

Organisation des différents fichiers

Vincent Descotes

22 avril 2011

Résumé

Cet article correspond à un fichier type LISEZ-MOI. Il a pour but d'expliquer comment sont organisés les fichiers qui m'ont servi durant ma maîtrise recherche sur les VHTR en 2010/2011. Ceci devrait permettre de retrouver rapidement les fichiers d'entrée DRAGON ou INSTANT comportant ce que vous souhaitez. J'expose aussi comment était géré le schéma de calcul de manière automatique (ou à peu près). Plusieurs scripts écrits en bash, fortran ou python peuvent ainsi s'avérer utiles comme source d'inspiration.

1 Introduction

J'ai souhaité rédiger ce court document afin de faciliter l'utilisation ultérieur de mes fichiers d'entrée DRAGON / INSTANT par ceux qui voudraient y retrouver des schémas de calcul. On trouvera dans ce dossier tous les fichiers utilisés pour la rédaction de mon mémoire de maîtrise.

Je présenterai dans un premier temps les fichiers ayant trait à DRAGON, puis les scripts bash, python ou fortran qui permettaient d'automatiser certaines tâches. Je termine par quelques conseils que j'espère utiles sur Dragon ou L^AT_EX 2_ε.

2 Fichiers d'entrée DRAGON

2.1 Description du travail

Ma maîtrise nécessitait des calculs de type « bloc simple » (single block) ou bien « supercellule » (supercell). J'ai utilisé les trackings SYBILT, EXCELT-MCCGT, SNT. Des calculs SPH ont été faits avec un solveur MOC et d'autres plus rapides de DONJON (diffusion, SP1/SP3).

On trouvera donc l'arborescence décrite ci-dessous.

2.2 VHTR_SB_moc

Contient tous les calculs de bloc simple (SB) en MOC. On trouve les sous-dossiers DH295 pour double-hétérogénéité, où les calculs sont faits en partant directement de la librairie. Les dossiers HOM295 et HOM26 correspondent à des calculs où on utilise des sections partiellement homogénéisées (cellules par cellules). On fait appel à NCR pour créer une librairie contenant le combustible et le reste séparément. La multicompo est générée par un calcul de bloc simple à part, placée directement dans VHTR_SB_moc. Plusieurs condensations sont disponibles : 295, 26, 23, 13, 10, 9, 6, 4 et 2 groupes. dans le cas des calculs à 26 groupes, on ne pouvait condenser qu'à 26, 10, 6, 4 et 2 groupes pour des raisons de frontières des groupes (chevauchement).

2.3 VHTR_SB_syb

Contient tous les calculs de bloc simple (SB) avec la méthode des probabilités de collision (SYB). On trouve les sous-dossiers DH295 pour double-hétérogénéité, où les calculs sont faits en partant directement de la librairie. Les dossiers MOC295 et MOC26 correspondent à des calculs où on utilise une multicompo pour faire le calcul. On fait appel à NCR pour créer une librairie contenant le combustible et le reste séparément. La multicompo utilisée provient de VHTR_SB_moc, pour rester cohérent.

2.4 VHTR_Refl_moc et VHTR_Refl_moc26

Les calculs de réflecteur ont été effectués à 295 ou 26 groupes. Chaque dossier contient la même organisation, détaillée ci-dessous.

2.4.1 VHTR_Refl_void

Contient le calcul de réflecteur fait avec une géométrie cylindrique. Des tests ont été faits avec SNT et SYB avant de choisir SNT.

2.4.2 VHTR_Refl_B10

Calcul de réflecteur où le vide est simulé par une couche de bore rajoutée à l'extérieur. Cette solution permettait de faire le calcul en gardant une condition de réflexion au bord, ce qui est indispensable pour produire des coefficients de diffusion. La solution a été abandonnée en cours de route, du fait du choix du logiciel INSTANT qui ne nécessitait pas de transférer les coefficients de diffusion (méthode P_N).

2.5 VHTR_SC_moc295

Contient les calculs de supercellule faits avec MOC en 295 groupes. Trois types de géométrie ont été utilisées, et donc regroupées dans les dossiers Cell_n13, Cell_n18, et Cell_n19. Il n'était pas possible d'effectuer le calcul directement avec une librairie. Une multicompo a donc été générée dans un calcul de bloc simple pour homogénéiser l'intérieur des particules TRISO. L'auto-protection n'est plus possible une fois que des mélanges ont été homogénéisés. On considère que celle effectuée lors du calcul du flux dans le bloc simple est suffisante. Dans chaque sous-dossier Cell..., on trouvera les scripts Fortran « buildsupercell.f » écrits par Michael Pope et utilisés pour générer la géométrie. Automatiser cette tâche s'est avéré être indispensable avec 3367 cellules ! Ce script générait un système de coordonnées non orthogonal dans le plan, puis remplissait les cases avec le type de cellule voulu

correspondant à un bloc simple. Il « suffisait » donc de connaître le centre de chaque bloc, et d'écrire à la main comment était fait un bloc. Cela reste un peu compliqué à mettre au point, mais une fois que cela a été fait, je n'y ai plus touché. Les scripts DragSupCell.py étaient chargés de bâtir le reste du fichier d'entrée pour DRAGON, avec en option le nombre de groupes voulus pour la condensation.

2.6 VHTR_SC_moc26

Même chose, mais la multicompo de départ ne comporte que 26 groupes.

2.7 VHTR_SC_syb295 et VHTR_SC_syb26

Même chose mais avec le module SYB plutôt que EXCELT-MCCGT pour le tracking. Les multicompos sont par contre générées par un calcul MOC : ce sont les mêmes que pour les autres super-cellules calculées en MOC.

2.8 Evol_VHTR

Calculs d'évolution sur des super-cellules. Ces calculs avaient pour but de sortir à différents burnups les concentrations d'une cellule homogénéisée sur une ligne en zig zag partant de la gauche d'un bloc hexagonal et se terminant à droite. On pouvait ainsi caractériser l'influence du réflecteur sur les concentrations, uniquement en utilisant DRAGON. Des calculs de cœur ont été tenté mais n'ont rien donné d'utilisable.

2.9 SPH_SC

Ce dossier contient tous les calculs SPH, et reprend la même organisation entre les calculs de super-cellule suivant le type de tracking et le nombre de groupes utilisés au départ du calcul. Au passage, SPH ne peut marcher que si vous fournissez une macro-géométrie avec les indices de mélanges utilisés

dans l'homogénéisation dans EDI. Il n'est pas possible de faire par exemple moitié mélanges homogènes de EDI, moitié mélanges issus d'une librairie ou d'une autre multicompo. Tous les mélanges de EDI doivent a priori être utilisés dans la macro-géométrie. Je n'ai jamais réussi à faire passer un fichier de tracking dans SPH à la place de la géométrie, mais en 2D, le tracking est une étape rapide. Il est conseillé de vérifier sa macro-géométrie dans un fichier à part avant de l'inclure dans SPH. D'une manière générale, mieux vaut d'ailleurs avoir un fichier d'entrée qui marche nickel avant d'introduire SPH.

2.10 test_TRIVAC, test_6gr_TRIVAC et test_2gr_TRIVAC

Ces dossiers contiennent les fichiers ayant permis de vérifier la bonne marche de SPH. Cela consistait à effectuer un calcul sans SPH, à récupérer les mélanges homogènes dans une multicompo, puis à les faire tourner dans la macro-géométrie et à étudier les taux de réaction à 2 groupes (condensation en fin de parcours) et le Keff. On obtient logiquement des différences avec le calcul hétérogène de départ. Lorsque la même démarche est effectuée avec des sections efficaces corrigées par un calcul SPH, on retombe sur les mêmes taux de réaction et le même Keff que dans le cas hétérogène, ce qui est bien le but de la correction SPH. Pour ces calculs, le flux hétérogène a été sauvé en ASCII afin d'accélérer le processus. Le calcul d'un flux en super-cellule nécessite en effet 3 jours de calcul sur la machine Doppler. . . Les calculs ont été faits sur 26 ou 6 groupes, suivant le dossier.

3 Production de spectres neutroniques

Un spectre neutronique décrit la répartition des neutrons en fonction de l'énergie. Il est nécessaire de normaliser correctement le flux afin de pouvoir comparer des spectres entre eux. L'hypothèse utilisée dans mon mémoire est que l'intégrale sous la courbe doit être égale à 1 dans tous les cas. La

difficulté vient du fait que le flux est donné en multi-groupe et non de manière continue : la normalisation n'est donc pas aussi triviale qu'il y paraît, cf mon mémoire pour plus de détails.

Un code a été écrit en Fortran par Michael Pope pour lire les flux produits par DRAGON et produire un spectre normalisé : flux_reader. C'est ici que sont rassemblés les codes pour obtenir le spectre à partir d'un fichier de sortie de DRAGON. Il fallait copier la partie contenant les résultats (taux de réaction) dans un fichier « flux ». Le script readit295.f lit le fichier « flux », extrait les valeurs de flux pour chaque groupe, puis utilise les valeurs situées en début de script et correspondant aux frontières de groupe pour faire la normalisation.

Attention! Il faut changer le début si on change de structure de groupe. De même, en fonction du nombre de mélanges homogénéisés, il faut changer la commande de lecture du fichier, pour qu'il saute le bon nombre de ligne entre chaque valeur du flux. Cela est fait dans la commande READ par le nombre de / : un « / » correspond à un saut de ligne avec arrêt au tout début de la ligne suivante.

4 Passage des sections efficaces à INSTANT

INSTANT nécessitait la donnée de sections efficaces macroscopiques.

Il a été décidé de produire en fin de calcul DRAGON un fichier de sortie de EDI au lieu d'utiliser par exemple les multicompos. Dans chaque fichier de sortie, 16 isotopes étaient conservés, correspondant aux isotopes présents au début du calcul. Le mot-clef MICR 16 « Nom ou alias des isotopes » SAVE permettait de faire cela. Les isotopes nécessaires à l'évolution étaient donc retirés, ce qui allège considérablement les fichiers. Un seul mélange était utilisé.

Le calcul de cœur se faisait dans les dossiers Full_Core de chaque grand dossier. Les fichiers EDI de sections efficaces microscopiques correspondant

à différents calculs (réflecteur, bloc simple, super-cellules) étaient copiés et regroupés sous la dénomination XSOUTi.edi où i est un nombre de 1 à 14 dans le dossier xs. Les fichiers XSOUTi.N contenaient les densités à utiliser pour le calcul de cœur, associées aux sections efficaces correspondantes.

Un programme DRAG2INS3.f90 (écrit par Javier Ortensi) avait été compilé précédemment (executable a.out). Il allait chercher dans xs les sections efficaces, les densités, et calculait les sections macroscopiques qui allaient être utilisées dans INSTANT. Ce code préparait l'ensemble du fichier d'entrée INSTANT, le reste étant invariant. Il produisait un fichier « inp.xml », qui était ensuite envoyé à INSTANT.

Pour tourner, DRAG2INS3.f90 avait besoin d'un fichier d'entrée supplémentaire : DRAG2INSm.inp (« m » comme modified, il y a eu de nombreuses versions de ce code...). Ce fichier indique entre autre le nombre de groupes, et comment les sections doivent être calculées : en utilisant un coefficient de diffusion (STRD), ou bien directement avec les sections totales données par DRAGON, avec une correction transport ou pas... Toutes les méthodes ne sont pas forcément valides, mais celle sans la diffusion marche. Le script WriteDrag2InsInp.py permettait de générer automatiquement le fichier, en lui donnant en entrée le nombre de groupes, la méthode et l'ordre pour les collisions (isotropique versus linéaire anisotropique). Je recommande « 1 1 », c'est-à-dire méthode P_N classique sans essai d'imitation de diffusion (ceci est probablement faux) et avec des collisions linéaire anisotrope. L'anisotropie ajoute du temps de calcul mais c'est plus précis.

Enfin, on trouvera dans les dossiers Full_core des dossiers avec les résultats pour chaque type de calcul et de structure de groupe. Dans les dossiers parents, des scripts « copy_P3.sh » allaient chercher au bon endroit les fichiers de sortie EDI avec un nombre de groupe donné et les copiaient dans le dossier xs. Les scripts Core_calc_P3.sh effectuaient l'ensemble de la chaîne de calcul automatiquement : copie des fichiers EDI dans xs, appel à WriteDrag2InsInp.py puis à DRAG2INS3.f via le script « transfer.sh », production

du fichier d'entrée INSTANT, déplacement dans un dossier réservé au calcul de cœur, calcul de cœur avec l'exécutable INSTANT, copie des résultats au bon endroit et nettoyage.

5 Quelques conseils en vrac

Voilà quelques remarques sur DRAGON et autre :

- La mise en place des schémas de calcul SPH doit se faire à la fin, une fois que le reste marche. Pas la peine d'introduire des difficultés supplémentaires dans le débogage.
- Le copier-coller est un peu dangereux dès qu'il y a beaucoup de lignes. C'est ce qui m'a conduit à écrire des scripts en Python pour générer les fichiers d'entrée de DRAGON pour les super-cellules. Cela s'avère être très pratique pour éviter les fautes.
- La mise en place de scripts pour automatiser les calculs nécessite de bien réfléchir à l'organisation de son ordinateur : c'est typiquement quelque chose qui aurait pu être amélioré chez moi. Par exemple, au lieu de n'avoir qu'un seul endroit où stocker l'exécutable d'INSTANT et de faire des liens symboliques, je le copiaais à la main dans chaque dossier. Au fur et à mesure de l'augmentation des cas à étudier durant la maîtrise, cela est devenu très embêtant, avec le risque d'oublier un dossier par exemple et donc de ne pas faire le calcul avec la bonne version de l'exécutable... Dans le même ordre d'idées, les noms devraient être pensés pour faire court tout en restant compréhensible...
- Les sauvegardes régulières de ce qui marche (ou pas) sont impératives, de même que les sauvegarde de vos fichiers $\text{\LaTeX} 2_{\epsilon}$ lorsque vous rédigerez le mémoire. Il n'y a rien de pire que de perdre tout son travail à cause d'une panne informatique ou bien d'un vol d'ordinateur. On peut utiliser les messageries type gmail pour garder une version en ligne par exemple.

- Il est possible de se connecter à distance sur les ordinateurs de l'école si vous avez Linux ou tout autre système avec un terminal : `ssh -X login@gardien.recherche.polymtl.ca`. Si vous êtes sur Windows, Putty vous permettra d'avoir un terminal et donc de dialoguer avec les machines de l'école. Voir Google pour la configuration.

- Pour transférer peu de fichiers :

```
scp fichier-à-transférer login@gardien.polymtl.ca:~/dossier-dans-lequel-ça-doit-atteirir.
```

Cela marche à l'envers pour passer des fichiers sur votre ordinateur :

```
scp login@gardien.polymtl.ca:~/fichier-à-transférer ~/dossier-dans-lequel-ça-doit-atteirir-chez-vous.
```

Voir Google pour plus de détails.

- Pour transférer plus de deux fichiers : utiliser FileZilla (libre). Ce n'est pas très difficile à configurer et cela marche sur tous les systèmes d'exploitation. Attention à l'encodage des fichiers que vous transférez si vous passez de Windows à Unix, il vaut mieux faire le changement durant le transfert. Sinon, il y a des soucis à l'affichage.
- Le script `rdragon4` fournit lorsque vous téléchargez Dragon mérite d'être étudié si vous allez en stage ailleurs et qu'il faut réinstaller Dragon là-bas. En gros, il faut bien renseigner les adresses pour aller chercher les executables, les librairies, et le dossier `tmp` où doivent être créés les dossiers temporaires « `rundir1`, `rundir2`... » dans lesquels on fait tourner Dragon. Il existe des machines où l'accès à `/tmp` est interdit, donc le mieux reste d'en créer un sur votre répertoire local : `~/tmp`.
- Noter que à la base, pour faire tourner Dragon, il suffit de se mettre dans un dossier avec le fichier d'entrée, l'exécutable (nommé Dragon par exemple), la librairie, et de lancer au terminal dans le dossier : `./Dragon <fichier_entrée librairie >> nom_fichier_sortie.result`. C'est ce qui est fait dans le script `rdragon4`, en plus compliqué puisqu'il crée un dossier temporaire où on peut faire le calcul sans mettre le bazar dans ses fichiers, puis va rechercher les librairies et executables sans

les copier (liens symboliques), et enfin se débrouille pour renvoyer les résultats intéressants.

- Pour la rédaction en $\text{\LaTeX} 2_{\epsilon}$, il existe une très bonne documentation traduite en français sur le site du CTAN : *Une pas si courte (?) introduction à $\text{\LaTeX} 2_{\epsilon}$* . Wikipedia fournit également de l'aide pour s'y retrouver. Faire surtout attention à l'encodage de ses fichiers : choisir une fois pour toute et s'y tenir. L'école utilise généralement l'encodage Windows pour les fichiers modèles en $\text{\LaTeX} 2_{\epsilon}$, mais si vous préférez utiliser UTF-8, il suffit de changer les accents dans le fichier de style de l'école. Cela a l'avantage de pouvoir passer ses fichiers sur Linux sans soucis.
- À titre purement indicatif, j'utilisais Texmaker comme éditeur de texte et MiKTeX sur Windows. Sur Ubuntu, TeXLive est très facile à installer via le gestionnaire de paquets.
- Le site du zéro (<http://www.siteduzero.com>) possède un tutoriel d'initiation à Linux qui est très pratique. Pour ceux qui comme moi n'avaient aucune idée des commandes du terminal, cela permet de devenir opérationnel sur les ordinateurs de l'école et même d'installer Linux sur le sien. Les tutoriels de bash sont plus rares, j'ai essentiellement repris ce qui est fait dans le script `rdragon4` pour faire les miens.

6 Conclusion

Bon courage à tous pour vos calculs avec Dragon et vos études à Poly !