

MODE D'EMPLOI DU CODE FLUKA ET DE L'INTERFACE FLAIR

CE DOCUMENT A POUR BUT DE PERMETTRE A UN UTILISATEUR DEBUTANT UNE PREMIERE PRISE EN MAIN DE FLUKA ET DE L'INTERFACE FLAIR. IL A ETE REDIGE DANS LE CADRE DE L'UTILISATION DU CODE FLUKA PAR LE SERVICE D'ASTROPHYSIQUE DU CEA. DES INFORMATIONS COMPLEMENTAIRES A CE MANUEL PEUVENT ETRE TROUVEES DANS « FLUKA : A MULTI-PARTICLE TRANSPORT CODE », REFERENCE PRINCIPALE DE CE DOCUMENT. NOUS RAPPELONS QU'IL NE S'AGIT PAS LA D'UN MANUEL OFFICIEL, MAIS D'UN DOCUMENT DE TRAVAIL, A MANIPULER AVEC TOUTES LES PRECAUTIONS NECESSAIRES. DE MANIERE GENERALE IL EST FORTEMENT CONSEILLE A L'UTILISATEUR DE SE REFERER A LA DOCUMENTATION OFFICIELLE EN CAS DE DOUTE.

i r f u

cea

s a c l a y

Natacha Combier

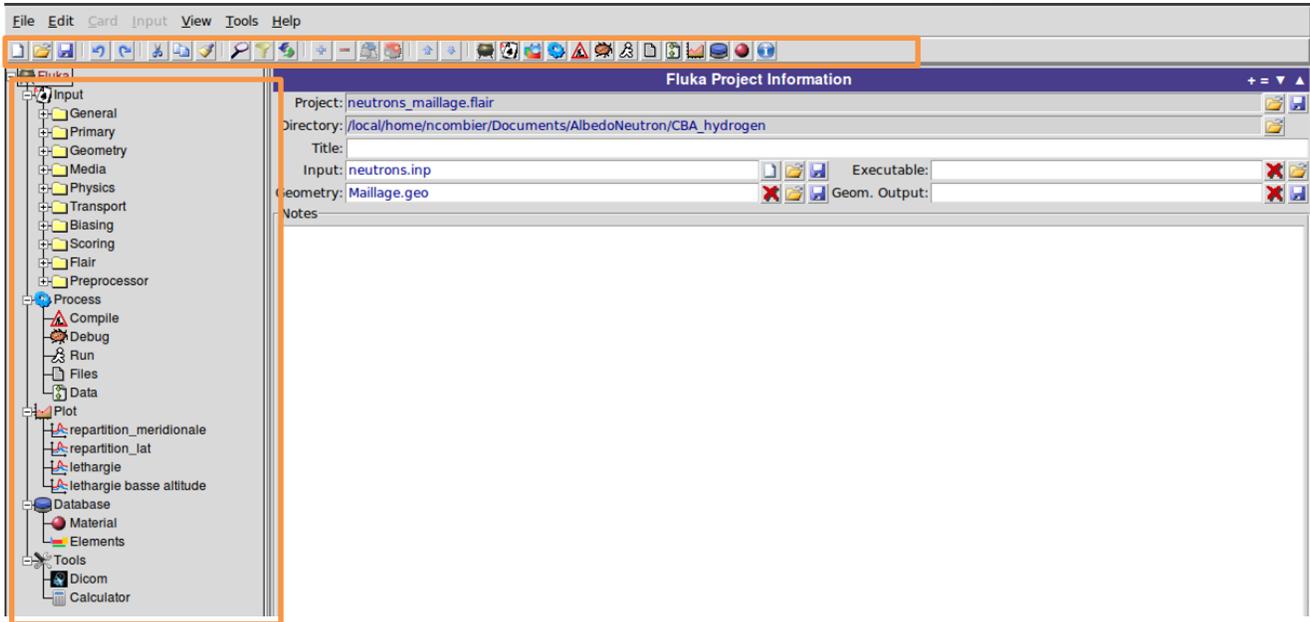
combier.natacha@gmail.com

Juin 2013

PRESENTATION DE L'INTERFACE FLAIR.....	3
Barre D'outils.....	3
Onglet Fluka.....	4
Onglet input.....	4
Onglet data.....	4
Onglet file.....	4
Onglet Compile.....	5
Debugging.....	6
editeur de geometrie.....	6
Onglet run.....	7
Onglet File.....	8
Onglet Data.....	8
Onglet Plot.....	8
INFORMATIONS GENERALES SUR L'UTILISATION DE FLUKA.....	12
Premiere simulation.....	12
Les cartes de scoring.....	15
les differentes techniques de biaisage.....	16
Les modeles physiques a haute energie.....	17
Les cartes PHYSICS et cartes associées.....	17
les cartes transport.....	19
QUELQUES ROUTINES UTILES.....	20
Utilisation de la route mgdraw.....	20
Utilisation de la routine spec_sour.....	22
EN CONCLUSION.....	24
REMERCIEMENTS.....	24

PRESENTATION DE L'INTERFACE FLAIR

Nous présentons ici de manière très concise l'utilisation de l'interface Flair. Le but de ce chapitre n'est pas de lister toutes les fonctionnalités de Flair mais de se familiariser avec celles nécessaires à une première approche du code. Observons tout d'abord la fenêtre telle qu'elle apparaît à l'ouverture d'un projet :



Ce chapitre s'organise en suivant cette fenêtre de haut en bas. Nous commencerons donc par présenter la barre d'outils puis nous suivrons l'ordre des onglets de l'arborescence située à gauche. Nous terminerons par une présentation de l'éditeur de géométrie. Pour plus d'informations sur chaque partie, il sera parfois nécessaire de se reporter au chapitre 2 qui décrit plus en détail le fonctionnement général du code Fluka.

BARRE D'OUTILS

La barre d'outils apparaît sous la forme suivante :



Les fonctionnalités les plus importantes sont :



Ajouter ou supprimer une carte.



Cloner la carte sélectionnée (la nouvelle carte apparaît en dessous de la première).



Commenter ou dé-commenter la carte.



Déplacer une carte sélectionnée vers le haut ou le bas. Il est également possible de le faire à la main en attrapant la carte (en cliquant sur son nom) et en la déplaçant manuellement (laisser appuyer).

ONGLET FLUKA

C'est ici que l'on enregistre le projet (.flair), l'input (.inp) et la géométrie (.geo, lorsqu'elle est enregistrée à part).

Avant de lancer une simulation, mieux vaut enregistrer tous ces champs l'un après l'autre, pour éviter par exemple d'enregistrer l'input lorsque la modification a été faite dans la géométrie.

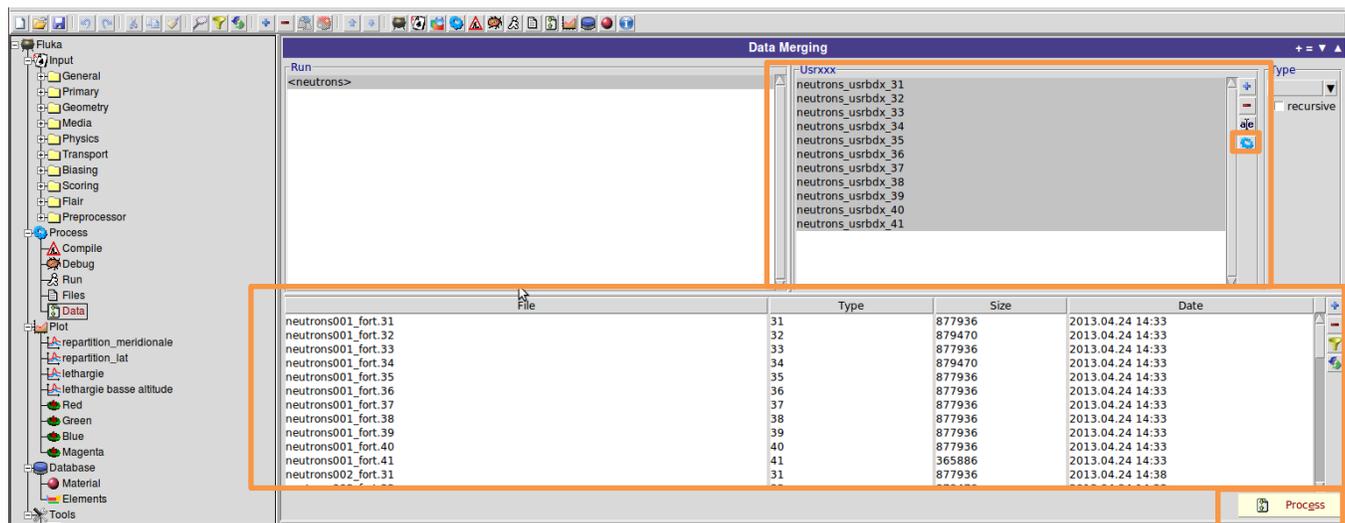
ONGLET INPUT

On trouve ici toutes les informations relatives à l'input c'est à dire aux cartes définissant la simulation (géométrie, beam, scoring, bais etc.)¹

ONGLET DATA

Apparaissent ici tous les fichiers relatifs au scoring (.fort), c'est-à-dire les fichiers créés par l'ajout d'une carte dans l'onglet « scoring » (par exemple USRBIN ou USRBDX). Dans la fenêtre en haut à droite sont listés tous les détecteurs de la simulation (éventuellement relancer le listing pour faire apparaître de nouveaux détecteurs en cliquant sur l'anneau bleu à droite).

Dans la fenêtre centrale on retrouve tous les fichiers .fort. On les process à l'aide du bouton du même nom. Sont alors créés les « _sum.lis » et « _tab.lis » (fichiers permettant le processing des résultats et le tracé des courbes) qui sont listés dans l'onglet « files ».



ONGLET FILE

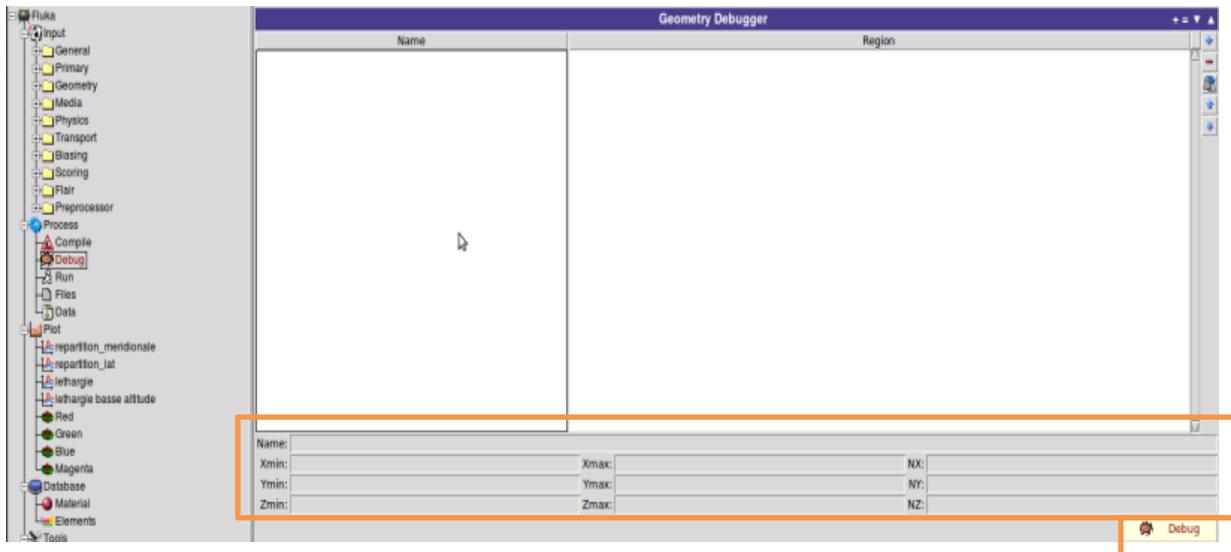
Apparaissent ici tous les fichiers créés². Certains sont propre à chaque cycle (cycle/001 etc.), d'autre sont issus des plots (cycles/plot) etc.

¹ Voir chap. Fluka/première simulation et dans chap. Fluka/les différentes parties relative à l'utilisation de chaque carte.

² Voir chap. Fluka/première simulation/les fichiers créés

DEBUGGING

Cet onglet permet un débogage rapide de la géométrie. A l'aide des coordonnées x, y et z on rentre la zone que l'on souhaite debugger. On choisit ensuite les bins correspondants et on debugge.

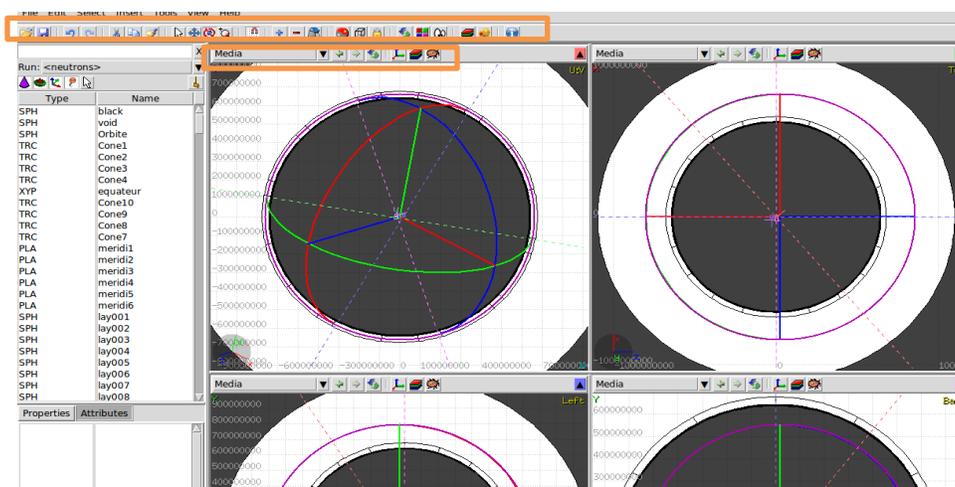


Attention: si les bins sont trop grands (par exemple des mailles de 1cm pour une géométrie de 5cm) les erreurs ne seront pas détectées. S'ils sont trop petits (par exemple des mailles de 1 μ m pour une géométrie de 5cm), le temps de calcul va être beaucoup trop important. Cette technique de débogage est donc limitée et l'on préfère généralement utiliser l'interface Geoviewer plutôt que cet onglet (qui repose sur le même principe).

EDITEUR DE GEOMETRIE

Cet éditeur permet de visualiser la géométrie mise au point et la debugger par la même occasion. L'affichage est divisé en 4 cadrans représentant 4 vues distinctes (les axes apparaissent en bas à gauche de chaque fenêtre).

Elle permet notamment un débogage rapide de la géométrie : les surfaces ou régions fausses (qui se superposent, qui n'ont pas de matériau associé etc.) apparaissent en rouge. Cependant si l'on est en vue trop générale, les petites erreurs n'apparaissent pas (i.e. si la maille est trop large par rapport à la géométrie). Le mieux est alors de zoomer (molette de la souris) sur les points délicats jusqu'à voir apparaître détails et erreurs éventuelles.



- La barre d'outils principale permet d'accéder aux actions suivantes:



Permet de faire tourner l'objet sur lui-même pour accéder à une vue plus complexe.



Permet de déplacer le point de vue de l'observation en restant dans un même plan.

- Chaque fenêtre a ensuite sa propre barre d'outils:



Permet d'aider au débogage une fois l'erreur repérée. On se place dans une vue où l'on voit une zone apparaître en rouge puis on clique sur l'icône. Apparaît un tableau indiquant les points posant problèmes et la raison précise de l'erreur (par exemple 1 point appartient à 2 régions à la fois).



Permet de rentrer les coordonnées des axes sous lesquelles on souhaite voir apparaître notre géométrie.



Permet de choisir si l'on souhaite afficher les couches, les régions etc.

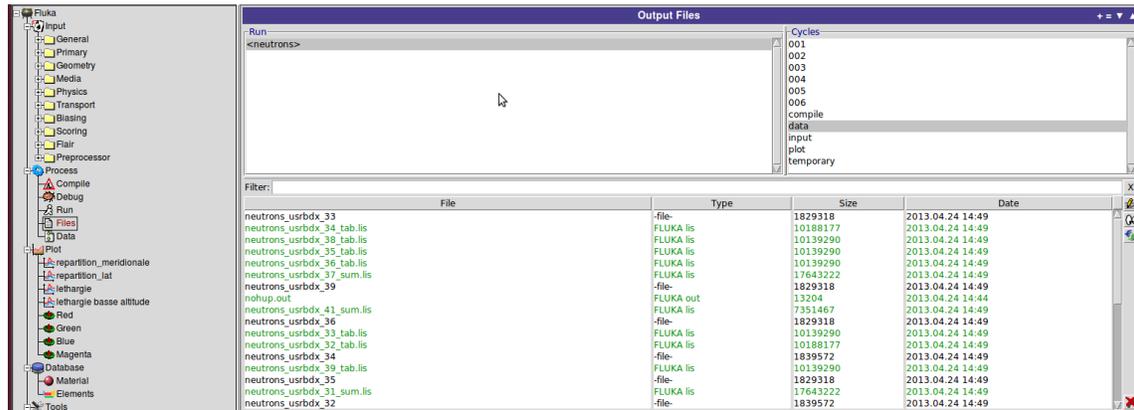
ONGLET RUN

A chaque run, il est conseillé de respecter les étapes suivantes :

1. Vérifier la géométrie avec l'éditeur de géométrie (les erreurs apparaissent striées en rouge, lorsque l'on zoom dessus)
2. Tester la simulation avec un petit nombre de particules afin de la valider (à l'aide des fichiers .log .out et .err) et de prévoir le temps de calcul nécessaire à un plus grand nombre de particules (CPU indiquée dans le .out, onglet « summary »).
3. Supprimer tous les fichiers issus des runs de test afin d'éviter les confusions.
4. Choisir un nombre de particules et un nombre de cycles adaptés au problème.
5. Lancer la simulation finale (onglet « run », icône « run »).

ONGLET FILE

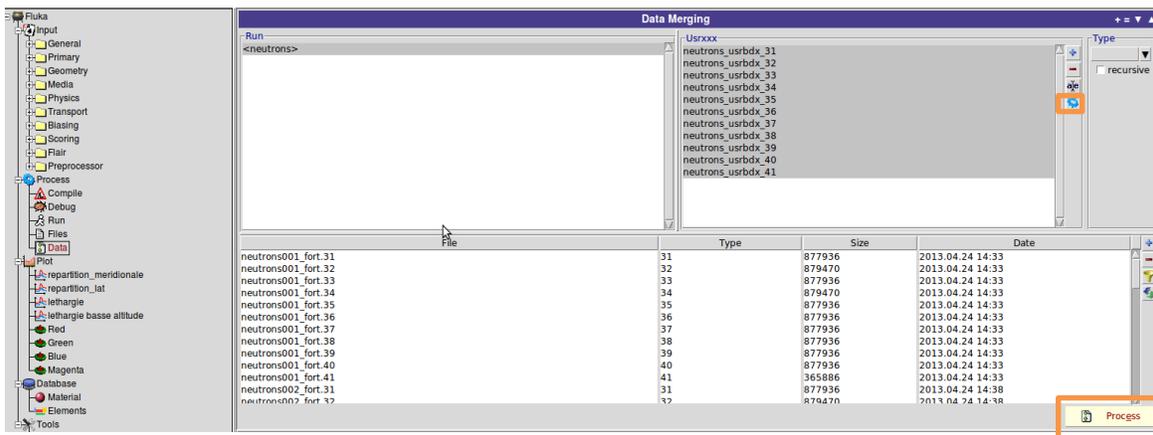
Sont répertoriés ici tous les fichiers créés. Il est conseillé de les effacer systématiquement à chaque nouvelle simulation⁵.



ONGLET DATA

Apparaissent ici tous les fichiers « .fort » créés à chaque cycle de chaque run. Ces fichiers bruts doivent être traités de la manière suivante:

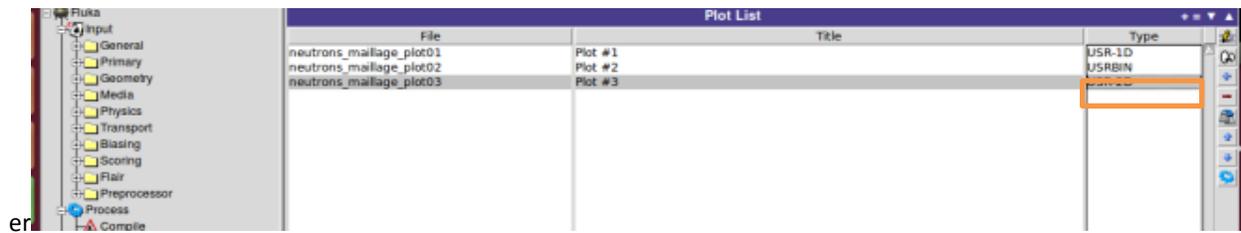
- dans l'onglet Data, dans la fenêtre Usrxxx on clique sur l'anneau bleu clair pour qu'apparaissent automatiquement toutes les unités que l'on souhaite traiter (par exemple si tous nos détecteurs utilisent l'unité 30 un seul fichier apparaîtra).
- on sélectionne tout ce qui apparaît dans la fenêtre puis on clique sur l'icône « process ».



ONGLET PLOT

Cet onglet permet d'afficher des courbes issues de nos détecteurs. Dans la fenêtre ci-dessous par exemple, nous avons ajouté 3 plots, chacun correspondant à un type de détecteurs différents⁵. Pour ajouter un plot, on utilise l'icône +. On en choisit ensuite le type en faisant un clic droit sur « geometry » qui est le type par défaut.

⁵ Voir chap. Fluka/première simulation/les fichiers créés



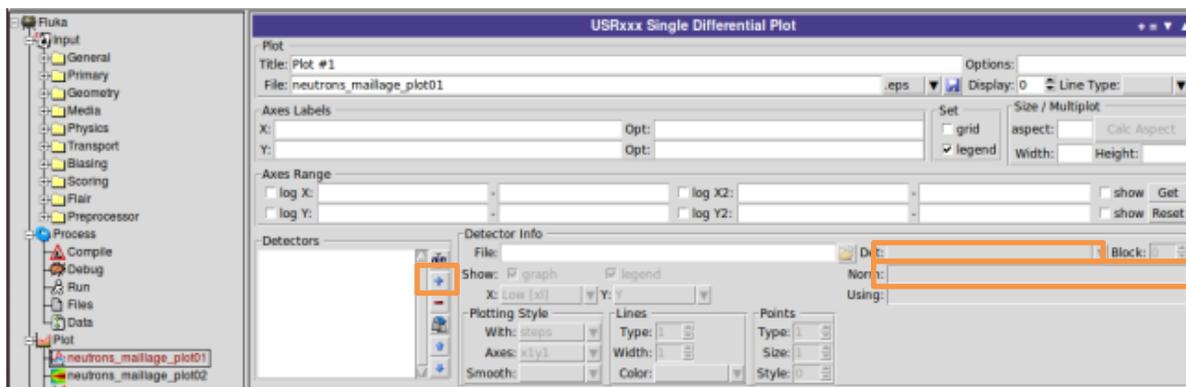
Il arrive que soient présents par défaut 4 plots (magenta, blue etc.). Ils sont associés aux 4 vues de Geoviewer et ne présentent généralement pas d'intérêt.

LE PLOT USR-1D

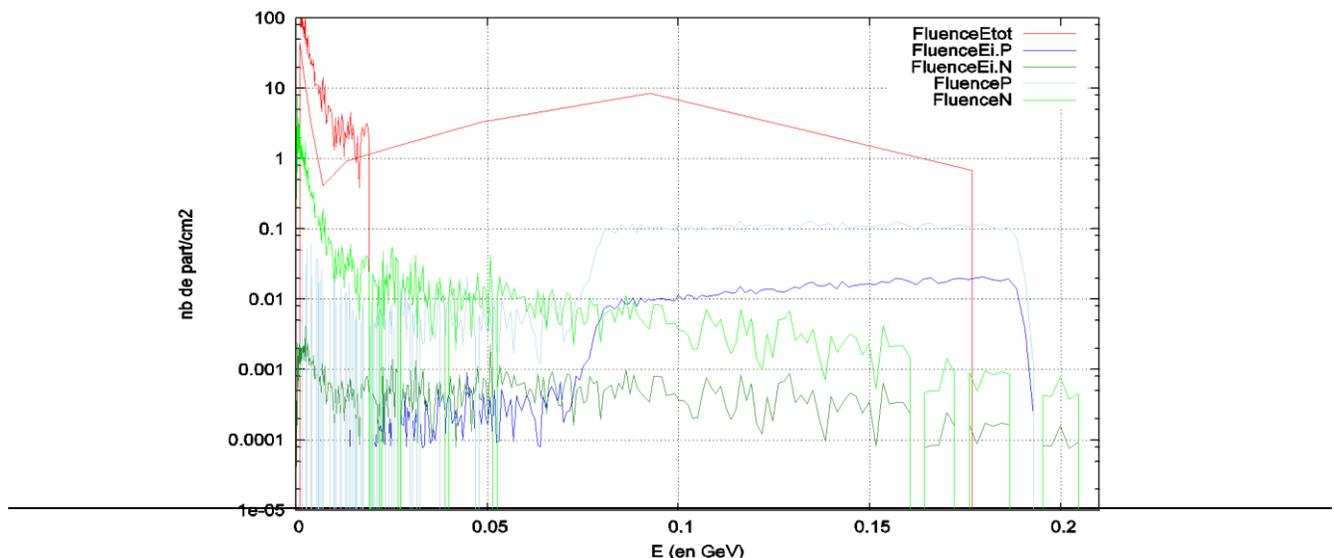
Il s'utilise par exemple dans le cas de détecteurs type USRBDX. On commence par ajouter le ou les détecteurs que l'on souhaite tracer :

- Dans detector, on utilise l'icone + qui nous permet de créer un nouveau détecteur associé à un fichier « _tab.lis » (il en existe un par unit).

Ensuite on précise le détecteur de l'unité que l'on souhaite tracer (« Det »). On peut éventuellement ajouter un facteur de normalisation («norm»). Puis on clique sur l'icone « Plot ».



Attention : dans le cas où une courbe aurait déjà été tracée lors d'une précédente simulation, il faut utiliser l'icône « replot ». Il arrive parfois que Flair bug et que, même comme cela, la courbe ne soit pas remise à jour. Dans ce cas une solution consiste à ajouter puis à supprimer un détecteur quelconque, ce qui forcera la mise à jour. On obtient des courbes de la forme :

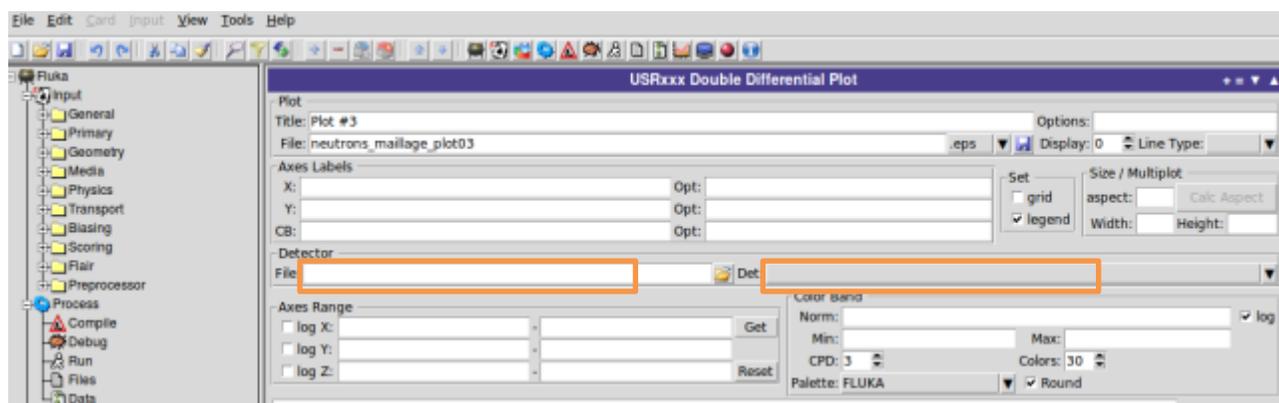


Les unités sont automatiquement en $\text{part}/(\text{cm}^2.\text{GeV}.\text{sr})$ à condition d'avoir correctement indiqué l'air du détecteur dans la carte USRBDX⁷

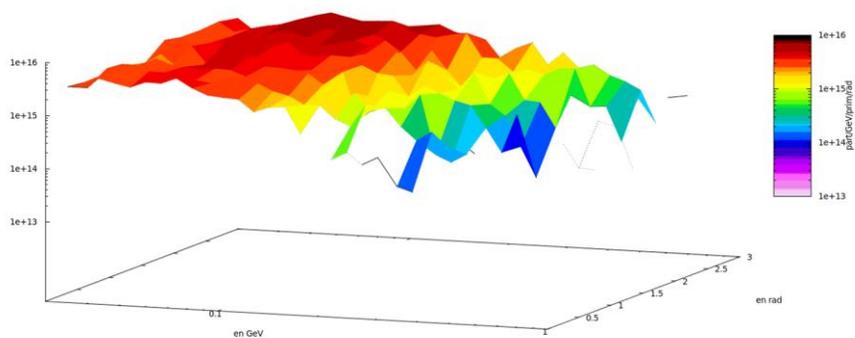
Il est possible de séparer les mesures en fonction de l'angle d'incidence des particules. Par exemple dans le cas où l'on a effectué un scoring de 100 bins d'énergie et de 4 bins d'angle, on peut tracer 4 courbes distinctes. Pour cela, on crée 4 fois le détecteur correspondant au scoring (« detector ») puis on utilise les champs « Block » situé à droite. Au premier on attribue la valeur 1, au second 2 etc.

LE PLOT USR-2D

Ce type de plot permet de tracer des courbes en 3D (Energie, angle et flux). On sélectionne un fichier .lis (champ « file ») et, à l'intérieur, le détecteur que l'on souhaite utiliser :



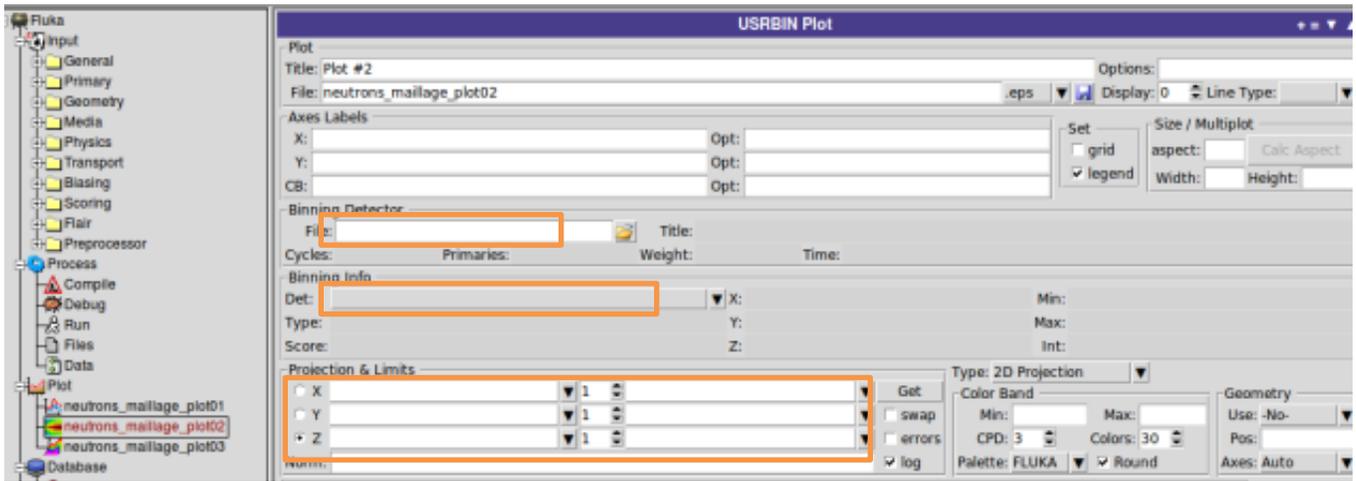
On obtient alors un résultat de la forme :



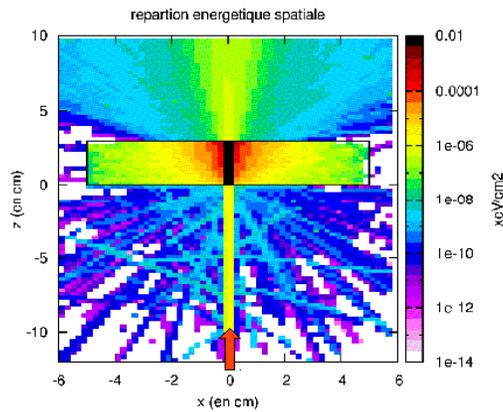
LE PLOT USRBIN

⁷ Voir chap. Fluka/USRBDX

Ce plot est associé au détecteur de type USRBIN. Comme précédemment on sélectionne le fichier tab.lis puis le détecteur choisit. Ensuite si l'on souhaite par exemple une vue de coupe en XY, on place le point à gauche (« projection & limits ») devant Z. On choisit X et Y max et min pour déterminer les limites de la vue que l'on va obtenir. Enfin on choisit l'épaisseur Z sur laquelle l'on va intégrer la valeur scorée.



On obtient des courbes de la forme :



INFORMATIONS GENERALES SUR L'UTILISATION DE FLUKA

Nous allons présenter ici quelques informations nécessaires une première prise en main du code Fluka. Encore une fois, il est conseillé de se reporter à la documentation officielle en cas de doute.

PREMIERE SIMULATION

Une simulation simple contient au minimum la description du faisceau incident et de la géométrie du modèle. Une fois ces éléments placés il est possible de complexifier le modèle en ajoutant du scoring, en biaisant sa géométrie etc. Tout ceci sera présenté dans le chapitre suivant. Nous étudions tout d'abord les différents onglets de l'input:

LES ONGLETS INPUT NECESSAIRES A UNE SIMULATION SIMPLE

Primary:

Regroupe toutes les informations concernant le faisceau incident. C'est à dire au minimum une carte BEAM (décrivant le faisceau) et une carte START (permettant de lancer la simulation avec un nombre de particules donné).

Geometry :

Contient les informations relatives à la géométrie. Il commence par une carte GEOBEGIN et se termine par GEOEND. On choisit un facteur de précision adapté à notre problème.

Media:

Assigne les matières aux régions définies dans la géométrie.

Physique:

Permet de choisir les lois physiques régissant notre simulation.

Transport:

Gère le transport des particules, par exemple au moyen de thresholds.

Biasing:

Voir chap. sur le sujet.

Scoring:

Voir chap. sur le sujet.

LES DIFFERENTS FICHIERS CREES:

Nous présentons ici les fichiers créés lors de chaque simulation ainsi que la manière dont ils doivent être exploités.

Les fichiers permettant le run:

Les 3 fichiers présentés ci-dessous sont plus ou moins interdépendants les uns des autres. Il est préférable lorsque l'on souhaite sauvegarder son travail de sauvegarder les 3 l'un après l'autre afin d'être sûr de ne pas perdre d'information.

- **.FLAIR**: Ce fichier contient le projet complet c'est à dire les liens qui permettent d'ouvrir le .in et le .geo associés ainsi que toutes les informations relative au post processing (comme par exemple les tracés des courbes).
- **.INP**: Ce fichier regroupe toutes les informations nécessaires à la simulation (hors géométrie lorsque celle-ci est enregistrée à part), c'est à dire le beam, le scoring, les biais etc.
- **.GEO**: Ce fichier contient les informations concernant les zones, les matières etc. Dans le cas où l'on crée un nouveau projet, les informations concernant la géométrie implémentée seront enregistrées dans le fichier .inp. Dans le cas de géométries complexes, on peut les sauvegarder à part (onglet « Fluka » puis champ « Geometry ») et alors le fichier .inp ne contiendra plus qu'un lien vers le nouveau fichier créé.

Les fichiers créés après le run:

Après chaque run, il faut effectuer une série de manipulations pour éviter de se retrouver surchargé de fichiers. Nous indiquons ici tous les fichiers créés, et nous précisons ceux à compiler et ceux à effacer.

1. Juste après le run, les fichiers de sorties sont les **_FORT.XX** (avec xx numéro de l'unité de chaque carte de scoring). Par exemple si vous avez 2 détecteurs USRBDX enregistré sur l'unité 21 et un USRBIN enregistré sur l'unité 22 et que votre simulation comporte 5 cycles, vous aurez créé 10 fichiers _fort. xx:

SimuName001_fort.21 SimuName002_fort.21 ... SimuName005_fort.21

SimuName001_fort.22 SimuName002_fort.22 ... SimuName005_fort.22

- Ces fichiers bruts doivent être compilés de la manière suivante:

-dans l'onglet Data, dans la fenêtre Usrxxx on clique sur l'anneau bleu pour rentrer automatiquement toutes les unités que l'on souhaite compiler.

-on sélectionne tout ce qui apparaît dans la fenêtre puis on clique sur « Process ».

- On obtient ainsi 3 nouveaux types de fichiers pour chaque unité compilée:

-les **_SUM.LIS** qui contiennent les informations "lisibles" de chaque détecteur de l'unité (flux pour chaque bin)

-les **_TAB.LIS** qui contiennent l'information brut et servent au tracé des courbes

-les `.BNX` qui ne nous servent pas et peuvent être effacés

Une fois compilés, les `_FORT.XX` peuvent être effacés.

2. Les autres fichiers créés sont ceux qui ne contiennent pas l'information obtenue par scoring mais un résumé général du run :

- Le `.LOG` qui contient la licence Fluka. En cas de bug, il arrive que des informations apparaissent au début ou à la fin de ce fichier. Dans notre simulation le fichier se termine par "l'erreur" suivante:

```
<Blckhl> : 0.0000000  
<Z_ion> : 1.1650909  
<Weight> : 0.99964058  
<E_k_sam> : 6.3414440  
<E_k> : 5.7402616  
<P_co> : 3.0337045
```

- Le `.OUT` qui contient les informations générales quant au déroulement de la simulation. Il contient entre autre:

-un rappel de l'input

-un rappel du scoring effectuée

-un run summary (avec de nombreuses informations comme par exemple la CPU)

S'il est inutile de garder le `.OUT` de chaque cycle, il est toujours intéressant d'en conserver au moins un qui va contenir toutes les informations relatives à notre simulation.

- Le `.ERR` qui regroupe les erreurs rencontrées (idéalement vide). Dans le cas de notre simulation ce fichier se terminera par un listing de particules du type:

```
100000      0      1      4.3125895E-03      1.0000000E+30      125389  
NEXT SEEDS:1801285D  0  0  0  0  0  0  181CD  3039  0  0
```

Et dans ce cas, les fichiers `.ERR` peuvent être supprimés.

- Les fichiers commençant par `RAN` contiennent les informations sur le nombre aléatoire utilisé et sont utiles si l'on veut reproduire exactement la simulation (ce qui n'est jamais le cas sauf en cas de bug). Si la simulation a marché, on peut les effacer.

EN CAS DE BUG:

En cas de bug, certains fichiers vont apparaître qui doivent être supprimés une fois le problème résolu:

- les fichiers `FLUKA_XXX`: qui contiennent une copie de l'input dont se sert Fluka lors de la simulation (ce qui permet de modifier les cartes sans interférer avec le run) et qui sont automatiquement effacés en cas de succès
- les `.EXXX` et `O.XXX`
- les `CORE.`
- les `NOHUP.OUT`
- les dernier `.ERR`
- les sources routines

EN CONCLUSION:

Après un run réussi, seuls doivent rester les fichiers:

- `SUM.LIS`
- `_TAB.LIS`
- `.OUT` (un seul suffit)

LES CARTES DE SCORING

Nous allons présenter ici les cartes permettant différents types de scoring. Certaines apparaissent dans notre simulation, d'autres non mais pourraient être rajoutées. Toutes sont situées dans l'onglet scoring de l'arborescence de gauche.

En général, les scoring de même espèce peuvent être enregistrés sur une même «unit sans que cela ne pose problème (dans la carte de scoring, le champ « unit » peut être le même).

USRBDX:

C'est le scoring le plus courant dans notre simulation, il permet d'intégrer doublement en énergie et en angle. On obtient ainsi une mesure de la fluence (« phi ») ou du courant (« I ») dans un sens (« 1 ») ou dans les 2 sens (« 2 »), en échelle logarithmique ou linéaire. Ce type de scoring est réalisé lors du passage d'une région à une autre.

SCORE:

Cette carte permet un scoring un peu particulier car, contrairement aux autres, elle ne crée pas de fichier `_fort`. Elle va rajouter de nouvelles informations dans le fichier `.out` (au niveau de l'onglet "event by regions") et permet de compter le nombre d'évènements provoqués par les particules de notre choix (jusqu'à 4 différentes).

Après avoir fait tourner la simulation, apparaissent dans le fichier .out 4 colonnes distinctes correspondant aux 4 particules choisies. Nous détaillons ci-dessous quelques types de particules spécifiques:

- -ENERGY énergie déposée par particule primaire uniquement
- -BEAMPART collision inélastique provoquées par les primaires
- -ALLPART collision inélastique provoquée par toutes les particules
- -UNBPART énergie déposée mais sans tenir compte du biais et de la normalisation qu'il est nécessaire de faire ensuite (une particule peut être transformée en 10 particules à un changement de région, il faut donc diviser l'énergie déposée par 10 pour avoir une valeur cohérente)

RESNUCLEI:

Cette carte compte les noyaux résiduels produits par les interactions inélastiques dans une région donnée. Lors du plot on obtient A en abscisse et Z en ordonnée avec en couleur la densité d'ions produits. Elle permet donc de se faire une idée du type d'ions créés.

USRBIN:

Cette carte permet un scoring en 3D du dépôt d'énergie, de la dose du flux etc. Le résultat est en particule/cm³. On définit le volume dans lequel on veut réaliser ce scoring à l'aide des coordonnées x, y et z.

SI1MEVNE :

Cette carte permet d'obtenir l'équivalent neutron de 1MeV de silicium et donc d'avoir un élément de comparaison pour l'effet des particules dans les composants.

USRYELD (RENDEMENT) :

Cette carte permet par exemple de différencier les particules issues d'interactions inélastiques des autres. Cependant cette technique ne donne pas forcément de très bons résultats car elle ne fait qu'envoyer le faisceau dans la matière et donner le spectre sortant.

LES DIFFERENTES TECHNIQUES DE BIAISAGE

PRINCIPE:

Il existe plusieurs techniques de biaisage différentes. Nous détaillons ici les deux principales:

- le biaisage d'importance: on attribue une importance différente à chaque région. Toute particule passant d'une région d'importance A à une autre d'importance B sera transformée en A/B particule de même nature.

- le biaisage de collision: on attribue un facteur RR (« russion roulette » si >1 , « surface splitting » si <1) à une région donnée. Chaque particule produite lors de collisions dans cette région est transformée en RR particules de même nature.

LES DIFFERENTES CARTES NECESSAIRES:

BIAISING: la plus courante, elle permet d'appliquer soit le biaisage d'importance, soit le biaisage de collision, dans une région choisie et à un type de particule choisi.

DISCARD: permet de supprimer automatiquement les neutrinos, les électrons et les muons.

DELTARAY: permet de supprimer les électrons produits par ionisation (l'énergie d'ionisation est entièrement déposée sur le chemin de la particule et aucun électron n'est créé).

LES MODELES PHYSIQUES A HAUTE ENERGIE

Flukahp est l'exécutable par défaut du code Fluka et marche très bien pour la plupart des applications. Dans certains cas où l'on manipule des ions lourds, il peut s'avérer nécessaire d'utiliser l'exécutable **flukadpm3** qui contient les 2 modèles que nous détaillons ci-dessous.

MODELE DE SUPERPOSTION RQMD:

Avec l'exécutable flukadpm3, ce modèle est utilisé automatiquement pour des énergies comprises entre 5GeV et 100MeV. Les ions primaires ne sont plus traités comme des noyaux mais comme des ensembles de nucléons indépendants. En dessous de 100MeV c'est le modèle BME qui est utilisé.

MODELE DPMJET

Au-dessus d'une certaine énergie (A-A : 5GeV/nucléon, hadrons-A : 20TeV) ce modèle simule les collisions noyau/noyau (qui sont désormais vu en tant que particules et plus en tant que groupe de nucléons). En dessous, c'est le modèle de superposition qui est utilisé. Dans ce cas, le générateur d'événements externes est DPMJET et l'exécutable est flukadpm3. Pour le sélectionner depuis Flair, on sélectionne « flukadpm3 » comme Link dans l'onglet « compile », puis on « build ».

LES CARTES PHYSICS ET CARTES ASSOCIEES

On présente dans ce paragraphe certaines cartes décrivant la physique utilisée dans notre modèle.

EVAPORAT:

Dans Fluka, la modélisation des phénomènes d'évaporation s'arrête normalement aux gammas. Cette carte permet d'activer cette modélisation dans le cas de particules plus lourdes. Elle est donc importante pour simuler les problèmes d'activation (transmutation des atomes (généralement en éléments radioactifs) par irradiation de particules nucléaires).

COALESCE:

Cette carte active une option qui permet la fabrication de Deutérium à très haute énergie (collision 2p et 2n).

PEATHRES:

Sans cette carte c'est le modèle Peanut qui est utilisé pour toutes les particules jusqu' à 5GeV. Lorsque l'on utilise cette carte le seuil est élevé à 30 TeV (particulièrement nécessaire lors des interactions A-A). Au-dessus de ce seuil, le modèle utilisé est moins bon mais plus rapide (plus ancien).

IONSPLIT:

Elle permet d'activer le modèle de superposition. Par exemple comme il n'existe pas de modèle pour le deutérium en dessous de 150 MeV le modèle de superposition permet de le traiter. Lorsque Dpmjet est actif on applique le modèle superposition seulement aux ions d'énergie comprise entre 100MeV et 150MeV. Dans le cas contraire, il faut étendre le modèle pour permettre de traiter tous les ions à haute énergie.

DECAYS:

Cette carte permet de choisir le mode de simulation du phénomène de désintégration que l'on applique ensuite aux particules d'indices compris entre les 2 bornes données (dans notre cas, entre le proton et la dernière des particules). Elle est principalement utilisée pour les pions et les kaons, afin de leur donner une polarisation.

DETARAY:

Cette carte permet de n'activer la modélisation du rayonnement delta (électron issu d'une ionisation) qu'au-dessus d'un seuil d'énergie donné (Ethres). En dessous les e- produit par ionisation ne sont pas modélisés et l'énergie d'ionisation est entièrement déposée sur le chemin de la particule incidente.

PHOTONUC:

Active la modélisation des interactions gamma-noyau.

MUPHOTON:

Contrôle les interactions photo nucléaires entre muons

PAIRBREM:

Contrôle la simulation de production de pair et le bremsstrahlung (rayonnement continu de freinage) issus de muons à haute énergie, de hadrons chargés et d'ions légers. Elle est principalement utilisée pour prédire le couple des muons.

DISCARD:

Cette carte permet de supprimer certaines des particules qui ne nous intéressent pas. Dans le cas où 2 cartes se suivent la seconde peut réactiver certaines des particules supprimées par la première (si elle a supprimé par exemple toutes les particules dont la référence est comprise entre 1 et 10).

LES CARTES TRANSPORT

PART-THR:

Cette carte permet de fixer le threshold des particules. En général on le met à un MeV sinon le CPU devient trop important. Dans le cas des neutrons, on peut le mettre plus bas car c'est ce qui nous intéresse et ils réagissent peu.

IONTRANS:

Active le transport des ions lourds⁸.

EMF:

-on: les photons et les électrons sont modélisés (valeur par défaut)

-off: on ne les prend pas en compte ce qui permet de réduire le temps de calcul (sans perdre l'information utile dans le cas de spectre neutrons).

MGNFIELD:

Permet de spécifier la précision du transport de particules chargées dans un champ magnétique (angle max par pas etc.). Lorsque les 3 composantes du champ sont nulles, elle indique à Fluka d'utiliser la routine qui définit ce champ.

EMFCUT:

Permet de choisir le threshold des électrons et photons en fonction des matériaux (inutile donc si EMF est sur off). Cette carte n'a de sens qu'avec une option EMF.

TIME-CUT:

Permet de choisir combien de temps est suivit une particule avant d'être considérée comme perdue (par défaut aucun cutoff). Dans le cas où il y aurait 2 cartes, la première peut spécifier un time-cut commun à toutes les particules et la seconde à rallonger le time-cut de l'une des particules (c'est pour cela qu'elle doit être placée derrière).

⁸ Voir chap. « Les modèles physiques à haute énergie »

QUELQUES ROUTINES UTILES

Nous présentons ici 2 routines que nous avons utilisées et qui donnent un aperçu du fonctionnement général des routines dans Fluka. Ainsi plusieurs des conseils que nous donnons ici sont valables de manière plus générale avec d'autres routines.

UTILISATION DE LA ROUTE MGDRAW

Il s'agit d'une routine qui doit permettre d'écrire dans un fichier des informations précises sur chaque interaction (comme par exemple s'il s'agit de collisions élastiques ou non, quelles sont les particules créées etc.).

CARTES NECESSAIRES

-USERDUM: appelle spécifiquement cette routine

Type: dump (écriture complète, a l'aide de mgdraw)

dump: user defined (car on a adapté la routine)

Unit: (vide) (car la routine prévoit déjà un enregistrement dans l'unit 61)

what: complète

Score: all

-OPEN: ouvre un fichier dans lequel écrire les informations que l'on souhaite retenir

Unit: 61

Statue: new

File: nomdufichier.txt

COMPILE:

Dans l'onglet « compile » et avant de lancer la simulation :

- on ajoute la routine.f (avec + à droite)
- on donne un nom à l'exécutable (exe:)
- on compile puis on build

FICHIERS CREEES

Les différents fichiers créés sont :

- un fichier/ cycle contenant les infos voulues
- un exécutable et un .map qui sont utilisés par fluka

Les fichiers en sortie sont :

-dump4: donne des infos sur les particules émises à chaque réaction

- dump3: pareil mais seulement pour les ions lourds
- dump2: pour chaque part émise, donne A et Z
- dump1: donne la particule incidente et les infos correspondantes

ANALYSE DU RESULTAT

Le résultat apparait dans les fichiers créés après chaque cycle. Le premier chiffre en partant de la gauche indique le rang de la particule étudiée (commence à 1 pour l'interaction d'un primaire puis 2 et). Une fois toute la cascade étudiée (c'est à dire toutes les particules issues de la particule primaire) on recommence à étudier une chaîne de réaction, en repartant donc de 1. Par exemple:

```
1...
2...
2...
1...
2...
2...
```

Signifie qu'une particule primaire (dont les caractéristiques ont données dans la première ligne) a donné 3 secondaires (2ieme 3iem et 6ieme ligne) et que la seconde particule secondaire a créé une autre particule secondaire (ligne 5).

- ligne de particule primaire (commençant par un 1):

le type d'interaction (100 élastique 101 inélastique)

Le num de la part

Le type de particule (par ex proton=5, neutron=8)

L'énergie

L'angle (en 3 composantes x, y et z)

Le poids

La région ou a eu lieu l'interaction

F= pas de fission

- ligne de particule secondaire (ne commençant pas par 1):

le numéro de la particule primaire mère

Le type de particule

A

Z

L'énergie

L'angle (en 3 composantes x, y et z)

Le poids

UTILISATION DE LA ROUTINE SPEC_SOUR

Cette routine permet d'envoyer un spectre quelconque depuis un beam dont la géométrie est simple (pour des géométries complexes une autre routine est nécessaire). Nous détaillons ici les fichiers nécessaires, les cartes à rajouter ainsi que le fonctionnement général de la routine.

FICHIERS NECESSAIRES:

-les fichiers spectres (un par variété de particules) qui apparaissent sous forme de 2 colonnes: le premier contenant la borne inférieure du bin en énergie et la seconde contenant le flux dans ce bin.

```
Ex:      1.00e-04 1.96e+05
         1.26e-04 1.66e+05
         1.58e-04 1.26e+05
```

-le fichier SPECT+analog.f qui permet la création du beam à partir des fichiers spectres et de la géométrie indiquée par la carte (à modifier comme indiqué ci-dessous).

-le fichier source_spect+analog.f qui permet d'associer le spectre à la géométrie du beam.

-dummy common qui est appelé par source_spect+analog.f .

CARTES NECESSAIRES:

La carte BEAM et la carte START (comme d'habitude) en prenant soin que la gamme d'énergie de la carte BEAM recouvre bien celle des spectres en entrée (les autres champs ne sont pas pris en compte).

Les cartes permettant de définir la géométrie du BEAM. Dans notre cas par exemple, si l'on souhaite tirer notre spectre depuis une sphère, de la surface vers l'intérieur, on utilisera une BEAMPOS de type négative (le flux va vers l'intérieur) et une BEAMPOS de type flood (définit le rayon de la sphère).

La carte SOURCE qui appelle la routine spec_sour (pas besoin de remplir les autres champs).

SPECIFICATION DE LA SOURCE.FOR

Dans le fichier spect+analog.f on adapte la routine à notre besoin en changeant les lignes suivantes :

-le nombre de particules différentes qui seront envoyées

```
PARAMETER (NROFPARTICLES=2)
```

-leur genre et la localisation du fichier contenant le spectre

```
FILENAMES(1) = '../CERF_POS6_Elias_ph'
```

PARTICLETYPE(1) = 7

-il faut ensuite implémenter les indices i dans particule(i) et filename(i)

LES DIFFERENTES ETAPES A SUIVRE:

-Il faut commencer par compiler les routines que l'on va utiliser. Dans l'onglet compile avec le +, on ajoute SPECT+analog.f, et source_spect+analog.f on choisit un nom d'exécutable (Exe:), on compile (création de fichiers .o et des exécutables) puis on build

-Ensuite on lance un run normal

-Après la simulation, des informations apparaîtront dans le fichier .log sous la forme suivante:

```
#-----  
# Mixed field sampling - V 1.0,  
# March 2008, Chris Theis & Helmut Vincke  
#-----  
# analogue sampling of the spectrum is performed  
# position, direction & spatial distribution parameters are taken from the BEAM card,  
# with the exception of the ENERGY!!!  
### reading file no. 1  
### reading file no. 2  
### reading file no. 3  
### reading file no. 4  
# accumulated total fluence: 4.82667423E-05  
# ID, acc. individual fluence, acc. contribution prob.  
8 1.60994556E-05 0.333551734  
6.82550824E-08 0.334965857
```

EN CONCLUSION

Nous avons tenté de présenter ici les informations de base permettant une première approche de l'utilisation du code Fluka et de l'interface Flair, dans le cas de simulation simples. Il s'agit d'une documentation non officielle qui s'appuie principalement sur les documents suivant :

- le site officiel de Fluka : <http://www.fluka.org/fluka.php>
- le manuel d'utilisation de Fluka disponible sur le site
- le manuel d'utilisation de Flair disponible sur le site
- le manuel d'utilisation Fluka/Flair rédigé par Sébastien Wurth

Il est conseillé de se reporter à ces ouvrages en cas de doute ou pour obtenir des informations supplémentaires.

REMERCIEMENTS

Un grand merci pour leur soutien et leur patience à Alfredo Ferrari et Markus Brugger (CERN) sans qui ce travail n'aurait pas été possible.

Merci aussi à Ruben Garcias (CERN) pour son aide, notamment en ce qui concerne l'utilisation des routines Fluka.

Tous mes remerciements à Arnaud Claret (CEA) pour son implication dans ce projet et la confiance qu'il m'a accordé.

Merci enfin à toute la Fluka Team, pour leur accueil et leur aide au cours de mes différents séjours au CERN.