

ÉCOLE PUBLIQUE D'INGÉNIEUR EN 3 ANS

RAPPORT DE STAGE DE DEUXIÈME ANNÉE

Modélisation d'un canal du réacteur de recherche CABRI lors d'un incident de réactivité

Mai-Août 2018

Amandin PAQUET

amandin.paquet@ecole.ensicaen.fr

Tuteur école : Gilles BAN

Tuteur stage : Alain HÉBERT



**POLYTECHNIQUE
MONTRÉAL**

**LE GÉNIE
EN PREMIÈRE CLASSE**

Table des matières

1	Introduction	1
1.1	Présentation de l'entité d'accueil	1
1.2	Présentation du projet	1
2	Description du Programme International Cabri (CIP)	3
2.1	Réacteur de recherche et programme expérimental	3
2.2	Simulation des RIA (Accident d'Insertion de Réactivité)	4
2.3	Transitoires simulés	5
3	État de l'existant	7
3.1	Méthode de calcul actuelle	7
3.1.1	Le code SCANAIR	7
3.2	Positionnement de la contribution	7
4	Outils utilisés	8
4.1	Code de calcul neutronique Monte Carlo - SERPENT	8
4.2	Code déterministe - DRAGON-DONJON	8
4.3	Code de calcul thermohydraulique et thermomécanique - TRIAD	9
5	Mise en place de la nouvelle méthode de calcul	9
5.1	Géométrie sous SERPENT sans maillage axiale	9
5.2	Géométrie sous SERPENT avec maillage axiale	11
5.3	Calcul de transitoire	12
5.4	Solution thermohydraulique	13
5.5	Le couplage	14
5.5.1	Méthodes de communication	15
5.5.2	Couplage global	16
5.6	Parallélisation du calcul	16
6	Résultats	18
7	Conclusion	19
	Annexes	20

Liste des figures

FIGURE 1 : Schéma du réacteur CABRI, figure de l'IRSN [5]	3
FIGURE 2 : Circuit expérimental de CABRI, figure de l'IRSN [5]	4
FIGURE 3 : Section efficace d'absorption de l'hélium-3	4
FIGURE 4 : Exemple de pic de puissance obtenu	6
FIGURE 5 : Courbes de dépressurisation des trois transitoires enveloppes	6
FIGURE 6 : Schéma des flux de données entre les différents modules de SCANAIR. Figure de l'IRSN [6]	7
FIGURE 7 : Définition des matériaux du crayon de combustible dans SERPENT	10
FIGURE 8 : Définition du combustible UOX dans SERPENT	10
FIGURE 9 : L'étage TBI (Toutes Barres Insérées) à gauche et l'étage TBH (Toutes Barres Hautes) à droite avec leur définition dans SERPENT	10
FIGURE 10 : Schéma représentant la géométrie avec sa définition dans SERPENT	11
FIGURE 11 : Exemple du fichier d'interface pour l'UOX	11
FIGURE 12 : Vue en coupe de la géométrie maillée selon l'axe vertical	12
FIGURE 13 : Liste de fichiers détecteurs après le calcul d'un transitoire	12
FIGURE 14 : Exemple du formatage des fichiers détecteurs	13
FIGURE 15 : Distribution selon l'axe vertical de la température du combustible .	13
FIGURE 16 : Puissance totale et distribution selon l'axe vertical de la puissance de l'assemblage	14
FIGURE 17 : Schéma de communication simplifié entre la solution thermohydraulique et la solution neutronique	14
FIGURE 18 : Schéma de communication entre SERPENT et l'interface en C sans la thermohydraulique	15
FIGURE 19 : Appel de librairie en C	16
FIGURE 20 : Schéma de communication entre la solution thermohydraulique et l'interface en C sans solution neutronique	16
FIGURE 21 : Schéma global de couplage entre la solution thermohydraulique, la solution neutronique et l'interface en C	17
FIGURE 22 : Calcul d'un transitoire sur un cœur à gauche et sur 22 cœurs à droite	17
FIGURE 23 : Transitoire simulé à l'aide du couplage SERPENT-THERMO:, cas du transitoire norm10	18
FIGURE 24 : Schéma simplifié de la récupération des données thermohydrauliques	20
FIGURE 25 : Schéma simplifié de la récupération des énergies calculées par SERPENT	23

Remerciements

Je tiens à remercier Alain Hébert pour m'avoir permis de réaliser mon stage à Montréal et m'avoir fait confiance sur ce projet.

Je remercie également Lucas Fede pour m'avoir grandement aidé à comprendre le projet et ses problématiques.

Je remercie Martin Guillet pour m'avoir conseillé et relu lors de la rédaction de ce rapport.

Enfin je remercie mes amis, Clément Gazon, Maxime Paradis, Guillaume De Pré, Vivian Salino et tous les autres qui ont aidé à ce que cette mobilité internationale soit une expérience inoubliable et à réitérer.

1 Introduction

1.1 Présentation de l'entité d'accueil

Fondée en 1873, Polytechnique Montréal est l'un des plus importants établissements d'enseignement et de recherche en génie au Canada. Il demeure le premier au Québec quant au nombre de ses étudiants et à l'ampleur de son activité de recherche. Le stage est mené au sein de l'Institut de Génie Nucléaire (IGN) de l'école Polytechnique de Montréal, sous la tutelle de M. Alain Hébert, Docteur-ingénieur et professeur, en collaboration avec deux autres étudiants ingénieurs. L'IGN, dirigé par Guy Marleau, est notamment reconnu pour le développement des codes de calcul DRAGON et DONJON, qui seront utilisés dans ce projet.

1.2 Présentation du projet

La présence de 58 réacteurs à eau pressurisée (REP) sur le sol français justifie la conception d'un programme permettant de simuler finement les diverses conditions d'accidents à même de se produire dans les réacteurs en fonctionnement. Le code SCANAIR (Système de Codes pour l'ANalyse d'Accidents d'Insertion de Réactivité) a été développé par l'IRSN (Institut de Radioprotection et Sûreté Nucléaire), en partenariat avec EDF (Électricité De France) et le CEA (Commissariat à l'Énergie Atomique et aux Énergies alternatives), dans ce but. En parallèle, des expériences ont été effectuées grâce au Programme International Cabri (CIP), dirigé par l'IRSN et cofinancé avec de multiples partenaires français et étrangers. Le dispositif CABRI permet de reproduire des conditions semblables à un RIA (Accident d'Insertion de Réactivité) dans les REP à l'aide d'un réacteur spécifique. Ce dernier est opéré par le CEA. Il permet entre autres d'étudier, le comportement mécanique des crayons de combustibles fortement irradiés issus de REP et ainsi qualifier les codes de simulation comme SCANAIR.

Le RIA (Accident d'Insertion de Réactivité) représente tout transitoire neutronique accidentelle qui entraîne une augmentation soudaine et localisée du flux de neutrons. Par exemple, une éjection de grappe de contrôle peut déclencher une élévation de la puissance produite par les fissions. Cette élévation a plusieurs conséquences, elle provoque une augmentation de la température du combustible ce qui entraîne une déformation mécanique du crayon de combustible pouvant aller jusqu'à la rupture du crayon. Ce genre d'incident se produit suite à une défaillance de l'enveloppe pressurisée du système de pilotage de la grappe de contrôle. La grappe de commande est éjectée à cause de la différence de pression entre le cœur du réacteur et l'enceinte de confinement. Comment simuler une éjection de grappe de manière expérimentale ? Diverses méthodes permettent de simuler physiquement l'injection de réactivité positive semblable aux conséquences d'une éjection de grappe de contrôle. La solution retenue dans le design du réacteur CABRI a été de dépressuriser (de manière contrôlée) des tubes d'hélium-3, un gaz neutrophage.

L'IRSN simule les comportements associés à ces RIA à l'aide d'un chaînage entre deux codes de calculs, DULCINEE et SCANAIR. le premier est un code de cinétique point du CEA prenant en compte les effets thermohydrauliques et le second a été développé spécifiquement pour l'étude thermohydraulique et thermo-mécanique des comportements des crayons de combustible lors de RIA.

Le comportement mécanique des crayons de tests n'a pu être modélisé de manière satisfaisante avec cette solution. L'IRSN cherche à améliorer la précision de ses logiciels pour représenter le comportement du réacteur, incluant les crayons de test, lors d'un RIA. L'une des voies explorées est la prise en compte des effets spatio-temporels lors du calcul neutronique. L'idée est d'effectuer un couplage de code permettant des calculs neutroniques à une dimension sur un assemblage de réacteur et pouvant à terme simuler le coeur complet.

Dans cette optique, l'objectif de ce stage était de proposer une solution de couplage entre le code de type Monte Carlo SERPENT et du module de thermohydraulique du code DONJON développé à l'École Polytechnique de Montréal.

2 Description du Programme International Cabri (CIP)

Le CIP s'inscrit dans une démarche d'amélioration de la sûreté en exploitation nucléaire du parc français. Le but de ce programme est d'étudier les comportements des crayons de combustible irradiés issus de REP lors d'injection de puissance en cas de RIA afin de mieux prévoir leurs conséquences sur un cœur en fonctionnement.

2.1 Réacteur de recherche et programme expérimental

Le réacteur CABRI est un réacteur de type piscine. Son cœur est immergé dans une cuve remplie d'eau, à une température de 318K et une pression de 2 bar et refroidit par la boucle d'eau indiquée en bleu sur la figure 1 ci-dessous.

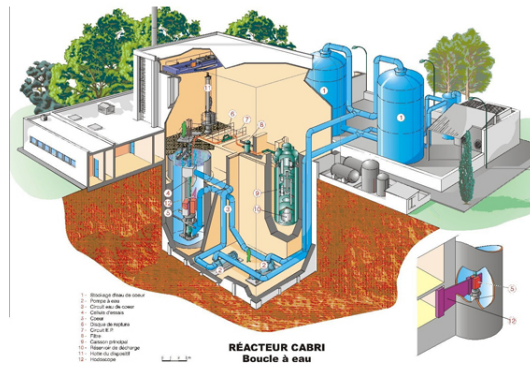
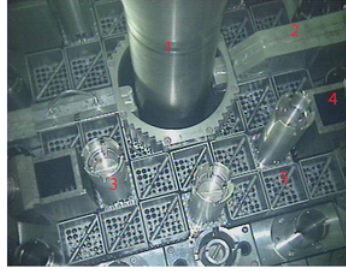


FIGURE 1 – Schéma du réacteur CABRI, figure de l'IRSN [5]

La simulation de RIA est réalisée dans le cœur du réacteur, appelé cœur nourricier. Ce cœur mesure 80 cm de hauteur pour 60 cm de côté et est composé de 1488 crayons de combustibles, 6 barres de commande et 4 barres remplies d'hélium-3 neutrophage avec un espace en son centre pour laisser passer le circuit expérimental comme le montre la figure 2.

Le circuit expérimental a été installé de manière à simuler les conditions physiques d'un REP en fonctionnement, la température est donc de 553,15K, la pression est de 155 bar et le débit est de 6 mètres cubes par heure.



Coeur de CABRI
1 - Cellule expérimentale EP
2 - Canal d'irradiation
3 - Barre de commande et de sécurité (BCS)
4 - Barre transitoire
5 - Assemblage de combustible

FIGURE 2 – Circuit expérimental de CABRI, figure de l'IRSN [5]

2.2 Simulation des RIA (Accident d'Insertion de Réactivité)

Afin d'obtenir les mêmes conditions que dans un REP en cas de RIA (comme pour une éjection de grappe par exemple) une augmentation brutale de la puissance du cœur doit être produite. Le fonctionnement normal du réacteur est atteint à l'aide des grappes de contrôle et des barres pressurisées à l'hélium-3.

Le pic de puissance est obtenu grâce à la dépressurisation rapide des barres d'hélium-3. Un gaz neutrophage particulièrement efficace en fonctionnement thermique, domaine de fonctionnement du réacteur, comme l'indique la figure 3. La réactivité va donc fortement augmenter car plus de neutrons entraîne plus de fissions.

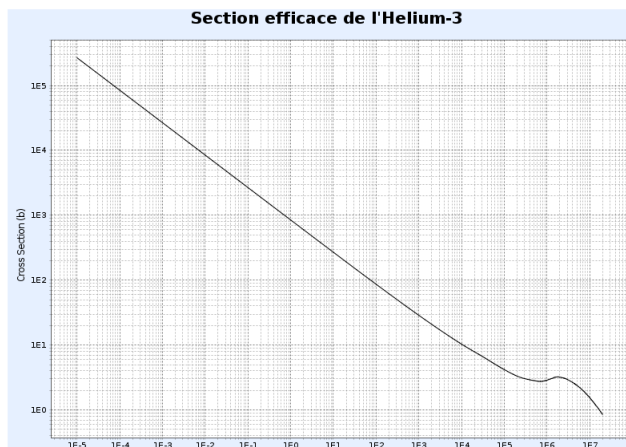


FIGURE 3 – Section efficace d'absorption de l'hélium-3

La dépressurisation des barres suit la loi suivante :

Le transitoire étant très rapide, il est correct de penser que la réaction est adiabatique. Cette hypothèse permet d'utiliser une loi de Laplace.

$$P(t) = \begin{cases} P_0 & \text{si } t \leq 0 \\ \frac{P_0}{(1+B.t)^\alpha} & \text{si } t \in]0, t_{cri}[\\ P_\infty & \text{si } t \geq t_{cri} \end{cases} \quad (1)$$

avec :

- P_0 = la pression initiale dans les barres.
- P_∞ = la pression finale dans les barres.
- α = $\frac{2\gamma}{\gamma-1}$ et $\gamma = \frac{C_p}{C_v} = 1.66$ pour l'He
- B = Coefficient de pilotage du transitoire
- t_{cri} = temps de dépressurisation totale de la barre, dépend de la voie de sortie.

En parallèle à la dépressurisation de l'hélium-3 dont l'évolution temporelle vient d'être détaillée le système réagit. Il est important de s'intéresser à la puissance. Comme le montre la figure 4, après une élévation exponentielle pouvant aller jusqu'à 100 000 fois la puissance initiale (due à l'augmentation instantanée de la réactivité), la température du combustible va augmenter fortement. L'augmentation de température en conséquence de cette élévation de puissance entraîne un retour à l'équilibre du système en raison de l'effet Doppler. Dans un second temps les grappes de contrôle sont abaissées pour contrer une deuxième hausse de la puissance due aux neutrons retardés.

2.3 Transitoires simulés

Le fait d'avoir un outil de simulation permet de tester un grand nombre de transitoires et ainsi vérifier le comportement des crayons de test selon différents RIA. Pour regrouper l'ensemble des transitoires, trois transitoires appelés enveloppe sont étudiés. Ces pics de puissance se différencient par leur largeur à mi-hauteur (10, 30 et 75 ms), la pression d'Hélium initiale et la valeur du coefficient B.

La dépressurisation de l'hélium-3 est différente selon le type de transitoire comme l'indique la figure 5.

Les transitoires struct30 et struct75 sont de type structuré, il y a deux valeurs de B au cours de ces derniers. La dépressurisation de l'hélium-3 étant différente, les dommages sur le crayons de tests le seront tout autant.

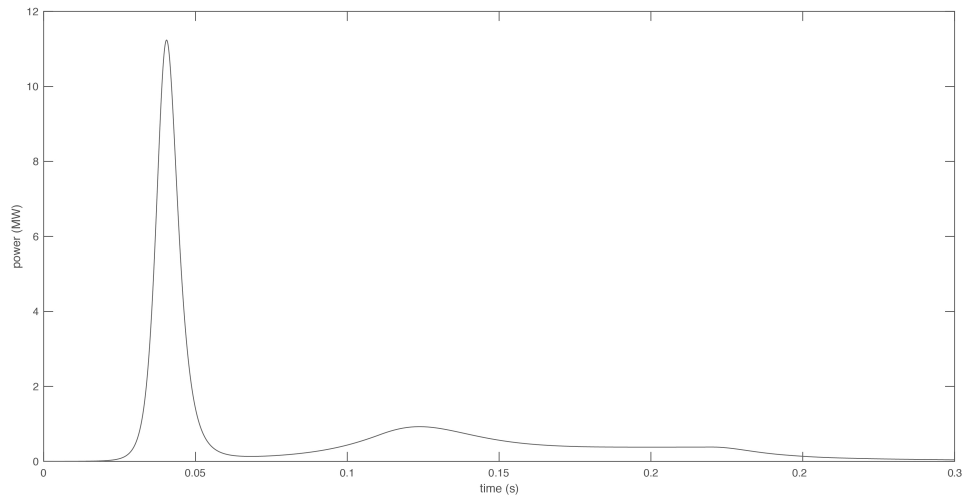


FIGURE 4 – Exemple de pic de puissance obtenue

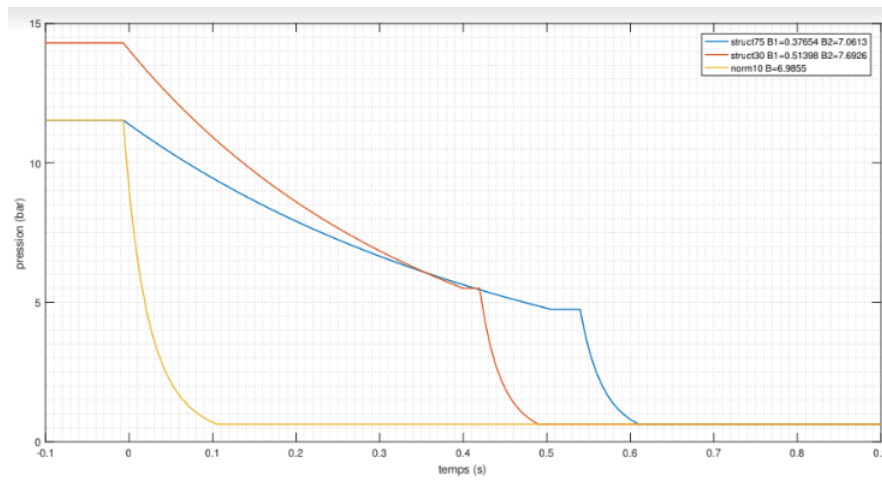


FIGURE 5 – Courbes de dépressurisation des trois transitoires enveloppes

norm10 : Pression initiale d'hélium-3 : 11,50 bar et $B = 6,99500 \text{ s}^{-1}$
struct30 : Pression initiale d'hélium-3 : 14,30 bar et $B1 = 0,51398 \text{ s}^{-1}$ puis $B2 = 7,6926 \text{ s}^{-1}$
struct75 : Pression initiale d'hélium-3 : 11,52 bar et $B1 = 0,37654 \text{ s}^{-1}$ puis $B2 = 7,0613 \text{ s}^{-1}$

3 État de l'existant

3.1 Méthode de calcul actuelle

3.1.1 Le code SCANAIR

Le code SCANAIR (Système de Codes pour l'ANalyse d'Accidents d'Insertion de Réactivité) développé par l'IRSN permet de simuler le comportement thermomécanique d'un crayon de combustible durant un RIA dans un REP. Pour réaliser une telle simulation, il est nécessaire de calculer l'évolution de différentes valeurs, mécaniques, neutroniques ou thermohydrauliques. Le couplage des trois modules principaux de SCANAIR est présenté sur la figure 6.

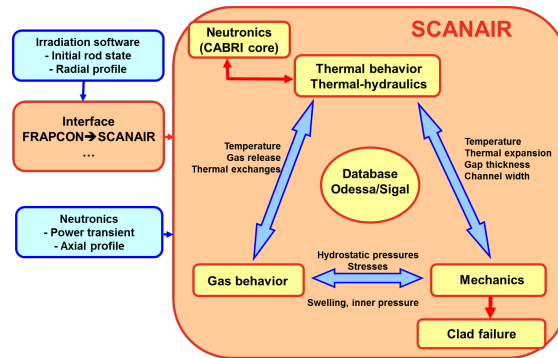


FIGURE 6 – Schéma des flux de données entre les différents modules de SCANAIR. Figure de l'IRSN [6]

3.2 Positionnement de la contribution

La simulation actuelle ne modélisant pas de manière satisfaisante le comportement mécaniques des crayons, l'IRSN cherche une solution plus viable. L'une d'elle est le couplage du code de calcul neutronique de type Monte Carlo SERPENT avec la solution de calcul thermohydraulique développée par l'IGN à L'École Polytechnique de Montréal.

Le but de ce stage est de réaliser un couplage fonctionnel SERPENT-DONJON afin de modéliser un assemblage représentatif du comportement du réacteur CABRI. Ces travaux doivent permettre d'envisager un couplage plus complet (en simulant le transitoire sur le coeur complet) dans le cadre d'un futur travail de doctorat encadré par l'IRSN.

4 Outils utilisés

4.1 Code de calcul neutronique Monte Carlo - SERPENT

SERPENT est un code open source de transport de particules de type MONTE-CARLO développé au Centre de recherche technique de Finlande (VTT Technical Research Centre of Finland, Ltd) depuis 2004 [7]. Il permet de réaliser différents types d'applications :

- Physique des réacteurs (étude de cycle de combustible, modélisation de réacteur de recherche...)
- Simulation multiphysique
- Simulation de transport de neutrons et de photons pour la radioprotection

La méthode de calcul MONTE-CARLO est une technique de calcul probabiliste, elle permet de simuler aléatoirement le comportement de différentes particules, par exemple dans un réacteur.

Le code SERPENT est un jeune code et le calcul de transitoire y est encore en version bêta néanmoins un forum est à disposition des utilisateurs pour toutes questions ainsi qu'un wiki.

4.2 Code déterministe - DRAGON-DONJON

DRAGON5 [3] est un code de réseau développé autour de la résolution numérique de l'équation du transport neutronique. Le code fonctionne via l'exécution de plusieurs modules de calculs liés les uns aux autres par le superviseur général GANLIB [4]. Les modules de calculs neutroniques n'ont pas été utilisés puisque SERPENT traite cette opération.

Le module **COMP0** : est utilisé pour constituer une base de donnée du réacteur pour des calculs en cœur complet.

Le module **GEO** : permet de créer ou de modifier une géométrie. **DONJON5** [1] est un code de modélisation cœur-complet développé autour de la résolution numérique de l'équation de la diffusion neutronique.

Le module **THM** : est un module de thermohydraulique construit autour de freesteam, une implémentation open source de la table IAPWS-IF97 de vapeur pour l'eau légère. **THM** : fonctionne en régime permanent et en conditions de transitoire.

4.3 Code de calcul thermohydraulique et thermomécanique - TRIAD

TRIAD est un code de calcul thermohydraulique et de thermo-mécanique basé sur les recommandations de la Référence [2] décrivant les modèles utilisés par le code SCANAIR. Les calculs sont réalisés sur un canal c'est à dire, un crayon de combustible entouré d'eau.

Le module **THERMO** : regroupe les modèles thermohydrauliques du code TRIAD alors que le module **MECA** : regroupe les modèles thermo-mécaniques. Les gaz de fission ne sont pas modélisés actuellement.

Dans ce projet, le module **THM** : et **THERMO** : ont été utilisés mais pas le module **MECA** :. Ils permettent de calculer l'évolution de la température du combustible et du modérateur ainsi que la densité du modérateur le long du crayon lors d'un transitoire.

Leur implémentation était différente mais les fichiers utilisés et les données nécessaires pour un couplage étaient les mêmes et facilitaient leur utilisation avec SERPENT.

5 Mise en place de la nouvelle méthode de calcul

La nouvelle méthode de calcul est présentée en deux étapes, chaque utilisation des logiciels est expliquée afin de comprendre comment est réalisé le couplage.

5.1 Géométrie sous SERPENT sans maillage axiale

Pour simuler l'assemblage du réacteur il était nécessaire de reproduire sa géométrie sur le logiciel SERPENT. Son fonctionnement est proche du code MCNP développé par les chercheurs du Los Alamos National Lab, cela a facilité sa prise en main étant donné qu'il est étudié durant la formation Génie Nucléaire à l'ENSICAEN.

À partir des informations disponibles [9] la géométrie a pu être mise en place, une déclaration de la géométrie et des matériaux étaient nécessaires.

Les données de géométrie et compositions des matériaux ont été récupérées depuis une géométrie existante sous DRAGON et adapté pour SERPENT.

Un crayon se définit comme il est possible de le voir sur la figure 7. Les matériaux sont définis par les différents isotopes les composant, comme le montre la figure 8, avec leur code atomique selon le format AAAZZZ. Ils sont définis en densité atomique ou en densité massique.

La géométrie du réacteur étant très hétérogène avec les barres d'hélium réparties autour du circuit expérimental, la simulation de l'assemblage a du être adaptée afin qu'il représente le cœur entier en terme de réactivité.

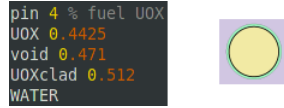


FIGURE 7 – Définition des matériaux du crayon de combustible dans SERPENT

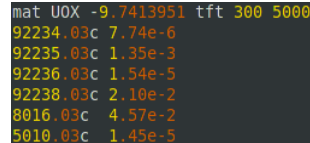


FIGURE 8 – Définition du combustible UOX dans SERPENT

L'assemblage simulé est un assemblage de 7x7 crayons, d'une hauteur de 80 cm et de 8,82 cm de côté. Le crayon d'hélium-3 est placé au centre. L'assemblage est défini en deux sections comme indiqué sur la figure 9 ci-dessous :

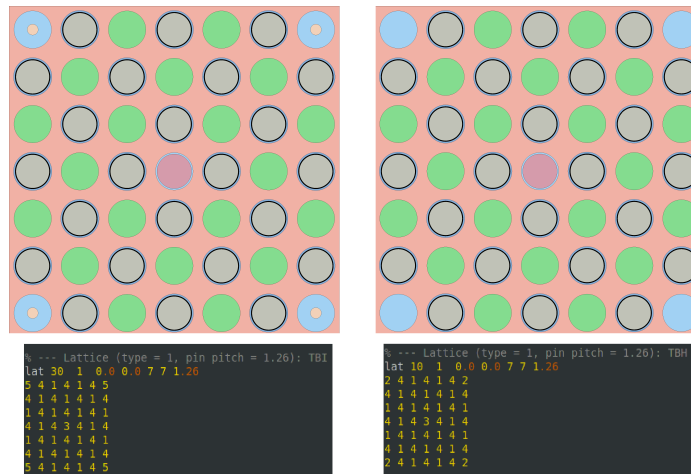


FIGURE 9 – L'étage TBI (Toutes Barres Insérées) à gauche et l'étage TBH (Toutes Barres Hautes) à droite avec leur définition dans SERPENT

Les sections sont définies à l'aide de l'option lattice, un carré infini sur l'axe vertical. Il faut donc les tronquer à l'aide de plan à différentes hauteurs afin de respecter la géométrie de l'assemblage.

Comme le montre la figure 10, la section TBI va remplir la zone haute de l'assemblage et la section TBH la zone basse de l'assemblage.

Il était nécessaire de faire des essais sur différentes hauteurs d'insertion de barres de contrôle afin de trouver la profondeur pour laquelle l'assemblage serait critique, c'est à dire tel que le coefficient de multiplication effectif soit égal à l'unité. Ces essais étaient réalisés en modifiant la hauteur du plan séparant les deux sections de l'assemblage.

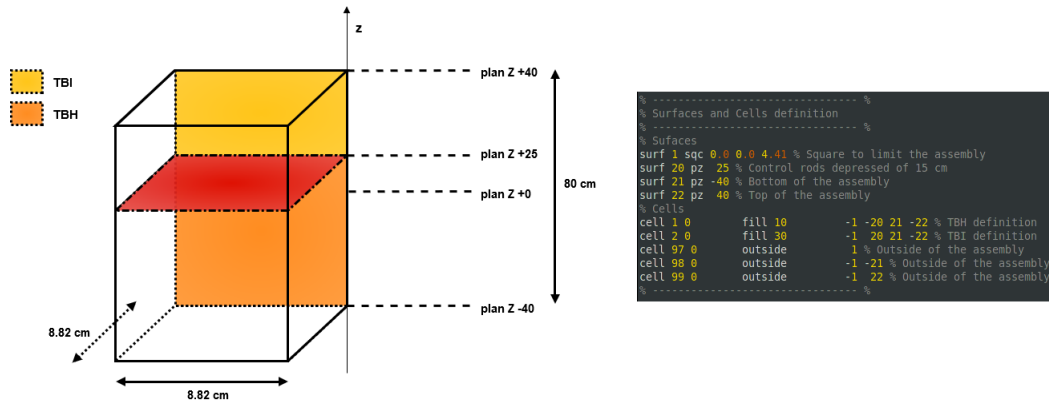


FIGURE 10 – Schéma représentant la géométrie avec sa définition dans SERPENT

5.2 Géométrie sous SERPENT avec maillage axiale

Il est possible de mailler la géométrie pour obtenir une répartition des densités et des températures correspondant au comportement physique d'un assemblage. SERPENT utilise des fichiers d'interface pour obtenir la densité et la température du matériau sur chaque maille. Ces fichiers d'interface sont formatés (comme on peut le voir sur la figure 11 ci-dessous) pour faciliter leur manipulation avec des langages tels que le C ou le Python. Ils permettent la prise en compte des densités et températures calculées par un code de thermohydraulique (externe) [10].

```

1 2 UOX 1
2 /home/u097152/serpent2/examples/TEST/fuelife.out 20 -40 40 1
3 1
4 1 -4.41 +4.41 1 -4.41 +4.41 20 -40 +40
5 -9.7413951 331.013733
6 -9.7413951 344.623535
7 -9.7413951 356.760620
8 -9.7413951 367.954590
9 -9.7413951 378.011414
10 -9.7413951 386.692749
11 -9.7413951 393.770625
12 -9.7413951 399.101257
13 -9.7413951 402.535004
14 -9.7413951 403.991150
15 -9.7413951 403.423584
16 -9.7413951 400.858765
17 -9.7413951 396.357056
18 -9.7413951 390.827527
19 -9.7413951 382.805005
20 -9.7413951 372.219574
21 -9.7413951 360.893982
22 -9.7413951 350.267212
23 -9.7413951 340.848492
24 -9.7413951 329.240540
    
```

FIGURE 11 – Exemple du fichier de maillage pour l'UOX

Des fichiers d'interfaces doivent être créés pour l'UOX (le combustible), l'eau (le modérateur) et l'hélium-3 (le crayon d'insertion de réactivité).

La visualisation de la géométrie permet de vérifier que le maillage a été correctement réalisé. On observe sur la figure 12 que la géométrie est effectivement scindée en 20 parties et que les températures et densités changent (couleurs plus ou moins sombres) avec la position spatiale.

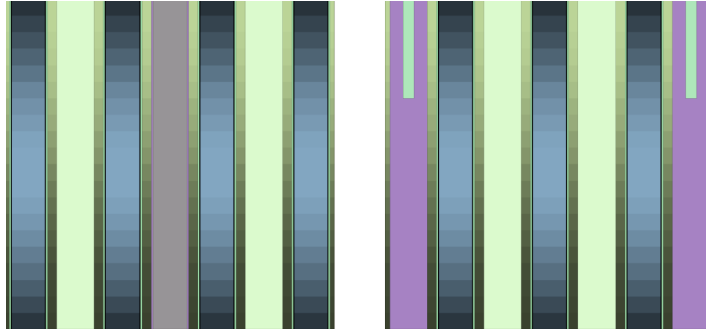


FIGURE 12 – Vue en coupe de la géométrie maillée selon l’axe vertical

5.3 Calcul de transitoire

Les calculs de transitoire sont possibles sur le logiciel SERPENT mais sont encore en version bêta, des discussions avec les développeurs ont été nécessaires [11].

Le mode transitoire se passe en deux temps. Une première étape consiste à créer une source de neutrons dans la situation critique (géométrie proche de la criticité). Dans un second temps, on introduit la perturbation de puissance sur une période de temps donnée via la modification de la concentration d’hélium dans le fichier interface he3.ifc.

Il est possible de suivre la puissance de l’assemblage lors du calcul du transitoire en adaptant les détecteurs mis à disposition par SERPENT.

```
det 1 dr -8 void dx -4.41 +4.41 1 dy -4.41 +4.41 1 dz -40 +40 20
det 2 dr -8 void
```

Ces détecteurs permettent par exemple d’obtenir l’énergie due aux fissions (en Joules) des 20 mailles selon l’axe vertical et l’énergie correspondant à l’assemblage complet. Lors d’un calcul de transitoire, un fichier de résultat détecteur est créé à chaque intervalle de temps comme le montre la figure 13.

CABRI_TRANSIENT_det0.m	2,4 Kio	13/07/2018
CABRI_TRANSIENT_det1.m	2,4 Kio	13/07/2018
CABRI_TRANSIENT_det2.m	2,4 Kio	13/07/2018
CABRI_TRANSIENT_det3.m	2,4 Kio	13/07/2018

FIGURE 13 – Liste de fichiers détecteurs après le calcul d’un transitoire

Le formatage de ces fichiers permet un traitement simple en C et ainsi déterminer la puissance pour le couplage. La figure 14 permet de visualiser ce formatage.


```

DET1 = [
1 1 1 1 1 1 1 1 1 1 1 1 4.07966E-03 0.16223
2 1 1 1 1 1 1 1 2 1 1 1 8.27018E-03 0.14718
3 1 1 1 1 1 1 1 3 1 1 1 1.27338E-02 0.16949
4 1 1 1 1 1 1 1 4 1 1 1 1.65310E-02 0.16654
5 1 1 1 1 1 1 1 5 1 1 1 1.95522E-02 0.15722
6 1 1 1 1 1 1 1 6 1 1 1 2.31364E-02 0.15859
7 1 1 1 1 1 1 1 7 1 1 1 2.44837E-02 0.15213
8 1 1 1 1 1 1 1 8 1 1 1 2.64059E-02 0.15864
9 1 1 1 1 1 1 1 9 1 1 1 2.88914E-02 0.11981
10 1 1 1 1 1 1 1 10 1 1 1 2.96897E-02 0.13157
11 1 1 1 1 1 1 1 11 1 1 1 3.08377E-02 0.13306
12 1 1 1 1 1 1 1 12 1 1 1 2.96726E-02 0.14303
13 1 1 1 1 1 1 1 13 1 1 1 2.98787E-02 0.15898
14 1 1 1 1 1 1 1 14 1 1 1 2.43383E-02 0.14806
15 1 1 1 1 1 1 1 15 1 1 1 2.39227E-02 0.14369
16 1 1 1 1 1 1 1 16 1 1 1 2.84382E-02 0.14607
17 1 1 1 1 1 1 1 17 1 1 1 1.61514E-02 0.12534
18 1 1 1 1 1 1 1 18 1 1 1 1.32748E-02 0.15123
19 1 1 1 1 1 1 1 19 1 1 1 9.24687E-03 0.15414
20 1 1 1 1 1 1 1 20 1 1 1 4.02671E-03 0.14227
];
DET2 = [
1 1 1 1 1 1 1 1 1 1 1 1 1.84428E-01 0.14344
];

```

FIGURE 14 – Exemple du formatage des fichiers détecteurs

5.4 Solution thermohydraulique

Le calcul de l'évolution thermohydraulique est réalisé par des procédures prenant la forme de fichiers .c2m et appelant les différents modules disponibles dans DRAGON-DONJON. Une procédure existait déjà pour un calcul de transitoire entièrement réalisé par DONJON via des données DRAGON. Elle devait donc être adaptée pour permettre le couplage avec une solution de calcul neutronique externe. La procédure a été scindée en deux fichiers, l'initialisation et le calcul.

Ces procédures créent deux fichiers utiles, l'objet FMAP et THM, ils contiennent les informations sur l'évolution de la température et de la densité lors du transitoire, un exemple de distribution de température est représenté sur la figure 15 et la distribution de puissance dans l'assemblage est représentée sur la figure 16.

Les températures et densités calculées par le code de thermohydraulique correspondent à un crayon de combustible entouré d'eau. Ces températures et densités sont attribués à chaque crayon composant l'assemblage.

Les puissances devront être mises à jour à chaque pas de temps puisque le calcul neutronique n'est pas utilisé dans les procédures.

Les fichiers Fmap.txt et THM.txt devront être transformés à chaque pas de temps en objet LCM. Des objets spécifiques au code DRAGON-DONJON pour communiquer en C. Ces objets LCM seront envoyés en entrée lors des appels de procédures pour poursuivre le calcul thermohydraulique.

```

P-VALUE
6.13906250E+02 9.27836426E+02 1.20763989E+03 1.48044666E+03 1.77775439E+03
1.99222766E+03 2.17294678E+03 2.29721484E+03 2.37209448E+03 2.42084375E+03
2.43340283E+03 2.39170435E+03 2.34219604E+03 2.24407764E+03 2.10861475E+03
1.93157581E+03 1.68632520E+03 1.40774719E+03 1.08492883E+03 6.96088684E+02

```

FIGURE 15 – Distribution selon l'axe vertical de la température du combustible

Les températures et densités seront récupérés à chaque pas de temps pour que le couplage fonctionne.

```
TOT-PW
0.258643
-> 1 12 2 20
BUND-PW
2.579400 4.662450 8.231825 10.728201 13.945301
14.186101 16.055552 18.092550 19.635900 18.972000
19.859951 18.591452 18.904625 16.876026 15.610101
13.332451 10.751575 8.825101 6.123125 2.686225
-> 1 12 2 40
```

FIGURE 16 – Puissance totale et distribution selon l’axe vertical de la puissance de l’assemblage

Les puissances devront être modifiées lors du couplage après avoir été récupérées dans SERPENT afin que l’évolution des températures et des densités puisse être calculée.

5.5 Le couplage

SERPENT est conçu pour pouvoir coupler des simulations neutroniques à des calculs de solutions thermohydrauliques ou thermomécaniques fournis par des codes externes en utilisant un mode de communication d’écriture et lecture de fichiers.

SERPENT requiert la distribution de température et de densité au sein du combustible et du modérateur ainsi que la densité d’hélium-3.

La solution thermohydraulique, qu’elle provienne des modules THM: ou THERMO:, nécessite la puissance totale ainsi que la puissance axiale de l’assemblage calculées par le code neutronique, en l’occurrence SERPENT.

La figure 17 représente de manière simplifiée les échanges d’informations nécessaires entre SERPENT et le code de calcul de thermohydraulique.

La manière dont ces échanges sont réalisés en C est expliquée dans les annexes.

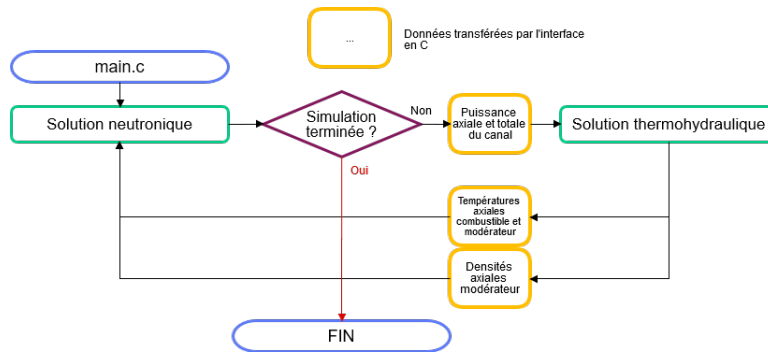


FIGURE 17 – Schéma de communication simplifié entre la solution thermohydraulique et la solution neutronique

Un langage de programmation tel que le C ou le Python prévoit des fonctions afin de récupérer des informations et de les transmettre à partir de fichiers textes. Le langage de programmation permet de lire ou de modifier ces fichiers.

5.5.1 Méthodes de communication

Une interface en C a été choisie pour réaliser le couplage entre la solution de calcul neutronique et thermohydraulique, c'est un langage bas niveau et répandu. L'interface en C va permettre de lier les solutions entre elles.

SERPENT : Le logiciel SERPENT utilise trois types de fichiers pour communiquer. Un fichier d'entrée appelé « com.in », un fichier de sortie appelé « com.out » et des fichiers d'interface. SERPENT est codé en C et utilise les signaux suivants pour communiquer [8] :

- SIGUSR1 : Étape terminée
- SIGTERM : Simulation terminée

SERPENT écrit dans le fichier de sortie et lit le fichier d'entrée et le programme en C écrit dans le fichier d'entrée et lit le fichier de sortie.

L'interface en C boucle jusqu'à lire un signal dans le fichier de sortie pour reprendre la main sur SERPENT, réaliser différents processus et écrire dans le fichier d'entrée afin que SERPENT reprenne la main. Si le programme en C lit le signal SIGTERM, la boucle est cassée et il réalisera le traitement des données avant de se fermer.

La figure 18 suivante schématise le processus de communication entre le programme en C et SERPENT.

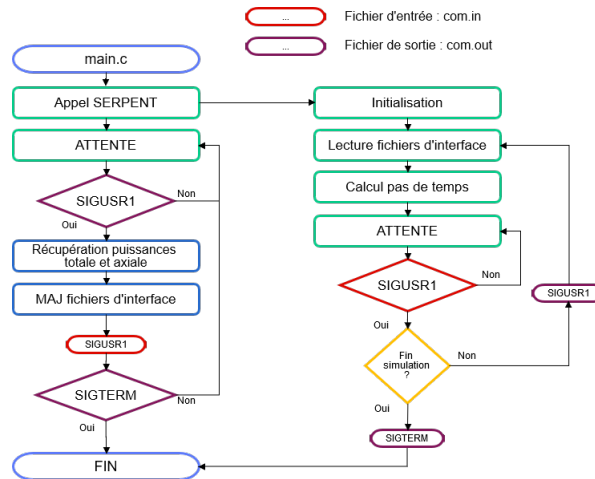


FIGURE 18 – Schéma de communication entre SERPENT et l'interface en C sans la thermohydraulique

Les modules THM: et THERMO: : La librairie GANLIB [4] présente dans la suite DRAGON-DONJON permet de communiquer avec des procédures .c2m à partir du C.

Tout est prévu pour l'implémenter rapidement dans un code C, la librairie est appelée comme une librairie standard du C comme l'indique la figure 19.

```
#include <signal.h>
#include <time.h>
#include "cle2000.h"
```

FIGURE 19 – Appel de librairie en C

L’interface en C permet de récupérer et transférer les objets nécessaires au bon fonctionnement du calcul de thermohydraulique à l’aide de fonctions créées autour des commandes GANLIB. Cette communication est schématisée de manière simplifiée sur la figure 20

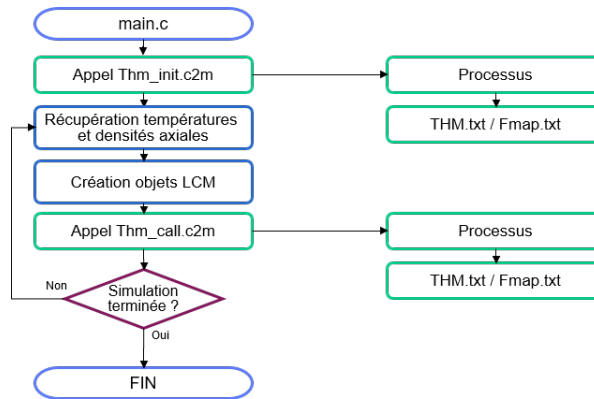


FIGURE 20 – Schéma de communication entre la solution thermohydraulique et l’interface en C sans solution neutronique

La création de fichiers textes par les procédures .c2m a deux utilités, la première, récupérer ces .txt sous forme d’objets LCM pour les transférer lors d’un appel de .c2m et la seconde, d’obtenir les données thermohydrauliques nécessaires au couplage.

5.5.2 Couplage global

Les trois solutions, le code de calcul de neutronique, le code de calcul de thermohydraulique et l’interface en C, fonctionnent ensemble suivant le schéma de la figure 21.

Ce couplage permet donc d’obtenir une simulation des comportements neutronique et thermohydraulique de l’assemblage et d’étudier des pics de puissances en cas de RIA.

5.6 Parallélisation du calcul

Les calculs Monté Carlo sont relativement chronophages. Pour obtenir une précision satisfaisante, le temps de calcul peut rapidement atteindre une dizaine d’heures. SERPENT propose un moyen de réduire ce temps de calcul, comme le montre la figure 22, en le répartissant sur plusieurs cœurs.

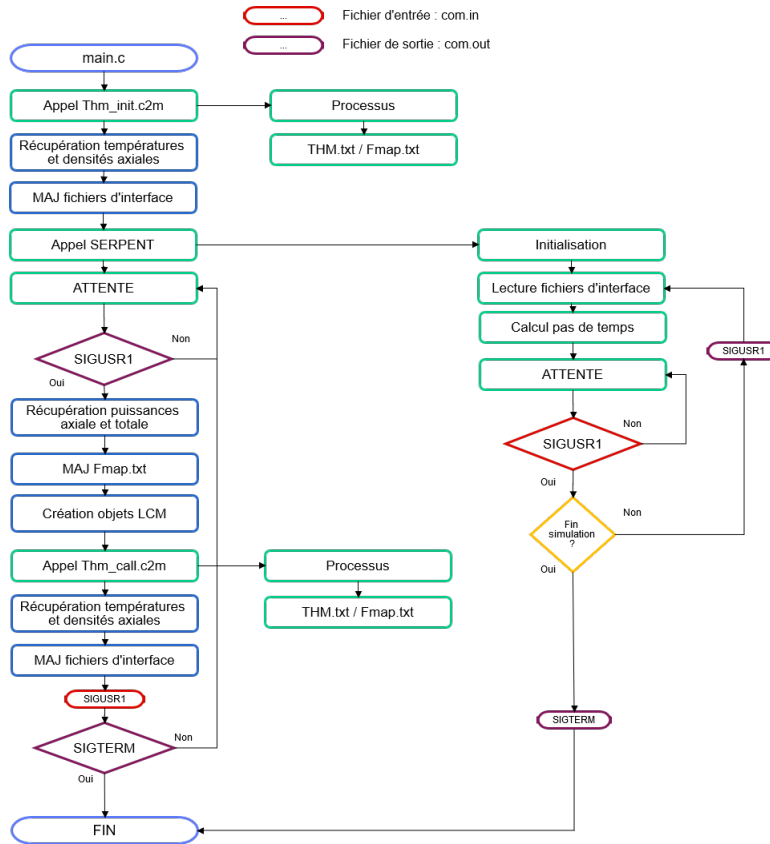


FIGURE 21 – Schéma global de couplage entre la solution thermohydraulique, la solution neutronique et l’interface en C

<pre> Serpent 2.1.30 -- Dynamic neutron external source simulation Title: "Calculation of a Lumped_Model_Reactor transient simulation" Source batch 1 / 5 (10000 histories per batch) Time interval 1 / 500 from 0.000000E+00 s to 4.000000E-04 s Running time : 0:03:36 Estimated running time : 122:02:15 Estimated running time left : 121:58:39 Estimated relative CPU usage : 99.7% Multiplication = 2.1047E+01 (0.00000) k-eff (analog) = 0.95249 +/- 0.00000 [0.95249 0.95249] k0 (source) = 1.15531 +/- 0.00000 [1.15531 1.15531] Time cut-off at 2.00E-01 seconds, 500 intervals (O4) (TMS) (OMP=1) (CC) </pre>	<pre> Serpent 2.1.30 -- Dynamic neutron external source simulation Title: "Calculation of a Lumped_Model_Reactor transient simulation" Source batch 4 / 5 (10000 histories per batch) Time interval 1 / 500 from 0.000000E+00 s to 4.000000E-04 s Running time : 0:02:07 Estimated running time : 8:16:36 Estimated running time left : 8:15:29 Estimated relative CPU usage : 1648.5% Multiplication = 1.7712E+01 (0.01660) k-eff (analog) = 0.94349 +/- 0.00092 [0.94169 0.94530] k0 (source) = 1.14878 +/- 0.00453 [1.13990 1.15767] Time cut-off at 2.00E-01 seconds, 500 intervals (O4) (TMS) (OMP=22) (CC) </pre>
--	---

FIGURE 22 – Calcul d’un transitoire sur un coeur à gauche et sur 22 coeurs à droite

Le temps de calcul est ainsi réduit d’un facteur 15 et permet de lancer des calculs et d’étudier les résultats chaque jour.

6 Résultats

Le couplage présenté précédemment a permis d'obtenir un transitoire. Il est représenté sur la figure 23.

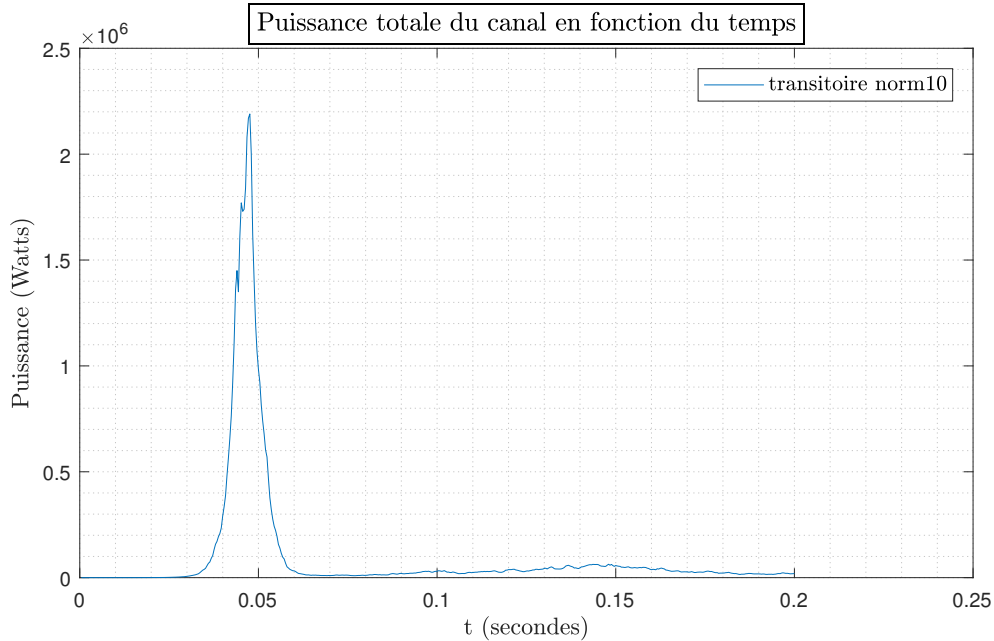


FIGURE 23 – Transitoire simulé à l'aide du couplage SERPENT-THERMO : , cas du transitoire norm10

Il est possible d'observer sur la figure 23 que la puissance augmente de manière exponentielle sur une période de temps très courte, environ 35 millisecondes. La forme du pic de puissance correspond au pic présenté dans la partie présentant le réacteur CABRI.

Cependant, des différences sont à observer puisque le maximum de puissance atteint par le pic (2,2 MW) est inférieur à ce qui est attendu (11,7 MW). Néanmoins l'ordre de grandeur, l'ordre du mégawatt, est respecté.

Obtenir un pic de puissance de cette ampleur, sur une échelle de temps aussi courte à l'aide d'un code de calcul neutronique Monte Carlo est rarissime voir même peut-être une première.

Il faut se remémorer que les calculs de transitoires sur le logiciel SERPENT sont encore en version bêta et que la méthode utilisée dans SERPENT, à savoir la cinétique, pourrait être améliorée.

7 Conclusion

Le but de ce projet était de proposer une solution de couplage entre un code de calcul neutronique Monte Carlo appelé SERPENT et un code de calcul thermohydraulique développé à l'École Polytechnique de Montréal via les modules **THM:** et **THERMO:** des codes DONJON et TRIAD afin que l'IRSN ait une base pour remplacer le code SCANAIR.

Une part importante du projet était de réussir à faire communiquer correctement les différents logiciels entre eux. Le couplage est fonctionnel et les données entre les entités sont échangées convenablement. Un transitoire a été correctement simulé et le pic de puissance correspond en terme d'ordre de grandeur au transitoire attendu.

Des pistes d'améliorations de la simulation sont possibles : la précision de la simulation peut être augmentée en simulant plus de neutrons, prendre en compte les comportements thermomécaniques via le module **MECA:**, utiliser des méthodes de cinétique plus précises dans le code SERPENT.

Ce travail permet de planifier la simulation du transitoire pour le coeur complet dans un avenir proche. Un travail de doctorat encadré par l'IRSN est d'ailleurs envisagé. Dans cette optique, poursuivre les discussions avec les développeurs de SERPENT serait profitable afin de comprendre correctement ce code Monte Carlo.

Annexes

Annexe A : Récupération des données thermohydrauliques

La manière dont les données thermohydrauliques sont récupérées est schématisée sur la figure 24.

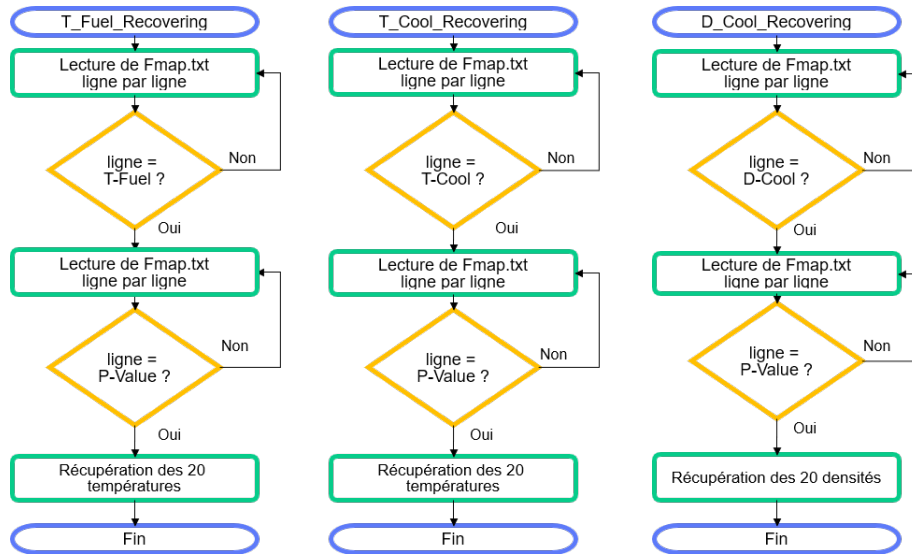


FIGURE 24 – Schéma simplifié de la récupération des données thermohydrauliques

Le programme lit ligne par ligne le fichier Fmap.txt. Pour trouver la donnée recherchée et être sûr de récupérer celle attendue, la température du combustible par exemple, des jetons sont utilisés. Le fichier texte étant toujours formaté de la même façon, la température du combustible se trouve après les lignes T-FUEL et P-VALUE.

Si ces lignes sont lues, deux variables sont mises à un et il est possible de récupérer les 20 températures réparti sur 4 lignes avec la commande :

```

if(token_Tfuel && token_Pvalue){
    while(i<=16){
        fscanf(FMAP, "%f %f %f %f %f", &(*t_T_fuel)[i], &(*t_T_fuel)[i+1],
            &(*t_T_fuel)[i+2], &(*t_T_fuel)[i+3], &(*t_T_fuel)[i+4]);
        i = i + 5 ;
    }
    token_Tfuel = token_Pvalue = 0;
    i = 1 ;
}
    
```

La commande fscanf permet de lire des lignes de texte et récupérer des variables selon un format précis.

Les données sont stockées dans un tableau pour être utilisées par la suite. La méthode

est la même pour récupérer la densité et la température du modérateur, il faut néanmoins chercher la ligne T-COOL ou D-COOL au lieu de T-FUEL.

Annexe B : Mise à jour des fichiers d'interface

Les données thermohydrauliques récupérées, la mise à jour des fichiers d'interface peut être réalisée.

Le programme va simplement créer ou écraser le fichier d'interface en écrivant les nouvelles données thermohydrauliques.

Les premières lignes des fichiers fuel.ifc, data.ifc et he3.ifc sont réservées aux informations devant être données à SERPENT, le matériau concerné ou le maillage.

```
fputs("2 WATER 1\n",ifc);
fputs(IFC_OUT,ifc);
fputs("1\n",ifc);
fputs("1 -4.41 +4.41 1 -4.41 +4.41 20 -40 +40\n",ifc);
```

Cette commande écrit dans le fichier que le matériau WATER sera divisé en 20 sur l'axe vertical. La commande fput permet d'écrire des données directement dans le fichier texte.

Les 20 densités et températures doivent être écrites à la suite, la difficulté est que ces valeurs sont variables. La commande fprintf est donc utilisée. Cette dernière permet d'écrire une ligne selon un format désiré à l'aide de variables.

```
for(i = 1 ; i<=20 ; i++){
    fprintf(ifc,"%f %f\n",(*t_D_cool)[i],(*t_T_cool)[i]);
}
```

La même méthode est utilisée pour le combustible dans le fichier fuel.ifc. L'hélium n'étant pas maillé, il suffit d'écrire une densité et une température pour la totalité du crayon d'insertion de réactivité.

Annexe C : Récupération des énergies des détecteurs SERPENT

La récupération des énergies provenant des fichiers détecteurs est réalisée de la même manière que la récupération des données thermohydrauliques comme le montre la figure 25.

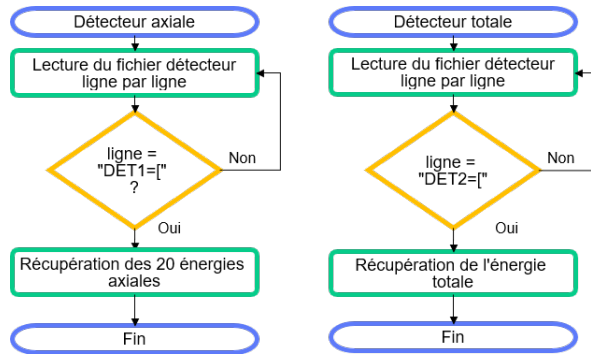


FIGURE 25 – Schéma simplifié de la récupération des énergies calculées par SERPENT

Les fichiers détecteurs ont un nom variable étant donné qu'ils sont créés à chaque pas de temps de la simulation.

Par exemple : CABRI_TRANSIENT_det0.m / CABRI_TRANSIENT_det1.m

Il était donc nécessaire de pouvoir lire des fichiers à noms variables pour récupérer les énergies axiales et l'énergie totale.

La commande suivante permet de créer des chaînes de caractères à partir d'une variable :

```
sprintf(file_Name, CABRI_TRANSIENT_det%d.m, i);
```

La variable *i* est incrémentée à chaque pas de temps et permet donc de créer une chaîne de caractère contenant le nom correct du détecteur. Il peut ensuite être lu de la même manière que le fichier *Fmap.txt*.

Annexe D : Modification de la distribution de puissance

A chaque pas de temps les puissances du fichier Fmap.txt doivent être mises à jour. Pour cela, les énergies venant des détecteurs SERPENT sont transformées en puissance en les divisant par le pas de temps de la simulation.

Par exemple : 0.65 divisé par 0.0004 va donner 1680 W.

Dans le fichier Fmap.txt, les puissances axiales sont définies après le mot-clé BUND-PW, elles sont en kW, et la puissance totale est définie deux fois. Après le mot-clé TOT-PW et après REACTOR-PW, elle est en MW.

Les puissances sont calculées d'après les commandes suivantes :

```
float power = ((*t_power)[i])/t_dt) * 1e-6 ; // total power in MW

float axial_power[20] ;
for(k = 0 ; k <=19 ; k++){
    axial_power[k] = ((*t_axial_p)[k])/t_dt) * 1e-3 ; // axial power in kW
}
```

La modification d'un fichier est différente de simplement récupérer une information dans ce dernier. Effectivement, il est nécessaire de réécrire le fichier dans son intégralité pour en modifier seulement une partie.

Le fichier initial Fmap.txt est recopié ligne par ligne dans un fichier Fmap_copy.txt jusqu'à arriver aux lignes à modifier, la déclaration des puissances.

Lorsque la ligne contenant BUND-PW est détectée, le programme écrit la nouvelle distribution de puissance. Le programme écrit donc 4 lignes alors qu'au niveau de la lecture du fichier, le programme est encore à la ligne BUND-PW. Il faut donc forcer le programme à lire les 5 lignes suivantes puis poursuivre la copie. Toute cette étape est réalisée par la commande ci-dessous :

```
fputs(line, fmap_copy);
while(j <= 15){
    fprintf(fmap_copy,"%f %f %f %f %f\n", axial_power[j], axial_power[j+1],
    axial_power[j+2], axial_power[j+3],axial_power[j+4]);
    j = j + 5 ;
}
fgets (line, MAX_SIZE, fmap); // Lecture ligne suivante sans copie
fgets (line, MAX_SIZE, fmap); // Lecture ligne suivante sans copie
fgets (line, MAX_SIZE, fmap); // Lecture ligne suivante sans copie
fgets (line, MAX_SIZE, fmap); // Lecture ligne suivante sans copie
fgets (line, MAX_SIZE, fmap); // Lecture ligne suivante sans copie
```

La commande fputs écrit la ligne en train d'être lu dans le fichier copié et fgets lit la prochaine ligne du fichier initial.

La même méthode est utilisée pour la puissance totale avec les mots-clé TOT-PW et REACTOR-PW mais une seule ligne est modifiée.

Références

- [1] D. Sekki A. Hébert and R. Chambon. "A user guide for DONJON version5" École polytechnique de montréal montréal qc, canada, tech. rep. ige-344 (2018). <http://www.polymtl.ca/merlin/downloads/IGE344.pdf>.
- [2] V. Georgenthum A. Moal and O. Marchan. "scanair : A transient fuel performance code, part one : General modelling description," nucl. eng. and design, 280, 150–171 (2014).
- [3] A. Hébert G. Marleau and R. Roy. "A user guide for DRAGON version5", École polytechnique de montréal montréal qc, canada, tech. rep. ige-335 (2018). <http://www.polymtl.ca/merlin/downloads/IGE335.pdf>.
- [4] A. Hébert and R. Roy. "the ganlib5 kernel guide (64-bit clean version)," École polytechnique de montréal, rapport ige-332, 2015. <http://www.polymtl.ca/merlin/downloads/IGE332.pdf>.
- [5] IRSN. L'installation cabri et les instruments de mesures. 2018. <https://www.irsn.fr/FR/Larecherche/Organisation/Programmes/CIP/Pages/Installation-CABRI.aspx>.
- [6] IRSN. Code de calcul scanair. <http://www.irsn.fr/FR/Larecherche/outils-scientifiques/Codes-de-calcul/Pages/logiciel-calcul-Scanair.aspx#.Wnt1ZVEiG1i>.
- [7] Leppänen J. et al. Serpent - a monte carlo reactor physics burnup calculation code. 2018. <http://montecarlo.vtt.fi/>.
- [8] Leppänen J. et al. Coupled multi-physics calculations - serpent wiki. 2018. http://serpent.vtt.fi/mediawiki/index.php/Coupled_multi-physics_calculations.
- [9] Leppänen J. et al. Input syntax manual - serpent wiki. 2018. http://serpent.vtt.fi/mediawiki/index.php/Input_syntax_manual.
- [10] Leppänen J. et al. Multi-physics interface - serpent wiki. 2018. http://serpent.vtt.fi/mediawiki/index.php/Multi-physics_interface.
- [11] Leppänen J. et al. Transient simulations - serpent wiki. 2018. http://serpent.vtt.fi/mediawiki/index.php/Transient_simulations.
- [12] International Group on Research Reactors CEA. Le programme Cabri (CIP) [archive]. 2003. <http://www.igorr.com/scripts/home/publigen/content/templates/Show.asp?P=741&L=EN>.

Résumé

Ce document présente de manière détaillée comment un couplage entre un code de calcul neutronique de type Monte Carlo et un code de calcul thermohydraulique a été réalisé. Ce couplage a été produit dans le cadre de simulation de RIA (Accident d'Insertion de Réactivité) du réacteur de recherche CABRI. Cette simulation s'appuie sur l'étude d'un assemblage représentatif du comportement neutronique du réacteur CABRI. Ce travail permet néanmoins la simulation du transitoire pour le cœur complet.

Mots clés : CABRI, SERPENT, Monte Carlo, École Polytechnique de Montréal, transitoire, couplage, programmation

Abstract

This report shows how was realize a coupling between a neutron transport numerical code and a thermohydraulic numerical code. The coupling aims to simulate a reactivity insertion accident in an assembly representative of the neutron behaviour of the CABRI research reactor. This coupling provides a way to simulate the transient on the whole core.

Keywords : CABRI, SERPENT, Monte Carlo, École Polytechnique de Montréal, transient, coupling, programming