSTER BC#5 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
425.7135 454.2885 211.400 554.300 196.3600 245.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
BC#5C-B-A !* DESCRIPTION #3 *)
!* POSITION *)
425.7135 454.2885 211.400 554.300 196.3600 245.8900
<<mRbi>> <<mNbi>> !* MIX *)
ENDDEV
DEVICE!* DATA #6 *)
CONTROLLED ADJUSTER BC#6 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
482.8635 511.4385 211.400 554.300 196.3600 245.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
BC#6C !* DESCRIPTION #1 *)
!* POSITION *)
482.8635 511.4385 211.400 297.125 196.3600 245.8900
<<mRco>> <<mNco>> !* MIX *)
BC#6B !* DESCRIPTION #1 *)
!* POSITION *)
482.8635 511.4385 297.125 468.575 196.3600 245.8900
<<mRci>> <<mNci>> !* MIX *)
BC#6A !* DESCRIPTION #1 *)
!* POSITION *)

```

```

482.8635 511.4385 468.575 554.300 196.3600 245.8900
<<mRco>> <<mNco>> !* MIX *)
ENDDEV
DEVICE!* DATA #7 *)
CONTROLLED ADJUSTER BC#7 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
540.0135 568.5885 325.700 440.000 196.3600 245.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
BC#7B !* DESCRIPTION #1 *)
!* POSITION *)
540.0135 568.5885 325.700 440.000 196.3600 245.8900
<<mRdi>> <<mNdi>> !* MIX *)
ENDDEV
DEVICE!* DATA #8 *)
CONTROLLED ADJUSTER BC#8 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
197.1135 225.6885 325.700 440.000 276.3600 325.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
BC#8B !* DESCRIPTION #1 *)
!* POSITION *)
197.1135 225.6885 325.700 440.000 276.3600 325.8900
<<mRdi>> <<mNdi>> !* MIX *)
ENDDEV
DEVICE!* DATA #9 *)
CONTROLLED ADJUSTER BC#9 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
254.2635 282.8385 211.400 554.300 276.3600 325.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
BC#9C !* DESCRIPTION #1 *)
!* POSITION *)
254.2635 282.8385 211.400 297.125 276.3600 325.8900
<<mRco>> <<mNco>> !* MIX *)
BC#9B !* DESCRIPTION #2 *)
!* POSITION *)
254.2635 282.8385 297.125 468.575 276.3600 325.8900
<<mRci>> <<mNci>> !* MIX *)

```

```

    BC#9A !* DESCRIPTION #3 *)
!* POSITION *)
    254.2635 282.8385 468.575 554.300 276.3600 325.8900
    <<mRco>> <<mNco>> !* MIX *)
ENDDEV
DEVICE!* DATA #10 *)
CONTROLLED ADJUSTER BC#10 !* TYPE FAMILY SEQUENCE NUMBER *)
    311.4135 339.9885 211.400 554.300 276.3600 325.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    BC#10C-B-A !* DESCRIPTION #1 *)
!* POSITION *)
    311.4135 339.9885 211.400 554.300 276.3600 325.8900
    <<mRbi>> <<mNbi>> !* MIX *)
ENDDEV
DEVICE!* DATA #11 *)
CONTROLLED ADJUSTER BC#11 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    368.5625 397.1375 211.400 554.300 276.3600 325.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    BC#11C !* DESCRIPTION #1 *)
!* POSITION *)
    368.5625 397.1375 211.400 297.125 276.3600 325.8900
    <<mRao>> <<mNao>> !* MIX *)
    BC#11B !* DESCRIPTION #2 *)
!* POSITION *)
    368.5625 397.1375 297.125 468.575 276.3600 325.8900
    <<mRai>> <<mNai>> !* MIX *)
    BC#11A !* DESCRIPTION #3 *)
!* POSITION *)
    368.5625 397.1375 468.575 554.300 276.3600 325.8900
    <<mRao>> <<mNao>> !* MIX *)
ENDDEV
DEVICE!* DATA #12 *)
CONTROLLED ADJUSTER BC#12 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    425.7135 454.2885 211.400 554.300 276.3600 325.8900

```

```

Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
  BC#12C-B-A !* DESCRIPTION #1 *)
!* POSITION *)
  425.7135 454.2885 211.400 554.300 276.3600 325.8900
  <<mRbi>> <<mNbi>> !* MIX *)
ENDDEV
DEVICE!* DATA #13 *)
CONTROLLED ADJUSTER BC#13 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
  482.8635 511.4385 211.400 554.300 276.3600 325.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
  BC#13C !* DESCRIPTION #1 *)
!* POSITION *)
  482.8635 511.4385 211.400 297.125 276.3600 325.8900
  <<mRco>> <<mNco>> !* MIX *)
  BC#13B !* DESCRIPTION #1 *)
!* POSITION *)
  482.8635 511.4385 297.125 468.575 276.3600 325.8900
  <<mRci>> <<mNci>> !* MIX *)
  BC#13A !* DESCRIPTION #1 *)
!* POSITION *)
  482.8635 511.4385 468.575 554.300 276.3600 325.8900
  <<mRco>> <<mNco>> !* MIX *)
ENDDEV
DEVICE!* DATA #14 *)
CONTROLLED ADJUSTER BC#14 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
  540.0135 568.5885 325.700 440.000 276.3600 325.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
  BC#14B !* DESCRIPTION #1 *)
!* POSITION *)
  540.0135 568.5885 325.700 440.000 276.3600 325.8900
  <<mRdi>> <<mNdi>> !* MIX *)
ENDDEV
DEVICE!* DATA #15 *)
CONTROLLED ADJUSTER BC#15 !* TYPE FAMILY SEQUENCE NUMBER *)

```

```

!* POSITON MAX *)
    197.1135 225.6885 325.700 440.000 356.3600 405.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    BC#15B !* DESCRIPTION #1 *)
!* POSITION *)
    197.1135 225.6885 325.700 440.000 356.3600 405.8900
    <<mRdi>> <<mNdi>> !* MIX *)
ENDDEV
DEVICE!* DATA #16 *)
CONTROLLED ADJUSTER BC#16 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    254.2635 282.8385 211.400 554.300 356.3600 405.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    BC#16C !* DESCRIPTION #1 *)
!* POSITION *)
    254.2635 282.8385 211.400 297.125 356.3600 405.8900
    <<mRco>> <<mNco>> !* MIX *)
    BC#16B !* DESCRIPTION #2 *)
!* POSITION *)
    254.2635 282.8385 297.125 468.575 356.3600 405.8900
    <<mRci>> <<mNci>> !* MIX *)
    BC#16A !* DESCRIPTION #3 *)
!* POSITION *)
    254.2635 282.8385 468.575 554.300 356.3600 405.8900
    <<mRco>> <<mNco>> !* MIX *)
ENDDEV
DEVICE!* DATA #17 *)
CONTROLLED ADJUSTER BC#17 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    311.4135 339.9885 211.400 554.300 356.3600 405.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    BC#17C-B-A !* DESCRIPTION #1 *)
!* POSITION *)
    311.4135 339.9885 211.400 554.300 356.3600 405.8900
    <<mRbi>> <<mNbi>> !* MIX *)
ENDDEV

```

```

DEVICE!* DATA #18 *)
CONTROLLED ADJUSTER BC#18 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    368.5625 397.1375 211.400 554.300 356.3600 405.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    BC#18C !* DESCRIPTION #1 *)
!* POSITION *)
    368.5625 397.1375 211.400 297.125 356.3600 405.8900
<<mRao>> <<mNao>> !* MIX *)
    BC#18B !* DESCRIPTION #2 *)
!* POSITION *)
    368.5625 397.1375 297.125 468.575 356.3600 405.8900
<<mRai>> <<mNai>> !* MIX *)
    BC#18A !* DESCRIPTION #3 *)
!* POSITION *)
    368.5625 397.1375 468.575 554.300 356.3600 405.8900
<<mRao>> <<mNao>> !* MIX *)
ENDDEV
DEVICE!* DATA #19 *)
CONTROLLED ADJUSTER BC#19 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    425.7135 454.2885 211.400 554.300 356.3600 405.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    BC#19C-B-A !* DESCRIPTION #1 *)
!* POSITION *)
    425.7135 454.2885 211.400 554.300 356.3600 405.8900
<<mRbi>> <<mNbi>> !* MIX *)
ENDDEV
DEVICE!* DATA #20 *)
CONTROLLED ADJUSTER BC#20 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    482.8635 511.4385 211.400 554.300 356.3600 405.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    BC#20C !* DESCRIPTION #1 *)
!* POSITION *)
    482.8635 511.4385 211.400 297.125 356.3600 405.8900

```

```

    <<mRco>> <<mNco>> !* MIX *)
    BC#20B !* DESCRIPTION #2 *)
!* POSITION *)
    482.8635 511.4385 297.125 468.575 356.3600 405.8900
    <<mRci>> <<mNci>> !* MIX *)
    BC#20A !* DESCRIPTION #3 *)
!* POSITION *)
    482.8635 511.4385 468.575 554.300 356.3600 405.8900
    <<mRco>> <<mNco>> !* MIX *)
ENDDEV
DEVICE!* DATA #21 *)
CONTROLLED ADJUSTER BC#21 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    540.0135 568.5885 325.700 440.000 356.3600 405.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    BC#21B !* DESCRIPTION #1 *)
!* POSITION *)
    540.0135 568.5885 325.700 440.000 356.3600 405.8900
    <<mRdi>> <<mNdi>> !* MIX *)
ENDDEV

!* LIQUID ZONE CONTROLLERS AT VARIOUS LEVELS *)
!* ZCU-1 *)

DEVICE!* DATA #22 *)
CONTROLLED LZC CONTL#1 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    197.1125 225.6875 <<yM01>> <<yP01>> 146.3600 195.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    EMPTY#1 !* DESCRIPTION #1 *)
!* POSITION *)
    197.1125 225.6875 <<yM01>> <<y001>> 146.3600 195.8900
    <<m12E>> <<m12M>> !* MIX *)
    FULL#1 !* DESCRIPTION #2 *)
!* POSITION *)
    197.1125 225.6875 <<y001>> <<yP01>> 146.3600 195.8900

```

```

    <<m12F>> <<m12M>> !* MIX *)
ENDDEV
DEVICE!* DATA #23 *)
    CONTROLLED LZC CONTL#2 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    197.1125 225.6875 <<yM02>> <<yP02>> 146.3600 195.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    EMPTY#2 !* DESCRIPTION #1 *)
!* POSITION *)
    197.1125 225.6875 <<yM02>> <<y002>> 146.3600 195.8900
<<m01E>> <<m01M>> !* MIX *)
    FULL#2 !* DESCRIPTION #2 *)
!* POSITION *)
    197.1125 225.6875 <<y002>> <<yP02>> 146.3600 195.8900
<<m01F>> <<m01M>> !* MIX *)
ENDDEV
DEVICE!* DATA #24 *)
    CONTROLLED LZC CONTL#3 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    368.5625 397.1375 <<yM03>> <<yP03>> 146.3600 195.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    EMPTY#3 !* DESCRIPTION #1 *)
!* POSITION *)
    368.5625 397.1375 <<yM03>> <<y003>> 146.3600 195.8900
<<m23E>> <<m23M>> !* MIX *)
    FULL#3 !* DESCRIPTION #2 *)
!* POSITION *)
    368.5625 397.1375 <<y003>> <<yP03>> 146.3600 195.8900
<<m23F>> <<m23M>> !* MIX *)
ENDDEV
DEVICE!* DATA #25 *)
    CONTROLLED LZC CONTL#4 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    368.5625 397.1375 <<yM04>> <<yP04>> 146.3600 195.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    EMPTY#4 !* DESCRIPTION #1 *)

```

```

!* POSITION *)
  368.5625 397.1375 <<yM04>> <<y004>> 146.3600 195.8900
  <<m12E>> <<m12M>> !* MIX *)
  FULL#4 !* DESCRIPTION #2 *)
!* POSITION *)
  368.5625 397.1375 <<y004>> <<yP04>> 146.3600 195.8900
  <<m12F>> <<m12M>> !* MIX *)
ENDDEV
DEVICE!* DATA #26 *)
  CONTROLLED LZC CONTL#5 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
  368.5625 397.1375 <<yM05>> <<yP05>> 146.3600 195.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
  EMPTY#5 !* DESCRIPTION #1 *)
!* POSITION *)
  368.5625 397.1375 <<yM05>> <<y005>> 146.3600 195.8900
  <<m01E>> <<m01M>> !* MIX *)
  FULL#5 !* DESCRIPTION #2 *)
!* POSITION *)
  368.5625 397.1375 <<y005>> <<yP05>> 146.3600 195.8900
  <<m01F>> <<m01M>> !* MIX *)
ENDDEV
DEVICE!* DATA #27 *)
  CONTROLLED LZC CONTL#6 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
  540.0125 568.5875 <<yM06>> <<yP06>> 146.3600 195.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
  EMPTY#6 !* DESCRIPTION #1 *)
!* POSITION *)
  540.0125 568.5875 <<yM06>> <<y006>> 146.3600 195.8900
  <<m12E>> <<m12M>> !* MIX *)
  FULL#6 !* DESCRIPTION #2 *)
!* POSITION *)
  540.0125 568.5875 <<y006>> <<yP06>> 146.3600 195.8900
  <<m12F>> <<m12M>> !* MIX *)
ENDDEV

```

```

DEVICE!* DATA #28 *)
  CONTROLLED LZC CONTL#7 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
  540.0125 568.5875 <<yM07>> <<yP07>> 146.3600 195.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
  EMPTY#7 !* DESCRIPTION #1 *)
!* POSITION *)
  540.0125 568.5875 <<yM07>> <<y007>> 146.3600 195.8900
<<m01E>> <<m01M>> !* MIX *)
  FULL#7 !* DESCRIPTION #2 *)
!* POSITION *)
  540.0125 568.5875 <<y007>> <<yP07>> 146.3600 195.8900
<<m01F>> <<m01M>> !* MIX *)
ENDDEV
DEVICE!* DATA #29 *)
  CONTROLLED LZC CONTL#8 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
  197.1135 225.6885 <<yM08>> <<yP08>> 406.3600 455.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
  EMPTY#8 !* DESCRIPTION #1 *)
!* POSITION *)
  197.1135 225.6885 <<yM08>> <<y008>> 406.3600 455.8900
<<m12E>> <<m12M>> !* MIX *)
  FULL#8 !* DESCRIPTION #2 *)
!* POSITION *)
  197.1135 225.6885 <<y008>> <<yP08>> 406.3600 455.8900
<<m12F>> <<m12M>> !* MIX *)
ENDDEV
DEVICE!* DATA #30 *)
  CONTROLLED LZC CONTL#9 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
  197.1125 225.6875 <<yM09>> <<yP09>> 406.3600 455.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
  EMPTY#9 !* DESCRIPTION #1 *)
!* POSITION *)
  197.1125 225.6875 <<yM09>> <<y009>> 406.3600 455.8900

```

```

<<m01E>> <<m01M>> !* MIX *)
  FULL#9 !* DESCRIPTION #2 *)
!* POSITION *)
  197.1125 225.6875 <<y009>> <<yP09>> 406.3600 455.8900
  <<m01F>> <<m01M>> !* MIX *)
ENDDEV
DEVICE!* DATA #31 *)
  CONTROLLED LZC CONTL#10 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
  368.5625 397.1375 <<yM10>> <<yP10>> 406.3600 455.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
  EMPTY#10 !* DESCRIPTION #1 *)
!* POSITION *)
  368.5625 397.1375 <<yM10>> <<y010>> 406.3600 455.8900
  <<m23E>> <<m23M>> !* MIX *)
  FULL#10 !* DESCRIPTION #2 *)
!* POSITION *)
  368.5625 397.1375 <<y010>> <<yP10>> 406.3600 455.8900
  <<m23F>> <<m23M>> !* MIX *)
ENDDEV
DEVICE!* DATA #32 *)
  CONTROLLED LZC CONTL#11 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
  368.5625 397.1375 <<yM11>> <<yP11>> 406.3600 455.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
  EMPTY#11 !* DESCRIPTION #1 *)
!* POSITION *)
  368.5625 397.1375 <<yM11>> <<y011>> 406.3600 455.8900
  <<m12E>> <<m12M>> !* MIX *)
  FULL#11 !* DESCRIPTION #2 *)
!* POSITION *)
  368.5625 397.1375 <<y011>> <<yP11>> 406.3600 455.8900
  <<m12F>> <<m12M>> !* MIX *)
ENDDEV
DEVICE!* DATA #33 *)
  CONTROLLED LZC CONTL#12 !* TYPE FAMILY SEQUENCE NUMBER *)

```

```

!* POSITON MAX *)
    368.5625 397.1375 <<yM12>> <<yP12>> 406.3600 455.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    EMPTY#12 !* DESCRIPTION #1 *)
!* POSITION *)
    368.5625 397.1375 <<yM12>> <<y012>> 406.3600 455.8900
<<m01E>> <<m01M>> !* MIX *)
    FULL#12 !* DESCRIPTION #2 *)
!* POSITION *)
    368.5625 397.1375 <<y012>> <<yP12>> 406.3600 455.8900
<<m01F>> <<m01M>> !* MIX *)
ENDDEV
DEVICE!* DATA #34 *)
    CONTROLLED LZC CONTL#13 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    540.0135 568.5885 <<yM13>> <<yP13>> 406.3600 455.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    EMPTY#13 !* DESCRIPTION #1 *)
!* POSITION *)
    540.0135 568.5885 <<yM13>> <<y013>> 406.3600 455.8900
<<m12E>> <<m12M>> !* MIX *)
    FULL#13 !* DESCRIPTION #2 *)
!* POSITION *)
    540.0135 568.5885 <<y013>> <<yP13>> 406.3600 455.8900
<<m12F>> <<m12M>> !* MIX *)
ENDDEV
DEVICE!* DATA #35 *)
    CONTROLLED LZC CONTL#14 !* TYPE FAMILY SEQUENCE NUMBER *)
!* POSITON MAX *)
    540.0125 568.5875 <<yM14>> <<yP14>> 406.3600 455.8900
Y 0.0 0.0 0.0 !* DIRECTION SPEED PARAMETERS *)
    EMPTY#14 !* DESCRIPTION #1 *)
!* POSITION *)
    540.0125 568.5875 <<yM14>> <<y014>> 406.3600 455.8900
<<m01E>> <<m01M>> !* MIX *)
    FULL#14 !* DESCRIPTION #2 *)

```

```

!* POSITION *)
  540.0125 568.5875 <<y014>> <<yP14>> 406.3600 455.8900
  <<m01F>> <<m01M>> !* MIX *)
ENDDEV
  ;

```

```
END: ;
```

```
QUIT .
```

Procédure PDEVPROC

```

!*****
!* Descriptions des DEVICE pour CANDU6 *
!* *
!* PROCEDURE := Pinproc DEVICE :: timbc ; *
!* *
!* timbc = Temps minimum d'insertion pour Barres de compensation *
!* *
!*****
PARAMETER PROCEDURE DEVICE ::
  :: LINKED_LIST PROCEDURE DEVICE ;
  ;

MODULE END: INPROC: ;
REAL timbc ;
REAL yful1 yful3 ;

!*****

:: >>timbc<< ;

EVALUATE yful1 yful3 := 170.1827 53.8898 ;
!* VALEURS POUR 100% D'UNE BL DE LONGUEUR 'DONJON' (voir Pdevice) *

PROCEDURE := INPROC: DEVICE ::

```

```
EDIT 0 TYPE CONTROLLED
END
ROD ADJUSTER !* DATA FOR SSAR *)
NAME BC#1 NB 1 NBANC 1 TIME <<timbc>>
END
ROD ADJUSTER
NAME BC#2 NB 2 NBANC 2 TIME <<timbc>>
END
ROD ADJUSTER
NAME BC#3 NB 3 NBANC 5 TIME <<timbc>>
END
ROD ADJUSTER
NAME BC#4 NB 4 NBANC 3 TIME <<timbc>>
END
ROD ADJUSTER
NAME BC#5 NB 5 NBANC 6 TIME <<timbc>>
END
ROD ADJUSTER
NAME BC#6 NB 6 NBANC 2 TIME <<timbc>>
END
ROD ADJUSTER
NAME BC#7 NB 7 NBANC 1 TIME <<timbc>>
END
ROD ADJUSTER
NAME BC#8 NB 8 NBANC 4 TIME <<timbc>>
END
ROD ADJUSTER
NAME BC#9 NB 9 NBANC 4 TIME <<timbc>>
END
ROD ADJUSTER
NAME BC#10 NB 10 NBANC 7 TIME <<timbc>>
END
ROD ADJUSTER
NAME BC#11 NB 11 NBANC 1 TIME <<timbc>>
END
ROD ADJUSTER
```

NAME BC#12 NB 12 NBANC 7 TIME <<timbc>>

END

ROD ADJUSTER

NAME BC#13 NB 13 NBANC 4 TIME <<timbc>>

END

ROD ADJUSTER

NAME BC#14 NB 14 NBANC 4 TIME <<timbc>>

END

ROD ADJUSTER

NAME BC#15 NB 15 NBANC 1 TIME <<timbc>>

END

ROD ADJUSTER

NAME BC#16 NB 16 NBANC 3 TIME <<timbc>>

END

ROD ADJUSTER

NAME BC#17 NB 17 NBANC 6 TIME <<timbc>>

END

ROD ADJUSTER

NAME BC#18 NB 18 NBANC 2 TIME <<timbc>>

END

ROD ADJUSTER

NAME BC#19 NB 19 NBANC 5 TIME <<timbc>>

END

ROD ADJUSTER

NAME BC#20 NB 20 NBANC 3 TIME <<timbc>>

END

ROD ADJUSTER

NAME BC#21 NB 21 NBANC 1 TIME <<timbc>>

END

!* LIQUID ZONE CONTROLLERS *

LZC LZC

NAME CONTL#1 NAMFUL FULL#1 NAMEMP EMPTY#1

YFULL <<yful1>> NB 1 BIAS 0.670 ZFLOW 0.670 0.670

TIME 60. 500. END

LZC LZC

NAME CONTL#2 NAMFUL FULL#2 NAMEMP EMPTY#2

NB 2 BIAS 0.602 ZFLOW 0.602 0.602

TIME 36. 500. END

LZC LZC

NAME CONTL#3 NAMFUL FULL#3 NAMEMP EMPTY#3

YFULL <<yful3>> NB 3 BIAS 0.544 ZFLOW 0.544 0.544

TIME 36. 500. END

LZC LZC

NAME CONTL#4 NAMFUL FULL#4 NAMEMP EMPTY#4

NB 4 BIAS 0.622 ZFLOW 0.622 0.622

TIME 36. 500. END

LZC LZC

NAME CONTL#5 NAMFUL FULL#5 NAMEMP EMPTY#5

NB 5 BIAS 0.585 ZFLOW 0.585 0.585

TIME 36. 500. END

LZC LZC

NAME CONTL#6 NAMFUL FULL#6 NAMEMP EMPTY#6

YFULL <<yful1>> NB 6 BIAS 0.651 ZFLOW 0.651 0.651

TIME 60. 500. END

LZC LZC

NAME CONTL#7 NAMFUL FULL#7 NAMEMP EMPTY#7

NB 7 BIAS 0.591 ZFLOW 0.591 0.591

TIME 36. 500. END

LZC LZC

NAME CONTL#8 NAMFUL FULL#8 NAMEMP EMPTY#8

YFULL <<yful1>> NB 8 BIAS 0.634 ZFLOW 0.634 0.634

TIME 60. 500. END

LZC LZC

NAME CONTL#9 NAMFUL FULL#9 NAMEMP EMPTY#9

NB 9 BIAS 0.601 ZFLOW 0.601 0.601

TIME 36. 500. END

LZC LZC

NAME CONTL#10 NAMFUL FULL#10 NAMEMP EMPTY#10

YFULL <<yful3>> NB 10 BIAS 0.671 ZFLOW 0.671 0.671

TIME 36. 500. END

```

LZC LZC
NAME CONTL#11 NAMFUL FULL#11 NAMEMP EMPTY#11
NB 11 BIAS 0.619 ZFLOW 0.619 0.619
TIME 36. 500. END
LZC LZC
NAME CONTL#12 NAMFUL FULL#12 NAMEMP EMPTY#12
NB 12 BIAS 0.526 ZFLOW 0.526 0.526
TIME 36. 500. END
LZC LZC
NAME CONTL#13 NAMFUL FULL#13 NAMEMP EMPTY#13
YFULL <<yfull>> NB 13 BIAS 0.628 ZFLOW 0.628 0.628
TIME 60. 500. END
LZC LZC
NAME CONTL#14 NAMFUL FULL#14 NAMEMP EMPTY#14
NB 14 BIAS 0.601 ZFLOW 0.601 0.601
TIME 36. 500. END
;

```

```
END: ;
```

```
QUIT .
```

Procédure PMACRO

```

!*****
!* *
!* ACCES AUX DONNEES DE DRAGON *
!* *
!*****

```

```

PARAMETER MACRO ::
    ::: LINKED_LIST MACRO ;
;

```

```
MODULE CRE: END: ;
```

```
LINKED_LIST NREFL NFUEL ZCR ADJ ;
```

```

SEQ_ASCII adjnat fnat rnat zcunat
      :: FILE 'cpo/adjnat'
          'cpo/fnat'
          'cpo/rnat'
          'cpo/zcunat' ;

INTEGER mRai mRao mRbi mRci mRco mRdi
      mNRai mNRao mNRbi mNRci mNRco mNRdi :=
      5 6 7 8 9 10
      11 12 13 14 15 16 ;

INTEGER m23M m23E m23F m12M m12E m12F m01M m01E m01F :=
      17 18 19 20 21 22 23 24 25 ;

ADJ := adjnat ;
ZCR := zcunat ;
NFUEL := fnat ;
NREFL := rnat ;

MACRO := CRE: NFUEL NREFL ADJ ZCR ::
      EDIT 0 NMIX 25
      READ
      COMPO NFUEL MIX 1 'MIXTRXE 1'
          T-BURNUP 0.0 8675.9 MICRO XE135 0.0
          ENDMIX
! MIX 2 'MIXTRXE 1'
! T-BURNUP 0.0 7124.6 MICRO XE135 0.0
! ENDMIX
! MIX 3 'MIXTRXE 1'
! T-BURNUP 0.0 7124.6 MICRO XE135 0.0
! ENDMIX
      COMPO NREFL MIX 2 'MIXTMOD 1' ENDMIX
      COMPO ADJ MIX <<mRai>> 'BCAINT 1' ENDMIX
          MIX <<mNRai>> 'BCAINTRE 1' ENDMIX
          MIX <<mRao>> 'BCAOUT 1' ENDMIX

```

```
MIX <<mNRao>> 'BCAOUTRE 1' ENDMIX
MIX <<mRbi>> 'BCBINT 1' ENDMIX
MIX <<mNRbi>> 'BCBINTRE 1' ENDMIX
MIX <<mRci>> 'BCCINT 1' ENDMIX
MIX <<mNRci>> 'BCCINTRE 1' ENDMIX
MIX <<mRco>> 'BCCOUT 1' ENDMIX
MIX <<mNRco>> 'BCCOUTRE 1' ENDMIX
MIX <<mRdi>> 'BCDINT 1' ENDMIX
MIX <<mNRdi>> 'BCDINTRE 1' ENDMIX
COMPO ZCR MIX <<m23M>> 'NOBL32 1' ENDMIX
MIX <<m23E>> 'BLIQE32 1' ENDMIX
MIX <<m23F>> 'BLIQF32 1' ENDMIX
MIX <<m12M>> 'NOBL21 1' ENDMIX
MIX <<m12E>> 'BLIQE21 1' ENDMIX
MIX <<m12F>> 'BLIQF21 1' ENDMIX
MIX <<m01M>> 'NOBL10 1' ENDMIX
MIX <<m01E>> 'BLIQE10 1' ENDMIX
MIX <<m01F>> 'BLIQF10 1' ENDMIX
;
END: ;
QUIT .
```