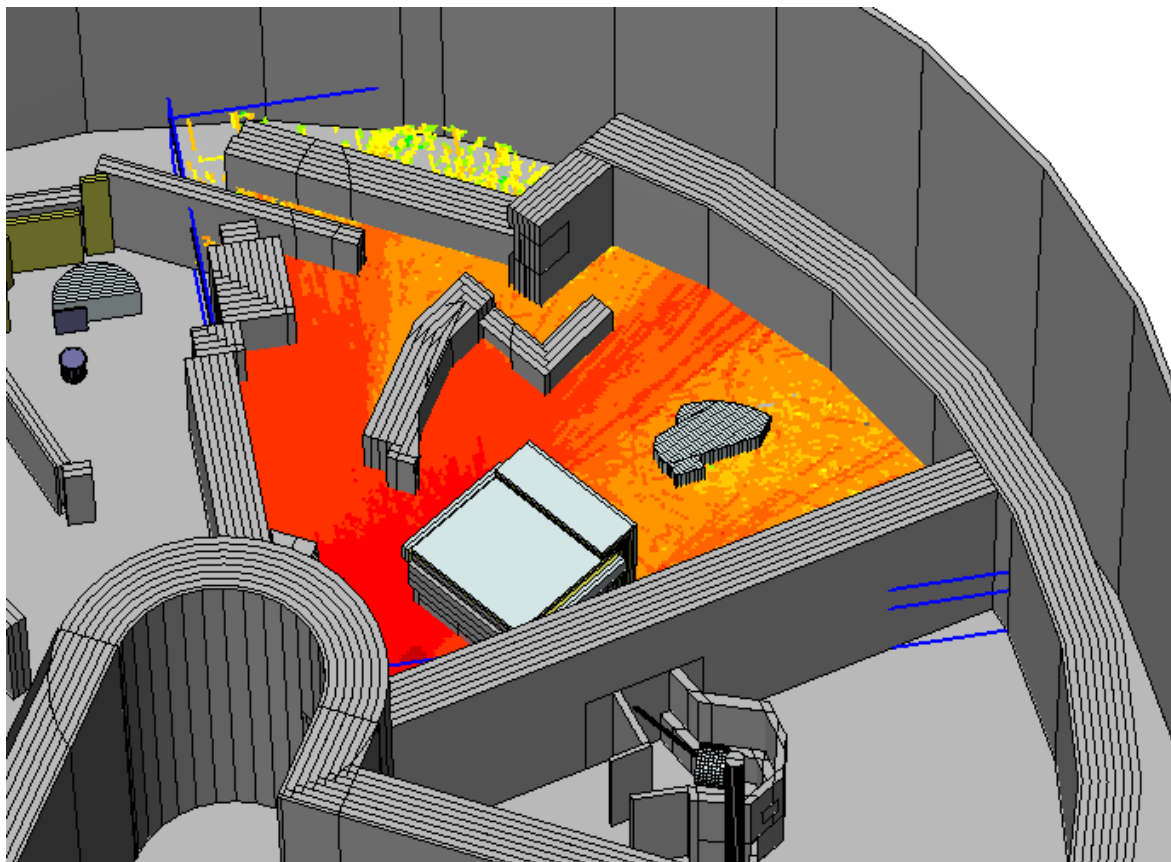




UTILISATION DU CODE FLUKA PAR LE SERVICE DE PRÉVENTION ET DE RADIOPROTECTION DE L'INSTITUT DE PHYSIQUE NUCLÉAIRE D'ORSAY



Rédaction		Vérification, approbation	
<i>Version n° 0.9.1</i>			
S. WURTH	20/04/2012	J.-F. LE DU	23/04/2012

Introduit par FAC SQE n° E972

Introduit par FAC SCR n° 60

ÉVOLUTION DES VERSIONS

V.	Date	Description des modifications
0.	19-04-2005	Version initiale du document pour la version FLUKA 2003.1.b du code.
1.	19-09-2005	Modification après la sortie de FLUKA 2005.6 <u>Suppression :</u> Ancien chapitre 3. Le pré processeur électromagnétique (PEMF) PEMF est intégré dans le code à partir de cette version. <u>Ajouts :</u> <ul style="list-style-type: none"> ➤ Chapitre 9. Utilisation de l'option SOURCE ➤ Annexe C. Routine source.f pour simuler une source isotrope de cobalt 60
2.	20-02-2006	Modification après la résolution par l'équipe de développement d'un bug de FLUKA concernant l'utilisation de la décroissance radioactive (RADDECAY). <u>Ajout :</u> <ul style="list-style-type: none"> ➤ Chapitre 10. Autre méthode pour tracer des graphes des fichiers de sortie de l'estimateur USRBIN
3.	20-06-2006	<u>Ajouts :</u> <ul style="list-style-type: none"> ➤ Description des outils pour tracer des spectres de fluence, USRTRACK et USRBDX. 6.3. et 6.4. ➤ 7.5. Utilisation de la routine deq99c.f ➤ Annexe D. Tracer un profil de géométrie avec ALIFE. ➤ Changement d'ordre des paragraphes 9 et 10 ➤ Dans le nouveau chapitre 10. : <ul style="list-style-type: none"> ✓ Ajout de l'algorithme pour obtenir une source d'Eu152 dans 10.1. ✓ Ajout des résultats de simulation de sources Co60 et Eu152. 10.2. et 10.3. ➤ Chapitre 11 décrivant les options de calculs de noyaux résiduels et décroissance radioactive.
4.	19-09-2006	Adapté pour la version 2006.3 du code (cf. page d'introduction pour les détails). <ul style="list-style-type: none"> ➤ Révision complète du chapitre 1. ➤ Dans le chapitre 2. suppression de la reproduction du README accompagnant la distribution du code. ➤ Suppression de l'ancien paragraphe 7.4. : étude incomplète et contredite par l'expérience ultérieure. ➤ <u>Ajout :</u> Chapitre 4. : description des formes géométriques RCC et PLA.
5.	06-02-2007	<u>Ajouts :</u> <ul style="list-style-type: none"> ➤ Une annexe consacrée à flukaGUI (Graphical User Interface) pour tracer des graphes à partir de résultats de l'estimateur USRBIN. ➤ Description d'une source de neutrons isotrope (Am-Be). ➤ Annexe : Étude comparative source monoénergétique et source réelle de neutrons. ➤ Annexe : Utilisation de l'option LATTICE pour la construction de la géométrie.
6.	05-03-2007	<u>Ajouts :</u> <ul style="list-style-type: none"> ➤ Description d'une source linéairement répartie.
7.	20-04-2007	Remaniement du document pour y incorporer des chapitres concernant des utilitaires ou interfaces graphiques d'importance décisive (SimpleGeo, FLAIR), modification de chapitres déjà existants. Version 2006.3b
7.1	16-05-2007	Pour versions 2006.3b.2 de FLUKA et 0.4 de FLAIR. Changement dans le tuto flair, l'option MATERIAL dans le tracé de géométrie fonctionne à présent.
8.	19-07-2007	Ajout d'un chapitre sur la carte DETECTOR : utilisation et exploitation des données de « sortie ». Versions 2006.3b.4 de FLUKA et 0.5.1 de FLAIR.

8.1	26-09-2007	Quelques précisions pour simuler une source cylindrique ou sphérique. Versions 2006.3b.5 de FLUKA, 0.6 de FLAIR et 3.0 de SimpleGeo.
8.2	17-01-2008	Complément au sujet des matériaux : ce qui change avec les nouvelles options. Ajout d'un paragraphe 5.3. Ajout d'un paragraphe 10.4.4. : 2 ^{ème} solution pour définir une source d'Am-Be (neutrons).
8.3	28-05-2008	Ajout d'une vraie étude avec la carte DETECTOR (Ge Canberra) (6.5.5) Routine fluscw.f pour des électrons (7.5) Décrire une forme elliptique dans la section géométrie (complément à 4.3)
8.4	02-10-2008	Évolutions avec la sortie de la version 2008.3 du code, principalement : La nouvelle bibliothèque de sections efficaces pour les neutrons d'énergie < 20 MeV (260 groupes de neutrons et 42 groupes de photons gamma). Nouvelle base de données de décroissance radioactive (prise en compte des conversions électroniques et des électrons Auger). La routine deq99c.f est désormais incluse dans le code, on peut y faire appel via le code 240.0 ou DOSE-EQ en ce qui concerne la quantité à calculer dans les estimateurs et via la carte AUXSCORE qui permet d'associer les estimateurs aux facteurs de conversion (modification du paragraphe 7.4 pour décrire la carte AUXSCORE). Versions 0.7 de FLAIR et 3.1 de SimpleGeo.
8.5	23-11-2009	Ajout d'un chapitre concernant les techniques de réduction de la variance (biaisage) et rajout d'un tuto concernant l'utilisation du plugin DaVis3D de SimpleGeo (version 4.0). Suppression de l'ancienne annexe F consacrée à une étude sur les neutrons. Allègement du paragraphe 10.4.
9.0	16-02-2011	Ajout d'un tutoriel pour installer et utiliser flupix. Ajout de la description du geoviewer de flair, dans le chapitre consacré à cette interface. Nouvelle version de la routine de conversion fluence/Hp(0.07) électrons.
9.1	20-04-2012	Nouveautés apportées par la version 2011.2 du code, principalement : - possibilité de décrire des distributions spatiales complexes lors de la définition des caractéristiques du faisceau : sphérique, cylindrique, cartésien, isotrope dans une sphère - possibilité d'effectuer des transformations géométriques automatiques sur des corps constituant la géométrie du problème - possibilité de ne considérer que certaines régions pour le calcul de la décroissance radioactive

TABLE DES MATIÈRES

Introduction	1
1. Présentation du code de calcul FLUKA	2
1.1. Généralités	2
1.2. Modèles physiques.....	2
1.3. Calculs de blindages et de neutroniques	3
1.4. Prédiction de la production de radioéléments	4
1.5. Techniques de biaisage et estimation des grandeurs physiques	5
2. Installation, compilation, exécution	6
2.1. Installation	6
2.2. Compilation	7
2.3. Exécution	7
3. Généralités sur les entrées du code FLUKA.....	9
3.1. Format des entrées et unités physiques	9
3.2. Définition des caractéristiques de la source ou du faisceau	10
4. La géométrie combinatoire.....	13
4.1. Présentations et recommandations.....	13
4.2. Procédure pour définir une géométrie dans FLUKA.....	14
4.2.1. La carte GEOBEGIN	14
4.2.2. La carte de titre de la géométrie	15
4.2.3. Body data	15
4.3. Description de quelques formes	15
4.4. Les régions et la géométrie combinatoire	18
4.4.1. Region data.....	18
4.4.2. Signification des opérateurs	19
5. Les matériaux.....	21
5.1. Définition des matériaux	21
5.2. L'assignation des matériaux aux régions	23
5.3 Remarque	24
6. Les estimateurs	25
6.1. Description de la carte USRBIN	25
6.1.1. Première carte	25
6.1.2. Seconde carte	26
6.2. Exemple d'estimateur USRBIN	26
6.2.1. Première carte	26
6.2.2. Seconde carte	27
6.2.3. Remarques	27
6.3. La carte USRTRACK.....	27
6.3.1. Première carte	27
6.3.2. Seconde carte	27
6.3.3. Notes	28
6.3.4. Exemples	28
6.4. La carte USRBDX.....	29
6.4.1. Première carte	29
6.4.2. Seconde carte	30
6.5. La carte DETECT.....	32
6.5.1. Description.....	32
6.5.2. Exemples.....	33
6.5.3. Remarque.....	34
6.5.4. Récupération des données de sortie	34
6.5.5. Étude de l'efficacité absolue de détection d'un cristal hyper pur de germanium.....	35
7. Calcul des doses	39
7.1. La carte USERWEIG	39
7.2. Utilisation de la routine FLUSCW	39
7.3. Utilisation de la routine COMSCW.....	40
7.4. Facteurs de conversion CIPR74	40
7.4.1 Utilisation de la routine deq99c.f.....	40
7.4.2 Carte AUXSCORE.....	41
7.5. Cas des électrons.....	42
8. Tracer des graphes en couleurs avec les fichiers de sortie d'USRBIN	44
8.1. L'option PLOTGEOM de FLUKA	44
8.2. Fichier « résumé » d'USRBIN	45
8.3. Fichiers auxiliaires pour le tracé du graphe.....	45
8.4. Tracé du graphe avec PAW	47
8.5. Exemple	48
9. Autre méthode pour tracer des graphes des fichiers de sortie USRBIN	49
9.1. Format des fichiers de sortie	49
9.2. Lien symbolique.....	49

9.3. Fichier de liste, propriétés du graphe	49
9.4. Tracé de la géométrie	50
9.5. Création du graphe	50
9.6. Tracé du graphe avec PAW	51
10. Utilisation de l'option SOURCE	52
10.1. Mode d'emploi de l'option SOURCE	52
10.2. Définir une source isotrope de cobalt 60	52
10.2.1. Créer une source produisant deux photons	52
10.2.2. Créer une source isotrope	53
10.3. Définir une source isotrope d'europium 152	54
10.4. Définir une source isotrope de neutrons (Am-Be)	57
10.4.1. Spectre d'émission typique d'une source d'américium-béryllium	57
10.4.2. Algorithme pour décrire les probabilités d'émissions par intervalle	58
10.4.3. Algorithme pour assurer l'isotropie de la source	58
10.4.4. Étude comparative	59
10.5. Définir une source uniformément répartie sur une ligne	60
10.5.1. Source linéaire uniformément répartie	60
10.5.2. Application : source « bilinéaire »	61
10.6. Définir une source de forme circulaire et sphérique	62
10.6.1. Définir une source circulaire	62
10.6.2. Définir une source sphérique	63
10.6.3. Définir une source cylindrique isotrope	63
10.7. Autres routines potentiellement utiles	64
11. Utilisation des options de production de noyaux et de décroissance radioactive	65
11.1. Un peu de physique	65
11.2. La carte RESNUCLEI	65
11.3. La carte RADDECAY	67
11.4. La carte IRRPROFILE	68
11.5. La carte DCYTIMES	68
11.6. La carte DCYSCORE	69
11.7. Application : l'ensemble cible-source d'ALTO	70
12. Utilisation d'un utilitaire de géométrie : SimpleGeo	74
12.1. Manipuler un fichier d'entrée de FLUKA dont la géométrie existe déjà	74
12.2. Créer une géométrie pour FLUKA avec SimpleGeo	78
12.3. Utilisation de la macro DaVis3D	81
13. Utilisation de l'interface FLAIR : initiation	85
13.1. Généralités	85
13.2. Charger un fichier et inspecter les données	86
13.3. Généralités sur les graphes et représentation de la géométrie	87
13.4. Représentation d'un graphe USRBIN	89
13.5. Représentation d'un graphe USR-1D	90
13.6. Représentation d'un graphe RESNUCLEI	92
13.7. Utilisation de l'éditeur de géométrie	94
13.7.1. Intallation	94
13.7.2. Utilisation	94
14. Techniques de réduction de la variance, biaisage	100
14.1. Biaisage d'importance	100
14.2. Biaisage de la cascade électromagnétique	102
14.3. Biaisage par fenêtres d'énergie et de poids statistique	103
14.3.1. Carte WW-THRESH	105
14.3.2. Carte WW-FACTOR	106
14.4. Biaisage de la longueur d'interaction	108
14.5. Combinaison de divers types de biaisage dans une application concrète	109
15. Utilisation de fluxpix : un cd image de flair et fluka	109
15.1. Installation	109
15.2. Configuration	110
15.2.1. Configuration générale	110
15.2.2. Configurations optionnelles : son, usb et disque dur image	111
15.2.3. Configuration des disques durs image	111
15.3. Exécuter FLUPIX	111
15.4. Trucs et astuces	112
15.4.1. Configuration clavier et souris	112
15.4.2. Transfert de données	112
Conclusion	113
Références pour la rédaction de ce document	114
Annexes	114
A. Quelques messages d'erreurs typiques	114
B. Exemple de fichier d'entrée	116
C. Routine <i>source.f</i> pour simuler une source isotrope de cobalt 60	119
D. Tracer un profil de géométrie avec ALIFE	122

D.1. Installation d'ALIFE.....	122
D.2. Tracé d'un profil de géométrie	122
E. Utilisation de l'interface graphique flukaGUI	126
E.1. Installation.....	126
E.2. Petit aperçu des possibilités	126
F. Utilisation de l'option LATTICE pour la construction de la géométrie.....	130
F.1. Généralités.....	130
F.2. Exemple 1.....	132
F.3. Exemple 2.....	135
G. Commandes UNIX et routines fortran utiles	138
H. Un peu de physique.....	139
I. Carte DETECT : fichiers utiles.....	141
I.1. Récupération de données de sortie : DETOUTPUT.....	141
I.2. Fichier d'entrée Ge-CANBERRA et source de type SG500	142
I.3. Fichier source de type SG500	146
J. Routine FLUSCW pour la conversion fluence/dose dans le cas des électrons	149

Introduction

Ce document a pour objectif de constituer un guide à l'utilisation du code FLUKA pour des études de simulations simples (ou un peu moins) dans le cadre de son utilisation par le Service de Prévention et de Radioprotection de l'Institut de Physique Nucléaire d'Orsay. Nous allons très fréquemment renvoyer vers le manuel du code **FLUKA**, la référence principale de ce document (bien que les auteurs n'en préconisent pas la mention en référence du fait des fautes pouvant s'y trouver). Il s'agit de la version de **2011**. Certains changements issus de versions plus récentes peuvent survenir.

Il est possible que certains points soient plus facilement résolubles qu'avec la méthode présentée. Le lecteur est fortement encouragé à faire part de toute observation ou nouvelle méthode de résolution, présentation qui améliorerait celle proposée dans ce document. Ceci est important pour sa constante évolution.

De la même façon que les développeurs de **FLUKA** préviennent que leur code contient très probablement un nombre conséquent de « bugs », ***il faut rappeler qu'il ne s'agit pas là d'un document « officiel » validé par quelque autorité en la matière, mais bel et bien d'un document de travail issu d'élucubrations personnelles, à manipuler avec toutes les précautions nécessaires par conséquent.***

L'utilisateur novice (ou un peu plus expérimenté) peut tester de nombreuses pistes d'analyse tant au niveau de la construction de sa géométrie, de la modélisation des propriétés physiques, des techniques de réduction de la variance, du choix de estimateurs à utiliser, du choix de l'interface graphique pour exploiter ses résultats.

S. Wurth, SPR, IPN d'Orsay.

1. Présentation du code de calcul FLUKA

1.1. Généralités

FLUKA est un code de calcul utilisant la méthode de Monte Carlo basée sur des tirages de nombres aléatoires, du traitement d'une particule simulée de son lancement jusqu'à sa « mort », en tenant compte de toutes les interactions qu'elle est susceptible de subir. Le code peut ainsi simuler une large gamme des interactions « particule-matière », et cela sur un vaste domaine d'énergie pour les différents types de particules.

Le développement de la version moderne du code **FLUKA** a débuté à la fin des années quatre-vingt avec comme objectif la mise au point d'un outil de simulation adapté aux calculs pour le **LHC**. À cette époque, la majorité des calculs impliquant des rayonnements était effectuée avec des versions antérieures du code issues de développements démarrés au milieu des années soixante. Le bouleversement qui a permis de transformer un programme spécifiquement dédié aux calculs de blindages en outil généraliste utilisé dans des domaines aussi variés que la calorimétrie, la mise au point de détecteurs, la simulation de rayonnement cosmiques, la dosimétrie ainsi qu'un nombre important d'autres applications, a commencé après la distribution de la version de 1987 et est dû essentiellement à un travail effectué par A. Fassò (actuellement au SLAC et précédemment au CERN), A. Ferrari (actuellement au CERN et précédemment à l'INFN), J. Ranft (Siegen) et P.R. Sala (INFN). Suite à cet effort initié par le nouveau défi que représentait la mise au point de la nouvelle génération de collisionneur de protons, le code s'est transformé en un outil capable de suivre de manière précise et fiable aussi bien le développement complet d'une cascade hadronique impliquant des énergies de plusieurs dizaines de milliers de TeV jusqu'à la contribution des neutrons thermiques que de transporter des muons de 1000 TeV jusqu'au repos. Actuellement, le domaine d'applications du code est très large et comprend la radioprotection, la dosimétrie, la simulation de détecteurs, la physique des rayonnements cosmiques et spatiaux, la neutronique ainsi que l'hadron-thérapie.

1.2. Modèles physiques

FLUKA permet de simuler les interactions des faisceaux avec la matière dans une gamme d'énergie très large. Une fois couplé au code DPMJET-3, le code s'est avéré capable de traiter les cas suivants :

- Les faisceaux de hadrons et d'ions d'énergie comprise entre quelques MeV/n jusqu'à 10000 TeV/n.
- Le transport de neutrons jusqu'aux énergies thermiques.
- Les rayonnements électromagnétiques de 1 keV à 10 000 TeV.
- Les muons jusqu'à une énergie de 10 000 TeV.
- Les neutrinos jusqu'à 1 TeV.

Le transport ainsi que les interactions des neutrons d'énergie inférieure à 20 MeV sont effectués grâce à deux bibliothèques couplées spécialement développées pour **FLUKA** qui sont basées sur une structure à 260 ou 72 groupes de neutrons et 42 ou 22 groupes de photons. Ces bibliothèques ont été produites par le code NJOY (la version actuelle NJOY99) à partir des évaluations les plus récentes (principalement ENDF/B-VI version 8, JENDL-3.2 et 3.3, JEF-3.0).

La structure en groupe de photons est utilisée uniquement pour échantillonner l'énergie des gammas émis, le transport est effectué à partir du module standard utilisé pour le calcul du rayonnement électromagnétique. Les bibliothèques utilisées contiennent les informations

nécessaires à la prédiction de la production de noyaux résiduels résultant de l'interaction d'un neutron ainsi que les facteurs de kerma. Pour certains éléments légers (typiquement l'H, l'He et le Li), il est possible d'effectuer le transport des neutrons dans un mode continu en tenant compte explicitement de la génération de particules chargées de recul (essentiellement des protons de recul). Pour les neutrons d'énergie supérieure à 20 MeV ainsi que pour toutes les autres particules quelque soient leurs énergies, les interactions nucléaires implémentées dans le code sont modélisées en suivant la séquence suivante :

- Modèle de cascade Glauber-Gribov (non utilisé en dessous de quelques GeV).
- Cascade intranucléaire généralisée (GINC, non utilisé en dessous de 50 MeV).
- Émission dans un état de pré équilibre.
- Évaporation, fission, fragmentation.
- Désexcitation gamma.

Les interactions individuelles entre les hadrons et les nucléons qui deviennent significatives une fois que le seuil utilisé pour la GINC explicite est dépassé, sont décrites selon la séquence suivante :

- Diffusion élastique et échange de charge basée sur une analyse de changement de phase (jusqu'à quelques GeV) et un modèle dit eikonal pour des énergies plus élevées.
- Génération de particules basée sur une production de résonance et de décroissance jusqu'à quelques GeV et utilisant un modèle de quark et de cordes (DPM, Dual Parton Model) pour des énergies plus élevées.

De nombreux détails sur ces modèles et leurs performances ainsi que la comparaison pour de nombreux cas des données expérimentales de production de particules disponibles aux résultats prédits par le code peuvent être trouvés (*cf. références à la fin de ce document*). Il est intéressant de rappeler que **FLUKA** était en 1991, le seul code, avec LAHET, à harmoniser dans un cadre cohérent les modèles nucléaires utilisés dans les basses énergies (i.e. pré équilibre) avec ceux généralement dédiés aux énergies intermédiaires (i.e. INC). La précision des modèles est particulièrement importante pour les calculs de production de noyaux résiduels ainsi que pour la génération de particules qui peut généralement être validée par l'intermédiaire des sections efficaces de production obtenues pour des cibles minces ou épaisses. Cette dernière fonction peut généralement être réalisée avec un très haut niveau de précision quand des modèles correspondant à l'état de l'art sont utilisés, les performances obtenues sont d'ailleurs assez fiables. L'estimation de la production de noyaux résiduels est cependant plus délicate étant donné que des modifications ou des faiblesses minimales dans la modélisation et tout particulièrement lors de la phase de pré équilibre, peuvent induire de larges imprécisions dans le taux de prédiction de noyaux résiduels sans toutefois affecter notablement le spectre des particules émises. De plus, les isotopes les plus dangereux sont souvent le résidu d'une faible fraction de l'interaction totale possible (représentant parfois 1/1000 de la section efficace de la réaction). À partir du milieu des années 90, les modèles d'interactions ont été progressivement améliorés afin de pouvoir reproduire de manière satisfaisante l'émission de particules mais également les taux de production de radioéléments.

1.3. Calculs de blindages et de neutroniques

Le développement longitudinal et latéral des cascades hadroniques est une caractéristique essentielle qui doit être reproduite correctement afin d'obtenir un résultat utilisable pour le calcul des blindages et particulièrement dans le cas où de larges facteurs d'atténuations doivent être considérés. Dans beaucoup de problèmes relatifs aux blindages latéraux, une fois la cascade hadronique entièrement développée, les neutrons dominent la composante

du rayonnement émergeant des écrans. Le spectre énergétique des neutrons correspond généralement à un spectre à l'équilibre avec une forme dépendant du type de blindage et un facteur d'atténuation défini par celui des neutrons dans une gamme en énergie comprise entre 50 et 200 MeV. **FLUKA** s'est révélé très performant pour reproduire avec une grande précision et dans une gamme très large en énergie, les spectres de particules (particulièrement des neutrons) issus de blindages. Un exemple illustrant la capacité du code à prédire correctement la distribution spatiale et énergétique du flux d'hadrons, ainsi que le profil de la courbe de dépôt d'énergie peuvent être trouvés dans une étude, où les résultats du code sont comparés à ceux obtenus durant les expériences de Rösti.

La capacité du code à reproduire la production ainsi que le transport de neutrons dans les matériaux lourds a également été testée à des niveaux jamais atteints jusque là durant les expériences FEAT et TARC principalement, qui se sont déroulées au **CERN**.

Le transport des neutrons d'énergie intermédiaire dans des écrans de fer ou de béton a donné lieu à la réalisation d'un benchmark international qui a conclu que **FLUKA** était le code le plus adapté pour ce type d'exercice. Peut être que l'exemple le plus révélateur des capacités du code dans le domaine de la radioprotection peut être trouvé dans la comparaison des calculs avec les résultats obtenus depuis plusieurs années dans l'installation CERF située au **CERN**. Cette installation permet d'obtenir un champ diffus de rayonnements à l'extérieur d'un blindage de béton et d'acier d'épaisseur variable, grâce à l'interaction d'un faisceau d'hadrons (de 120 à 200 GeV) interagissant avec une cible de cuivre. L'objectif de l'installation et de tester la validité des appareils de mesures ainsi que des codes de calculs dans un environnement défini qui s'avère représentatif du points de vue du champ de radiations de celui attendu dans un accélérateur de hadrons de haute énergies ou de celui induit par les rayonnements cosmiques pour des altitudes correspondant à celles des vols commerciaux. Des équipes de physiciens de plusieurs pays sont impliquées dans ces mesures au **CERN** ainsi que dans leur comparaison systématique aux résultats des simulations : l'accord entre les deux s'est révélé si convaincant que les spectres calculés avec **FLUKA** sont considérés comme références pour l'étude de cette installation. **FLUKA** a été utilisé non seulement pour simuler l'installation, mais également pour caractériser la fonction réponse des différents appareils utilisés pour la détection des neutrons. Dans une configuration différente utilisée cette fois pour la caractérisation des particules très pénétrantes émises vers l'avant, l'installation a pu être utilisée pour vérifier les prédictions du bruit de fond et du blindage de l'expérience ATLAS.

1.4. Prédiction de la production de radioéléments

La prédiction précise de la production de radioéléments grâce à l'utilisation de modèles nucléaires est de loin la tâche la plus compliquée. **FLUKA** se révélait déjà très adapté pour reproduire les données expérimentales concernant la propagation ainsi que la génération de particules en 1993 lorsque des générateurs d'événements (**PEANUT**) plus « sophistiqués » ont été introduits dans la gamme d'énergie intermédiaire. Par contre, l'obtention de taux de production de radioéléments précis a encore nécessité un effort supplémentaire. Le code a commencé à produire des résultats utilisables dans le domaine de la radioprotection après l'implémentation d'un modèle d'évaporation beaucoup plus complexe et performant au cours des années comprises entre 1997 et 1999 ce qui permet la prédiction d'inventaires de noyaux résiduels fiables. **FLUKA** a été utilisé pour résoudre les problèmes liés au démantèlement du **LEP**. Afin de valider le code, une série d'expériences a été menée et les données expérimentales ont été comparées aux simulations. Le bon accord obtenu est particulièrement significatif étant donné qu'il implique la bonne modélisation de toute la physique mise en jeu, soit le développement de la cascade électromagnétique, les données nucléaires relatives aux interactions photonucléaires, les interactions avec les noyaux et la

production de noyaux. Plus récemment, un vaste programme expérimental ayant pour objectif la validation du calcul de la production de radioéléments avec les codes a été mené au **CERN** en prévision du nouveau défi posé par la construction du **LHC**. La comparaison des calculs de production de radioéléments et de débit de dose résiduelle avec les résultats expérimentaux a clairement montré la fiabilité du code ainsi que des algorithmes associés qui permettent de faire évoluer les inventaires de radioéléments. Des améliorations apportées dans les modèles d'évaporation ont d'ailleurs encore perfectionné cet accord. Étant donné que la précision sur la distribution spatiale ou énergétique du flux de particules est toujours meilleure que celle associée à la production de radioéléments, chaque fois que des sections efficaces expérimentales de production de radioéléments existent pour le type de particules et la gamme d'énergie considérés, la convolution de ces spectres avec les sections efficaces de production est l'approche la plus exacte.

1.5. Techniques de biaisage et estimation des grandeurs physiques

Le code **FLUKA** peut être utilisé dans un mode analogue, généralement lorsque les fluctuations entre deux événements présentent un intérêt (cas de la simulation des détecteurs) mais il est également possible d'avoir recours à des techniques de biaisages (généralement pour les problèmes de pénétrations profondes de particules). Les techniques disponibles incluent notamment, la possibilité d'associer un poids aux particules et de gérer ces poids grâce à l'utilisation de fourchette de poids acceptés, pour les neutrons l'absorption non analogue ainsi que la diffusion forcée, la sélection des particules d'intérêt, et l'augmentation arbitraire de l'échantillonnage de réactions ou décroissances.

Le **chapitre 14** est entièrement consacré à la description de quelques-unes de ces techniques de biaisage.

Le code inclut également les algorithmes et les données permettant de faire évoluer les inventaires de radioéléments suite à n'importe quel type de profil d'irradiation et de refroidissement, ainsi que de déterminer le débit de dose résiduelle induit par le rayonnement bêta et gamma. Ces opérations faisaient auparavant partie de la phase de post-traitements des résultats de la simulation. Elles font désormais partie intégrante du code afin de pouvoir permettre un calcul direct, lors de la simulation du développement du rayonnement prompt, de l'inventaire de radioéléments et des débits de dose résiduelle pour un profil d'irradiation et un temps de refroidissement donnés. L'algorithme d'évolution permet d'obtenir une solution exacte des équations traduisant l'accumulation et la décroissance des radioéléments sans aucune restriction sur le nombre de noyaux ou d'isomères mis en jeu. Les grandeurs physiques souhaitées (flux, dépôt d'énergie, production de radioéléments) peuvent être évaluées à partir de divers estimateurs incluant la traversée de surfaces, les longueurs de trajectoires, la densité de collision, et la possibilité de calculer des cartes en 2D ou 3D des différents estimateurs calculés.

Les références indispensables accompagnant cette partie, effacées pour alléger le chapitre sont décrites à la fin du document technique qui a servi de support à ce chapitre : Alfredo Ferrari et Joachim Voltaire (CERN), « Présentation générale du code de calcul FLUKA », 03-03-2006.

2. Installation, compilation, exécution

Nous n'expliquerons ici que la procédure à employer sur un système **UNIX**.

2.1. Installation

Il faut compter environ 100 Mo d'espace disque pour installer et exécuter **FLUKA**. De plus, un compilateur **FORTRAN** est nécessaire sur votre plate-forme pour créer le fichier exécutable. Après le téléchargement de la distribution **FLUKA**, vous aurez un fichier compressé de « format UNIX », avec l'extension ***.tar** qui contient le « package » **FLUKA**. À partir de ce point, il faut utiliser un système **UNIX**.

Important

Créez un nouveau répertoire en préparation de l'installation. Déplacez le fichier que vous venez de télécharger dans ce répertoire avant d'extraire ! Supposons que ce répertoire se nomme **myFlukaArea** et le fichier **fluka.tar.gz**. L'installation se fait à l'aide des commandes suivantes :

```
mkdir myFlukaArea
mv fluka.tar.gz myFlukaArea/
cd myFlukaArea
gunzip fluka.tar.gz
tar -xf fluka.tar
setenv FLUPRO /somePath/myFlukaArea
```

Cette dernière ligne de commande implémente une variable d'environnement qui est requise par les scripts utilisés par **FLUKA** (sous **\$FLUPRO/flutil**). Cette ligne est spécifique de la SHELL UNIX utilisée sur votre machine. L'exemple ci-dessus est en syntaxe correcte pour « tcsh/csh ». Dans le cas d'une shell « bash », la commande équivalente sera :

```
export FLUPRO =/somePath/myFlukaArea
```

Pour que ces paramètres soient pérennes, il faut modifier les fichiers **.My_login_redhat** et **.My_tcshrc_redhat** (ou tout autre nom d'OS selon votre cas) dans votre répertoire racine de la forme **/home/user**.

Ou alors, créez un fichier **flupro.env** (contenant les lignes suivantes) à enregistrer par exemple dans le dossier où pointe **\$FLUPRO**.

```
--
# !/bin/csh -v
# source flupro.env initialisation pour FLUKA
setenv FLUPRO ~/Fluka

alias rfluka ${FLUPRO}/flutil/rfluka
alias lfluka ${FLUPRO}/flutil/lfluka
alias fff ${FLUPRO}/flutil/fff
alias alife ${FLUPRO}/ALIFE/alife
--
```

Les 2 dernières commandes définissent des alias, ou raccourcis de commande pour les scripts **rfluka**, **lfluka** et **fff**, et pour l'utilitaire **alife**.

À chaque nouvelle session ou ouverture de terminal, rendez vous dans le répertoire dans lequel pointe votre fichier et tapez la commande **source flupro.env** pour configurer votre environnement.

Techniquement, **FLUKA** est installé.

Jetez un coup d'œil aux fichiers **README** et **RELEASE-NOTES**, en particulier si vous êtes un nouvel utilisateur.

Avec la distribution se trouve également une version remise à jour du manuel. Celui-ci est la source d'information et d'aide majeure pour préparer et exécuter **FLUKA**.

Remarques

Dans les versions précédentes de ce mémo figuraient la reproduction du fichier **README**. Par souci d'allègement, ceci a été supprimé, mais il est très fortement conseillé de lire ce fichier. Il contient la description rapide des fichiers qui y sont contenus.

La partie **Usage notes** contient les instructions principales pour faire fonctionner le code.

En ce qui concerne les changements entre les versions successives, les détails figurent dans le fichier **RELEASE-NOTES.fluka200X.X** accompagnant la distribution.

À partir de la version **2008.3**, la création de l'exécutable standard ainsi que la compilation des routines auxiliaires figurant dans le répertoire **\$FLUPRO/flutil** est effectué en exécutant le fichier **makefile** figurant à la racine, dans **\$FLUPRO**.

```
[wurth@ipnscr01 ~/fluka2008] $ make
```

2.2. Compilation

Pour créer l'exécutable **FLUKA**, il faut lancer la commande suivante :

```
$FLUPRO/flutil/lfluka -m fluka  
Le nom du fichier sera flukahp
```

Si l'on désire tenir compte d'interactions d'ions lourds, il faut utiliser l'autre script :

```
$FLUPRO/flutil/ldpmqmd  
Le nom du fichier sera flukadpm
```

Ou **\$FLUPRO/flutil/ldpm3qmd** qui produit un exécutable **flukadpm3** (nouvelle version).

Par la suite, pour vos applications, vous serez amenés à utiliser des routines disponibles dans **\$FLUPRO/usermwax** et peut-être que vous devrez les modifier sensiblement pour l'usage que vous désirez en faire. Par exemple, pour calculer des doses, vous serez amenés à utiliser les routines **FLUSCW** ou **COMSCW** (*cf. paragraphes 7.2. et 7.3. de ce document*).

Après les avoir modifiées, selon votre choix, à l'aide d'un éditeur UNIX (**nedit** par exemple), il faut les compiler à l'aide du script disponible :

```
$FLUPRO/flutil/fff nom_de_la_routine.f  
Ceci générera un nouveau fichier nom_de_la_routine.o
```

Pour que ces changements soient effectifs dans votre future simulation, il faut recompiler l'exécutable de **FLUKA** en y liant la routine.

```
$FLUPRO/flutil/lfluka -o myfluka -m fluka nom_de_la_routine.o
```

Le fichier exécutable sera alors **myfluka** dans ce cas. Il est préférable de ne pas écraser votre **flukahp** de départ. Il s'agit de savoir ce que l'on fait en modifiant ces routines, sous peine de se retrouver avec des résultats dénués de signification au final ou dont on ignore le sens.

2.3. Exécution

La ligne de commande pour lancer une simulation est la suivante par exemple :

```
$FLUPRO/flutil/rfluka -e myfluka -N0 -M2 irradié
```

Le drapeau **-e** appelle le fichier exécutable, par défaut (si le drapeau n'apparaît pas) il s'agit de **flukahp**. Ici, il s'agit de **myfluka**.

Les drapeaux **-Nx -My** précisent le nombre de cycles : N est suivi du numéro de début (x) et M de celui de fin (y). Ici, deux cycles sont requis.

Le dernier nom est celui du fichier d'entrée écrit sans extension ***.inp**.
Ici, le fichier s'appelle **irradie.inp**.

Les fichiers de sortie rendant compte des cycles sont construits de la façon suivante :
nom_du_fichier_d'entréeXXX.out où **XXX** est le numéro du cycle.

Dans notre exemple, nous aurons deux fichiers de sortie **irradie001.out** et **irradie002.out**.
La taille de ces derniers est directement liée au nombre d'histoires (ou particules) devant être suivies.

Important pour rester en bons termes avec votre éventuel hôte

Lorsque vous êtes logués sur la machine de quelqu'un, l'emploi de la commande **nice** est apprécié par votre hôte qui peut ainsi poursuivre son travail sur la même machine sans voir son environnement « ramer ». Concrètement, cela modifie les priorités de chaque tâche.

La commande s'insère au début :

wurth@ipnpacs17<101> nice \$FLUPRO/flutil/rfluka ... etc.

Plus d'informations : tapez **man nice** dans votre terminal.

3. Généralités sur les entrées du code FLUKA

3.1. Format des entrées et unités physiques

L'entrée consiste en un fichier texte contenant une séquence de lignes d'options, souvent nommées « cartes » qui sont suivies quelquefois de cartes de données spécifiques de l'option choisie. Les cartes ont toutes la même structure et peuvent être lues en format « fixé » ou « libre ». Les formats sont définis par l'option **GLOBAL** (cf. *manuel*). **Le format par défaut est « fixé simple précision ».**

Format des entrées

La structure générale est la suivante : **CODEWD**, (**WHAT(I)**, **I = 1, 6**), **SDUM**

Où :

- **CODEWD** est le mot-clé représentant l'option choisie.
- Les paramètres **WHAT** sont des données numériques ou logiques sous forme numérique. Même s'il s'agit d'entiers, on y met un point décimal.
À partir de la version 2006.3, certains « what(s) » peuvent être les mots-clés ayant la même signification que les chiffres, exemple « 7.0 » pour « PHOTON ».
- **SDUM**, s'il est présent contient, sauf exception, des caractères.

L'ordre des cartes d'entrée est indifférent, à l'exception de :

- **GLOBAL** : si cette option est présente, elle doit précéder toute autre option exécutable.
- **DEFAULTS** : cette option doit également intervenir très tôt, seules peuvent précéder les cartes **GLOBAL**, **COMMENT**, **TITLE**.
- La commande **START** initialise l'exécution. Toute entrée donnée après **START** est ignorée à l'exception de **STOP**.
- La commande **STOP** fait cesser l'exécution du programme. Toute commande intervenant après la carte **STOP** est ignorée.

Certaines options nécessitent de nombreuses données d'entrée de telle sorte que deux ou plusieurs lignes sont requises pour en donner la description complète.

La plupart du temps, des données par défaut sont préenregistrées. Si elles conviennent pour la simulation voulue, il n'est pas nécessaire que la carte correspondante apparaisse dans la séquence d'entrée. De plus, pour la plupart des paramètres **WHAT** et / ou **SDUM**, une valeur par défaut / ! \ **Peut-être différente de la valeur par défaut si la carte n'est pas définie** / ! \ est appliquée si le champ correspondant est laissé vide (ou réglé à 0.0) dans la carte d'entrée.

Plusieurs cartes d'option peuvent apparaître plus d'une fois dans une séquence d'entrée. Dans la plupart des cas, chacune de ces cartes additionnelles ajoutent évidemment davantage de définitions de celles déjà données à condition qu'elles soient différentes et non contradictoires. En cas de conflit, les dernières données écrasent généralement les précédentes.

Unités physiques utilisées dans FLUKA

- Distance : cm (et donc cm² et cm³, pour les surfaces et volumes).
- Énergie : GeV (exceptions : l'eV est utilisé pour le potentiel d'ionisation moyen dans l'option **MAT-PROP** ; g/MeV/cm² est utilisé pour les coefficients de Birks dans l'option **TCQUENCH**).

- Quantité de mouvement : GeV/c.
- Température : degrés Kelvin.
- Angle solide : sr (exception : sur demande de l'utilisateur, les degrés peuvent être utilisés dans l'option **USRYIELD**).
- Champ magnétique : T.
- Champ électrique : V/cm.
- Temps : s (options **TCQUENCH** et celles liés à la décroissance radioactive) ou ns (option **TIME-CUT**).

Le format « fixé » est de la forme A8, 2X, 6E10.0, A8

Avec A : caractère alphanumérique

X : espace laissé vide

E : nombre décimal simple précision dans ce cas-là

3.2. Définition des caractéristiques de la source ou du faisceau

Cartes **BEAM**, **BEAMPOS**, **SOURCE**

BEAM définit plusieurs caractéristiques de faisceaux, de types de particules, d'énergie, de divergence, de profil et de poids statistique.

	WHAT (1)	WHAT (2)	WHAT (3)	WHAT (4)	WHAT (5)	WHAT (6)	SDUM
BEAM	-661.7E-6	0.0	1.E4	0.0	0.0	1.0	PHOTON

- **WHAT(1)** : > 0.0 : la quantité de mouvement moyenne en **GeV/c**
< 0.0 : l'énergie cinétique moyenne du faisceau en **GeV** (valeur absolue).
- **WHAT(2)** : > 0.0 : déviation de la quantité de mouvement en GeV/c (considérée comme rectangulaire)
< 0.0 : largeur à mi-hauteur de la distribution gaussienne (FWHM)
- **WHAT(3)** : spécifie la divergence du faisceau en mrad
> 0.0 : largeur de la distribution angulaire de forme rectangulaire
< 0.0 : FWHM de la distribution angulaire de forme gaussienne
> 2000 π mrad (soit 2 π rad) : la distribution est considérée comme isotrope
- **WHAT(4)** : > 0.0 :
si **WHAT(6)** > 0.0, c'est la largeur du faisceau dans la direction X en cm
si **WHAT(6)** < 0.0, c'est le rayon maximal d'un faisceau annulaire
< 0.0 : FWHM d'un profil gaussien en X, quelque soit **WHAT(6)**
- **WHAT(5)** :
> 0.0 : si **WHAT(6)** > 0.0, largeur du faisceau dans la direction Y en cm (le profil du faisceau est considéré comme étant rectangulaire)
si **WHAT(6)** < 0.0, **WHAT(5)** est le rayon minimum d'un faisceau annulaire
< 0.0 : **WHAT(5)** est la FWHM d'un profil gaussien en y, quelque soit **WHAT(6)**
- **WHAT(6)** : poids des particules. Si **WHAT(6)** < 0.0, **WHAT(4)** et **WHAT(5)**, s'ils sont positifs, sont interprétés comme les rayons maximal et minimal d'un faisceau annulaire. S'ils sont négatifs, ils sont interprétés comme étant les FWHM des profils gaussiens, indépendamment de la valeur de **WHAT(6)**.
- **SDUM** : nature des particules constituant le faisceau (*cf. liste des codes des particules pour FLUKA, chap. 5 du manuel*).

Dans l'exemple donné en ouverture du paragraphe, nous avons donc défini une source de césium 137, isotrope, de photons de poids statistique 1.

Autres exemples :

	WHAT (1)	WHAT (2)	WHAT (3)	WHAT (4)	WHAT (5)	WHAT (6)	SDUM
BEAM	-40.E-3	0.0	0.0	1.5	0.0	-1.0	DEUTERON

Ici, nous définissons un faisceau circulaire de diamètre 3 cm de deutons de 40 MeV.

	WHAT (1)	WHAT (2)	WHAT (3)	WHAT (4)	WHAT (5)	WHAT (6)	SDUM
BEAM	-50.E-3	0.0	0.0	3.0	3.0	1.0	PROTON

Ici, nous définissons un faisceau carré de 3 cm de côté en x et y de protons de 50 MeV.

Note : Attention donc au signe de **WHAT(1)** de la carte **BEAM**, ce ne sont pas les mêmes valeurs !

La quantité de mouvement s'écrit $p = m * v$ et l'énergie cinétique s'écrit $E_c = \frac{1}{2} * m * v^2$

On a donc la relation $p = \sqrt{2 * m * E_c}$

Exemple avec un faisceau d'Hélium 4 : $m({}_2^4\text{He}) = 4,00260 \text{ u} = 3728,4219 \text{ MeV}$

Si l'on donne à ce faisceau 5 MeV/c de quantité de mouvement moyenne, l'énergie cinétique moyenne sera $E_c = \frac{p^2}{2 * m} = 3,35 \text{ keV}$ environ !

BEAMPOS définit les coordonnées du centre du faisceau, soit le point à partir duquel le transport commence, ainsi que la direction.

	WHAT (1)	WHAT (2)	WHAT (3)	WHAT (4)	WHAT (5)	WHAT (6)	SDUM
BEAMPOS	0.0	0.0	1.0				

- **WHAT(1)** : coordonnée en X du centre du faisceau. **Par défaut = 0.0**
- **WHAT(2)** : coordonnée en Y du centre du faisceau. **Par défaut = 0.0**
- **WHAT(3)** : coordonnée en Z du centre du faisceau. **Par défaut = 0.0**
- **WHAT(4)** : cosinus directeur du faisceau par rapport à l'axe X. **Par défaut = 0.0**
- **WHAT(5)** : cosinus directeur du faisceau par rapport à l'axe Y. **Par défaut = 0.0**
- **WHAT(6)** : non usité
- **SDUM = NEGATIVE** signifie que le cosinus directeur par rapport à l'axe Z est négatif.

Par défaut, le faisceau est dirigé dans le sens des Z positifs, SDUM est laissé vide.

Dans notre exemple ci-dessus, la source a pour coordonnées (0, 0, 1).

Tous les paramètres décrits précédemment peuvent être « écrasés » par l'emploi d'une variable adéquate pour chaque paramètre dans le cadre de l'utilisation de la carte **SOURCE** dans le cas où les propriétés du faisceau à décrire sont trop compliquées pour être correctement définies à l'aide des cartes **BEAM** et **BEAMPOS** seules (*cf. manuel*).
Reportez-vous également au chap. 10. du présent document.

Note : Dans la mesure du possible, il vaut mieux éviter de définir l'origine du faisceau sur une limite entre deux régions, cela peut causer des problèmes de transport et des crashes intempestifs (*c'est du vécu !*).

À partir de la version 2011.2, il est désormais possible de définir directement des paramètres supplémentaires quant à la description du faisceau grâce à l'ajout d'une seconde carte **BEAMPOS**.

Cette seconde carte doit comporter un identifiant de la forme de la distribution, dans **SDUM**.

⇒ Si **SDUM = SPHE-VOL**, la distribution sera de forme coquille sphérique centrée en (x,y,z) définis dans une autre carte **BEAMPOS** (avec **SDUM** = laissé vide ou **NEGATIVE**).

- **WHAT(1)** :
 - Si $>$ ou $= 0$: rayon en cm de la sphère interne définissant la coquille.
 - Si < 0 : valeur par défaut.
 - Par défaut : 0.0 cm.
- **WHAT(2)** :
 - Si $>$ ou $= 0$: rayon en cm de la sphère externe définissant la coquille.
 - Si < 0 : valeur par défaut.
 - Par défaut : 1.0 cm.

⇒ Si **SDUM = CYLI-VOL**, la distribution sera de forme coquille cylindrique centrée en (x,y,z) définis dans une autre carte **BEAMPOS** (avec **SDUM** = laissé vide ou **NEGATIVE**) et sa hauteur sera parallèle à l'axe des z dans le référentiel d'origine.

- **WHAT(1)** :
 - Si $>$ ou $= 0$: rayon en cm du cylindre interne définissant la coquille.
 - Si < 0 : valeur par défaut.
 - Par défaut : 0.0 cm.
- **WHAT(2)** :
 - Si $>$ ou $= 0$: rayon en cm du cylindre externe définissant la coquille.
 - Si < 0 : valeur par défaut.
 - Par défaut : 1.0 cm.
- **WHAT(3)** :
 - Si $>$ ou $= 0$: hauteur en cm du cylindre interne définissant la coquille.
 - Si < 0 : valeur par défaut.
 - Par défaut : 0.0 cm.
- **WHAT(4)** :
 - Si $>$ ou $= 0$: hauteur en cm du cylindre externe définissant la coquille.
 - Si < 0 : valeur par défaut.
 - Par défaut : 1.0 cm.

⇒ Si **SDUM = CART-VOL**, la distribution sera de forme coquille parallélépipédique (cartésienne) centrée en (x,y,z) définis dans une autre carte **BEAMPOS** (avec **SDUM** = laissé vide ou **NEGATIVE**) et ses côtés seront parallèles aux axes du référentiel d'origine.

- **WHAT(1)** :
 - Si $>$ ou $= 0$: longueur en cm définissant le côté en x de la coquille interne.
 - Si < 0 : valeur par défaut.
 - Par défaut : 0.0 cm.
- **WHAT(2)** :

- Si $>$ ou $= 0$: longueur en cm définissant le côté en x de la coquille externe.
 - Si < 0 : valeur par défaut.
 - Par défaut : 1.0 cm.
 - **WHAT(3) :**
 - Si $>$ ou $= 0$: longueur en cm définissant le côté en y de la coquille interne.
 - Si < 0 : valeur par défaut.
 - Par défaut : 0.0 cm.
 - **WHAT(4) :**
 - Si $>$ ou $= 0$: longueur en cm définissant le côté en y de la coquille externe.
 - Si < 0 : valeur par défaut.
 - Par défaut : 1.0 cm.
 - **WHAT(5) :**
 - Si $>$ ou $= 0$: longueur en cm définissant le côté en z de la coquille interne.
 - Si < 0 : valeur par défaut.
 - Par défaut : 0.0 cm.
 - **WHAT(6) :**
 - Si $>$ ou $= 0$: longueur en cm définissant le côté en z de la coquille externe.
 - Si < 0 : valeur par défaut.
 - Par défaut : 1.0 cm.
- ⇒ Si **SDUM = FLOOD**, la distribution sera définie sur une surface sphérique centrée en (x,y,z) définis dans une autre carte **BEAMPOS** (avec **SDUM** = laissé vide ou **NEGATIVE**) de sorte à produire une distribution uniforme et isotrope **dans** la sphère.
- **WHAT(1) :**
 - Si $>$ ou $= 0$: rayon de la sphère en cm.
 - Si < 0 : valeur par défaut.
 - Par défaut : $1/\sqrt{\pi}$ cm.

Rappel : en l'absence complète de carte **BEAMPOS**, le faisceau démarre en $(0,0,0)$ et est dirigé selon le sens des z positifs.

Note : Ces possibilités peuvent avantageusement remplacer certaines décrites dans le **chapitre 10** consacré à l'utilisation de l'option source et l'édition de la routine associée, pour peu que la situation à décrire puisse coïncider avec les axes du référentiel lorsque ceux-ci interviennent (cas de la distribution cartésienne et cylindrique).

4. La géométrie combinatoire

4.1. Présentations et recommandations

La première étape du travail consiste en l'étude géométrique du problème. Il s'agit de modéliser la situation réelle à l'aide de formes simples ou plus complexes. Selon l'étude à effectuer et ses impératifs de précision quant au résultat final, cette géométrie sera plus ou moins élaborée. Il semble préférable de commencer par un modèle ne comportant pas tous les détails mais l'essentiel permettant de déterminer un ordre de grandeur cohérent du calcul

à effectuer. Il sera encore temps, par la suite, d'affiner le modèle en tenant compte de telle caractéristique géométrique de la situation réelle négligée dans un premier temps.

La séquence d'entrée doit être complètement contenue entre les cartes **GEOBEGIN** et **GEOEND** (nous y reviendrons ultérieurement).

Il y a deux concepts fondamentaux en géométrie combinatoire : les corps (**bodies**) et les régions (**regions**). En plus des formes finies, l'utilisation de formes infinies (cylindre, plans) est encouragée car elle favorise un transport plus précis et moins susceptible de provoquer des erreurs.

Les régions sont définies comme la combinaison de corps obtenue par des opérations booléennes : union, soustraction, intersection.

Chaque région n'est pas forcément définie de façon simple, elle peut être constituée de deux ou plusieurs parties non contiguës, mais doivent être homogènes dans leur composition (même matériau).

Région « Trou Noir »

Il est préférable (c'est très fortement conseillé) que la géométrie soit entourée d'une région « trou noir », autrement dit à laquelle on assigne le matériau 1.0 dans le code prédéfini de **FLUKA**, soit « blackhole » ou « external vacuum » afin de « tuer » toute particule pénétrant dans la zone par absorption. Cette région doit impérativement être constituée d'un volume « fermé », sphère, parallélépipède, par exemple. Il est recommandé de lui donner des dimensions beaucoup plus grandes que le minimum requis pour ne pas interférer avec d'éventuelles évolutions de la géométrie par la suite.

Important : tout point de l'espace, compris à l'intérieur de la zone « trou noir » doit être compris dans une région. La géométrie ne sera pas validée par **ALIFE** et donc par **FLUKA** si une zone reste indéterminée, si des points appartiennent à deux ou plusieurs régions, ou à aucune.

Il est également conseillé de créer une région constituée de vide cette fois-ci englobée dans la précédente à laquelle on assigne le matériau 2.0 de section efficace d'absorption nulle. Ceci permet un meilleur fonctionnement.

4.2. Procédure pour définir une géométrie dans FLUKA

Il s'agit d'abord de déclarer les corps formant l'espace à décrire. Pour cela, nous avons le choix entre plusieurs formes prédéfinies pour lesquelles il s'agit simplement de donner les caractéristiques nécessaires.

L'ordre des séquences est le suivant, et doit être respecté :

GEOBEGIN card

Geometry title : un titre qui explicite ce qui suit

Body data : la liste des corps, formes géométriques utilisées

END card

Region data : une suite d'opérations booléennes associant les corps précédents

END card

LATTICE cards (*optionnel*)

Regions volumes (*optionnel*)

GEOEND card

4.2.1. La carte GEOBEGIN

- **WHAT(1)** : drapeau pour imprimer les messages d'erreurs
Différent de 0.0 : seul un résumé global est donné
Par défaut = 0.0 : tous les messages d'erreurs de géométrie sont imprimés.

- **WHAT(2)** : paramètre de précision, réservé au développement du code
- **WHAT(3)** : unité logique d'où la géométrie est lue. Si > 0.0 et différent de l'unité d'entrée standard **LUNIN** (normalement = 5), le nom du fichier correspondant doit être entré à la carte suivante. Dans le cas contraire, l'entrée de la géométrie suit.
- **WHAT(4)** : unité logique vers laquelle le fichier de sortie est dirigé. Si > 0.0 et différent de l'unité de sortie standard **LUNOUT** (normalement = 11), le nom du fichier correspondant doit être entré à la carte suivante. Dans le cas contraire, le fichier de sortie de la géométrie est imprimé dans le fichier de sortie standard.
- **WHAT(5), WHAT(6)** : non usités
- **SDUM** : vide = **COMBINAT** ou **COMBNAME**

Avec la version 2006.3, nous avons la possibilité d'utiliser COMBNAME qui utilise la géométrie combinatoire en format libre, des noms peuvent être employés pour les corps et les régions. Nous ne détaillons pas cette option, le principe étant le même. Il faut faire la correspondance entre les associations de corps décrivant chaque région. À l'usage, il semble que cette option soit plus conviviale.

4.2.2. La carte de titre de la géométrie

3 variables sont entrées sur cette carte dont le format est : **215, 10X, A60**

Le premier entier IVOPT (Input Volume OPTion) est un drapeau indiquant comment normaliser les quantités obtenues dans les différentes régions par l'option **SCORE** (cf. *manuel*).

4.2.3. Body data

Liste des formes géométriques utilisées (cf. *manuel*).

Il s'agit tout simplement d'entrer la nature et les coordonnées demandées pour chaque forme voulue en respectant le format d'entrée (il peut différer d'une forme à une autre). Encore une fois, il est préférable d'avoir étudié son problème sur le papier auparavant afin d'éviter les mauvaises surprises et de devoir recommencer.

Prenons un exemple pour le format « fixé » par défaut (simple précision) : **2X, A3, I5, 6D10.3**

```
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
GEOBEGIN
                                Irradiateur cylindrique
RPP 1  -9999.0  +9999.0  -9999.0  +9999.0  -9999.0  +9999.0
RPP 2  -900.0   +900.0   -900.0   +900.0   -900.0   +900.0
RPP 3  -32.7   +32.7    -32.7    +32.7    -12.7    +77.7
RPP 4  -32.5   +32.5    -32.5    +32.5    -12.5    +77.5
RPP 5  -32.0   +32.0    -32.0    +32.0    -12.0    +65.0
ZCC 6  0.0      0.0       20.0
XYP 7  0.0
TRC 8  0.0      0.0       0.0      0.0      0.0      25.0
      8.25    20.0
XYP 9  25.0
END
```

4.3. Description de quelques formes

- Le code **RPP** représente un parallélépipède rectangle dont les arêtes sont parallèles aux axes de coordonnées. Il est défini par 6 nombres. Dans l'ordre : **Xmin, Xmax, Ymin, Ymax, Zmin, Zmax**, coordonnées minimales et maximales.
- Le code **ZCC** représente un cylindre infini parallèle à l'axe Z. Ses pendants pour les axes X et Y sont **XCC** et **YCC**. Il est défini par trois nombres A_x, A_y (pour **ZCC**), A_y, A_z (pour **XCC**), A_x, A_z (pour **YCC**), les coordonnées de l'axe du cylindre et R, le rayon.

- Le code **RCC** (Right Circular Cylinder) représente un cylindre pouvant avoir n'importe quelle orientation dans l'espace. Il est délimité par une surface cylindrique et deux faces planes perpendiculaires à son axe. Si l'axe du cylindre est à l'un des axes de coordonnées, il est préconisé d'employer un cylindre infini (**XCC**, **YCC** ou **ZCC**) afin d'optimiser la vitesse de transport des particules. *Néanmoins, à l'usage, personnellement, je préfère définir des « RCC » plutôt que des cylindres infinis délimités par des plans eux-mêmes infinis. La durée de votre simulation sera peut-être plus longue, mais la description de la géométrie sera beaucoup plus simple (avis personnel qui n'engage que moi).*

Un **RCC** est caractérisé par 7 nombres : **Vx, Vy, Vz**, les coordonnées du centre de l'une des faces circulaires, **Hx, Hy, Hz**, composantes en x, y et z d'un vecteur correspondant à la longueur du cylindre pointant vers l'autre cercle et **R**, le rayon.

Exemples :

```
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
RCC      5      0.0      0.0      -0.2      0.0      0.0      -11.4      6.3
```

Le corps n° 5 est un cylindre parallèle à l'axe des z, de centre (0, 0, -0,2), de longueur 11,4 cm vers le sens des z négatifs et de rayon 6,3 cm.

```
RCC      9      3.2  5.542562      -6.8      0.35  0.606218      0.0      1.0
```

Le corps n° 9 est un cylindre de rayon 1 cm, de centre (3,2, 5,542562, -6,8), de longueur 0,7 cm et son axe fait un angle 60° dans le plan (x, y).

- Le code **XYP** désigne un plan infini sur un demi-espace délimité par un plan perpendiculaire à l'axe de coordonnées Z.

Délimité par un plan perpendiculaire à l'axe X : code **YZP**

Délimité par un plan perpendiculaire à l'axe Y : code **XZP**

Chaque demi-espace est défini par un seul nombre.

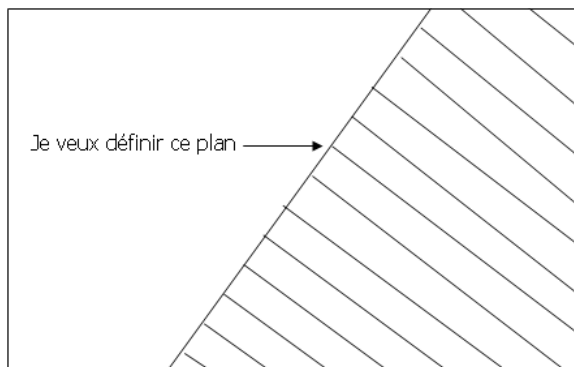
Vx (pour **YZP**), **Vy** (pour **XZP**), **Vz** (pour **XYP**), la coordonnée du plan sur l'axe correspondant.

Le demi-espace à l'« intérieur du corps » est l'ensemble des points dont $x < V_x$ (**YZP**), $y < V_y$ (**XZP**), $z < V_z$ (**XYP**), selon le cas de figure.

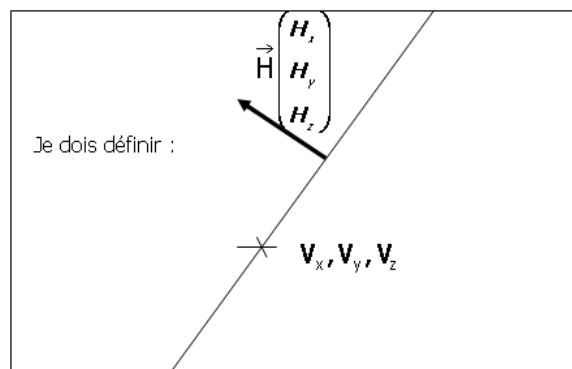
- Le code **PLA** désigne un plan général couvrant un demi-espace.

Il est défini par 6 nombres : **Hx, Hy, Hz**, composantes en x, y et z d'un vecteur de longueur arbitraire perpendiculaire au plan, et **Vx, Vy, Vz**, coordonnées d'un point quelconque appartenant au plan. Le demi-espace compris dans la définition de ce corps (signe + donc) est celui duquel pointe le vecteur (ils pointent vers l'extérieur).

Si je veux définir ce plan (avec le demi-espace en partie hachurée) :



Je dois définir :



Avec $\vec{H} = \begin{pmatrix} H_x \\ H_y \\ H_z \end{pmatrix}$ vecteur perpendiculaire pointant à l'extérieur du plan et V_x, V_y, V_z ,

les coordonnées d'un point appartenant au plan.

Exemples :

```
PLA 39      0.0      1.0      1.0      200.0      -300.0      240.0
```

Le corps n° 39 désigne tous les points compris « sous » un plan à 45° dans le plan (y, z) qui passe par le point x = 200, y = -300 et z = 240. Notez que pour un tel plan la valeur en x n'est pas du tout significative.

```
PLA 30      +0.866025403784439      -0.5      0.0
              -95.0      0.0      0.0
```

Le corps n° 30 (décrit en double précision) désigne tous les points compris « sous » un plan à (90 - 30 = 60 °) dans le plan (x, y) qui passe par le point x = -95, y = 0 et z = 0.

- Le code **TRC** désigne un cône tronqué d'angle droit.

Un **TRC** peut avoir toute orientation dans l'espace. Il est limité par une surface conique et deux faces circulaires, perpendiculaires à l'axe du cône.

Il est défini par 8 nombres : **Vx, Vy, Vz**, coordonnées du centre de la base principale, **Hx, Hy, Hz**, composantes d'un vecteur correspondant à la hauteur de **TRC**, dirigé de la base principale à la base secondaire, **R1** (rayon de la base principale) et **R2** (rayon de la base secondaire).

Un cône **TRC** est toujours défini sur deux cartes pour le format fixé par défaut.

Remarque : dans la version originale du manuel, on parle de « *major* » base et « *minor* » base. Cela pourrait laisser sous-entendre une notion de taille : grande base, petite base. Mais la taille des bases importe peu en l'occurrence, ce que l'on définit c'est bien le sens, car le cône est dirigé de la base « **principale** » (major) vers la base « **secondaire** » (minor). Dans le premier exemple, la base principale a pour coordonnées (0, 0, 0), la base secondaire a pour coordonnées (0, 0, 25), R1 = 8,25 cm et R2 = 20,0 cm.

```
TRC 8      0.0      0.0      0.0      0.0      0.0      25.0
          8.25      20.0
```

Dans ce second exemple (décrit en double précision), la base principale a pour coordonnées (543, -1757, 300), le cône est défini dans le plan (x,y) par une longueur de 100 cm et un angle de 30°. Les rayons sont respectivement de 171,5 et 100 cm.

```
TRC 300      543.0      -1757.0      300.0
          -86.602540378443865      50.0      0.0
              171.5      100.0
```

- Le code **ELL** désigne une forme ellipsoïde que l'on peut obtenir en faisant tourner une ellipse autour de son axe principal et pouvant avoir toute orientation dans l'espace. Chaque **ELL** est défini par 7 nombres : **Fx(1), Fy(1), Fz(1), Fx(2), Fy(2), Fz(2)**, les coordonnées des deux foyers sur l'axe principal, et **L** (la longueur totale de l'axe principal).

Exemple :

```
* . . . . . 1 . . . . . 2 . . . . . 3 . . . . . 4 . . . . . 5 . . . . . 6 . . . . . 7 . .
ELL      003      -400.0      0.0      0.0      400.0      0.0      0.0
                1000.
```

Une ellipsoïde obtenu par révolution autour de l'axe X d'une ellipse centre à l'origine, avec son axe principal parallèle à X de longueur 1 000 cm et de petit axe de longueur 600 cm.

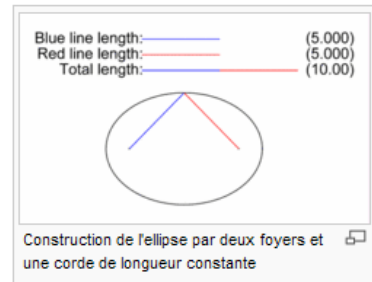
Un peu de géométrie :

Définition bifocale de l'ellipse [modifier]

Soient F et F' deux points distincts du plan. On appelle ellipse de foyers F et F' , l'ensemble des points M du plan vérifiant la propriété suivante :

$$d(M, F) + d(M, F') = 2a = 2\sqrt{c^2 + b^2} \quad a \in \mathbb{R}_+^*, \quad b \in \mathbb{R}_+^*$$

où $2a$ est la longueur du grand axe, $2c = d(F, F')$ et $2b$ est la longueur du petit axe (perpendiculaire au grand axe). Cette relation exprime que la somme des distances d'un point M aux foyers est constante et vaut la longueur du grand axe.



Dans la réalité, si l'on part d'un plan, il est possible de connaître les longueurs des petit et grand axes ($2a$ et $2b$). On en déduit $2c$, la distance séparant les deux foyers et les coordonnées choisies pour ces deux foyers. On aura ainsi les données nécessaires à la description d'une telle forme dans **FLUKA**.

$c = (a - b) * (a + b)$ et la distance entre deux points $F(x, y, z)$ et $F'(x', y', z')$ est donnée par $\sqrt{(x' - x)^2 + (y' - y)^2 + (z' - z)^2}$

- Le code **SPH** désigne une sphère, définie par trois nombres : les coordonnées de son centre (x, y, z) et son rayon R .

Ne pas oublier de terminer la liste d'entrée des corps par une carte **END**.

4.4. Les régions et la géométrie combinatoire

4.4.1. Region data

La création des régions par association des corps (cf. manuel).

Les différentes régions sont décrites en termes de différences, d'intersections et d'unions de corps. Comme pour la description des corps, l'utilisateur a le choix entre plusieurs formats (un libre et deux formats fixés).

Nous ne détaillerons que le format fixé simple précision, celui par défaut :

- Chaque région est décrite par une combinaison d'un ou plusieurs corps par l'un des trois opérateurs -, +, **OR** en référence aux opérations booléennes de soustraction (complément), d'intersection et d'union.
- Chaque corps est identifié par le numéro qu'il porte dans la séquence d'entrée.

Le format, lorsque toute entrée est en format fixé par défaut : **(2X, A3, I5, 9 (A2, I5))**

- Les 3 caractères alphanumériques en colonnes 3 à 5 sont à la discrétion de l'utilisateur pour identifier sa région, il peut également laisser des numéros.
- Les entiers des colonnes 6 à 10 : c'est le nombre de régions dans lesquelles peut pénétrer une particule quittant la région décrite.

Ce nombre est utilisé pour la mémoire, il n'est pas essentiel qu'il soit exact (par défaut, il est fixé à 5). Tout nombre est accepté mais si la somme finale de ces entiers est proche de la somme réelle, la vitesse du transport peut être légèrement augmentée.

- Les colonnes 11 à 73 : il s'agit d'alterner autant de champs **OR** ou laissés vides (blancs) et de champs **+**, **-**, numéro de corps que nécessaires pour compléter la description de cette région. Si une ligne ne suffit pas, d'autres peuvent suivre, identifiées par un champ vide dans les colonnes 3 à 5.

4.4.2. Signification des opérateurs

Si un numéro de corps est précédé d'un opérateur **+**, cela signifie que la zone décrite est complètement contenue à **l'intérieur** du corps.

Si un numéro de corps est précédé d'un opérateur **-**, cela signifie que la zone décrite est complètement contenue à **l'extérieur** du corps.

De toute évidence, le symbole **+** doit apparaître au moins une fois dans toute description de région.

Quelquefois, une région peut être décrite en termes de sous-régions collées ensemble. L'opérateur **OR** est utilisé pour combiner ces sous-régions, pouvant partiellement se chevaucher ou non. Les sous-régions sont formées comme expliqué ci-dessus, la région est ensuite formée par une union de ces sous-régions. Ces dernières incluent tout numéro de corps jusqu'au **OR** suivant ou à la fin de la description de la région.

Rappel : en ce qui concerne les plans semi infinis, un signe **+**, soit l'intérieur du corps, est défini comme étant l'ensemble des points de coordonnées inférieures à la valeur utilisée pour définir ce plan. Au contraire, un signe **-**, soit l'extérieur du corps, est défini comme étant l'ensemble des points de coordonnées supérieures à la valeur utilisée pour définir ce plan.

- Exemple 1 : Un corps n° 23 défini par un plan **XYP** avec $z = 12,5$
+23 inclut tous les points de coordonnées $z < 12,5$
-23 inclut tous les points de coordonnées $z > 12,5$

- Exemple 2 : La suite du cas décrit dans le paragraphe **4.2.**, cette séquence suit immédiatement la carte **END** de la séquence d'entrée des corps.

```
*XAAA.....OR.....OR.....OR.....OR.....OR.....OR.....OR.....OR.....OR.....
  1      5      1      -2
  2      5      2      -3
  3      5      3      -4
  4      5      4      -5
  5      5      5      -6      -8
  6      5      5      6      -8      -9
  7      5      5      8
  8      5      6      -8      9      -7
  9      5      6      -8      7
END
GEOEND
```

- La région 1 est celle du trou noir, elle est constituée du premier corps à l'exception du second.
- La région 2, qui sera considérée comme vide, est constituée du second corps hormis ce qui se trouve à l'intérieur du troisième corps. C'est dans ce dernier que sera contenue la forme qui nous intéresse, le modèle cylindrique d'un irradiateur.
- La région 3 sera l'enveloppe d'inox de très faible épaisseur (2 mm) entourant l'irradiateur. Elle est définie comme le volume constitué par la différence entre les corps 3 et 4.

- La région 4 sera la première partie de l'enveloppe de plomb constituée par la différence entre les formes 4 et 5. Jusqu'à présent, tous les corps considérés sont des parallélépipèdes rectangles.
- La région 5 est la seconde partie de l'enveloppe de plomb, on soustrait le cône (corps n° 8) et le cylindre (corps n° 6) du parallélépipède (corps n° 5).
- La région 6 est la première partie de l'intérieur de l'irradiateur, la partie cylindrique. Elle est définie par l'intersection entre les corps n° 5 (parallélépipède) et n° 6 (cylindre) et la soustraction avec le corps n° 8 (cône) et le corps n° 9 (plan **XYP**). Cette région est donc un cylindre de rayon 20 centré sur l'axe Z et avec z compris entre 25 (le plan **XYP**) et la limite du parallélépipède n° 5, soit 65.
 - La région 7 est l'autre partie de l'irradiateur, la partie conique. Elle est définie comme l'intersection entre les corps n° 5 (parallélépipède) et n° 8 (cône).
 - Les deux dernières régions sont constituées par les points non encore définis par notre géométrie. Or, nous avons dit précédemment que tout point à l'intérieur de la région « trou noir » doit être défini, c'est-à-dire doit appartenir à une région.
 - ✓ La région 8 est constituée de l'intersection entre les corps n° 6 et 9 et soustraction avec les corps n° 8 et 7. Autrement dit, c'est la partie du cylindre (+6) dont les coordonnées en Z sont inférieures à 25 (+9), supérieures à 0 (-7) et à l'extérieur du cône (-8).
 - ✓ La région 9 est constituée par la partie du cylindre (+6) située à l'extérieur du cône (-8) et dont les coordonnées en Z sont inférieures à 0 (+7).

Ceci donne la figure suivante, option limites des matériaux, dans le plan (z, y).

Coordonnées du tracé :

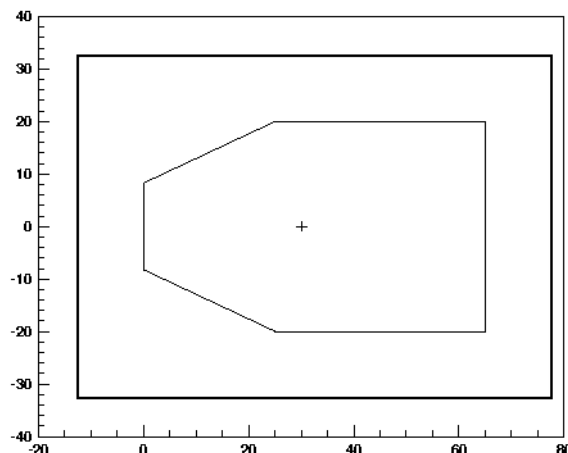
Xmin : 0.0, Ymin : -40.0, Zmin : -20.0, Xmax : 0.0,

Ymax : 40.0, Zmax : 80.0

TXX : 0.0, TXY : 0.0, TXZ : 1.0,

TYX : 0.0, TYY : 1.0, TYZ : 0.0

Note : **TXX**, **TXY**, **TXZ** sont les cosinus directeurs de l'axe des abscisses, respectivement en X, Y et Z. De même **TYX**, **TYY**, **TYZ** sont les cosinus directeurs de l'axe des ordonnées respectivement en X, Y et Z.



À partir de la version 2011.2, des transformations concernant la géométrie, appliqués aux corps directement sont possibles. Retenons l'expansion (facteur multiplicatif ou diviseur), la translation, la roto-translation (à l'aide de la carte **ROT-DEFI** décrite dans la partie consacrée aux réseaux (lattices, cf. annexe F.)

Les directives concernant ces transformations ont des commandes spéciales encadrées entre deux lignes du type \$Start_xxx \$End_xxx

Où "xxx" est soit "expansion", "translat" or "transform". Elles permettent d'obtenir respectivement une expansion des coordonnées, une translation des coordonnées, une roto-translation des coordonnées des corps compris entre les directives des lignes "start" et "end". Les descriptions des corps comprises entre les lignes "\$Start_xxx" and "\$End_xxx" sont automatiquement modifiées selon la syntaxe adéquate.

On se reportera au manuel pour de plus amples informations (section géométrie).

5. Les matériaux

Il s'agit à présent de définir les matériaux utilisés pour cette simulation et de les assigner aux régions.

5.1. Définition des matériaux

La carte **MATERIAL** définit un matériau constitué d'un seul élément ou un composé si cette carte est couplée avec une carte **COMPOUND**.

- **WHAT(1)** = numéro atomique (n'a de sens que s'il s'agit d'une carte non couplée avec une carte **COMPOUND**, sinon **WHAT(1) = 0.0**).
- **WHAT(2)** = masse atomique (n'a de sens que s'il s'agit d'une carte non couplée avec une carte **COMPOUND**, sinon **WHAT(2) = 0.0**).
- **WHAT(3)** = masse volumique en g/cm³. Si cette valeur est inférieure à 0,01, le matériau est considéré comme étant un gaz à la pression atmosphérique.
- **WHAT(4)** = numéro (index) du matériau.

Par défaut = NMAT+1, où NMAT est le dernier numéro des matériaux prédéfinis.

Sa valeur est 25 avant qu'une carte **MATERIAL** soit définie et ne change pas si **WHAT(4)** écrase un numéro qui a déjà été assigné.

- **WHAT(5)** : non usité.
 - **WHAT(6)** = nombre de masse du matériau. Cette valeur est laissée à 0.0 à moins qu'un isotope en particulier ne soit requis.
 - **SDUM** = nom du matériau. Il doit correspondre à celui tabulé dans le manuel de **FLUKA**, dans le cas contraire certaines données (celles utilisées pour les interactions avec les neutrons par exemple) ne seront pas trouvées par le code.
- Le numéro **1.0** est toujours « Blackhole », appelé également vide externe et ne peut être redéfini. Toutes les particules disparaissent lorsqu'elles atteignent le trou noir qui a une section efficace d'absorption infinie.
 - Le numéro **2.0** est toujours « Vacuum », soit le vide et ne peut être redéfini. Ici, la section efficace d'absorption est nulle.
 - Tous les autres matériaux peuvent être redéfinis par l'utilisateur.

Exemple : les matériaux constitués d'éléments simples :

*23456789	123456789	123456789	123456789	123456789	123456789	123456789
MATERIAL	1.0	1.0079	0.0000899	3.0		HYDROGEN
MATERIAL	6.0	12.011	2.0	4.0		CARBON
MATERIAL	7.0	14.0067	0.0012506	5.0		NITROGEN
MATERIAL	8.0	15.9994	0.001429	6.0		OXYGEN
MATERIAL	11.0	22.9898	0.971	7.0		SODIUM
MATERIAL	12.0	24.305	1.738	8.0		MAGNESIU
MATERIAL	13.0	26.982	2.6989	9.0		ALUMINUM
MATERIAL	14.0	28.0855	2.330	10.0		SILICON
MATERIAL	16.0	32.0650	1.960	11.0		SULFUR
MATERIAL	18.0	39.948	0.0016629	12.0		ARGON
MATERIAL	19.0	39.0983	0.862	13.0		POTASSIU
MATERIAL	20.0	40.0780	1.550	14.0		CALCIUM
MATERIAL	24.0	51.9961	7.140	15.0		CHROMIUM
MATERIAL	26.0	55.847	7.877	16.0		IRON
MATERIAL	28.0	58.6934	8.908	17.0		NICKEL
MATERIAL	29.0	63.546	8.92	22.0		COPPER
MATERIAL	82.0	207.2	11.35	18.0		LEAD
MATERIAL	92.0	238.02891	19.05	24.0		238-U

La carte **COMPOUND** définit un composé ou une mixture, constitué de plusieurs matériaux ou même une mixture de divers isotopes.

- Si **WHAT(1) > 0.0** et **WHAT(2) > 0.0** :
 - **WHAT(1)** = nombre d'atomes relatifs du premier matériau dans le composé
 - **WHAT(2)** = index du premier matériau
- Si **WHAT(1) < 0.0** et **WHAT(2) > 0.0** :
 - **Valeur absolue de WHAT(1)** = fraction massique du premier matériau du composé
 - **WHAT(2)** = index du premier matériau
- Si **WHAT(1) < 0.0** et **WHAT(2) < 0.0** :
 - **Valeur absolue de WHAT(1)** = fraction volumique du premier matériau du composé
 - **Valeur absolue de WHAT(2)** = index du premier matériau

Il n'y a pas de valeurs par défaut.

De façon similaire, **WHAT(3)** et **WHAT(4)** sont consacrés au second matériau du composé, **WHAT(5)** et **WHAT(6)** au troisième.

- **SDUM** : nom du composé

Pour des composés de plus de trois matériaux, il est possible d'utiliser autant de cartes qu'il est nécessaire avec le même **SDUM** (avec une limite à 80 composants par composé et le nombre des composants est limité à 1000).

Exemple :

```
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
MATERIAL                                7.84      19.0                                INOX
COMPOUND  -0.78000      16.0  -0.15000      15.0  -0.07000      17.0INOX
*
MATERIAL                                0.001205    20.0                                AIR
COMPOUND  -0.75500      5.0  -0.23200      6.0  -0.01300      12.0AIR
*
MATERIAL                                2.35      21.0                                CONCRETE
COMPOUND  -0.00560      3.0  -0.49830      6.0  -0.01710      7.0CONCRETE
COMPOUND  -0.00240      8.0  -0.04560      9.0  -0.31580      10.0CONCRETE
COMPOUND  -0.00120     11.0  -0.01920     13.0  -0.08260     14.0CONCRETE
COMPOUND  -0.01220     16.0                                CONCRETE
```

- Nous avons défini l'inox (matériau n° 19 dans notre problème), de densité 7,84, composé (en fractions massiques) de 78 % de fer (matériau n° 16), de 15 % de chrome (matériau n° 15) et de 7 % de nickel (matériau n° 17).
- Nous avons également défini l'air (matériau n° 20 dans notre problème), de densité 0.001205, composé (en fractions massiques) de 75,5 % d'azote (matériau n° 5), de 23,2 % d'oxygène (matériau n° 6) et de 1,3 % d'argon (matériau n° 12).
- De la même façon, le béton (concrete), matériau n° 21, de densité 2,35 est composé de 10 matériaux (H : 0,56 %, O : 49,83 %, Na : 1,71 %, Mg : 0,24 %, Al : 4,56 %, Si : 31,58 %, S : 0,12 %, K : 1,92 %, Ca : 8,26 %, Fe : 1,22 %).

Ces cartes doivent évidemment être précédées de celles définissant les matériaux simples (exemple précédent).

5.2. L'assignation des matériaux aux régions

La carte **ASSIGNMAT**

Cette carte définit la correspondance entre les indices de régions et les indices de matériaux.

- **WHAT(1)** = index du matériau ou nom utilisé dans la description
- **WHAT(2)** = limite inférieure des indices de région dont l'index de matériau vaut **WHAT(1)**
- **WHAT(3)** = limite supérieure des indices de région dont l'index de matériau vaut **WHAT(1)**
- **WHAT(4)** = longueur du pas dans l'assignation des indices (par défaut 1.0)
- **WHAT(5)** = 1.0 : un champ magnétique est présent dans la(les) région(s) définie(s) par **WHAT(2), (3) et (4)**.
 = 0.0 : ignoré
 < 0.0 : réinitialise les valeurs par défaut (pas de champ) dans la(les) région(s) définie(s) par **WHAT(2), (3) et (4)**.

Se référer au manuel pour tous les autres combinaisons possibles (incluant la présence de champs électrique et magnétique à considérer alternativement pour les radiations prompte ou de décroissance radioactive).

- **WHAT(6)** = index du matériau ou nom du matériau, pour une potentielle alternative dans le cas du transport de produits de la décroissance radioactive (cas d'objets activés déplacés de leur emplacement d'origine et entreposés ailleurs que sur le lieu même de l'irradiation). Seuls vacuum et blackhole sont autorisés pour cette alternative, et dans ce cas, la région à laquelle cette carte est liée ne sera pas considérée dans le calcul de la décroissance radioactive. Par défaut = le même que pour la radiation prompte (cf. **WHAT(1)** et **chapitre 11** de ce document).
- **SDUM** : non usité.
- **ATTENTION** : À partir de la version 2011.2, il n'y a plus de matériau par défaut, il faut attribuer un matériau à toute région du système défini (même vacuum ou blackhole), le code s'arrêtera sinon.

Exemple :

```
*23456789 123456789 123456789 123456789 123456789 123456789 123456789
* External Black Hole
ASSIGNMAT      1.0      1.0 (ou ASSIGNMAT BLCKHOLE  1.0)
* Vacuum
ASSIGNMAT      2.0      2.0 (ou ASSIGNMAT VACUUM    2.0)
* inox
ASSIGNMAT     19.0      3.0 (ou ASSIGNMAT INOX      3.0)
* plomb
ASSIGNMAT     18.0      4.0
ASSIGNMAT     18.0      5.0
ASSIGNMAT     18.0      8.0
ASSIGNMAT     18.0      9.0
* air
ASSIGNMAT     20.0      6.0
ASSIGNMAT     20.0      7.0
```

Nous retrouvons ce que nous disions déjà lors de la partie consacrée à la géométrie combinatoire.

- La région 1 est celle qui entoure la géométrie de tout le problème et permet un bon fonctionnement, toute particule y pénétrant disparaît, c'est le « Blackhole », on lui assigne donc le matériau prédéfini et non modifiable 1.0.
- La région 2 est celle composée de vide, également préconisée pour un bon fonctionnement du code. On lui assigne le matériau prédéfini et non modifiable 2.0.
- La région 3 est composée d'inox (matériau n° 19 ici), c'est l'enveloppe de l'irradiateur.
- Les régions 4, 5, 8 et 9 sont composées de plomb (matériau n° 18), le blindage de l'irradiateur.
- Les régions 6 et 7 constituent l'intérieur de l'irradiateur, la partie qui nous intéresse particulièrement, elles sont composées d'air.

5.3 Remarque

À partir des versions mises à jour à l'été 2007, une conséquence de l'utilisation de noms en lieu et place du numéro correspondant du matériau pour ceux définis par l'utilisateur ou pour les matériaux préenregistrés peut entraîner des conflits sur d'anciens fichiers d'entrée (crash et message d'erreur assez explicite). En effet, il peut se produire un conflit si un matériau auquel **FLUKA** a déjà attribué un numéro (l'un des 25 matériaux préenregistrés) est à nouveau défini par l'utilisateur. Il y a alors deux solutions :

- Introduire une carte **GLOBAL** juste après le titre (début du fichier d'entrée) avec **4.0** en **WHAT(4)** (*cf. manuel option GLOBAL*). Ceci force le format d'entrée sous forme numérique pour toutes les commandes d'entrée. Avec une telle option, il est impossible d'utiliser les options **COMBNAME** pour la géométrie, les noms des matériaux au lieu de leurs codes et également en ce qui concerne les options de calcul (le nom de **PHOTON** plutôt que son code **7.0** par exemple).
- Utiliser les matériaux préenregistrés dans FLUKA et compléter par les matériaux de votre choix à partir de l'indice **26.0**.

La liste est la suivante :

FLUKA name	FLUKA number	Common name	A	Z	Density (g/cm ³)
BLCKHOLE	1	Blackhole or External Vacuum	0.	0.	0.
VACUUM	2	Vacuum or Internal Vacuum	0.	0.	0.
HYDROGEN	3	Hydrogen	1.00794	1.	0.0000837
HELIUM	4	Helium	4.002602	2.	0.000166
BERYLLIU	5	Beryllium	9.012182	4.	1.848
CARBON	6	Carbon	12.0107	6.	2.000
NITROGEN	7	Nitrogen	14.0067	7.	0.00117
OXYGEN	8	Oxygen	15.9994	8.	0.00133
MAGNESIU	9	Magnesium	24.3050	12.	1.740
ALUMINUM	10	Aluminium	26.981538	13.	2.699
IRON	11	Iron	55.845	26.	7.874
COPPER	12	Copper	63.546	29.	8.960
SILVER	13	Silver	107.8682	47.	10.500
SILICON	14	Silicon	28.0855	14.	2.329
GOLD	15	Gold	196.96655	79.	19.320
MERCURY	16	Mercury	200.59	80.	13.546
LEAD	17	Lead	207.2	82.	11.350
TANTALUM	18	Tantalum	180.9479	73.	16.654
SODIUM	19	Sodium	22.989770	11.	0.971
ARGON	20	Argon	39.948	18.	0.00166
CALCIUM	21	Calcium	40.078	20.	1.550
TIN	22	Tin	118.710	50.	7.310
TUNGSTEN	23	Tungsten	183.84	74.	19.300
TITANIUM	24	Titanium	47.867	22.	4.540
NICKEL	25	Nickel	58.6934	28.	8.902

- À partir de la version 2011.2, certains matériaux composés prédéfinis d'intérêt pour les problèmes liés à la dosimétrie (ICRU) ont été rajoutés aux matériaux de la liste précédente (cf. ci-dessous). Elle est « named based », il n'y a pas de carte **MATERIAL** ou **COMPOUND** à définir du moment que l'on utilise le nom prédéfini. Si l'on veut utiliser les index numériques, il faut leur assigner un numéro via une carte **MATERIAL**, attention bien entendu à ne laisser aucun blanc dans l'assignation de ces index numériques.

FLUKA name	Common name	Density (g/cm ³)
WATER	Water	1.0
POLYSTYR	Polystyrene	1.06
PLASCINT	Plastic scintillator	1.032
PMMA	Polymethyl methacrylate, Plexiglas, Lucite, Perspex	1.19
BONECOMP	Compact bone	1.85
BONECORT	Cortical bone	1.85
MUSCLESK	Skeletal muscle	1.04
MUSCLEST	Striated muscle	1.04
ADTISSUE	Adipose tissue	0.92
KAPTON	Kapton polyimide film	1.42
POLYETHY	Polyethylene	0.94
AIR	Dry air at NTP conditions	0.00120479

6. Les estimateurs

Les estimateurs sont nombreux dans FLUKA et permettent de calculer divers paramètres. Nous nous intéresserons principalement à ceux qui permettent de calculer des fluences, des densités d'énergie déposée et donc des doses et notamment à celui qui permet de définir une région **indépendante** de la géométrie d'entrée, l'estimateur **USRBIN**.

À partir de la version 2008.3, on peut calculer directement la dose équivalente (pSv/particule primaire) grâce au code **240.0** ou **DOSE-EQ** (cf. aussi 7.4).

6.1. Description de la carte USRBIN

La carte **USRBIN** donne la distribution de plusieurs quantités dans une structure spatiale régulière (binning) indépendante de la géométrie (cf. manuel pour la description complète).

Selon les options décrites ci-dessous, cet estimateur permet de calculer la distribution spatiale de l'énergie déposée, la fluence totale, dans une maille régulière cylindrique ou cartésienne décrite par l'utilisateur.

La définition complète requiert normalement deux cartes successives (la seconde carte identifiée par le caractère « & » dans toute colonne comprise entre 71 et 78, i.e. **SDUM**), sauf si les valeurs par défaut sont acceptables pour l'utilisateur.

6.1.1. Première carte

- WHAT(1)** : c'est le code indiquant le type de « binning » sélectionné. Chaque type est caractérisé par différentes propriétés. Structure de la maille : spatiale (**R-Z**, **R-Z-Phi**), cartésienne, ou spéciale (par région, ou définie par l'utilisateur).

Retenons deux codes principalement utilisés :

10.0 : Maille : Cartésienne, pas de symétrie

Grandeur calculée : dépend de **WHAT(2)**.

11.0 : Maille : R-Z ou R-Phi-Z, pas de symétrie

Grandeur calculée : dépend de **WHAT(2)**.

• **WHAT(2)** : particule ou famille de particules intéressantes pour le calcul (code ou nom).
 Si **WHAT(2) = 208.0, 211.0, 229.0, 230.0**, il s'agit de la densité d'énergie avec le nouvel algorithme.

Si **WHAT(2) =** code de la particule (*cf. manuel*), il s'agit de la fluence de particules. Pour calculer la dose avec cette option, il faut invoquer la routine **FLUSCW** (*cf. 7. 2) du présent document*). Voir également la remarque ci-dessous.

• **WHAT(3)** : unité logique du fichier de sortie.

Si > 0.0 : données formatées

Si < 0.0 : données non formatées

• **WHAT(4)** : Pour le « binning » cartésien : **Xmax**

Pour le « binning » R-Z, R-Phi-Z : **Rmax**

• **WHAT(5)** : Pour le « binning » cartésien : **Ymax**

Pour le « binning » R-Z, R-Phi-Z : coordonnée **Y** de l'axe de « binning »

• **WHAT(6)** : Pour le « binning » cartésien : **Zmax**

Pour le « binning » R-Z, R-Phi-Z : **Zmax**

• **SDUM** : tout caractère sauf « & » identifiant le « binning » (maximum 10 caractères)

6.1.2. Seconde carte

• **WHAT(1)** : Pour le « binning » cartésien : **Xmin**

Pour le « binning » R-Z, R-Phi-Z : **Rmin**

• **WHAT(2)** : Pour le « binning » cartésien : **Ymin**

Pour le « binning » R-Z, R-Phi-Z : coordonnée **X** de l'axe de « binning »

• **WHAT(3)** : Pour le « binning » cartésien : **Zmin**

Pour le « binning » R-Z, R-Phi-Z : **Zmin**

• **WHAT(4)** : Pour le « binning » cartésien : nombre de mailles en **X**

Pour le « binning » R-Z, R-Phi-Z : nombre de mailles en **R**

• **WHAT(5)** : Pour le « binning » cartésien : nombre de mailles en **Y**

Pour le « binning » R-Z, R-Phi-Z : nombre de mailles en **Phi**

• **WHAT(6)** : Pour le « binning » cartésien : nombre de mailles en **Z**

Pour le « binning » R-Z, R-Phi-Z : nombre de mailles en **Z**

• **SDUM = &** (n'importe où entre les colonnes 71 à 78)

6.2. Exemple d'estimateur USRBIN

```
*23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789
USRBIN      10.      208.      -82.      80.      80.      80.total
USRBIN      0.       0.      -20.      100.     100.     100.&
```

6.2.1. Première carte

• **WHAT(1) = 10.0** : maille cartésienne sans symétrie.

• **WHAT(2) = 208.0** : on veut calculer la densité d'énergie déposée avec le nouvel algorithme.

• **WHAT(3) = -82.0** : données de sortie non formatées inscrites dans l'unité logique 82 (fichier de sortie de la forme : *nom_du_fichier_d'entrée_numéro_de_cycle_fort.82*).

• **WHAT(4) = WHAT(5) = WHAT(6) = 80.0**, soit **Xmax = Ymax = Zmax = 80.0**, coordonnées maximales de la maille.

• **SDUM = total** : nom du « binning ».

6.2.2. Seconde carte

- **WHAT(1) = WHAT(2) = 0.0**, soit **Xmin = Ymin = 0.0**, coordonnées minimales en X et Y de la maille.
- **WHAT(3) = Zmin = -20.0**, coordonnée minimale en Z.
- **WHAT(4) = WHAT(5) = WHAT(6) = 100.0** : nombre de « bins », soit le nombre de mailles en X, Y et Z.
- **SDUM = &**, pour toutes les cartes secondaires.

6.2.3. Remarques

À partir de la version 2006.3 de **FLUKA**, il est désormais possible de calculer l'activité par unité de volume (code **ACTIVITY** ou **234.0**) ou l'activité par unité de masse (code **ACTOMASS** ou **235.0**) qui n'ont de sens que s'ils sont requis pour un estimateur 2D ou 3D défini par une carte **USRBIN** associée aux cartes activant le calcul de la décroissance radioactive Bêta et Gamma à des temps de décroissances choisis (cartes **RADDECAY**, **IRRPROFILE**, **DCYTIMES**, **DCYSCORE**, cf. chap. **11**).

Les activités seront exprimées respectivement en Bq/cm^3 et en Bq/g à moins qu'un « binning » spécial ou un « binning » par région ne soit requis, auquel cas les unités seront Bq et $Bq.cm^3/g$.

Attention, il fait bien être conscient de ce que l'on fait, le calcul de l'activité est ponctuel alors que celui de la fluence est une distribution.

Il faut donc bien veiller à changer le type de « binning » (défini en **WHAT(1)**) en 0.0 ou 1.0 et ne pas laisser les « habituels » 10.0 ou 11.0.

Grand merci à Alfredo FERRARI pour m'avoir éclairé sur ce point.

Cf. le manuel de FLUKA à la section USRBIN pour plus de précisions.

6.3. La carte USRTRACK

La carte **USRTRACK** définit un détecteur permettant d'estimer la fluence dans un volume. La définition complète requiert normalement deux cartes successives (la seconde carte identifiée par le caractère « & » dans toute colonne comprise entre 71 et 78), sauf si les valeurs par défaut sont acceptables pour l'utilisateur.

6.3.1. Première carte

- **WHAT(1) = 1.0** : intervalles d'énergie linéaires
= -1.0 : intervalles d'énergie logarithmiques
Par défaut = 1.0.
- **WHAT(2)** : type de particules dont on veut estimer la fluence (code de **FLUKA**, cf. *manuel*). Par défaut = 201.0 = **ALL-PART** (toutes les particules transportables).
- **WHAT(3)** : unité logique du fichier de sortie.
Si > 0.0 : données formatées
Si < 0.0 : données non formatées
- **WHAT(4)** : indice de la région définissant le détecteur. Par défaut : 1.0
- **WHAT(5)** : volume du détecteur en cm^3 , par défaut = 1.0.
- **WHAT(6)** : nombre d'intervalles d'énergie, par défaut = 10.0.
- **SDUM** : tout caractère sauf « & » identifiant le détecteur (maximum 10 caractères).

6.3.2. Seconde carte

- **WHAT(1)** : énergie maximale pour le calcul en GeV. Par défaut, la valeur définie dans la carte **BEAM** est considérée (et la valeur par défaut de cette carte si la carte **BEAM** n'a pas été définie, soit 200 GeV/c).

- **WHAT(2)** : énergie minimale pour le calcul. 0.0 GeV pour le calcul en intervalles linéaires, 0.001 GeV sinon.
- **WHAT(3)-WHAT(6)** : non usités.
- **SDUM** : & (n'importe où entre les colonnes 71 à 78).

6.3.3. Notes

- 1. Les résultats d'un estimateur **USRTRACK** sont toujours donnés sous formes de distributions différentielles de fluence en fonction de l'énergie en $/cm^2/GeV/particule$ primaire. Lorsque l'on désire obtenir un spectre en énergie de la fluence intégrée en $/cm^2/particule$ primaire, il faut multiplier la valeur de chaque intervalle d'énergie par la taille de cet intervalle (même en cas de structure logarithmique).
- 2. En cas de calcul de la fluence des neutrons, lorsque les intervalles d'énergie choisis par l'utilisateur chevauchent ceux des neutrons de basses énergies, les limites des intervalles sont forcées à coïncider avec les limites des groupes et aucun intervalle ne peut être plus petit que le groupe correspondant. En fait, le programme utilise les limites requises et le nombre d'intervalles pour estimer la largeur de l'intervalle désiré. Le nombre d'intervalles au-dessus de la limite supérieure du premier groupe de neutrons de basses énergies est recalculé d'après une telle largeur.

Notez que les limites d'énergies du groupe des neutrons thermiques sont 10^{-14} GeV (10^{-5} eV) et 4.14×10^{-10} GeV (0.414 eV) pour les données de l'ENEA. Toutes les limites d'énergies des groupes de neutrons sont listées dans le manuel (chapitre consacré aux neutrons de basses énergies).

- 3. Les résultats obtenus avec des données non formatées peuvent être analysés par le programme **ustsuw**, qui permet de compiler les résultats de plusieurs cycles. Ce programme figure dans la distribution de **FLUKA** dans le répertoire **\$FLUPRO/flutil**.
- 4. En fait, le calcul de la fluence suit le même principe que pour la carte **USRBIN**, « tracklength fluence ». Il y a donc un calcul du parcours de la particule dans un volume donné. L'unité brute est $particule.cm/cm^3/particule$ primaire soit bien équivalente à $particule/cm^2/particule$ primaire. Si l'on n'indique pas le volume exact, si on le laisse égal à 1.0, la valeur calculée sera équivalente à une probabilité uniquement dans le cas où la trajectoire maximale de la particule est d'1 cm, typiquement dans une sphère (vide) d'1 cm de rayon. Sinon, il faut procéder à une normalisation, méfiance ! (Cf. **annexe H.**)

6.3.4. Exemples

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
*23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789
USRTRACK      1.0      7.0     -24.0      16.0     4500.0      150.PhotFlu
USRTRACK      1.5      0.0      0.0      0.0      0.0      0.0      0.&
```

Ici, on désire calculer le spectre en énergie de la fluence des photons dans la région 16 de 0 à 1.5 GeV, avec 150 intervalles d'énergie linéaires. Les résultats sont inscrits en données non formatées dans l'unité logique 24. Le volume de la région 16 est 4500 cm³.

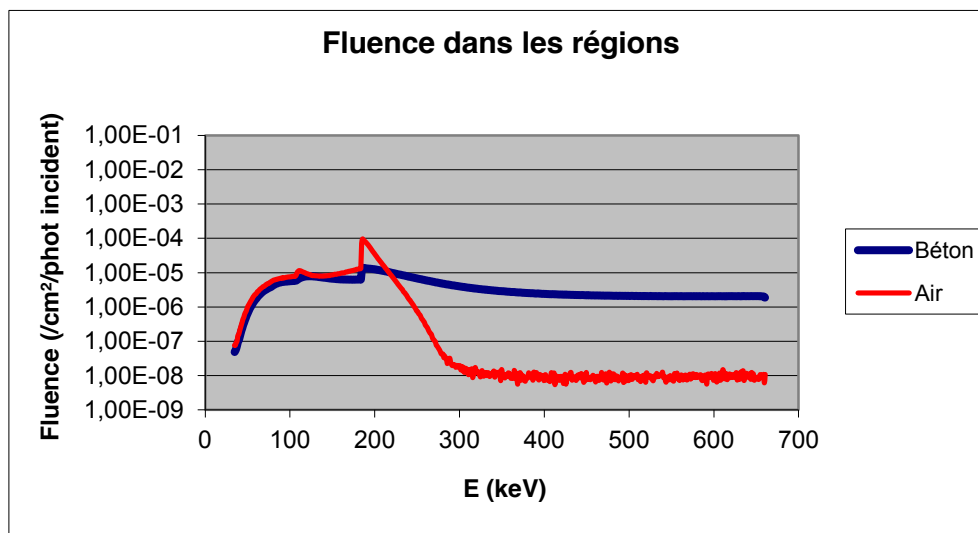
```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
USRTRACK      1.0      7.0     -50.0      3.0     458.15      700.PHOT_CONC
USRTRACK      0.7E-3     0.0
*
USRTRACK      1.0      7.0     -51.0      4.0     65.45      700.PHOT_AIR
USRTRACK      0.7E-3     0.0      &
```

Ici, nous avons défini 2 sphères concentriques de rayons respectifs 5,0 et 2,5 cm. La partie externe est constituée de béton, et la partie interne d'air. La région en béton a un volume égal à celui de la boule de rayon 5 cm moins celui de la boule de rayon 2,5 cm. La région

d'air a un volume compris dans la boule de rayon 2,5 cm. La source de Cs137, ponctuelle, isotrope, est définie au centre des boules.

Nous désirons calculer le spectre en énergie des photons dans la région 3 (constituée de béton) et dans la région 4 (constituée d'air) de 0 à 700 keV avec 700 intervalles linéaires. Les résultats non formatés seront inscrits respectivement dans les unités logiques 50 et 51.

Les résultats sur 5 cycles de 20 millions de particules ont été traités par le programme **ustsuw**. On obtient ainsi deux fichiers **ASCII** par détecteur, l'un comportant toutes les informations contenues dans les fichiers d'origine (**nom_choisi_par_l'utilisateur_sum.lis**), et l'autre est un tableau (comportant les intervalles d'énergie, la valeur du flux, et l'incertitude sur le calcul (**nom_choisi_par_l'utilisateur_tab.lis**). Un fichier binaire est également créé. Pour tracer le graphique ci-dessous, nous avons calculé l'énergie moyenne de chaque intervalle d'énergie et multiplié la valeur de la fluence dans l'air par la taille de chaque intervalle (1 keV, dans notre exemple).



6.4. La carte USRBDX

La carte **USRBDX** définit un détecteur permettant d'estimer la fluence ou le courant lors d'un passage à une limite entre deux régions.

La définition complète requiert normalement deux cartes successives (la seconde carte identifiée par le caractère « & » dans toute colonne comprise entre 71 et 78), sauf si les valeurs par défaut sont acceptables pour l'utilisateur.

6.4.1. Première carte

- **WHAT(1)** = $i1 + (i2 \times 10) + (i3 \times 100)$, où $i1$, $i2$, $i3$ ont les significations suivantes :

$i1 = +1.0$: intervalles linéaires en énergie et en angle solide

= -1.0 : intervalles logarithmiques en énergie, linéaire en angle solide

= $+2.0$: intervalle logarithmique en angle solide, linéaires en énergie

= -2.0 : intervalles logarithmiques en énergie et en angle solide

$i2 = 0.0$: calcul dans un "sens" (demi-espace)

= $+1.0$: calcul dans les deux "sens" (espace entier)

$i3 = 0.0$: calcul du courant

= $+1.0$: calcul de la fluence (normalisée par $1/\cos\theta$)

Par défaut = 1.0 (courant dans un sens, intervalles linéaires en énergie et en angle solide)

- **WHAT(2)** : type de particules dont on veut estimer la fluence (code de **FLUKA**, cf. *manuel*). Par défaut = 201.0 = **ALL-PART** (toutes les particules transportables).
- **WHAT(3)** : unité logique du fichier de sortie.
 Si > 0.0 : données formatées
 Si < 0.0 : données non formatées
- **WHAT(4)** : première région définissant la limite (en cas de calcul dans un sens, ceci est la région en amont). Par défaut = 1.0.
- **WHAT(5)** : seconde région définissant la limite (en cas de calcul dans un sens, ceci est la région en aval). Par défaut = 2.0.
- **WHAT(6)** : aire du détecteur en cm². Par défaut = 1.0.
- **SDUM** : tout caractère sauf « & » identifiant le détecteur (maximum 10 caractères).

6.4.2. Seconde carte

- **WHAT(1)** : énergie maximale pour le calcul en GeV. Par défaut, la valeur définie dans la carte **BEAM** est considérée (et la valeur par défaut de cette carte si la carte **BEAM** n'a pas été définie, soit 200 GeV/c).
- **WHAT(2)** : énergie minimale pour le calcul. 0.0 GeV pour le calcul en intervalles linéaires, 0.001 GeV sinon.
- **WHAT(3)** : nombre d'intervalles d'énergie pour le calcul. Par défaut = 10.0.
- **WHAT(4)** : angle maximal pour le calcul en sr.
 Par défaut = 2π pour un calcul dans un seul sens, 4π pour un calcul dans les deux sens.
- **WHAT(5)** :
 - Si les angles sont distribués en intervalles linéaires, c'est la valeur minimale de l'angle solide pour le calcul (sr). Par défaut = 0.0.
 - Si les angles sont distribués en intervalles logarithmiques, c'est la valeur d l'angle solide du premier intervalle (sr). Par défaut = 0.001.
- **WHAT(6)** = nombre d'intervalles pour distribuer les angles.
 Par défaut : 1.0 pour des intervalles linéaires, 3.0 sinon.
- **SDUM = &** (n'importe où entre les colonnes 71 à 78).

6.4.3. Notes (la note 2. du paragraphe **USRTRACK** est également valable ici).

- 1. Les résultats d'un estimateur **USRBDX** sont toujours donnés en termes de distributions double différentielles de fluence (ou de courant) en énergie et en angle solide, en $/\text{cm}^2/\text{GeV}/\text{sr}/\text{particule primaire}$ même si un seul intervalle a été requis ce qui est souvent le cas pour les distributions angulaires. Lorsque l'on désire obtenir un spectre en énergie de la fluence (ou du courant) sans distribution angulaire, avec les résultats intégrés en $/\text{cm}^2/\text{particule}$, il faut multiplier chaque intervalle d'énergie par la taille de cet intervalle même en cas de structure logarithmique et par 2π ou 4π (selon le type de calcul choisi dans un ou deux sens) si l'on fait l'intégration soi-même. Le programme **usxsuw** fait déjà ce calcul.
- 2. Les distributions angulaires doivent être considérées comme distributions en $\cos\theta$, où θ est l'angle entre la trajectoire de la particule et la normale à la limite à l'endroit du passage. Lorsqu'une structure logarithmique est requise pour les distributions angulaires, tous les intervalles ont la même largeur logarithmique (ratio égal entre la limite supérieure et inférieure de l'intervalle), sauf pour le premier. Les limites de ce premier intervalle sont $\theta = 0$ et la valeur indiquée par l'utilisateur en **WHAT(5)** dans la seconde carte.
- 3. Les résultats obtenus avec des données non formatées peuvent être analysés par le programme **usxsuw**, qui permet de compiler les résultats de plusieurs cycles. Ce programme figure dans la distribution de **FLUKA** dans le répertoire **\$FLUPRO/flutil**.

- 4. Le calcul est différent ici, « boundary crossing fluence ». C'est le calcul de la densité de trajectoires des particules dans un volume constitué de la base, l'aire de l'interface et une épaisseur infinitésimale. L'unité brute est donc *particule/cm²/particule primaire*. Si l'on n'indique pas la surface exacte, si on la laisse égale à 1.0, la valeur calculée sera équivalente à une probabilité à une constante près, méfiance ! La fluence calculée est proportionnelle à $1/\cos\theta$ où θ est l'angle entre la normale à la surface à traverser et la trajectoire de la particule (cf. [annexe H](#)).

6.4.4. Exemples

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
USRBDX      101.0      9.0      21.0      3.0      4.0      400.0AntiNeu
USRBDX      5.0       0.0     200.0      0.0      0.0      0.0&
```

Ici, on désire calculer le spectre de fluence d'antineutrinos de 0 à 5 GeV, en 200 intervalles linéaires d'énergie, de la région 3 à la région 4 (et pas de la région 4 à la région 3). Les résultats formatés sont inscrits dans l'unité logique 21. L'aire de la limite est 400 cm². Un seul intervalle angulaire est requis (de 0 à 2 π).

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
*23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789
USRBDX      101.0      7.0     -63.0      4.0      3.0      150.4-3_2PIP
USRBDX      0.7E-3      0.0      700.      &
*
USRBDX      101.0      7.0     -64.0      4.0      5.0      25.4-5_2PIP
USRBDX      0.7E-3      0.0      700.      &
```

Avec un exemple du même ordre que le second exemple de la carte **USRTRACK**, on désire calculer le spectre de fluence de photons de 0 à 700 keV en 700 intervalles linéaires d'énergie, de la région 4 à la région 3 et de la région 4 à la région 5.

La région 3 est l'intersection entre un cube centré autour de (0, 0, 0) et de côté 5 cm et un cube concentrique de côté 2,5 cm. La région 4 (contenant la source de rayonnement) est la partie du cube intérieur dont z est compris entre -1 et 2,5. La région 5 est la partie du cube intérieur dont z est compris entre -2,5 et -1.

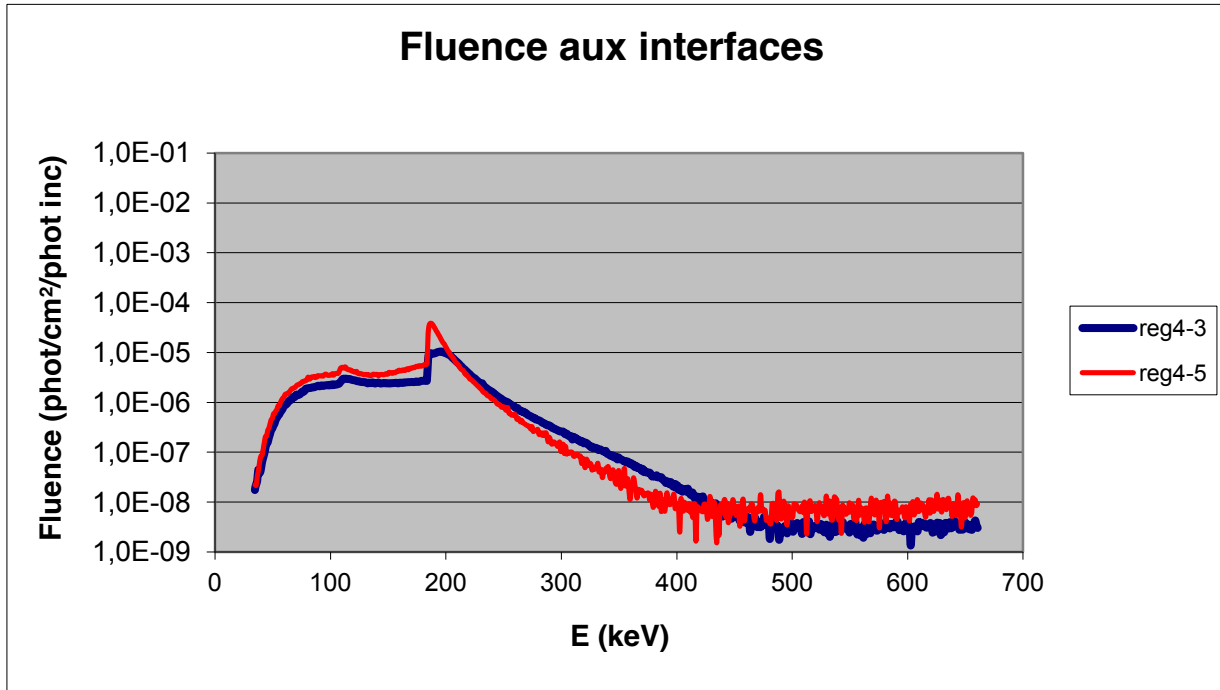
La source de Cs137, ponctuelle, isotrope, est définie au point de coordonnées (0, 0, 0).

La région 3 est constituée de béton, les régions 4 et 5 d'air.

Les résultats non formatés sont inscrits dans les unités logiques 63 et 64. Les aires des interfaces : pour le premier détecteur l'interface est toute la surface du petit cube soit 150 cm² (6*5²) et pour le seconde, l'aire de l'interface est 25 cm² (5*5).

Les résultats sur 5 cycles de 20 millions de particules ont été traités par le programme **usxsuw**. On obtient ainsi deux fichiers **ASCII**, l'un comportant toutes les informations contenues dans les fichiers d'origines (**nom_choisi_par_l'utilisateur_sum.lis**), et l'autre est un tableau (avec les intervalles d'énergie, la valeur du flux, et l'incertitude sur le calcul, (**nom_choisi_par_l'utilisateur_tab.lis**). Un fichier binaire est également créé.

Pour tracer le graphique ci-dessous, nous avons calculé l'énergie moyenne de chaque intervalle d'énergie, multiplié la valeur de la fluence par la taille de chaque intervalle (1 keV dans notre exemple), les résultats étant déjà intégrés sur les angles solides respectifs.



6.5. La carte DETECT

6.5.1. Description

La carte **DETECT** permet d'obtenir l'énergie déposée événement par événement tel un détecteur avec des options de coïncidence et d'anti-coïncidence telles que celles proposées par un trigger. Dans ce qui suit, un « événement » signifiera « énergie déposée » dans une ou plusieurs régions constituant le *détecteur* par une particule primaire et ses descendants entre deux appels successifs de la routine **FEEDER** (pour un faisceau incident décrit avec les cartes **BEAM**, **BEAMPOS**, etc.) ou de la routine **SOURCE** (dans le cas de son utilisation). Un « signal » signifie « énergie déposée » dans une ou plusieurs régions constituant le *trigger* par la même particule primaire et ses descendants.

Les données de sortie de **DETECT** : il s'agit d'une distribution d'énergie déposée par événement dans la ou les régions constituant le détecteur en (anti)coïncidence avec un signal supérieur à une énergie de coupure donnée ou un seuil donné pour les régions du trigger.

Dans le futur, il devrait également être possible de définir des combinaisons de détecteurs. Cette option peut être décrite sur plusieurs cartes.

La signification de la première :

- **WHAT(1)** ≤ 0.0 pour un détecteur, > 0.0 pour une combinaison de détecteurs.
 - Si **WHAT(1)** ≤ 0.0 (nous ne décrivons pas l'autre cas, pas encore opérationnel) :
- **WHAT(2)** : énergie minimale détectable en un événement, soit la limite inférieure de la distribution. Par défaut, elle vaut 0.0.
- **WHAT(3)** : énergie maximale détectable en un événement, soit la limite supérieure de la distribution. Pas de valeur par défaut.

- **WHAT(4)** : énergie de coupure pour le signal (seuil de coïncidence/anti-coïncidence). La distribution de la déposition d'énergie n'est comptée que si plus de **WHAT(4)** GeV sont ou ne sont pas déposés dans les régions du trigger. Par défaut, elle vaut $1 \text{ eV} = 10^{-9} \text{ GeV}$.
- **WHAT(5)** : ≥ 0.0 : les régions du détecteur, toutes prises en compte, doivent être considérées en coïncidence avec les régions du trigger (elles aussi toutes prises en compte).
 < 0.0 : les régions du détecteur sont considérées en anti-coïncidence avec les régions du trigger. Par défaut : anti-coïncidence.
- **WHAT(6)** : ≥ 0.0 : première région du détecteur.
 < 0.0 : la valeur absolue de **WHAT(6)** est la première région du trigger (les autres régions sont données par des cartes supplémentaires). Pas de valeur par défaut.
- **SDUM** : nom du détecteur (10 caractères maximum) sans que le caractère « & » soit présent.

Signification de la seconde carte (si elle est présente) :

- **WHAT(1)** : la même chose que le **WHAT(1)** de la première carte.
- **WHAT(2) – WHAT(6)** : régions successives (avec signe). Si > 0.0 , elles sont considérées comme des régions constituant le détecteur. Sinon, elles sont considérées comme des régions constituant le trigger.
- **SDUM** : « & » dans n'importe quelle position dans les colonnes 71 à 78.

Note : si aucune région de trigger n'est donnée, i.e. si aucune région n'a de signe négatif, un simple comptage événement par événement a lieu.

6.5.2. Exemples

- Premier exemple (celui du manuel) :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
DETECT      0.0      1.E-4      1.E-2      5.E-5      1.0      7.0Coincid_1
DETECT      0.0      -9.0      -12.0      10.0      11.0      &
```

Un signal égal au dépôt d'énergie dans les régions 7, 10 et 11 sera compté si :

- 1) Ce signal est supérieur à 100 keV et inférieur à 10 MeV.
- 2) Au même moment (en coïncidence) un dépôt d'énergie d'au moins 50 keV se produit dans les régions 9 et 12.

Le fichier de sortie sera un spectre d'événements compris entre 100 keV et 10 MeV distribués selon un nombre de canaux fixés (1024 dans la version standard).

- Second exemple (source de Cs137) :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
DETECT      0.0      5.E-5      7.E-4      5.E-5      1.0      4.0Spec
```

Ici, on aura un spectre d'événements comptés dans la région 4 entre 50 et 700 keV.

- Troisième exemple (coïncidences avec une source d'Eu152) :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
DETECT      0.      5.E-5      1.45E-3      5.E-5      1.      4.SCINT
DETECT      0.      5.      6.      7.      -8.      -9.&
```


Le détecteur est constitué des régions 4, 5, 6, 7, le trigger est constitué des régions 8 et 9. Un signal supérieur à 50 keV et inférieur à 1450 KeV sera compté dans le détecteur si au même moment un signal d'au moins 50 keV est compté dans le trigger.

6.5.3. Remarque

L'option **DETECT** ne donnera des résultats significatifs que si **FLUKA** est utilisé en mode complètement analogue parce que les corrélations sont détruites par le biaisage. En conséquence, l'option **DETECT** ne peut être utilisée avec des options suivantes : **BIASING**, **EMF-BIAS**, **LOW-BIAS**, **LAM-BIAS**, **WW-FACTOR**, **EMFCUT** avec **WHAT(3) < 0.0**, **EMF** avec **WHAT(6)** différent de 1.0, **EXPTRANS**, **LOW-DOWN**.

Il est d'ailleurs recommandé, dans le but d'empêcher tout biaisage d'utiliser une commande **GLOBAL** avec **WHAT(2) < 0.0** au début du fichier d'entrée (cf. option **GLOBAL** dans le manuel).

6.5.4. Récupération des données de sortie

Les résultats sont inscrits dans l'unité logique 17 sous forme binaire (non formaté). Il n'existe pas de routine jointe à la distribution de **FLUKA** pour récupérer ces données sous format **ASCII** comme avec les programmes **ustsuw**, **usxsuw** ou **usbrea** pour les estimateurs **USRTRACK**, **USRBDX** et **USBIN** décrits précédemment. Il faut donc écrire un fichier fortran contenant les commandes permettant d'ouvrir le fichier de données non formatées, de lire les variables intéressantes, de les écrire dans un fichier texte plutôt que directement dans le terminal. Sur le modèle de la routine **usbrea.f** par Alfredo Ferrari et Alberto Fasso, j'ai donc pris la décision de créer ma routine **detoutput.f**.

Le manuel donne quelques indications au sujet des variables intéressantes :

- **RUNTIM** : date de la simulation (également inscrit au début du fichier *.out).
- **WEIPRI** : poids total des particules primaires.
- **NCASE** : nombre de particules primaires.
- **NDET** : numéro du détecteur.
- **CHNAME** : nom du détecteur (= **SDUM** de la première carte **DETECT**).
- **NBIN** : nombre d'intervalles d'énergie (fixé à 1024 pour le moment).
- **EMIN** : énergie minimale totale (= **WHAT(2)** de la première carte **DETECT**).
- **EBIN** : taille de chaque intervalle d'énergie (égal à (Emax-Emin)/1024).
- **ECUT** : énergie de coupure ou seuil d'énergie pour le signal (= **WHAT(4)** de la première carte **DETECT**).
- **NBIN** valeurs **IV(I)** : ce sont les canaux du spectre ou les intervalles d'énergie.

On introduit aussi Emax, l'énergie maximale totale (= **WHAT(3)** de la première carte **DETECT**). La relation pour passer du numéro de canal (noté c) à l'énergie qu'elle représente est la suivante :

$$E(c) = \frac{(E_{\max} - E_{\min}) * c}{1024} + E_{\min}$$

La routine **detoutput.f** est décrite ci-après. Pour obtenir l'exécutable **detoutput**, on la compilera ainsi : **g77 -o detoutput detoutput.f**

Elle pourra être placée avec les autres routines de ce type dans **\$FLUPRO/flutil**.

La commande **\$FLUPRO/flutil/detoutput** lance le programme, les variables **RUNTIM**, **RUNTIM**, **WEIPRI** et **NCASE** sont affichées dans le terminal et le nom du fichier de données

non formatées est demandé (**type the input file name:**). Le nom entré et validé, le nom du fichier texte de sortie est demandé (**type the output file name:**). Le nom entré et validé, le programme créé ou écrase le fichier s'il existe déjà et se termine. Il n'est pas possible ici de procéder à des compilations de plusieurs cycles d'une simulation. La moyenne sur différents cycles peut être effectuée par la suite (dans Excel ou autre), les résultats étant manipulables par n'importe quel éditeur de texte ou tableur.

Présentation de la version 1 du 12/07/2007 de la routine detoutput : cf. **Annexe I.1.**

Ci-après, nous proposons un exemple de fichier obtenu pour une simulation d'un million de particules primaires ayant pour but d'estimer l'efficacité absolue de détection d'un scintillateur NaI(Tl) cylindrique de 3 pouces par 3 pouces avec une source isotrope de Cs137 centrée et à 3 cm de la face d'entrée du détecteur. Il s'agit ici d'un spectre simple (événement par événement). La carte utilisée pour obtenir ces résultats est la même que celle décrite dans le second exemple du paragraphe **6.5.2.**

```

***** Title of the run:  Sampling Cs137 source                *****

***** Detector number:      1 *****
***** Detector name: SCINT          *****
***** Number of energy bins:  1024 *****
***** Minimum total energy (GeV):  5.0000E-05 *****
***** Width of each energy bin (GeV):  6.3477E-07 *****
***** Cutoff energy for the signal (GeV):  0.0000E+00 *****
***** The spectrum follows: *****

1:      50
2:      55
3:      60
4:      61
5:      60
...
960:    0
961:    0
962:    0
963:    0
964:  53870
965:    0
...
1022:   0
1023:   0
1024:   0
    
```

On a donc le numéro du canal et le nombre d'événements enregistrés. Ici, avec cette source de Cs137, le maximum est atteint pour le canal n° 964 soit $50 + ((700 - 50) * 964 / 1024) = 661,9$ keV (taille des intervalles $(700-50)/1024 = 0,635$ keV). L'efficacité sur 4π à 661,7 keV vaut donc $53\,870/1\,000\,000 = 5,4\%$ environ.

6.5.5. Étude de l'efficacité absolue de détection d'un cristal hyper pur de germanium

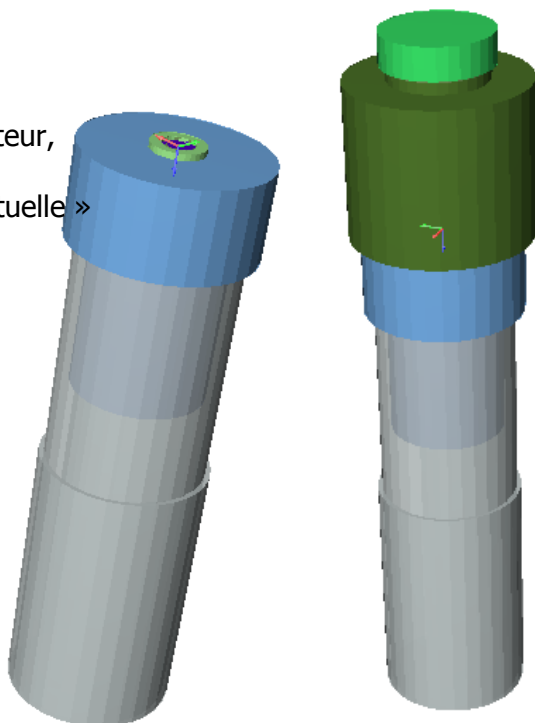
Référence pour ce travail : *SIMULATION OF AN HPGe DETECTOR AND A MARINELLI'S BEAKER SOURCE TO DERIVE EFFICIENCY CURVES WITH FLUKA 2005.6, A. A. Porta , F. Campi, Nuclear Engineering department – CeSNEF, Polytechnic of Milan.*

Les caractéristiques du détecteur :

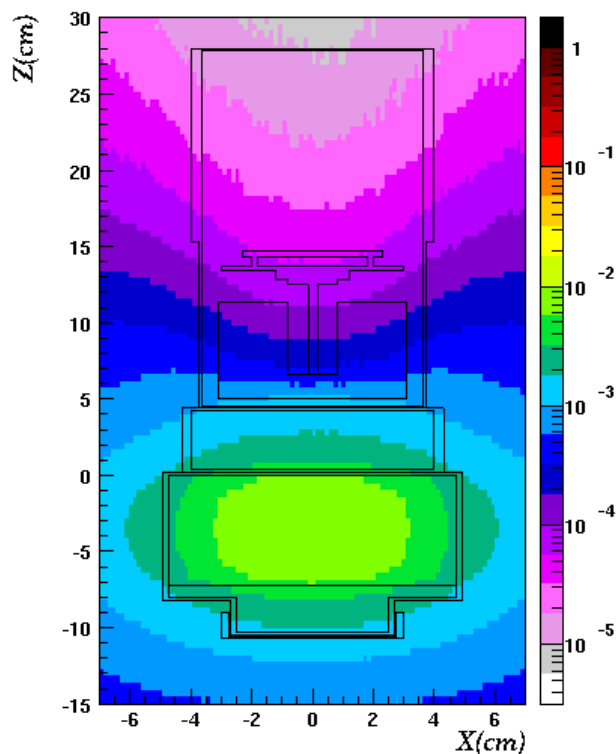
- Modèle : GC4019-7500SL (Canberra), numéro de série b96093.

- Coaxial, 62 mm de diamètre, 64 mm de longueur, distance de la fenêtre en aluminium : 5 mm. Efficacité relative : 40 %.
 - Les autres caractéristiques : matériaux, dimensions ont été relevées sur la documentation technique fournie à la livraison de l'appareil.
 - La taille de la zone morte au sommet du cylindre de germanium, épaisseur de matériau non active dans la détection est très importante. En l'absence de données précises quant à sa taille, nous avons procédé par dichotomie, choisissant une valeur réaliste (par rapport à l'étude citée en référence).
- Géométrie : La modélisation du détecteur doit être la plus précise possible, notamment les couches métalliques de plus de 0,1 mm (Aide décisive d'Alessandro A. Porta pour ce travail). Cf. **Annexe I.2.** : fichier d'entrée complet pour une source de type SG500.
 - Source : nous avons testé deux types de sources, l'une ponctuelle, l'autre étendue, de type SG500. Cette dernière, couvrant davantage la surface sensible du détecteur donne de meilleurs résultats. Il s'agit d'un cylindre (rayon 4,7 cm longueur 7,2 cm compris entre $Z = -0,025$ et $-7,225$) constitué de matériau de densité proche de 1, de la résine d'époxy contenant un mélange de radionucléides aux activités connues. La source utilisée est pourvue d'un certificat d'étalonnage (N° CT/070764/07/1436). Il s'agit de la source multi-gamma type 9ML01EGRH15 n°75036/2. Ci-après, nous présentons le fichier décrivant la source, lié à l'exécutable de **FLUKA**.

Ci-contre : détecteur, support plexiglas et source « ponctuelle »



Ci-contre : détecteur, support plexiglas et source SG500



- Les résultats :
 - En termes d'isodoses, nous avons l'allure suivante (la source est tête en bas, ci-contre) :

➤ Courbe d'efficacité :

Chaque énergie a fait l'objet d'une simulation, la statistique (5 cycles de 6 millions) a été choisie afin que chaque pic ait un nombre de coups supérieur à 10 000. Une carte **DETECT** a été définie, couvrant la région 17 (la partie active du cristal de germanium).

À partir du nombre de coups mesurés (logiciel de spectrométrie GENIE2000) et des valeurs d'activités radioactives (toutes données à + ou - 6 %), nous en déduisons la courbe d'efficacité mesurée. Concernant les résultats de la simulation, le rapport entre le nombre de coups à l'énergie E et le nombre de particules primaires donne l'efficacité à l'énergie E.

On rappelle que l'efficacité absolue ε se définit ainsi : $\varepsilon = \frac{N}{A * t * (\frac{I}{100})}$, avec :

N : Le nombre de coups mesurés à la date T, de l'élément X, à l'énergie E.

A : L'activité radioactive en Bq, à la date T, de l'élément X.

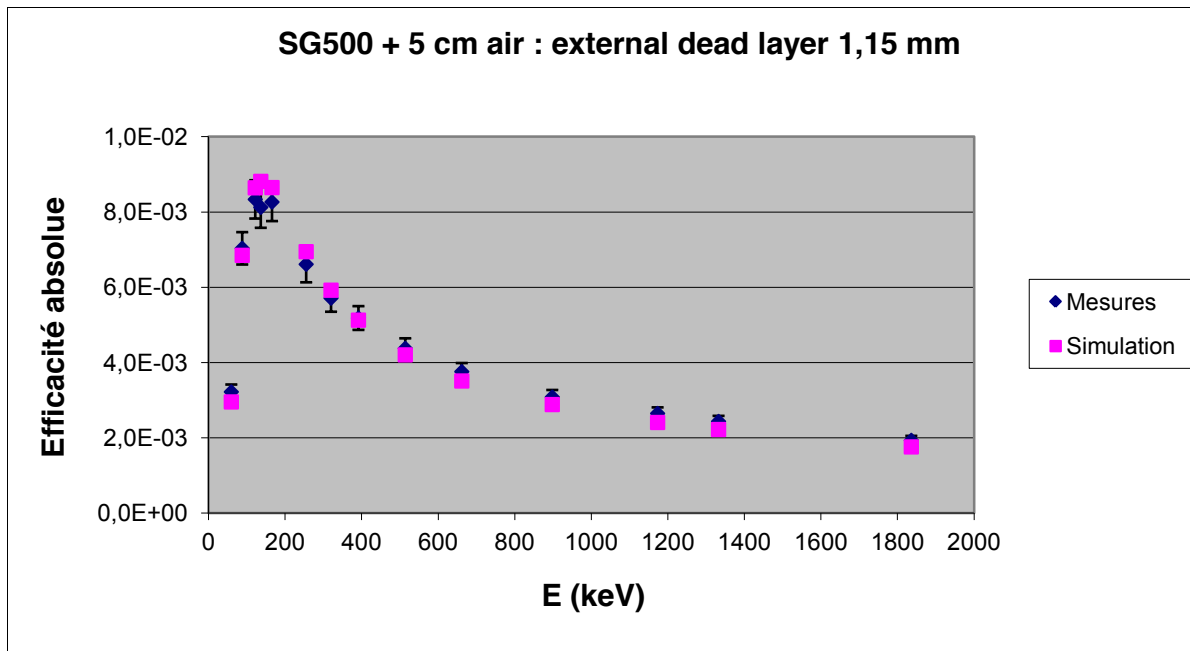
t : La durée d'acquisition en secondes

I : L'intensité relative de la raie d'énergie E en %

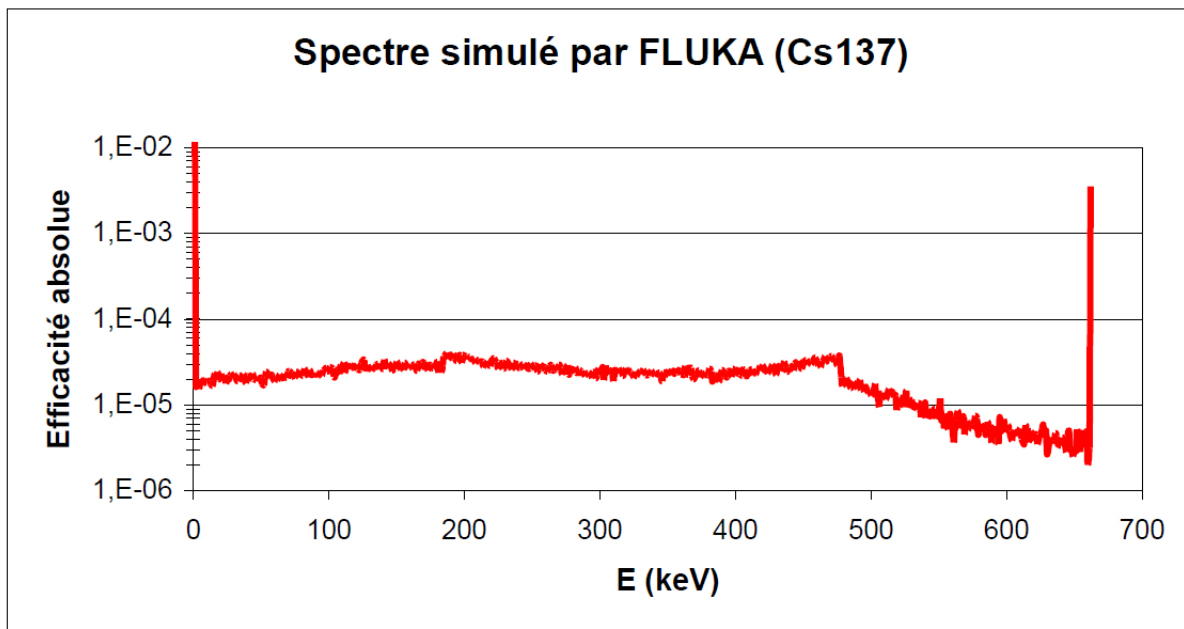
Le tableau suivant présente les résultats de ces simulations pour chaque raie émise par la source.

E (keV)	Élément	I (%)	Nombre de coups mesurés 07/03/08	Incertitude	Activité (Bq) 07/03/08	Efficacité	Incertitude (%)	Efficacité simulée	Différence (%)
59,54	Am241	35,78	2,12E+05	1110	1839	3,22E-03	6,11	2,96E-03	8,28
88,03	Cd109	3,626	1,82E+05	1100	7132	7,04E-03	6,11	6,85E-03	2,64
122,06	Co57	85,51	2,31E+05	1210	324	8,34E-03	6,11	8,65E-03	-3,69
136,47	Co57	10,71	2,82E+04	786	324	8,13E-03	6,69	8,82E-03	-8,49
165,86	Ce139	79,9	1,46E+05	1050	221	8,27E-03	6,13	8,65E-03	-4,65
255,13	Sn113	2,11	1,69E+04	666	1211	6,61E-03	7,25	6,95E-03	-5,05
320,08	Cr51	9,87	3,75E+04	632	665	5,71E-03	6,31	5,92E-03	-3,70
391,7	Sn113	64,97	4,08E+05	1360	1211	5,19E-03	6,09	5,13E-03	1,10
514	Sr85	98,5	1,57E+05	891	364	4,38E-03	6,11	4,21E-03	3,97
661,7	Cs137	84,99	8,21E+05	1860	2569	3,76E-03	6,09	3,51E-03	6,61
898,04	Y88	93,9	3,21E+05	1200	1108	3,09E-03	6,09	2,89E-03	6,48
1173,23	Co60	99,85	6,59E+05	1640	2491	2,65E-03	6,09	2,41E-03	9,06
1332,49	Co60	99,9826	6,07E+05	1570	2491	2,44E-03	6,09	2,22E-03	8,80
1836,05	Y88	99,32	2,13E+05	925	1108	1,94E-03	6,10	1,76E-03	9,06

Le graphe suivant représente la comparaison entre efficacité mesurée et efficacité simulée.



Le graphe suivant représente un spectre simulé par FLUKA dans le cas du Cs137.



7. Calcul des doses

En association avec les cartes **USRBIN**, **USRBDX**, **USRTRACK**.

Pour calculer directement la dose pendant la simulation, il faut invoquer la routine adéquate :

- **COMSCW** dans le cas du calcul de la densité d'énergie déposée
- **FLUSCW** dans le cas du calcul de la fluence particulaire

En pratique, ces routines sont activées par la carte **USERWEIG** (*cf. manuel*).

7.1. La carte USERWEIG

- **WHAT(1), WHAT(2)** : non usités.
- **WHAT(3)** : > 0.0 : multiplie la donnée obtenue par les différents estimateurs concernés par **FLUSCW** pendant la durée de la simulation.

Si **WHAT(3)** est compris entre 1.0 et 2.0 : **FLUSCW** est invoquée sans vérification du détecteur.

Si **WHAT(3)** > 2.0 : **FLUSCW** est invoquée après vérification que le détecteur utilisé est appliqué à ce calcul.

Si **WHAT(3)** = 2.0 ou 4.0, la routine **FLDSCP** est également invoquée appliquant un décalage dans le calcul de la fluence en cours de simulation.

Si **WHAT(3)** < 0.0 : remise des valeurs par défaut, pas de multiplication.

Si **WHAT(3)** = 0.0 : option ignorée.

Par défaut, **WHAT(3)** = -1.0, pas de multiplication.

- **WHAT(4)** : non usité
- **WHAT(5)** : si > 0.0 , la fonction **USCRNC** est invoquée chaque fois qu'un noyau résiduel est généré. Le calcul des noyaux résiduels est multiplié par la valeur retournée par cette fonction.
- **WHAT(6)** : > 0.0 : multiplie la donnée obtenue par les différents estimateurs concernés par **COMSCW** pendant la durée de la simulation.

Si **WHAT(6)** est compris entre 1.0 et 2.0 : **COMSCW** est invoquée sans vérification du détecteur.

Si **WHAT(6)** > 2.0 : **COMSCW** est invoquée après vérification que le détecteur utilisé est appliqué à ce calcul.

Si **WHAT(6)** = 2.0 ou 4.0, la routine **FLDSCP** est également invoquée appliquant un décalage de la perte d'énergie en cours de calcul.

Si **WHAT(6)** < 0.0 : remise des valeurs par défaut, pas de multiplication.

Si **WHAT(6)** = 0.0 : option ignorée.

Par défaut, **WHAT(6)** = -1.0, pas de multiplication.

- **SDUM** : non usité.

On peut déterminer la valeur de ce facteur par le calcul, à l'aide de formules théoriques, empiriques, en tenant compte des changements d'unités, du milieu traversé, ou en s'aidant de facteurs préconisés par les organismes référents en la matière (CIPR, AIEA, etc.).

7.2. Utilisation de la routine FLUSCW

Si l'on calcule la fluence particulaire, cette dernière sera donnée en nombre de particules par unité de surface (cm²) par particule incidente en fonction de la distance par rapport à la

source. Pour remonter à un débit de dose absorbée, il faudra alors se servir des relations suivantes (*photons monénergétiques et équilibre électronique atteint*) :

$$\dot{D} = 1,6 * 10^{-14} * \dot{F} * E * \frac{\mu_{en}}{\rho}, \text{ avec :}$$

- Le débit de fluence $\dot{F} = \frac{A * v}{4\pi * d^2}$ en *particules par m² par seconde* où A est l'activité en Bq, v le rapport d'embranchement pour la raie considérée, d, la distance considérée en m.
- L'énergie du rayonnement E en MeV.
- Le coefficient massique d'absorption d'énergie $\frac{\mu_{en}}{\rho}$ pour la matière traversée en *cm²/g*.

Or, ce que donne FLUKA est le rapport $\frac{\dot{F}}{A}$, exprimé en *particules par cm²*.

On en déduit la valeur à attribuer à **FLUSCW**, on modifie le fichier **fluscw.f** disponible dans **\$FLUPRO/usermvax** que l'on compile à l'aide du script disponible et que l'on relie à l'exécutable de **FLUKA** (cf. **2.2.**) du présent document).

7.3. Utilisation de la routine COMSCW

Le calcul de la densité d'énergie déposée propose un résultat exprimé en *GeV/cm³/particule incidente* en fonction de la distance à la source de rayonnement. Pour remonter à un débit de dose absorbée, il s'agit de diviser le résultat de **FLUKA** par la masse volumique ρ (en *g/cm³*) dans le milieu considéré et de convertir les unités pour obtenir un résultat en *J/kg*, en *Gy*. Une fois encore, il faudra multiplier par l'activité de la source, ce qui donnera des résultats en *Gy/s*.

On en déduit la valeur à attribuer à **COMSCW**, on modifie le fichier **comscw.f** disponible dans **\$FLUPRO/usermvax** que l'on compile à l'aide du script disponible et que l'on relie à l'exécutable de **FLUKA** (cf. **2. 2.**) du présent document).

7.4. Facteurs de conversion CIPR74

7.4.1 Utilisation de la routine deq99c.f

Une autre solution consiste à utiliser la routine **deq99c.f** développée par le CERN et qui contient en substance tous les facteurs de conversions en fonctions des énergies des particules pour différentes grandeurs.

Il suffit pour cela de compiler la routine **deq99c.f** en **deq99c.o** et de la lier à l'exécutable de **FLUKA** par les procédures décrites au **2. 2.**

La routine est appelée par la carte **USERWEIGH** (ex-**EXTRAWEIGH**), en donnant la valeur 1.0 (> 0.0 en tout cas) à **WHAT(3)**. C'est donc la routine **FLUSCW**, qu'il n'est pas nécessaire de modifier ici, qui intervient pour appeler **deq99c**.

Au début de la simulation, dans votre fichier résumé de la simulation, ***.out**, vous trouverez une ligne confirmant que **FLUSCW** est activée.

Exemple :

```

NEXT SEEDS:      0      0      0      0      0      0      0 181CD  3039      0      0
FLUSCW:  direct conversion of fluence to ambient dose eq. requested with
           - ICRP74 and Pelliccioni conversion factors
    
```

```

1          999999          999999          1.0004044E-03
1.0000000E+30          0
***
    
```

Rappel : Lorsque l'on utilise **USRBIN** pour calculer une fluence, les résultats sont exprimés en *particules/cm²/particule primaire*. Sachant que les facteurs de conversion de **deq99c** sont exprimés en *pSv.cm²*, les résultats obtenus à la fin de la simulation seront exprimés en *pSv/particule primaire*.

Les différentes grandeurs que l'on peut calculer par cette méthode :

"Effective dose from ICRP74 and Pelliccioni data calculated with ICRP radiation weighting factors W_r "

- Hp(10), irradiation antérieure-postérieure, code **EAP74**.
- Hp(10), irradiation rotationnelle, code **ERT74**.
- Hp(10), géométrie pénalisante (la pire) pour l'irradiation, code **EWT74**.
- H*(10), débit de dose gamma ambiant, code **AMB74**.

La grandeur est choisie en renseignant correctement le **SDUM** de la première carte de votre estimateur **USRBIN**, **USRTRACK**, **USRBDX**. Les codes ad hoc sont indiqués au début de **deq99c.f**.

Exemple : Je veux calculer le débit de dose gamma ambiant d'après les facteurs de conversion de la CIPR74 dans un « binning » cartésien. Le code à mettre en **SDUM** est **AMB74**.

```

USRBIN      10.      7.      -50.      20.      10.      20.AMB74
USRBIN      -20.     -10.     -20.      50.      50.      50.&
    
```

À présent, je désire calculer la dose efficace la plus pénalisante possible, toujours selon les données de la CIPR74 pour toutes les particules, le code à mettre en **SDUM** est **EWT74**.

```

USRBIN      10.     201.     -50.      20.      10.      20.EWT74
USRBIN      -20.     -10.     -20.      50.      50.      50.&
    
```

Attention, cette routine est donc une extension de **FLUSCW**, elle ne concerne que les conversions fluence/dose !

7.4.2 Carte AUXSCORE

À partir de la version 2008.3, les facteurs présents dans la routine **deq99c.f** sont directement intégrés au code et la dose équivalente est calculable directement grâce au code **240.0** ou **DOSE-EQ** renseigné à l'emplacement adéquat dans chaque carte d'estimateur.

Les estimateurs sont liés aux facteurs de conversion via la carte **AUXSCORE**. Les codes appelant les différents types de facteurs figurent à présent en **SDUM** de cette carte **AUXSCORE**.

Description de la carte **AUXSCORE** :

- **WHAT(1)** : type d'estimateur à associer à cette carte :
 - **1.0** : **USRBDX**
 - **2.0** : **USRBIN**, **EVENTBIN**
 - **3.0** : **USRTRACK**

- **4.0 : USRCOLL**
- **5.0 : USRYIELD**

Par défaut, la valeur **2.0** est donnée.

- **WHAT(2)** : particule ou famille de particules à considérer en tant que filtre pour l'estimateur associé.
 - **>-100** : code de la particule ou de la famille de particules
 - **< ou = -100** : codage des isotopes. Sélectionner le numéro atomique Z, le nombre de masse A et l'état isomérique M,
WHAT(2) = -(Z*100+A*100000+M*100000000).
 Z = 0 signifie : tous les numéros atomiques pour un A donné,
 A = 0 signifie : tous les nombres de masse pour un Z donné,
 M = 0 signifie : tous les états (fondamental et isomérique).
 Pour sélectionner l'état fondamental uniquement : M = 9.

Par défaut, la valeur **201.0 = ALL-PART** est donnée.

- **WHAT(3)** : non usité.
- **WHAT(4)** : index le moins élevé (ou nom correspondant) de la liste des estimateurs à laquelle la carte est associée (par défaut, **1.0**).
- **WHAT(5)** : index le plus élevé (ou nom correspondant) de la liste des estimateurs à laquelle la carte est associée (par défaut, **WHAT(4)**).
- **WHAT(6)** : pas à considérer dans l'assignation décrite aux deux cartes précédentes (par défaut, **1.0**).
- **SDUM** : pour le calcul de doses équivalentes (**DOSE-EQ**), l'utilisateur dispose des facteurs de conversion fluence / dose efficace ou équivalent de dose ambiant (en fonction de l'énergie) pour les neutrons, protons, pions chargés, muons, photons et électrons (à partir de 5 MeV et 2 MeV respectivement, cf. *paragraphe 7.5*).

Par défaut, **AMB74** est considéré.

Si l'on reprend le dernier exemple du paragraphe précédent, cela devient :

```

USRBIN          10.      240.      -50.      20.      10.      20.usrdose
USRBIN          -20.     -10.     -20.      50.      50.      50.&
AUXSCORE        USRBIN  ALL-PART          1.      1.      1.EWT74
    
```

Les résultats de cet estimateur **USRBIN** : dose équivalente en pSv par particule incidente avec les facteurs les plus pénalisants en ce qui concerne la géométrie d'irradiation.

7.5. Cas des électrons

En ce qui concerne les électrons, la routine **deq99c.f** ne comporte que les facteurs de conversion à partir d'une énergie de 5 MeV pour Hp(10) et de 2 MeV pour H*(10).

Si lors d'une étude, la dose (à la peau et aux extrémités notamment) induite par des électrons d'énergie inférieure à ces valeurs est d'un quelconque intérêt, il s'agit de construire soi-même une routine **fluscw.f** selon le principe énoncé en **7.2**.

La référence de ce paragraphe : Évaluation de l'exposition professionnelle due aux sources externes de rayonnements, coparrainé par l'Agence Internationale pour l'Énergie Atomique et le Bureau International du Travail, guide de sûreté n° RS-G-1.3, Annexe V, tableau V-6 : coefficients de conversion de référence pour les électrons, incidence normale, IAEA, 2004.

Le tableau présenté dans la référence susnommée donne notamment les facteurs de conversion fluence / $H'(0,07,0^\circ)$ à partir de 70 keV.

Énergie des électrons (MeV)	$H'(0,07,0^\circ)/F$ (nSv cm ²)	$H'(3,0^\circ)/F$ (nSv cm ²)	$H'(10,0^\circ)/F$ (nSv cm ²)
0,07	0,221		
0,08	1,056		
0,09	1,527		
0,1	1,661		
0,1125	1,627		
0,125	1,513		
0,15	1,229		
0,2	0,834		
0,3	0,542		
0,4	0,455		
0,5	0,403		
0,6	0,366		
0,7	0,344	0	
0,8	0,329	0,045	
1	0,312	0,301	
1,25	0,296	0,486	
1,5	0,287	0,524	
1,75	0,282	0,512	0
2	0,279	0,481	0,005
2,5	0,278	0,417	0,156
3	0,276	0,373	0,336
3,5	0,274	0,351	0,421
4	0,272	0,334	0,447
5	0,271	0,317	0,43
6	0,271	0,309	0,389
7	0,271	0,306	0,36
8	0,271	0,305	0,341
10	0,275	0,303	0,33

La routine **fluscw.f** modifiée de la sorte (seule la première colonne du tableau est intégrée, celle donnant $H'(0,07,0^\circ)/F$) est présentée en **Annexe K**.

De cette façon, lorsqu'un détecteur **USRBIN** avec calcul de la fluence des électrons est défini et si la carte **USERWEIG** (paragraphe **7.1.**) est définie correctement, les résultats sur les courbes isodoses (cf. paragraphe **8.**) seront exprimés en pSv/primary. Nous aurons a priori un estimateur majorant de la grandeur de protection $H_p(0,07)$.

Notons que cette routine ne fonctionne (dans sa configuration actuelle) qu'avec les détecteurs de type **USRBIN** (pas **USRTRACK**, ni **USRBDX**).

8. Tracer des graphes en couleurs avec les fichiers de sortie d'USRBIN

Les étapes :

- 1 - Dessiner une section de la géométrie pour y superposer les données : pour cela lancer **FLUKA** avec l'option **PLOTGEOM** ou tracer cette section avec **ALIFE** (cf. *annexe D*).
- 2 - Obtenir un fichier moyennant les données des fichiers de sortie disponibles de l'estimateur **USRBIN** : compiler, lier et exécuter le programme **USBSUW**.
- 3 - Produire le graphe en couleur : compiler, lier et exécuter le programme **PAWLEVBIN**.
- 4 - Tracer la figure : utiliser les programmes **PAW**, **GNU PLOT**, **ROOT** (nous ne décrivons que la procédure avec **PAW** dans ce chapitre).

8.1. L'option PLOTGEOM de FLUKA

La carte **PLOTGEOM** (cf. *manuel*) fait appel aux fonctions de constructions géométriques permettant de découper en tranches la géométrie du problème et de produire des fichiers auxiliaires pour le tracé.

Elle doit être placée après la définition de la géométrie (après **GEOEND**).

Si un tracé des limites dues aux matériaux est requis, elle doit être placée après l'assignation des matériaux aux régions.

Il est préférable de la placer avant les éventuelles options de biaisage.

La sortie doit être sous la forme d'un fichier **ASCII** ou binaire.

Illustration

```
*23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789
PLOTGEOM      0.0      1.0      0.0      0.0      1.0      0.0
              Irradiateur Curie modèle cylindrique plan z,y
              0.0      -40.0      -20.0      0.0      40.0      80.0
              0.0      1.0      0.0      0.0      0.0      1.0
              1.0      1.0      0.0      0.0
```

Légende

- **WHAT(1)** = 0.0 : les axes sont tracés.
= tout autre chose : pas d'axes.
- **WHAT(2)** = 0.0 : toutes les limites de régions sont tracées.
= tout autre chose : seules les limites entre les matériaux sont tracées.
- **WHAT(3)** = 0.0 : pas de numérotation des régions ou limites.
= tout autre chose : numérotation des régions et limites (pas encore activé).
- **WHAT(4)** > 0.0 : longueur maximale de chaque ver (segment ou série de segments).
< 0.0 : compression des vers effective.
Valeur par défaut : 2000.0.
- **WHAT(5)** = 0.0 : pas d'impression de diagnostic.
= tout autre chose : écriture de l'historique de l'unité logique de sortie identifiée dans **WHAT(6)**, le fichier de sortie sera **PLOTGEOM.OUT**.
- **WHAT(6)** = numéro de l'unité logique pour l'entrée de **PLOTGEOM**. Si elle est différente de 0.0 ou 5.0, l'entrée **PLOTGEOM** doit être contenue dans un fichier noté PLG.GFXINDAT. Par défaut = **LUNIN** (= 5.0), c'est-à-dire, l'entrée **PLOTGEOM** suit immédiatement.
- **SDUM** = FORMAT : le fichier **PLOTGEOM.STORE** sera formaté (**ASCII**).
= tout autre chose : le fichier ne sera pas formaté (binaire).

Si **WHAT(6)** = 0.0 ou 5.0, les 4 lignes suivantes sont obligatoires :

Première ligne, format A80 : **le titre de la section géométrique tracée**

Seconde ligne, format 6E10.5 : **X0, Y0, Z0, X1, Y1, Z1**.

Troisième ligne, format 6E10.5 : **TYX, TYY, TYZ, TXX, TXY, TXZ.**

Quatrième ligne, format 4E10.5 : **EXPANY, EXPANX, NX, NY.**

Avec :

X0, Y0, Z0 : coordonnées réelles du coin inférieur gauche de l'image.

X1, Y1, Z1 : coordonnées réelles du coin supérieur droit de l'image.

TYX, TYY, TYZ : cosinus directeurs de l'axe Y du graphe.

TXX, TXY, TXZ : cosinus directeurs de l'axe X du graphe.

EXPANY, EXPANX : facteurs d'expansion (> ou = 1) sur les échelles X et Y.

NX, NY : grille de « scan », par défaut NX = NY = 300, pour les matériaux, régions, champs magnétiques.

L'étape suivante est l'exécution de **FLUKA** avec l'obtention du fichier **PLOTGEOM.STORE**

Remarque : On peut l'obtenir très rapidement en simulant l'historique d'une seule particule, l'important en l'occurrence étant d'avoir le fichier.

⇒ Note : on peut également obtenir ce fichier en utilisant le programme **ALIFE**, en sélectionnant **unformatted** en format de sortie (description complète en [annexe D.](#)).

Signification des couleurs dans ce qui suit :

gras noir : texte écrit dans la fenêtre de votre terminal **UNIX**,

Italique vert : texte à entrer ou touche à presser par l'utilisateur,

Bleu : explications concernant les diverses options,

Rouge : points et remarques importants.

8.2. Fichier « résumé » d'USRBIN

Il s'agit d'exécuter la commande **\$FLUPRO/flutil/usbsuw** (pour des fichiers binaires, donc).

Pour cela, il faut compiler le fichier **usbsuw.f** disponible dans la distribution **FLUKA**.

La commande est la suivante : **g77 -o usbsuw usbsuw.f**

Lorsqu'**usbsuw** est lancé :

Type the input file: *irradie1001_fort.76*

Type the input file: *irradie1002_fort.76*

Type the input file: *irradie1003_fort.76*

Type the input file: *<return>* (= "pas d'autre fichier")

Type the output file name: *irradie1.sum*

Dans l'exemple ci-dessus, les fichiers de sortie **USRBIN** sont inscrits dans l'unité 76, décidé dans la carte **USRBIN**, cf. la partie dédiée à cet estimateur de ce document, pour trois cycles, le fichier d'entrée de **FLUKA** s'appelle ici **irradie1.inp**

Le fichier de sortie de la routine **usbsuw** est binaire et est nommé (choix de l'utilisateur) :

irradie1.sum

8.3. Fichiers auxiliaires pour le tracé du graphe

Cette étape va créer le graphe.

Il s'agit d'exécuter la commande **\$FLUPRO/flutil/pawlevbin**.

Il faut compiler et lier le fichier **pawlevbin.f** disponible dans la distribution **FLUKA**.

La commande est la suivante :

\$FLUPRO/flutil/lflukac -C -o pawlevbin pawlevbin.f

Il peut survenir un problème lors de la compilation, un message du genre :

**/cern/pro/lib/libpacklib.a(yppetacctent.o): In function `ypgetacctent':
undefined reference to `yp_match' ...**

Il s'agit d'un problème de configuration avec les librairies du CERN dont se sert le script.
Le problème peut être résolu en ajoutant le drapeau `-Insl` à la commande.

Lorsque `pawlevbin` est lancé :

**PLOTGEOM file (def. PLOTGEOM.STORE:) <return> ou
nom du fichier PLOTGEOM.STORE s'il est différent**

Rotate geometry? (y,n), def. n: <return>

Type the "X" expansion factor: <return>

Type the "Y" expansion factor: <return>

Peut écraser le choix effectué dans l'option PLOTGEOM

(Possible) compressed output file (def=""=no): <return>

Swap plotgeom axis (def=n): <return> ou y

En cas de « y », la ligne suivante apparaît :

Swap : both geo and bin axis (def=0), geo only (1) bin only (2)?

**Il s'agit de préciser ce qu'il faut changer : la géométrie (tapez 1), le fichier bin (tapez 2),
ou les deux (tapez 0)**

**Swap = échange : permet d'inverser les axes du graphe, souvent cela sert à faire
correspondre ceux du graphe de géométrie et ceux du graphe de données.**

-40. 50.

X min – X max (def. provided): -30. 50.

-65. 60.

Y min – Y max (def. provided): -60. 60.

Pour ajuster les limites du graphe

Bin file : *irradie.sum*

**Pour tracer la géométrie uniquement, il suffit d'appuyer sur <return> au lieu de donner le
nom du fichier**

Density file (name) or density value (pos.) or norm. (neg.): <return>

**Pour calculer les doses, on peut utiliser cette méthode en produisant un fichier des
densités, ou donner une valeur (positive pour une densité ou négative pour une
normalisation). La normalisation est donnée ainsi : -1/facteur d'échelle**

Threshold density: <return>

Rho threshold: 1.E-30

Which binning: 1

**Il peut y avoir plus d'un « binning » dans un même fichier de sortie, dans l'exemple, il
s'agit d'un maillage cartésien, défini dans USRBIN.**

***** SUM FILE *****

NB 1 n

-700. 200. 100

**Xmin – Xmax (def. provided): <return> ou au choix de l'utilisateur dans les limites
définies**

Nx (def. N_x_max): <return>

-300. 400. 100

Ymin – Ymax (def. provided): <return>

Ny (def. N_y_max): <return>

-200. 300. 100

Zmin – Zmax (def. provided): <return>

**Nz (def. N_z_max): 1 Sélection de l'axe de projection, ici le graphe représentera le plan
(x, y) IMPORTANT, à ne pas oublier.**

IRAXIS, IZAXIS 1 2

ZTALOW, ZTAHGH, DZTABN -300. 400. 7.

RADLOW, RADHGH, DRADBN -700. 200. 9.

0.000292883371 0.

Binning integral: 2441.39397

Valeur de l'intégrale

3.62995155E-11 0.000292883371

Valeurs minimale et maximale représentées sur le graphe sauf instruction contraire par la suite

<return> Pour terminer

Le programme **pawlevbin** produit les fichiers suivants :

pawlevh.poi : coordonnées (x, y) de chaque segment séparés par une ligne vide ;
pawlevh.npo : nombre de points décrivant chaque segment ;
pawlevh.lim : limites du tracé ;
pawlevh.his : histogramme **paw** avec les données de la projection sélectionnée ;
pawlevh.dat : valeurs **ASCII** de **pawlevh.his** ;
plotgeom.his : histogrammes **paw** avec les régions, matériaux, réseaux, composantes du champ magnétique (Bx, By, B) ;
plotgeom.dat : valeurs **ASCII** de **plotgeom.his**.

8.4. Tracé du graphe avec PAW

Certains programmes nécessitent de les configurer à l'environnement local avant de les utiliser à l'aide de la commande **use** (pas systématique, quelquefois **paw** est déjà configuré).

```
> use paw
> paw
```

Calling CERN version of paw-X11

```
*****
*
*          WELCOME   To   PAW          *
*
*          Version 2.13/08   19 September 2002   *
*
*****
Workstation type (?=HELP) <CR>=1: <return>
Version 1.28/07 of HIGZ started
***No default PAWLOGON file "/home/wurth/.pawlogon.kumac found"
```

PAW > exe \$FLUPRO/flutil/bpawlev

Cette commande ouvre le graphe créé par pawlevbin dans l'utilitaire de tracé paw, elle lance la macro « bpawlev » proposée dans la distribution de FLUKA
Les lignes qui suivent ne constituent que des options de tracé

Unit 1 is not active

Input file (<CR>=pawlevh) <return> Les données d'entrée sont lues dans pawlevh
Postscript output file (<CR>=pawlevh.ps) <return> ou au choix de l'utilisateur : toto.ps

Le nom du fichier de sortie postscript créé, par défaut : pawlevh.ps

Metafile type (PS-PORT = -111, PS-LAND = -112, EPS = -113) (<CR> = -112) <return>

Formats possibles du graphe, PS-portrait, PS-paysage, EPS, par défaut PS-paysage

Title (<CR>=) <return> ou au choix de l'utilisateur : Le graphe de toto

Le titre du graphique tel qu'il apparaîtra sur la figure (accents non acceptés)

Input X/Y stretching factor

Pour étirer l'un des axes ou les deux

Input factor X (<CR>=1) <return>

Input factor Y (<CR>=1) <return>

Plot type:

D = Data*

P = R-Z-Phi

G = Geometry only

R = Regions

M = Materials

F = Magnetic field

L = Lattice

Macro: plt ? (<CR>=) **D** ou au choix de l'utilisateur selon ce qu'il désire tracer
Type de graphe, données (D), Diagramme polaire (R-Z-Phi), tracé de la géométrie
uniquement (G), des régions (R), des champs magnétiques (F), des réseaux (L)

Min and Max value in matrix 8.8588E-05, 10.2613

Macro: minval ? (<CR>=8.8588E-05) <return> ou au choix de l'utilisateur

Valeur minimale pour le graphe

Macro: maxval ? (<CR>=10.2613) <return> ou au choix de l'utilisateur

Valeur maximale pour le graphe

Enter n. of divisions per decade (<CR>=3) <return> ou au choix de l'utilisateur

Nombre de divisions de couleur par décade

X axis label

Macro plotgeom: titxaxis ? (<CR>=Z(cm)) <return> ou au choix de l'utilisateur

Légende de l'axe des X

Y axis label

Macro plotgeom : tityaxis ? (<CR>=R(cm)) <return> ou au choix de l'utilisateur

Légende de l'axe des Y

Une fenêtre s'ouvre, montrant le graphe et en même temps un fichier **postscript** est créé portant le nom spécifié précédemment dans le répertoire sur lequel pointe **\$FLUPRO**.

Il y a également la possibilité de faire une capture d'écran, en sélectionnant la touche **impr écran**.

8.5. Exemple

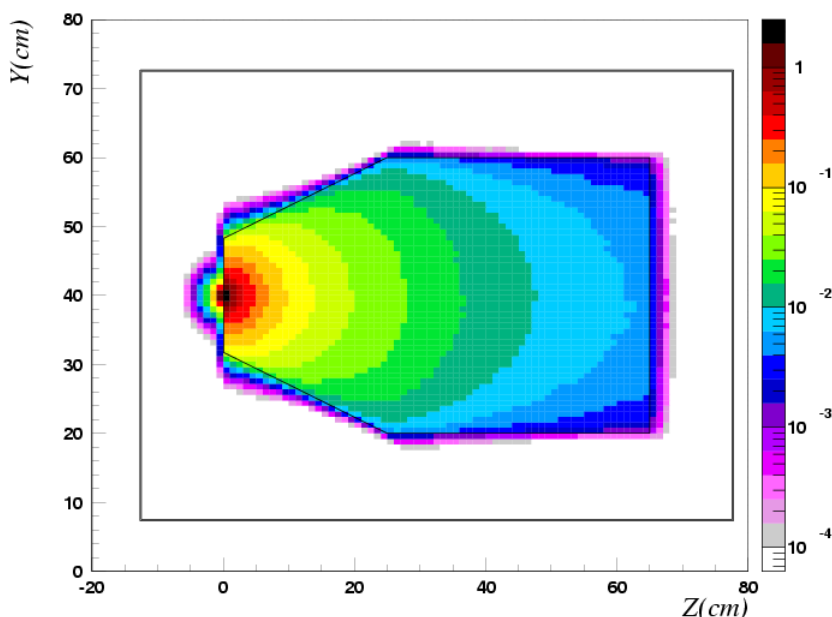
Simulation de l'irradiateur de l'institut Curie, données en Gy/s.

Source de césium 137 de $1.66 \cdot 10^{14}$ Bq en juin 2004.

Plan (z, y) :

La géométrie employée est :
Xmin : 40. Ymin : 0. Zmin : -20.

Xmax : 40. Ymax : 80. Zmax : 80.



On voit que pour cette méthode de calcul, la statistique est satisfaisante avec 5 millions d'histoires prises en compte. Les domaines d'isodoses se découpent de façon presque régulière. Le calcul a été effectué sur une maille assez conséquente englobant tout l'irradiateur (celle décrite dans la partie consacrée à la carte **USRBIN**, cf. **7. 2.** du présent document).

Plus de précisions à propos de ce cas dans le document spécialement dédié à cette étude de modélisation de l'irradiateur de type IBL637 de l'Institut Curie.

9. Autre méthode pour tracer des graphes des fichiers de sortie USRBIN

(Merci à Maher Cheikh MHAMED, utilisation de la routine deq99c.f dans l'exécutable)

Étapes :

- 1 – Format des fichiers de sortie
- 2 – Lien symbolique
- 3 – Fichier de liste, propriétés du graphe
- 4 - Tracé de la géométrie
- 5 – Création du graphe
- 6 – Tracé du graphe avec PAW

9.1. Format des fichiers de sortie

Dans les données de la première carte **USRBIN**, l'unité logique du fichier de sortie généré après calcul est donnée par **WHAT(3)**.

Si **WHAT(3) > 0.0**, les données sont formatées (c'est le cas ici).

Si **WHAT(3) < 0.0**, les données sont non formatées.

Il est conseillé de ne pas choisir de valeur inférieure à 21.0 (en valeur absolue), à l'exception de 11.0.

Exemples :

WHAT(3) = 60.0 pour la fluence totale, le fichier sera **inputname00X_fort.60**

WHAT(3) = 61.0 pour la fluence des gammas, le fichier sera **inputname00X_fort.61**

WHAT(3) = 62.0 pour la fluence des neutrons, le fichier sera **inputname00X_fort.62**

9.2. Lien symbolique

Supposons que l'on veuille tracer les graphes dans le répertoire **home/wurth/graphes**, et que les résultats soient placés dans **home/wurth/graphes/sim**

Nous allons créer un lien symbolique dans **home/wurth/graphes**, indiquant que les fichiers à lire sont dans **home/wurth/graphes/sim**

La syntaxe est la suivante (si l'on se trouve dans **home/wurth/graphes**)

ln -s nom_du_fichier_réel nom_du_fichier_symbolique

Dans notre exemple : **ln -s sim lien_sim**

Dans **home/wurth/graphes**, un **lien_sim** apparaît alors, en cliquant dessus dans une fenêtre d'exploration ou en l'activant par le terminal, on accède au répertoire **home/wurth/graphes/sim** et à son contenu.

9.3. Fichier de liste, propriétés du graphe

Ce fichier à créer en utilisant un éditeur de texte **unix/linux** (**gedit**, **nedit**, ...) contient les instructions nécessaires au tracé du graphe. De la même façon que pour le fichier d'entrée de **FLUKA**, il est nécessaire de suivre une syntaxe stricte afin que ces instructions soient bien interprétées.

Voyons un exemple :

```
1
preproc.out
trkl
1.0e-12
```

```

2.25e+17
5
61 140
3 1
1.0 lien_sim/inputname001_fort.62
1.0 lien_sim/inputname002_fort.62
1.0 lien_sim/inputname003_fort.62
1.0 lien_sim/inputname004_fort.62
1.0 lien_sim/inputname005_fort.62
0
0
    
```

Explications :

- **preproc.out** sera le fichier de sortie.
- **trkl** indique qu'il s'agit d'un « track length estimator » (*cf. manuel dans la partie dédiée à USRBIN et USRTRACK*).
- **1.0e-12** et **2.25e+17** sont des facteurs multiplicatifs, la valeur du fichier de sortie **USRBIN** sera multipliée par ces facteurs sur le graphe.
- **5** : le nombre de fichiers de sortie à moyenner (ce nombre peut changer, mais un nombre correspondant de noms de fichiers doit suivre).
- **61 140** : intervalle de « bins » de la projection sur lequel on calcule la moyenne.

Arrêtons-nous un peu là-dessus :

Je trace un graphe sur (z, x), l'axe de projection est donc y.

Le maillage sur y dans le fichier d'entrée est : Ymin = -2000, Ymax = 2200 avec 210 bins (intervalles), soit un pas de 20 cm.

Je désire tracer (z, x), mais en moyennant les données entre Y = -800 et Y = +800, on calcule l'intervalle à indiquer :

$$-2000 + 60 \cdot 20 = -800$$

$$-2000 + 140 \cdot 20 = 800$$

Il s'agit d'un intervalle, on écrit donc 61 140

- **3 1** : ce sont les codes pour le tracé du graphe, on donne (x, y) du graphe avec
 - 1 = X de la géométrie du problème
 - 2 = Y
 - 3 = Z

Ici, on demande le tracé de la section (z, x) de la géométrie.

Mon fichier s'appellera par exemple **inputname-62.list**

9.4. Tracé de la géométrie

On utilise **ALIFE** (*cf. annexe D*. pour l'explication détaillée).

Lancer **ALIFE**, ouvrir le fichier *.inp, choisir les bornes du tracé et l'axe de projection dans l'onglet **DRAW**, appuyer **OK**.

Si la géométrie est correctement définie et si **ALIFE** a tracé le graphe voulu, un fichier **PLOTGEOM.STORE** a été créé dans le répertoire d'**ALIFE**.

Déplacer ce fichier dans le répertoire voulu par exemple **home/wurth/graphes**

Le renommer en **plotgeom.store** : **cp PLOTGEOM.STORE plotgeom.store**

9.5. Création du graphe

Avant tout, éditer le fichier **mybins.out**.

Il permet de définir les échelles de valeurs attribuées à chaque couleur.

Par exemple :

1e0 --> soit de 1 à 1e-1 : NOIR

1e-1 --> soit de 1e-1 à 1e-2 : ROUGE

```

1e-2 --> soit de 1e-2 à 1e-3 : ORANGE
1e-3 --> soit de 1e-3 à 1e-4 : JAUNE
1e-4 --> soit de 1e-4 à 5e-5 : VERT CLAIR
5e-5 --> soit de 5e-5 à 1e-5 : VERT FONCÉ
1e-5 --> soit de 1e-5 à 5e-6 : TURQUOISE
5e-6 --> soit de 5e-6 à 1e-6 : BLEU CLAIR
1e-6 --> soit de 1e-6 à 5e-7 : BLEU CIEL
5e-7 --> soit de 5e-7 à 1e-7 : BLEU MARINE
1e-7 --> soit de 1e-7 à 0 : BLANC
0e0
    
```

Lorsque l'échelle est choisie, lancer **prepxyz** et le fichier ***.list** :
> prepxyz < inputname-62.list

Le programme va tourner, produire des messages de bonne exécution (ou d'erreur si quelque chose ne va pas), faire autant de cycles que de fichiers de sortie **USRBIN** attachés au ***.list**.

9.6. Tracé du graphe avec PAW

Dans le terminal, se placer dans le répertoire de travail choisi précédemment.

Dans notre exemple : **home/wurth/graphes**

> use paw

> paw

.... (laisser les options par défaut, taper **enter**)

PAW > exec cplot

PAW > exit

> cp cplot.eps le_nom_de_mon_graphe.eps

> gv le_nom_de_mon_graphe.eps

Ghostview affichera **cplot.eps**, mais il s'agit bien de votre fichier.

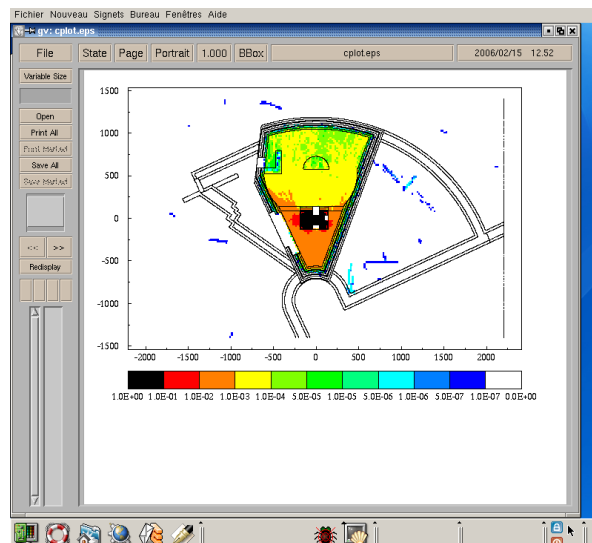
Cf. exemple à la page suivante.

Configuration minimale pour tracer des graphes :

Placer dans un répertoire, par exemple **home/wurth/graphes** les fichiers suivants :

- **prepxyz.f, prexyz, preproc.out, mybins.out, *.list**
- Un lien symbolique pointant vers les fichiers de sortie listés dans le fichier ***.list**
- **plotgeom.store, cplot.kumac, cplg.ftn, cplinit.ftn, dosepl.ftn, legend.ftn, plg.ftn, pltleg.ftn, usrplt.ftn, zoom.ftn**

Exemple :



10. Utilisation de l'option SOURCE

10.1. Mode d'emploi de l'option SOURCE

Elle invoque l'utilisation de la routine **source.f** préalablement modifiée par l'utilisateur selon les nécessités de son étude, compilée via le script **fff** et liée à l'exécutable de **FLUKA**. Cette option est employée pour simuler les particules primaires lorsque les caractéristiques de la distribution sont trop complexes pour être décrites à l'aide des seules cartes **BEAM** et **BEAMPOS** (néanmoins, à partir de la version 2011.2, certaines distributions peuvent être directement décrites dans la définition des caractéristiques, cf. § 3.2 du présent document).

Elle permet d'entrer jusqu'à 12 paramètres en double précision, choisis par l'utilisateur et pouvant être reliés à la routine **SOURCE**. Si le nombre de paramètres à entrer est supérieur à 6, 2 cartes **SOURCE** sont requises.

Première carte

- **WHAT(1), ..., WHAT(6)** : paramètres à passer dans la routine **SOURCE** représentés par les valeurs en double précision **WHASOU(1), ..., WHASOU(6)** (via COMMON SOURCM).
- **SDUM** = tout caractère (ne contenant pas "&"), à passer dans la routine **SOURCE** en tant que variable **SDUSOU** (via COMMON CHEPSR).

Seconde carte (si présente)

- **WHAT(1), ..., WHAT(6)** : paramètres à passer dans la routine **SOURCE** représentés par les valeurs en double précision **WHASOU(7), ..., WHASOU(12)**.
- **SDUM** = "&" n'importe où dans les colonnes 71 à 78

Par défaut, si l'option **SOURCE** n'est pas donnée : la routine **SOURCE** n'est pas invoquée.

Exemple :

```
SOURCE      0.5      1.173E-3  1.333E-3
```

Ici, la routine **SOURCE** est appelée et trois paramètres sont définis.

Dans le fichier **source.f**, ces trois paramètres sont identifiés par **WHASOU(1)**, **WHASOU(2)** et **WHASOU(3)**.

Avec $WHASOU(1) = 0.5$, $WHASOU(2) = 1,173E-3$ et $WHASOU(3) = 1,333E-3$

Ceci peut s'avérer très utile en cas de changements fréquents de ces paramètres pour les besoins de l'étude, par exemple. Cela permet de changer les valeurs dans le fichier d'entrée sans avoir à recompiler la routine **SOURCE** et l'exécutable **FLUKA**.

10.2. Définir une source isotrope de cobalt 60

Il s'agit de simuler une source émettant dans tout l'espace, avec la même probabilité, deux photons, l'un d'énergie 1,173 MeV, l'autre d'énergie 1,333 MeV.

10.2.1. Créer une source produisant deux photons

Il s'agit d'affecter une probabilité à une énergie. Pour cela, nous proposons un test par rapport à un nombre aléatoire compris entre 0 et 1.

Dans le cas du cobalt 60, les deux raies sont émises avec une probabilité égale, valant 1.

Il suffit donc de proposer une distribution « moitié-moitié ». Le nombre aléatoire est comparé à 0,5. S'il est inférieur ou égal à cette valeur, on affecte à la particule l'énergie 1,173 MeV, sinon on lui affecte l'énergie 1,333 MeV.

Nous suivons plus d'un million de particules dites primaires, ceci nous garantit une distribution uniforme. On peut affirmer qu'une particule sur deux sera affectée de l'énergie 1,173 MeV. La réalité est donc simulée avec fidélité.

On peut étendre ce principe à des radioéléments polyénergétiques en affectant des probabilités liées aux rapports d'embranchements pour les raies considérées.

Illustration : reproduction de la section décrivant l'énergie des particules dans **source.f**

[...]

* Kinetic energy of the particle (GeV)

* *Stratagème de G. Battistoni pour simuler une source non monoénergétique*

* *FLRNDM(XDUMMY) : nombre aléatoire compris entre 0 et 1*

* *WHASOU(n) : les champs de la carte SOURCE du fichier d'entrée (cf.1).*

* *Cela permet de modifier les paramètres sans avoir à recompiler.*

* *WHASOU(1) est la valeur test pour le nombre aléatoire.*

* *WHASOU(2) est la première valeur d'énergie.*

* *WHASOU(3) est la seconde valeur d'énergie.*

```
IF (FLRNDM(XDUMMY) .LE. WHASOU(1)) THEN
    PBEAM = WHASOU(2)
```

```
ELSE
```

```
    PBEAM = WHASOU(3)
```

```
ENDIF
```

```
TKE (LSTACK) = SQRT ( PBEAM**2 + AM (IJBEAM)**2 ) - AM (IJBEAM)
```

* Particle momentum

```
PMOM (LSTACK) = PBEAM
```

```
* PMOM (LSTACK) = SQRT ( TKE (LSTACK) * ( TKE (LSTACK) + TWOTWO
```

```
* & * AM (ILO(LSTACK)) ) )
```

[...]

10.2.2. Créer une source isotrope

Il s'agit d'affecter des valeurs aux cosinus directeurs du faisceau de particules afin d'obtenir une distribution angulaire sur 4π .

$A = 1 - 2 * X$, avec X : nombre aléatoire compris entre 0 et 1

⇒ Si $A > 1$, $A = 1$

⇒ Si $A < -1$, $A = -1$

$B = 2\pi * Y$ avec Y : nombre aléatoire compris entre 0 et 1

$$T_x = \sqrt{(1-A^2)} * \cos(B)$$

$$T_y = \sqrt{(1-A^2)} * \sin(B)$$

$$T_z = A$$

On décrit ainsi tout l'espace sur 4π .

Illustration : reproduction de la section définissant les cosinus directeurs du faisceau

[...]

* Cosines (tx,ty,tz)

*** *John Clem's method to define an isotropic source* ***

* *ONEONE = 1, TWOPIP = 2\pi*

* *FLRNDM(nDUMMY) avec n = Y,Z : nombres aléatoires compris entre 0 et 1*

```
COSTH = ONEONE - 2.D0*FLRNDM(YDUMMY)
```

```

IF (COSTH.GT.ONEONE) THEN
COSTH = ONEONE
ELSEIF (COSTH.LT.-ONEONE) THEN
COSTH = -ONEONE
ENDIF
ANG = TWOPI*FLRNDM(ZDUMMY)
UBEAM = SQRT(ONEONE - COSTH*COSTH)*COS(ANG)
VBEAM = SQRT(ONEONE - COSTH*COSTH)*SIN(ANG)
WBEAM = COSTH
*** End of John's code ***
TX      (LSTACK) = UBEAM
TY      (LSTACK) = VBEAM
TZ      (LSTACK) = WBEAM
[...]
```

La reproduction complète du fichier **source.f** pour simuler une source isotrope de cobalt 60 figure en **annexe C**.

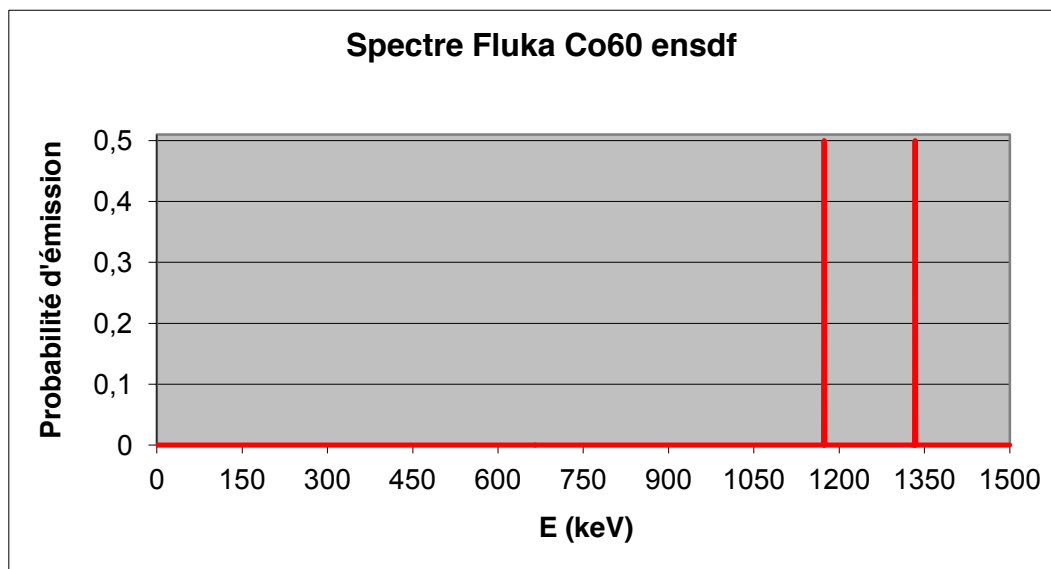
Remerciements chaleureux à Giuseppe Battistoni, John Clem, Alberto Fasso.

Compilation :

\$FLUPRO/flutil/fff source.f, obtention de **source.o**

\$FLUPRO/flutil/lfluka -o myfluka -m fluka source.o, la routine **SOURCE** sera fonctionnelle avec l'exécutable **myfluka** dans ce cas.

Ci-dessous : Un spectre en énergie représentant les résultats d'une simulation de 100 millions de particules sur 5 cycles de 20 millions. Pour calculer la fluence des photons, nous avons utilisé un estimateur **USRTRACK** avec 1 500 intervalles d'énergie linéaires de 0 à 1 500 keV dans une sphère de vide d'1 cm de rayon. La valeur de l'énergie portée sur le graphe est celle du milieu de chaque intervalle (largeur : 1 keV).



10.3. Définir une source isotrope d'euporium 152

Nous procédons de la même façon que précédemment, sauf que le problème est plus complexe ici. Les raies sont nombreuses et les rapports d'embranchement plus complexes. Néanmoins, en appliquant un raisonnement similaire, il est possible de simuler une telle source.

Nous étudions tout d'abord la littérature, la base de données **ENSDF-MIRD** au lien http://www.nndc.bnl.gov/useroutput/152eu_mird.html afin de relever les énergies des raies

et leurs probabilités d'émission. Nous sélectionnons les 14 raies les plus intenses, calculons leurs probabilités absolues d'être émises pour un photon incident et y associons la valeur normalisée à 100 % (cf. tableau ci-dessous).

	E (keV)	Intensité (%)	Probabilité absolue (%)
1	121,8	28,7	19,18
2	244,7	7,61	5,09
3	344,3	26,6	17,78
4	411,1	2,24	1,5
5	444,0	2,83	1,89
6	778,9	13,0	8,69
7	867,4	4,26	2,85
8	964,1	14,6	9,76
9	1086	10,2	6,82
10	1090	1,73	1,16
11	1112	13,7	9,16
12	1213	1,43	0,96
13	1299	1,63	1,09
14	1408	21,1	14,1
Somme		149,63	100

Il s'agit à présent de créer les intervalles pondérés par ces probabilités afin de rendre le test conforme à cette distribution sur des milliers / millions de tirages d'un nombre aléatoire.

Soit A, le nombre aléatoire compris entre 0 et 149,63 :

- Si A est inférieur à 28,7, l'énergie vaut 121,8 keV.
- Si A est supérieur ou égal à 28,7 et inférieur à 36,31, l'énergie vaut 244,7 keV.
- Si A est supérieur ou égal à 36,31 et inférieur à 62,91, l'énergie vaut 344,3 keV.
- Si A est supérieur ou égal à 62,91 et inférieur à 65,15, l'énergie vaut 411,1 keV.
- Si A est supérieur ou égal à 65,15 et inférieur à 67,98, l'énergie vaut 444,0 keV.
- Si A est supérieur ou égal à 67,98 et inférieur à 80,98, l'énergie vaut 778,9 keV.
- Si A est supérieur ou égal à 80,98 et inférieur à 85,24, l'énergie vaut 867,4 keV.
- Si A est supérieur ou égal à 85,24 et inférieur à 99,84, l'énergie vaut 964,1 keV.
- Si A est supérieur ou égal à 99,84 et inférieur à 110,04, l'énergie vaut 1086 keV.
- Si A est supérieur ou égal à 110,04 et inférieur à 111,77, l'énergie vaut 1090 keV.
- Si A est supérieur ou égal à 111,77 et inférieur à 125,47, l'énergie vaut 1112 keV.
- Si A est supérieur ou égal à 125,47 et inférieur à 126,9, l'énergie vaut 1213 keV.
- Si A est supérieur ou égal à 126,9 et inférieur à 128,53, l'énergie vaut 1299 keV.
- Si A est supérieur ou égal à 128,53 et inférieur à 149,63, l'énergie vaut 1408 keV.

Dans la routine **source.f**, cela donne :

[...]* Kinetic energy of the particle (GeV)

* *Stratagème de Sébastien Wurth (20/03/2007) pour simuler une source d'Eu152*

```
* Kinetic energy of the particle (GeV)
  XXX = 149.63d0 * FLRNDM(XDUMMY)
  IF (XXX .LT. 28.7d0) THEN
    PBEAM = 121.8d-6
  ELSE IF (XXX .GE. 28.7d0 .AND. XXX .LT. 36.31d0)
    & THEN
    PBEAM = 244.7d-6
  ELSE IF (XXX .GE. 36.31d0 .AND. XXX .LT. 62.91d0)
    & THEN
    PBEAM = 344.3d-6
  ELSE IF (XXX .GE. 62.91d0 .AND. XXX .LT. 65.15d0)
    & THEN
```

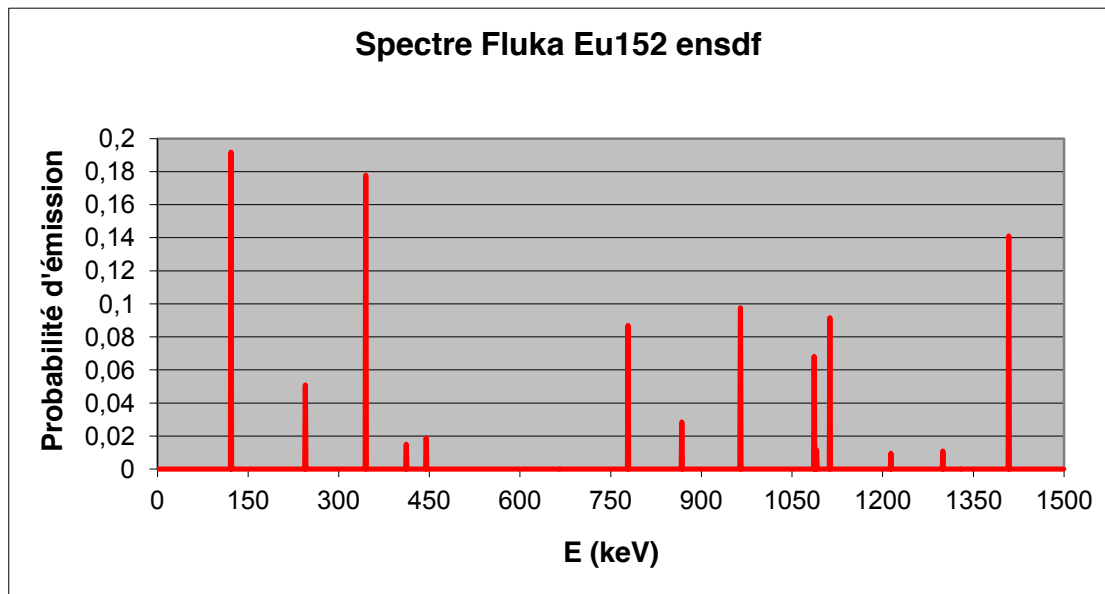


```

PBEAM = 411.1d-6
ELSE IF (XXX .GE. 65.15d0 .AND. XXX .LT. 67.98d0)
& THEN
PBEAM = 444.0d-6
ELSE IF (XXX .GE. 67.98d0 .AND. XXX .LT. 80.98d0)
& THEN
PBEAM = 778.9d-6
ELSE IF (XXX .GE. 80.98d0 .AND. XXX .LT. 85.24d0)
& THEN
PBEAM = 867.4d-6
ELSE IF (XXX .GE. 85.24d0 .AND. XXX .LT. 99.84d0)
& THEN
PBEAM = 964.0d-6
ELSE IF (XXX .GE. 99.84d0 .AND. XXX .LT. 110.04d0)
& THEN
PBEAM = 1086.0d-6
ELSE IF (XXX .GE. 110.04d0 .AND. XXX .LT. 111.77d0)
& THEN
PBEAM = 1090.0d-6
ELSE IF (XXX .GE. 111.77d0 .AND. XXX .LT. 125.47d0)
& THEN
PBEAM = 1112.d-6
ELSE IF (XXX .GE. 125.47d0 .AND. XXX .LT. 126.9d0)
& THEN
PBEAM = 1213.d-6
ELSE IF (XXX .GE. 126.9d0 .AND. XXX .LT. 128.53d0)
& THEN
PBEAM = 1299.d-6
ELSE IF (XXX .GE. 128.53d0 .AND. XXX .LT. 149.63d0)
& THEN
PBEAM = 1408.d-6
END IF
TKEFLK (NPFLKA) = SQRT ( PBEAM**2 + AM (IONID)**2 ) - AM (IONID)

```

Ci-dessous : Un spectre en énergie représentant les résultats d'une simulation de 100 millions de particules sur 5 cycles de 20 millions. Pour calculer la fluence des photons, nous avons utilisé un estimateur **USRTRACK** avec 1 500 intervalles d'énergie linéaires de 0 à 1 500 keV dans une sphère de vide d'1 cm de rayon. La valeur de l'énergie portée sur le graphe est celle du milieu de chaque intervalle (largeur : 1.0 keV).



10.4. Définir une source isotrope de neutrons (Am-Be)

Il s'agit de simuler un spectre d'une source isotrope d'Américium-Béryllium émettrice de neutrons en rendant discrètes les variations continues sur un certain nombre d'intervalles, le problème est plus ardu puisqu'il s'agit de simuler le transport de neutrons.

Il convient de se munir d'un spectre représentant la probabilité d'émission de la source en fonction de l'énergie des neutrons.

10.4.1. Spectre d'émission typique d'une source d'américium-béryllium

Il faut considérer que le code de calcul **FLUKA** utilise la technique « multigroupe » qui consiste à définir les neutrons lents en 260 ou 72 groupes d'énergie dont les intervalles sont de tailles équivalentes (en échelle logarithmique).

Les énergies considérées sont comprises entre $1e-11$ MeV et 20 MeV (pour 260 groupes) et 19,6 MeV (pour 72 groupes).

Il s'agit à présent de décrire les probabilités d'émissions par intervalles d'énergies tels qu'ils sont présentés dans la norme **ISO 85-29** (53 intervalles). Nous utilisons pour cela la routine **source.f**.

FLUKA effectuera tout seul l'adaptation aux 260 ou 72 groupes. Notons que cette solution permet de reproduire le spectre avec davantage de précision.

Les valeurs des intervalles d'énergie donnés par la norme **ISO 85-29** tels que définis dans la routine **source.f**.

```
* Neutron energy group boundaries
  DATA ENEDGE /
& 4.140000000E-10, 1.100000000E-04, 3.300000000E-04, 5.400000000E-04,
& 7.500000000E-04, 9.700000000E-04, 1.180000000E-03, 1.400000000E-03,
& 1.610000000E-03, 1.820000000E-03, 2.040000000E-03, 2.250000000E-03,
& 2.470000000E-03, 2.680000000E-03, 2.900000000E-03, 3.110000000E-03,
& 3.320000000E-03, 3.540000000E-03, 3.750000000E-03, 3.970000000E-03,
& 4.180000000E-03, 4.390000000E-03, 4.610000000E-03, 4.820000000E-03,
& 5.040000000E-03, 5.250000000E-03, 5.470000000E-03, 5.680000000E-03,
& 5.890000000E-03, 6.110000000E-03, 6.320000000E-03, 6.540000000E-03,
& 6.750000000E-03, 6.960000000E-03, 7.180000000E-03, 7.390000000E-03,
& 7.610000000E-03, 7.820000000E-03, 8.030000000E-03, 8.250000000E-03,
& 8.460000000E-03, 8.680000000E-03, 8.890000000E-03, 9.110000000E-03,
& 9.320000000E-03, 9.530000000E-03, 9.750000000E-03, 9.960000000E-03,
& 1.018000000E-02, 1.039000000E-02, 1.060000000E-02, 1.082000000E-02,
& 1.103000000E-02/
```

Les valeurs de probabilité par intervalle d'énergie en cumulé tel que définies dans **source.f**.

```
* Cumulative Am-Be spectrum
  DATA AMBE / 0.D0,
* . . . + . . . 1 . . . + . . . 2 . . . + . . . 3 . . . + . . . 4 . . . + . . . 5 . . . + . . . 6 . . . + . . . 7 . .
& 0.000000000E+00, 1.436481239E-02, 4.776543140E-02, 7.904037101E-02,
& 1.071627445E-01, 1.321669549E-01, 1.535298579E-01, 1.733626318E-01,
& 1.908343064E-01, 2.100845311E-01, 2.323385094E-01, 2.537980206E-01,
& 2.762823182E-01, 2.990502398E-01, 3.285590304E-01, 3.641472340E-01,
& 4.010032447E-01, 4.355894638E-01, 4.662507516E-01, 4.962406827E-01,
& 5.231495538E-01, 5.517784701E-01, 5.835652529E-01, 6.143046473E-01,
& 6.476476615E-01, 6.780626285E-01, 7.054456396E-01, 7.287669079E-01,
& 7.493534454E-01, 7.675069776E-01, 7.851818693E-01, 8.055774907E-01,
& 8.238785353E-01, 8.401788111E-01, 8.569538269E-01, 8.737621455E-01,
& 8.925971352E-01, 9.109730861E-01, 9.278550110E-01, 9.422084224E-01,
& 9.518865392E-01, 9.584078896E-01, 9.626634488E-01, 9.663306583E-01,
& 9.701368796E-01, 9.751953065E-01, 9.814492343E-01, 9.869689002E-01,
& 9.916447948E-01, 9.953409068E-01, 9.981225415E-01, 9.996366693E-01,
& 1.000000000E+00/
```

10.4.2. Algorithme pour décrire les probabilités d'émissions par intervalle

L'algorithme permettant la simulation d'une source de neutrons de type Am-Be est le suivant (Merci à **Nisy Ipe**) :

```
* Samples the energy group
  XI = FLRNDM(XDUMMY)
  DO 500 K = 1, 53
    IF(XI .LE. AMBE(K)) THEN
      ENERGY = ENEDGE(K-1) +
      & (XI-AMBE(K-1)) * (ENEDGE(K) - ENEDGE(K-1)) / (AMBE(K) - AMBE(K-1))
      GO TO 600
    END IF
  * CONTINUE
  K=K+1
```

- Nous définissons **ENEDGE**, les 53 valeurs formant les intervalles d'énergie décrits dans la norme.
- Nous définissons **X**, un nombre aléatoire compris entre 0 et 1.
- Nous définissons **AMBE**, les 54 valeurs représentant la probabilité d'émission cumulée en fonction des intervalles d'énergie (La valeur pour le dernier intervalle vaut un).
- Nous effectuons une boucle pour des valeurs de **K** allant de 1 à 53 (par incrément de 1) et effectuons le test suivant pour une valeur de **K** donnée : si **X** est inférieur ou égal à **AMBE(K)**, alors son énergie (variable **ENERGY**) est définie ainsi :

$$ENERGY = ENEDGE(K) - \frac{(X - AMBE(K - 1)) * (ENEDGE(K) - ENEDGE(K - 1))}{(AMBE(K) - AMBE(K - 1))}$$

Cette valeur d'énergie est l'énergie cinétique du neutron attribué à **FLUKA** pour sa simulation, son transport.

10.4.3. Algorithme pour assurer l'isotropie de la source

```
* Samples the polar angle
Y = FLRNDM(YDUMMY)
COSTHE = ONEONE - TWOTWO*Y
  SIN THE = SQRT(ONEONE - COSTHE**2)
  IF(Y .LE. HLFHLF) SIN THE = -SIN THE
* Samples the azimuthal angle
  Z = FLRNDM(ZDUMMY)
  PHI = TWOPIP * Z
  COSPHI = COS(PHI)
  SINPHI = SQRT(ONEONE - COSPHI**2)
  UBEAM = SIN THE * COSPHI
  VBEAM = SIN THE * SINPHI
  WBEAM = COSTHE
* Cosines (tx,ty,tz) (make sure they are properly normalized)
  TZFLK (NPFLKA) = WBEAM
  TYFLK (NPFLKA) = VBEAM
  TXFLK (NPFLKA) = UBEAM
```

- Angle polaire : nous définissons **COSTHE**, le cosinus de l'angle polaire défini pour être compris entre 1 et -1. $COSTHE = 1 - 2*Y$ où **Y** est un nombre aléatoire compris entre 0 et 1. À l'aide d'une relation bien connue, $(\cos(x))^2 + (\sin(x))^2 = 1$, nous définissons **SIN THE**, le sinus de l'angle polaire.

Nous ajoutons le test suivant, visant à uniformiser la distribution dans les valeurs positives et négatives (symétrie). Si Y est inférieur ou égal à 0.5, la valeur de **SINTHE** prend le signe négatif.

➤ Angle azimutal : nous définissons **PHI**, l'angle azimutal compris entre 0 et 2π .

$PHI = 2\pi * Z$ où **Z** est un nombre aléatoire compris entre 0 et 1.

Nous définissons **COSPHI** et **SINPHI**, les cosinus et sinus de cet angle **PHI**.

Les trajectoires (cosinus directeurs de la particule **UBEAM** selon X (**Tx**), **VBEAM** selon Y (**Ty**) et **WBEAM** selon Z (**Tz**) seront respectivement égales à :

$Tx = SINTHE * COSPHI$, $Ty = SINTHE * SINPHI$, $Tz = COSTHE$

La normalisation est vérifiée :

$Tx^2 + Ty^2 + Tz^2 = ((SINTHE)^2 * (COSPHI)^2) + ((SINTHE)^2 * (SINPHI)^2) + (COSTHE)^2 = (SINTHE)^2 * ((COSPHI)^2 + (SINPHI)^2) + (COSTHE)^2 = 1$

10.4.4. Étude comparative

À titre d'information, nous avons effectué le test sur une géométrie ultra-simple pour comparer les valeurs obtenues par l'utilisation des deux bibliothèques (72 et 260 groupes).

Il est à noter que la bibliothèque des 260 groupes est à présent celle par défaut dans **FLUKA**. Si l'on désire utiliser celle à 72 groupes, il faut forcer son utilisation, via la carte **LOW-NEUT** de la façon suivante :

```
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
LOW-NEUT      72.0      22.0      19.6E-3      0.0      1.0
```

Avec

- **WHAT(1)** : nombre de groupes de neutrons dans la bibliothèque utilisée.
- **WHAT(2)** : nombre de groupes de photons dans la bibliothèque utilisée.
- **WHAT(3)** : énergie maximale en GeV pour la librairie considérée.
- **WHAT(4)** : options d'impressions dans le fichier de sortie, 0.0 = impression d'un minimum de paramètres.
- **WHAT(5)** : nombre de groupes de neutrons thermiques pour la librairie considérée.

Pour une description détaillée de cette carte, cf. *manuel de FLUKA*.

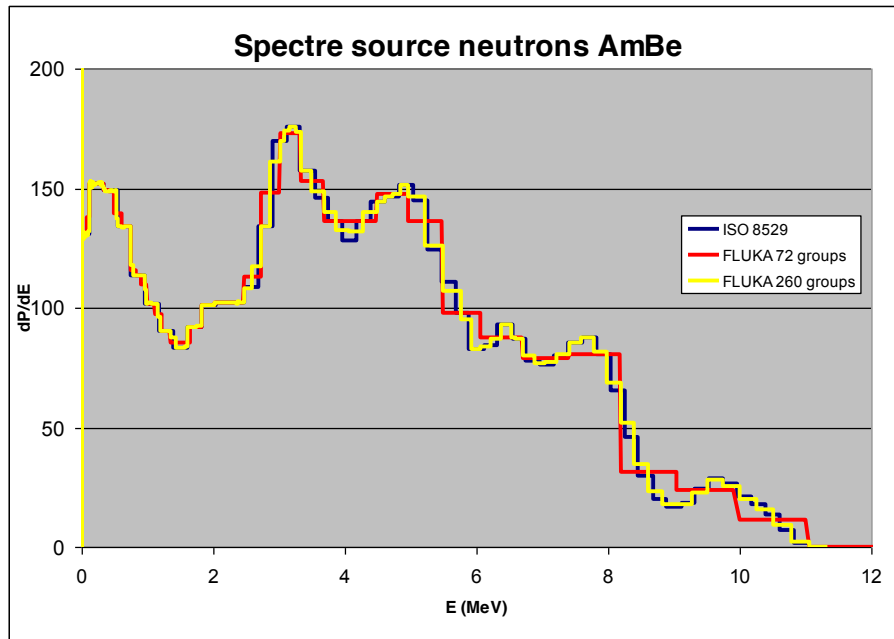
L'étude a été effectuée dans une géométrie des plus simples : une sphère vide de rayon 1 cm centrée autour du point source (considérée comme isotrope ici).

Les résultats proposés sont issus de simulations de 10 cycles de 10 millions d'histoires.

Ils sont présentés en dP/dE, c'est-à-dire en probabilité d'émission divisée par l'énergie, pour s'affranchir de la taille de ces intervalles qui varie d'une série de données à l'autre.

Bien entendu, la librairie à 260 groupes de neutrons s'accorde quasi parfaitement au spectre typique de l'AmBe.

Il est à noter qu'à partir de la version 2011.2, la librairie à 72 groupes a été déclarée obsolète et n'est plus distribuée avec le code.



Note : pour ce qui est des calculs de doses, en première approximation, il est possible de considérer une énergie moyenne des neutrons de 4,4 MeV. Une simulation simple, deux matériaux (air, béton) dans une géométrie sphérique, donne des résultats proches en termes de doses efficaces EWT74 (CIPR 74), avec une légère surestimation pour le faisceau monoénergétique par rapport à la source « réelle ».

10.5. Définir une source uniformément répartie sur une ligne

Pour définir une source isotrope, mais non ponctuelle dont l'activité est répartie de façon uniforme sur une ligne, voire sur deux lignes perpendiculaires, il faut une nouvelle fois éditer le fichier **source.f**, le compiler et le lier à l'exécutable qui servira dans le cadre de cette étude.

10.5.1. Source linéaire uniformément répartie

Pour cela, le manuel de **FLUKA** propose un exemple dans la section consacrée aux routines modifiables par l'utilisateur, section **SOURCE**.

Pour modéliser une distribution uniforme, il s'agit d'utiliser la fonction **FLRNDM(DUMMY)**, qui retourne un nombre pseudo-aléatoire en format double précision uniformément réparti entre 0 (inclus) et 1 (non inclus).

En fait, **DUMMY** peut être n'importe quel nom de variable.

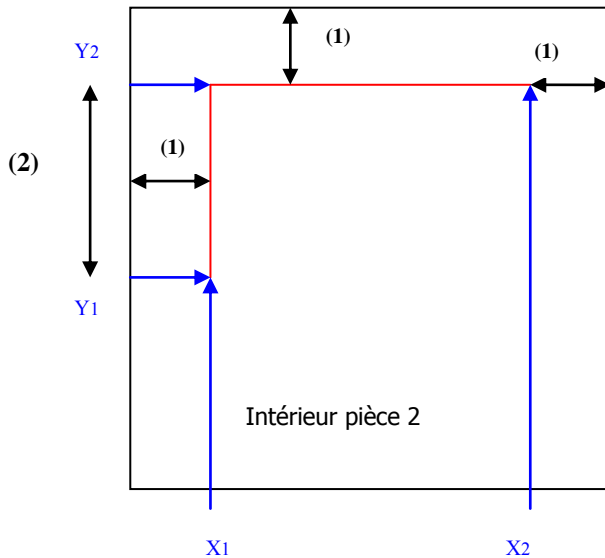
Un exemple simple de modélisation d'une distribution uniforme sur une ligne le long de l'axe Z entre $Z = 10$ et $Z = 80$:

```
Z1 = 10.D0
Z2 = 80.D0
ZFLK (NPFLKA) = 10.D0 + (Z2 - Z1) * FLRNDM(XXX)
```

À partir de cet exemple, nous allons appliquer la même méthode au cas de l'étude du bâtiment CIPA (Centre d'Imagerie Positons chez l'Animal) du CNRS Orléans (*cf. document de référence Cipa_validation_blindages_v1*).

10.5.2. Application : source « bilinéaire »

La situation à reproduire est la suivante :



Légende :

(1) : 75 cm

(2) : 150 cm

X1 = 414 cm

X2 = 650 cm

Y1 = 1435 cm

Y2 = 1585 cm

Z = 80 cm

Ligne rouge : position de la source

Ligne noire : parois internes de la pièce

Il s'agit de modéliser deux lignes ayant la même probabilité d'émission.

Le code dans le fichier **source.f** est donc modifié de la sorte :

```
* SW 08/02/07: Sampling bilinear source
* Y,Z fixed, X between 414 and 650
  ZBEAM = 80.d0
  X1 = 414.d0
  X2 = 650.d0
  Y1 = 1435.d0
  Y2 = 1585.d0
  IF ( FLRNDM(XDUMMY) .LT. WHASOU(1) ) THEN
    YBEAM = 1585.d0
    XBEAM = 414.d0 + ((X2 - X1) * FLRNDM(XXX))
* X,Z fixed, Y between 1435 and 1585
  ELSE IF ( FLRNDM(XDUMMY) .GE. WHASOU(1) ) THEN
    XBEAM = 414.d0
    YBEAM = 1435.d0 + ((Y2 - Y1) * FLRNDM(YYY))
  END IF
* End of bi-linear sampling
  XFLK (NPFLKA) = XBEAM
  YFLK (NPFLKA) = YBEAM
  ZFLK (NPFLKA) = ZBEAM
```

XBEAM, **YBEAM**, **ZBEAM** sont les coordonnées de l'origine de la source.

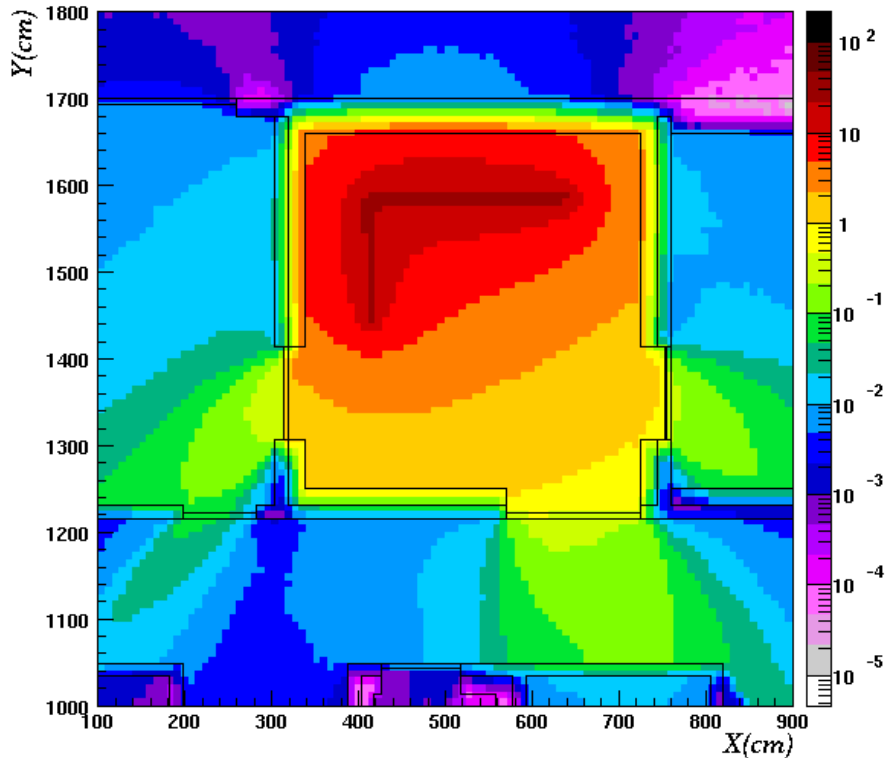
Dans notre application, **ZBEAM** est fixé pour tout le problème.

Ensuite, il s'agit de définir les deux lignes de façon équiprobable.

Dans la carte **SOURCE** du fichier d'entrée de ce problème, **WHASOU(1)** = 0.5.

On définit un test sur **FLRNDM(XDUMMY)**. Si le nombre aléatoire est strictement inférieur à 0.5, la source sera uniformément répartie sur la ligne « horizontale » (**YBEAM** fixé à 1585 et **XBEAM** varie entre 414 et 650). Si le nombre aléatoire est supérieur ou égal à 0.5, la source sera uniformément répartie sur la ligne « verticale » (**XBEAM** fixé à 414 et **YBEAM** sera uniformément réparti entre 1435 et 1585).

L'isotropie de la source est définie de la même façon que dans les applications précédentes, cf. **10.2.2** et **10.4.2**. Illustration ci-dessous :



10.6. Définir une source de forme circulaire et sphérique

On utilisera ici des routines déjà intégrées au code, qu'il suffit d'appeler et une application de la carte **BEAM** (la façon d'obtenir une source isotrope étant décrite au début du chapitre).

10.6.1. Définir une source circulaire

Dans la carte **BEAM**, avec **WHAT(6) < 0.0**, **WHAT(4)** et **WHAT(5)** sont les rayons maximal et minimal d'un faisceau annulaire. Si le rayon minimal est défini comme étant égal à **0.0**, on a donc bien une source circulaire.

On peut également appeler la routine **SFECFE** dans le fichier **source.f**. Elle retourne deux nombres aléatoires compris entre 0 et 1, X et Y de telle sorte que $X^2+Y^2 = 1.D+00$.

```
CALL SFECFE (SINT, COST)
XBEAM = COST
YBEAM = SINT
ZBEAM = 0.0
XFLK (NPFLUKA) = XBEAM
YFLK (NPFLUKA) = YBEAM
ZFLK (NPFLUKA) = ZBEAM
```

Ici la source est uniformément répartie sur un disque de rayon 1 cm de rayon et de centre (0,0) dans le plan (X,Y).

10.6.2. Définir une source sphérique

On appelle la routine **RACO** dans le fichier **source.f**. Elle retourne trois nombre aléatoires compris entre 0 et 1, **Txx**, **Tyy**, **Tzz** de telle sorte que $Txx^2 + Tyy^2 + Tzz^2 = 1.D+00$.

On définit ensuite **XBEAM = Txx**, **YBEAM = Tyy** et **ZBEAM = Tzz**, par exemple.

Ceci est valable pour une sphère de rayon 1 et de centre (0, 0, 0).

Si l'on désire une distribution sur une sphère de rayon **R**, il faut définir les coordonnées de la particule de la façon suivante : **iBEAM = Tii * R**, avec **i = x,y,z** et **R** le rayon de la sphère en cm en double précision (500.D0 pour 500 cm par exemple).

```
CALL RACO (TXX, TYY, TZZ)
XBEAM = 500.D0*TXX
YBEAM = 500.D0*TYY
ZBEAM = 500.D0*TZZ
XFLK (NPFLUKA) = XBEAM
YFLK (NPFLUKA) = YBEAM
ZFLK (NPFLUKA) = ZBEAM
```

Ici la source est uniformément répartie dans une sphère de rayon 500 et de centre (0, 0, 0).

10.6.3. Définir une source cylindrique isotrope

Le disque est défini sur (X, Y) et la hauteur sur Z.

La partie circulaire :

Le rayon du disque est de R cm et la hauteur comprise entre Zmin et Zmax.

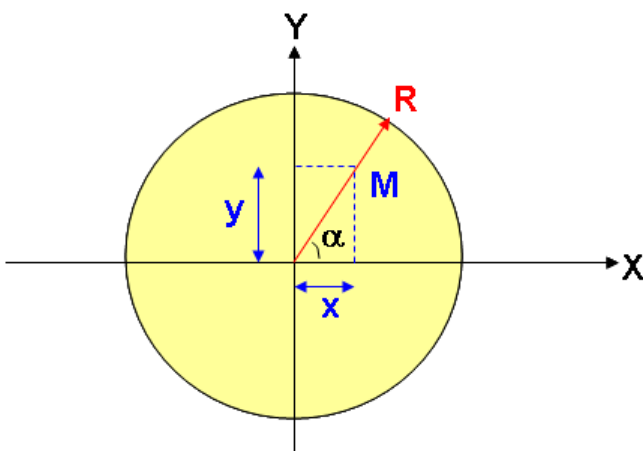
On sait que tout point est compris dans le disque de rayon R cm, soit sur une distance allant de 0 à R par rapport à l'origine sur 2π (360°).

Pour définir la partie angulaire, on se réfère au paragraphe **10.6.1**, en appelant la routine **SFECFE** qui retourne deux nombre aléatoires **SINT** et **COST**.

On écrit donc :

```
XBEAM = R*SQRT(A)*COST
YBEAM = R*SQRT(A)*SINT
```

Détail :



La probabilité $p(r)$ de trouver un point dans le disque de rayon R avec un certain rayon r est proportionnelle à r.

$$p(r) = 2*\pi*r/A, \text{ avec } A = \pi*R^2, p(r) = 2*r/R^2$$

La distribution de la probabilité est l'intégrale de cette fonction, $P(r) = r^2/R^2$.

La distribution des valeurs du rayon r est donc donnée par :

$$r = \text{sqrt}(P(r)*R^2) = R* \text{sqrt}(P(r))$$

Et $P(r)$ n'est rien d'autre qu'un nombre aléatoire compris entre 0 et 1.

Remerciements à Anton Empl pour cette partie.

La partie azimuthale a déjà été vue dans le paragraphe **10.5.2** :
 Les valeurs de Z sont uniformément comprises sur une ligne délimitée par Zmin et Zmax.
 Soit B un nombre aléatoire compris entre 0 et 1 :
 On écrit donc $Z_{BEAM} = Z_{min} + (Z_{max} - Z_{min}) * B$

L'isotropie est assurée par la même méthode que dans le paragraphe **10.6.2**

Application : modélisation d'une source de type SG500 (résine epoxy de densité 1,15, activité uniformément répartie dans un cylindre correspondant aux dimensions suivantes :
 Le rayon du disque est de 4,7 cm et la hauteur comprise entre -7,225 et -0,025.

```

****
DOUBLE PRECISION RANDY
DOUBLE PRECISION RADIUS
DOUBLE PRECISION MYRHO
DOUBLE PRECISION ZMIN
DOUBLE PRECISION ZMAX
* Cosines (tx,ty,tz)
* Ajout : isotropy
CALL RACO(TXX,TYY,TZZ)
UBEAM = TXX
VBEAM = TYY
WBEAM = TZZ
TXFLK (NPFLKA) = UBEAM
TYFLK (NPFLKA) = VBEAM
TZFLK (NPFLKA) = WBEAM
* TZFLK (NPFLKA) = SQRT ( ONEONE - TXFLK (NPFLKA)**2
* & - TYFLK (NPFLKA)**2 )
* Polarization cosines:
TXPOL (NPFLKA) = -TWOTWO
TYPOL (NPFLKA) = +ZERZER
TZPOL (NPFLKA) = +ZERZER
* Particle coordinates
CALL SFECFE(SINT,COST)
RADIUS = 4.699999D0
ZMIN = -7.224999D0
ZMAX = -0.0250001D0
MYRHO = RADIUS * SQRT (FLRNDM(XDUMMY))
XBEAM = MYRHO * COST
YBEAM = MYRHO * SINT
ZBEAM = ZMIN - ((ZMIN - ZMAX) * FLRNDM(ZDUMMY))
XFLK (NPFLKA) = XBEAM
YFLK (NPFLKA) = YBEAM
ZFLK (NPFLKA) = ZBEAM
    
```

10.7. Autres routines potentiellement utiles

- **CALL FLNRNN(RGAUSS)** retourne un nombre d'une distribution gaussienne **RGAUSS** ;
- **CALL FLNRN2(RGAUS1,RGAUS2)** retourne deux nombres de distribution gaussiennes indépendants **RGAUS1** et **RGAUS2** ;
- **CALL SFLOOD(XXX,YYY,ZZZ,UXXX,VYYY,WZZZ)** retourne **XXX**, **YYY**, **ZZZ** une position aléatoire SUR la surface d'une sphère de rayon 1 et de centre 0 (multiplier **XXX**, **YYY**, **ZZZ** par le rayon est additionner les coordonnées du centre) et **UXXX**, **VYYY**, **WZZZ** sont des cosinus directeurs parfaitement distribués afin de générer une distribution uniforme et isotrope de la fluence dans la sphère (bien entendu, en l'absence de matériel dans la sphère = vide), numériquement donnée par $1 / (\pi R^2)$, R étant le rayon de la sphère (Alfredo Ferrari).

11. Utilisation des options de production de noyaux et de décroissance radioactive

Cf. manuel, sections : **RESNUCLEI**, **IRRPROFILE**, **RADEDECAY**, **DCYTIMES**, **DCYSCORE**.

11.1. Un peu de physique

Pour obtenir des noyaux résiduels et produire des fissions, il faut activer les options physiques dans **FLUKA**.

Dans l'exemple que nous allons développer, nous simulons la photofission d'une cible de carbure d'uranium par un faisceau d'électrons (ALTO).

```

*** Physical properties ***
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
EMF
EMFCUT      -1.0E-3    0.1E-3      0.0      1.0      26.0
LOW-NEUT
PHYSICS      2.0      0.0      0.0      0.0      0.0      0.0EVAPORAT
PHOTONUC     1111.0      1.0      26.0
*IONTRANS
EVAP
    
```

Telles qu'elles sont données ci-dessus

- La carte **EMF** active le transport d'électrons, de positrons et de photons.
- La carte **EMF-CUT** donne les limites inférieures d'énergies du transport des électrons, positrons et photons pour les matériaux d'indices 1 à 26 sans biaisage. Les limites de productions sont données avec la même carte, mais avec un **SDUM = PROD-CUT** et non pas laissé vide.
- La carte **LOW-NEUT** active le transport de neutrons « lents».
- La carte **PHYSICS** active l'évaporation avec le nouveau modèle (algorithme du code de FLUKA) sans les fragments lourds.
- La carte **PHOTONUC** active les interactions des photons avec les noyaux avec les options interactions à haute énergie, delta résonance, quasi deutérons et résonance géante pour les matériaux d'indices 1 à 26.
- À partir de la version 2011.2, la carte **EVENTYPE** indiquant quel modèle est utilisé pour la production des hadrons est rendue obsolète. Une nouvelle carte **IONTRANS** est introduite pour contrôler le transport et les interactions liés aux ions lourds.

Pour de plus amples informations concernant ces cartes, *cf. le manuel de FLUKA*.

Avec ce jeu de données, la fission sera simulée, nous pouvons poursuivre par la description des cartes qui nous intéressent.

11.2. La carte RESNUCLEI

Cette option produit les noyaux résiduels après des interactions inélastiques dans une région donnée.

Description

- **WHAT(1)** : type de produits
 = 1.0, seuls les produits de spallation sont produits (toutes les interactions inélastiques sauf celles induites par les neutrons ayant une énergie inférieure au seuil pour le traitement multigroupe).

= 2.0, produits issus des neutrons lents, ceux dont l'énergie est inférieure au seuil pour le traitement multigroupe (si les données sont disponibles, voyez la note 1 du paragraphe **RESNUCLEI** ainsi que le chapitre consacré aux neutrons lents dans le manuel).

= 3.0, tous les noyaux résiduels sont produits (si les données sont disponibles, même remarque que précédemment).

- **WHAT(2)** : unité logique du fichier de sortie.
- > 0, le fichier de sortie sera «formaté».
- < 0, le fichier de sortie sera non «formaté».
- **WHAT(3)** : numéro atomique Z maximal pour la distribution des noyaux résiduels. Par défaut, ce nombre est choisi selon le numéro atomique du(des) élément(s) qui constitue(nt) le matériau assigné à la région dans laquelle le calcul est demandé.
- **WHAT(4)** : nombre M maximal pour la distribution des noyaux résiduels. M est un nombre égal à $N - Z - (NMZ)_{\min}$. Par défaut, ce nombre est choisi selon les numéros de masse A et atomique Z du(des) élément(s) qui constitue(nt) le matériau assigné à la région dans laquelle le calcul est demandé. Pour minimiser le stockage, les nucléides sont indexés par Z et par NMZ.

Sauf cas particulier, vous n'avez rien à entrer pour WHAT(3) et WHAT(4), les valeurs par défaut conviennent très bien.

- **WHAT(5)** : numéro de la région dans laquelle on désire faire le calcul.
- **WHAT(6)** : volume de la région en cm^3 , les résultats seront alors exprimés en nombre de noyaux / cm^3 / particule primaire. Si l'on laisse le volume à 1.0, on s'affranchit du paramètre volumique, les résultats seront exprimés en nombre de noyaux / particule primaire.
- **SDUM** : un nom identifiant le détecteur, 10 caractères au maximum.

Exemple

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
RESNUCLEI      3.0      -70.0      0.0      0.0      4.0      1.0UC2
RESNUCLEI      1.0       51.0      0.0      0.0      5.0      12.53C
```

Le premier détecteur : la production des noyaux issus de toutes les interactions inélastiques est demandée, le fichier de sortie sera : **nom_du_fichier_d'entrée_XXX_fort.70** où **XXX** est le numéro du cycle, ce fichier sera non formaté. La région n° 4 est concernée, son volume est laissé à 1.0 cm^3 . Il s'agit d'une région constituée de carbure d'uranium.

Le second détecteur : seuls les produits de spallation sont requis ici, la région n° 5 est concernée, le fichier de sortie sera : **nom_du_fichier_d'entrée_XXX_fort.51**, ce fichier sera formaté (fichier texte). Le volume du détecteur est de 12.53 cm^3 , les résultats seront en **noyaux/cm³/particule primaire**. Il s'agit d'une région constituée de graphite.

Exploitation des résultats

Les fichiers de sortie non formatés peuvent être lus et transformés en fichiers textes lisibles et exploitables par différents programmes :

- **rdresn**, lancé par **\$FLUPRO/flutil/rdresn**, qui produit un fichier texte très clair en *colonnes*, mais ne permet pas de gérer les statistiques sur plusieurs cycles.
- **ursuw**, lancé par **\$FLUPRO/flutil/ursuw**, produit des fichiers en matrices A, Z, nombre de noyaux / cm^3 / particule primaires, incertitudes associées, qu'il est un peu pénible de manipuler sans les scripts adéquats, mais qui permet de gérer différents cycles d'une même simulation.
- **ursuwev**, lancé par **\$FLUPRO/flutil/ursuwev**, est similaire à **ursuw** dans la forme, mais il permet de calculer l'activité radioactive induite et son évolution dans le temps. Les résultats obtenus seront en **Bq/cm³/particule primaire**. Cette option est maintenant un peu

obsoleète du fait de la possibilité de calculer la décroissance radioactive « online » avec les cartes que nous allons décrire ci-après. Il est recommandé dans le manuel de **FLUKA** de préférer le traitement *online* à celui *offline*.

Néanmoins, petit bémol personnel, pour avoir comparé des résultats obtenus par les deux méthodes, je n'ai pas vu de différence majeure. L'avantage du traitement par **usrsuwev**, c'est que l'on peut écrire les résultats des activités radioactives à différents temps de décroissance pour un détecteur donné dans le même fichier de sortie. Ceci permet de limiter le nombre de fichiers produits après traitement par **usrsuw / usrsuwev**.

Remarque

À partir de la version 2006.3 de **FLUKA** le calcul de l'hydrogène 1 est effectué.

Il est également préconisé d'utiliser les deux cartes suivantes si le calcul des noyaux résiduels est requis :

PHYSICS	3.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	EVAPORAT
PHYSICS	1.0	1.0	1.0	0.0	0.0	0.0	0.0	0.0	COALESC

11.3. La carte RADDECAY

Celle-ci est essentielle, elle active la simulation de la décroissance radioactive et permet de paramétrer le biaisage et les conditions de transport.

Description

- **WHAT(1)** : drapeau pour activer la décroissance radioactive.
 - = 1.0 : décroissance radioactive active pour les différents temps requis ;
 - > 1.0 : décroissance radioactive active en mode semi-analogue ;
 - = 0.0 : ignoré ;
 - < 0.0 : paramètre par défaut ;Par défaut : pas de décroissance radioactive.
- **WHAT(2)** : drapeau pour activer la production d'isomères, en attendant un meilleur modèle.
 - > 0.0 : production d'isomères "patching" activée ;
 - < 0.0 : production d'isomères "patching" désactivée ;
 - = 0.0 : ignoré ;Par défaut : activée si la décroissance non analogue est activée, désactivée sinon.
- **WHAT(3)** : nombre de "répliques" de la décroissance de chaque résidu
 - = 0.0 : ignoré ;
 - < 0.0 : paramètre par défaut ;Par défaut = 1.0 pour les décroissances analogues, 3.0 sinon.
- **WHAT(4)** : permet d'appliquer différentes techniques de biaisage seulement aux particules promptes ou seulement à celles produites par la décroissance radioactive ou les deux. Cf. le manuel **FLUKA** pour le descriptif complet.
- **WHAT(5)** : facteurs multiplicatifs à appliquer aux seuils d'énergie du transport des e^\pm et γ , respectivement pour les radiations promptes et la décroissance. Cf. le manuel **FLUKA** pour le descriptif complet.
- **WHAT(6), SDUM** : non usités.

Par défaut (option **RADDECAY** non donnée) : la décroissance radioactive n'est pas activée, aucun facteur multiplicatif n'est appliqué aux seuils d'énergie pour le transport.

11.4. La carte IRRPROFILE

Cette carte définit un profil d'irradiation pour les calculs de décroissance radioactive. Il est possible de définir jusqu'à 20 intervalles d'irradiation différents.

Description

- **WHAT(1)** : durée du nouvel intervalle (en s).
 > 0.0 : un nouvel intervalle de durée **WHAT(1)** est ajouté ;
 = 0.0 : ignoré ;
 < 0.0 : l'intervalle précédent est effacé (s'il existe) ;
 Par défaut : aucun nouvel intervalle d'irradiation n'est défini.
 - **WHAT(2)** : intensité du faisceau de l'intervalle défini (dans **WHAT(1)**)
 = 0.0 : intensité du faisceau en particules / s (0.0 est accepté) ;
 < 0.0 : considéré comme 0.0 ;
 Par défaut = 0.0 particules / s.
 - **WHAT(3)** : la même chose que **WHAT(1)**.
 - **WHAT(4)** : la même chose que **WHAT(2)**.
 - **WHAT(5)** : la même chose que **WHAT(1)**.
 - **WHAT(6)** : la même chose que **WHAT(2)**.
 - **SDUM** : non usité.
- Par défaut : (option **IRRPROFILE** non donnée) : aucun intervalle d'irradiation n'est défini.

Exemple

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
IRRPROFILE      1800.    1,5E12    250.    3,0E10    4500.    4,2E12
```

Le profil défini consiste en 1 800 s d'irradiation à une intensité de 1,5E12 particules / s, suivi de 250 s à faible intensité (3,0E10 particules / s), puis de 4 500 s à 4,2E12 particules / s.

11.5. La carte DCYTIMES

Cette carte permet de définir les temps de décroissance après la fin de l'irradiation auxquels les quantités demandées, activités radioactives, doses sont calculées.

Description

- **WHAT(1)** : temps de décroissance (en s) après la fin de l'irradiation, à associer avec un détecteur (*Cf. note 1 ci-dessous*).
- >=< 0.0 : un nouveau temps de décroissance est ajouté, il est égal à **WHAT(1)**. C'est le temps après la fin de l'irradiation.
 Par défaut : le calcul à la fin de l'irradiation est associé à l'indice 1.
- **WHAT(2)** : la même chose que **WHAT(1)** (un autre temps de décroissance).
 - **WHAT(3)** : la même chose que **WHAT(1)** (un autre temps de décroissance).
 - **WHAT(4)** : la même chose que **WHAT(1)** (un autre temps de décroissance).
 - **WHAT(5)** : la même chose que **WHAT(1)** (un autre temps de décroissance).
 - **WHAT(6)** : la même chose que **WHAT(1)** (un autre temps de décroissance).
 - **SDUM** : non usité.
- Par défaut (option **DCYTIMES** non demandée) : aucun temps de décroissance n'est défini.

Notes :

1. Un indice est assigné à chaque temps de décroissance. Il suit l'ordre dans lequel les temps ont été entrés. Cet indice peut être utilisé dans l'option **DCYSCORE** pour assigner ce temps de décroissance en particulier à un ou plusieurs détecteurs.
2. De multiples cartes peuvent être données, jusqu'à 20 temps de décroissance au maximum. Tous les temps de décroissance sont comptés à partir de la fin de la dernière période d'irradiation définie dans la commande **IRRPROFILE**.
Un temps de décroissance nul active le calcul à l'exacte fin de l'irradiation. Un temps de décroissance négatif est admis : le calcul est effectué au temps voulu « pendant l'irradiation ».

Exemple

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
*23456789 123456789 123456789 123456789 123456789 123456789 123456789
DCYTIMES      10.      30.      3600.      43200.      86400.      172800.
DCYTIMES      2592000. 31557600.
```

8 temps de décroissance ont été définis, chacun avec un indice correspondant à son ordre d'entrée, le temps de décroissance n° 1 est 10 s, le n° 2 est 30 s, les temps d'indices 3 à 8 sont respectivement d'1 h, 1/2 jour, 1 jour, 2 jours, 30 jours, 1 an.

11.6. La carte DCYSCORE

Cette commande associe divers détecteurs permettant d'estimer telle ou telle quantité à des temps définis par l'utilisateur dans la carte **DCYTIMES**. Voyez l'avertissement dans la note 1 ci-dessous.

- **WHAT(1)** : indice de temps de décroissance à associer avec le(s) détecteur(s) dont le type est donné par **SDUM** et dont le numéro correspond à **WHAT(4)–WHAT(6)** (cf. note 2, ci-dessous).

Par défaut = 0.0 : pas de calcul !

- **WHAT(2) – WHAT(3)** : non utilisés.
- **WHAT(4)** : limite inférieure des indices des détecteurs associés aux temps de décroissance spécifiés (« Du détecteur de type **SDUM** de numéro **WHAT(4)**... »)

Par défaut = 1.0.

- **WHAT(5)** : limite supérieure des indices des détecteurs associés aux temps de décroissance spécifiés (« ... au détecteur de type **SDUM** de numéro **WHAT(5)**... »)

Par défaut = **WHAT(4)**.

- **WHAT(6)** = longueur du pas dans l'assignation de indices (« ... par pas de **WHAT(6)** »)

Par défaut = 1.0.

- **SDUM** : identifie le type d'estimateur considéré : **EVENTBIN, RESNUCLEI, USRBDX, USRBIN, USRCOLL, USRTRACK, USRYIELD**

Par défaut : pas de donnée par défaut !

Par défaut (option **DCYSCORE** non requise) : aucun détecteur n'est associé avec un quelconque indice de temps de décroissance.

Notes :

1. **Avertissement** : lorsque l'option **DCYSCORE** est appliquée à un détecteur, toutes les quantités sont exprimées par unité de temps. Par exemple, l'estimateur **RESNUCLEI** sera exprimé en Bq, les estimateurs de dose seront exprimés en débit de dose, etc.

2. Le temps de décroissance indiqué par **WHAT(1)** doit être l'un de ceux définis par l'intermédiaire de l'option **DCYTIMES**.
3. Les détecteurs **USRBIN** et **EVENTBIN** sont comptés ensemble (ils appartiennent à la même séquence), c'est également le cas pour **USRTRACK** et **USRCOLL**.

Exemple

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...
DCYTIMES      86400.   172800.
DCYSCORE      1.           4.           8.           2. USRBIN
DCYSCORE      2.           1.           5.           USRTRACK
```

Deux temps de décroissance ont été définis. Les « binnings » numéros 4, 6 et 8 seront associés au premier temps de décroissance, 1 jour. Les détecteurs de fluence (track-length) numéros 1, 2, 3, 4 et 5 seront associés au second temps de décroissance, 2 jours.

11.7. Application : l'ensemble cible-source d'ALTO

Géométrie proposée par Maher Cheikh MHAMED

ALTO : Accélérateur Linéaire auprès du Tandem d'Orsay

Dans ce cas, nous désirons étudier la production des noyaux par activation et fission de la cible de carbure d'uranium sur laquelle parvient un faisceau d'électrons d'énergie 50 MeV et d'intensité 10 μ A pendant 3 semaines. Nous souhaitons également calculer les activités radioactives résultantes à divers temps après la fin de l'irradiation : 1 semaine pour le déchargement de la cible et son transfert dans un caisson de transport plombé, 1, 2 et 3 mois pour connaître les divers niveaux d'activités lorsque plusieurs caissons seront stockés (avec des temps de décroissance décalés) ensemble dans l'aire de stockage temporaire et 2 ans, afin de connaître le niveau d'activité de l'ensemble avant son transfert vers un stockage de plus long terme. Nous désirons également calculer les débits de doses équivalentes résultants autour du caisson à ces différents temps de décroissance.

Les estimateurs utilisés sont **RESNUCLEI** pour le calcul des noyaux produits pendant l'irradiation, ainsi que pour les calculs d'activités aux différents temps de décroissance, mais couplé cette fois avec les 3 cartes **IRRPROFILE**, **DCYTIMES**, **DCYSCORE**.

Les débits de dose sont estimés par **USRBIN** associé aux temps de décroissance également. *Il pourrait également être utile de connaître les spectres en énergie des particules produites pendant l'irradiation ou après décroissance avec des estimateurs **USRTRACK**.*

Noyaux résiduels produits pendant l'irradiation

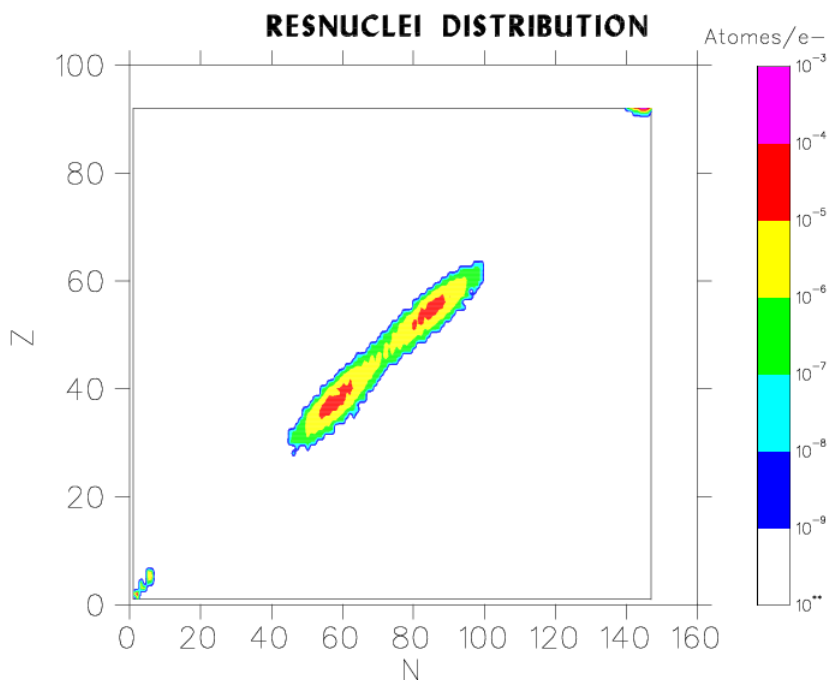
La « gamelle » contenant la source est composée de 10 matériaux significatifs, 10 estimateurs **RESNUCLEI** sont donc nécessaires pour la production pendant l'irradiation :

```
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
RESNUCLEI      3.0      -60.0      0.0      0.0      4.0      1.0UC2
RESNUCLEI      3.0      -61.0      0.0      0.0      5.0      1.0C
RESNUCLEI      3.0      -62.0      0.0      0.0      6.0      1.0Ta1
RESNUCLEI      3.0      -63.0      0.0      0.0      9.0      1.0Ta2
RESNUCLEI      3.0      -64.0      0.0      0.0     10.0      1.0Cu1
RESNUCLEI      3.0      -65.0      0.0      0.0     11.0      1.0Cu2
RESNUCLEI      3.0      -66.0      0.0      0.0     12.0      1.0Cu3
RESNUCLEI      3.0      -67.0      0.0      0.0     13.0      1.0Al
RESNUCLEI      3.0      -68.0      0.0      0.0     15.0      1.0Inox
RESNUCLEI      3.0      -69.0      0.0      0.0      2.0      1.0Air
```

Bien entendu, la région n° 4 (formée de carbure d'uranium) est la seule dans laquelle se produira la photofission. Les produits formés dans les autres régions seront dus à d'autres réactions nucléaires, à l'activation neutronique notamment.

Résultats : noyaux résiduels dans la région UC2 en nombre d'atomes / électron incident en fonction du nombre de neutrons et du nombre de protons.

Les points dans la région des atomes légers sont dus aux réactions sur le carbone et ceux dans la région des atomes lourds sont dus aux réactions autres que la fission sur les atomes d'uranium. Les produits de fission sont représentés aux masses intermédiaires. Le domaine de prévision s'étend du nickel 74 ($Z = 28$) à l'euporium 162 ($Z = 63$).



Activités calculées après l'irradiation

En ce qui concerne les activités, nous définissons tout d'abord les différents éléments activant la décroissance radioactive aux temps désirés, ainsi que le profil d'irradiation.

```
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
IRRPROFILE 1814400. 6.25E13
RADDECAY 1.0 1.0
DCYTIMES 0. 604800. 2592000. 5184000. 7776000.
DCYTIMES 63115200.
DCYSCORE 1.0 1. 1. 1.RESNUCLEI
DCYSCORE 2.0 2. 2. 1.RESNUCLEI
DCYSCORE 3.0 3. 3. 1.RESNUCLEI
DCYSCORE 4.0 4. 4. 1.RESNUCLEI
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
DCYSCORE 5.0 5. 5. 1.RESNUCLEI
DCYSCORE 6.0 6. 6. 1.RESNUCLEI
```

Dans l'exemple, l'intensité du faisceau est de $6,25E13$ électrons / s pendant 21 jours.

La décroissance radioactive est activée ainsi qu'une production approximative d'isomères.

Les temps de décroissance à partir de la fin de l'irradiation sont 0 s, 1 semaine, 1, 2, 3 mois et 2 ans. À chacun de ces temps est associé un estimateur **RESNUCLEI** repéré par son numéro d'ordre. Il pourra s'agir par exemple de suivre l'évolution des activités radioactives dans la région constituée d'UC2.

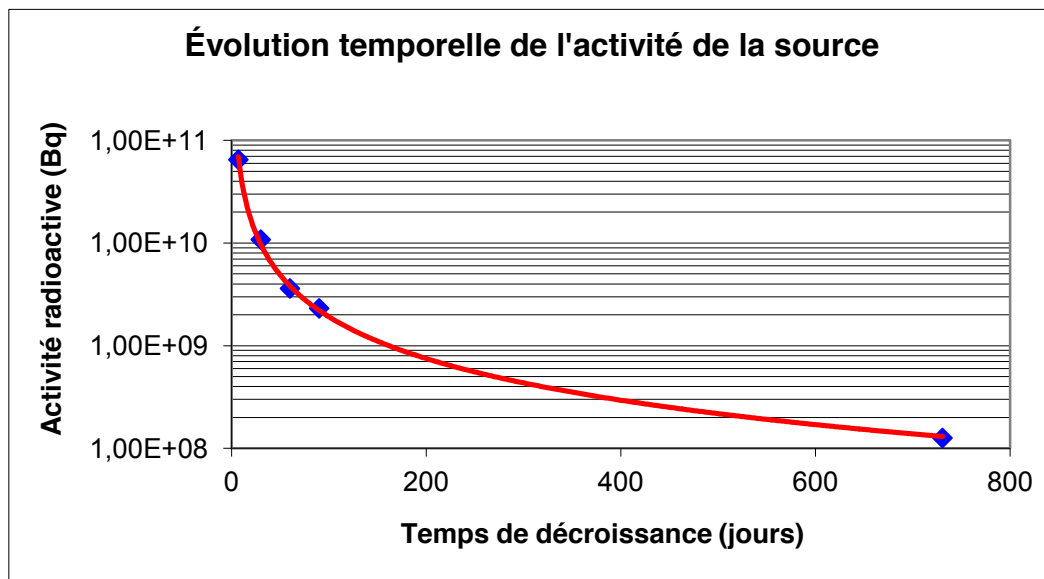
```
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
RESNUCLEI 3.0 -60.0 0.0 0.0 4.0 1.0UC2
RESNUCLEI 3.0 -61.0 0.0 0.0 4.0 1.0UC2
RESNUCLEI 3.0 -62.0 0.0 0.0 4.0 1.0UC2
RESNUCLEI 3.0 -63.0 0.0 0.0 4.0 1.0UC2
RESNUCLEI 3.0 -64.0 0.0 0.0 4.0 1.0UC2
RESNUCLEI 3.0 -65.0 0.0 0.0 4.0 1.0UC2
RESNUCLEI 3.0 -66.0 0.0 0.0 4.0 1.0UC2
```

Attention à ne pas succomber à la facilité d'assigner plusieurs temps de décroissances au même détecteur, chaque carte écrasant la précédente. Il faut assigner un seul temps de décroissance par détecteur alors que l'on peut assigner un temps donné à plusieurs détecteurs différents.

On peut tromper une fois mille personnes, mais on ne peut pas tromper une personne mille fois, non, non, ce n'est pas ça, on peut tromper mille fois mille personnes...

Il n'est pas vraiment intéressant de représenter les résultats sous forme de graphique, radioélément par radioélément. Des tableaux, fastidieux certes, rendent mieux compte des niveaux d'activité.

Néanmoins, à titre d'information, nous avons choisi de représenter l'évolution de l'activité totale (quelle que soit la forme physico-chimique des radioéléments) dans la région constituée d'UC2 (plus de 99 % de l'activité de l'ensemble cible-source après une semaine de décroissance).



À noter qu'après deux ans de décroissance, la région constituée d'UC2 ne représente plus que 42 % de l'activité totale. Ceci est dû au tantale 179 de période radioactive d'1,82 ans et produit de manière importante (constituant du four de l'ensemble cible-source).

Débits de dose après l'irradiation

Nous avons calculé les débits de dose aux différents temps de décroissance définis précédemment. Les commandes sont les suivantes :

```
IRRPROFILE 1814400. 6.25E13
RADDECAY 1.0 1.0
DCYTIMES 0. 604800. 2592000. 5184000. 7776000. 63115200.
DCYSCORE 1.0 2. 2. 1.USRBIN
DCYSCORE 2.0 3. 3. 1.USRBIN
DCYSCORE 3.0 4. 4. 1.USRBIN
```

```

DCYSCORE      4.0          5.      5.      1.USRBIN
DCYSCORE      5.0          6.      6.      1.USRBIN
DCYSCORE      6.0          7.      7.      1.USRBIN
* total during irradi. (no decay)
USRBIN        10.      201.      -50.      20.      10.      20.EWT74
USRBIN        -20.      -10.      -20.      50.      50.      50.&
* total at the end of irradi.
USRBIN        10.      201.      -51.      20.      10.      20.EWT74
USRBIN        -20.      -10.      -20.      50.      50.      50.&
* total decay 1 week
USRBIN        10.      201.      -52.      20.      10.      20.EWT74
USRBIN        -20.      -10.      -20.      50.      50.      50.&
* total decay 1 month
USRBIN        10.      201.      -53.      20.      10.      20.EWT74
USRBIN        -20.      -10.      -20.      50.      50.      50.&
* total decay 2 month
USRBIN        10.      201.      -54.      20.      10.      20.EWT74
USRBIN        -20.      -10.      -20.      50.      50.      50.&
* total decay 3 month
USRBIN        10.      201.      -55.      20.      10.      20.EWT74
USRBIN        -20.      -10.      -20.      50.      50.      50.&
* total decay 2 years
USRBIN        10.      201.      -56.      20.      10.      20.EWT74
USRBIN        -20.      -10.      -20.      50.      50.      50.&
    
```

6 temps de décroissance ont été définis, appliqués respectivement aux détecteurs 2 à 7. La forme dans laquelle s'effectue le calcul (le binning) est identique pour tous.

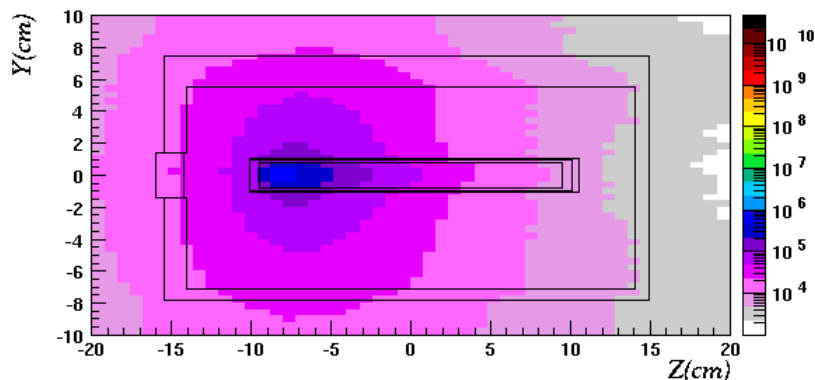
Le premier estimateur **USRBIN** est utilisé pour calculer la même quantité, mais pendant l'irradiation. La conversion fluence / dose efficace est effectuée via la routine *fluscw* et *deq99c* avec les facteurs les plus pénalisants en termes de géométrie.

Le tableau ci-dessous donne les tendances et les valeurs maximales des débits de doses après la traversée des 5 cm de plomb.

Temps de décroissance	Débit de dose gamma pénalisant ($\mu\text{Sv/h}$)	
	Tendance	Valeur maximale (point chaud)
1 semaine	600	1000
1 mois	200	300
2 mois	50	50
3 mois	20	30
2 ans	0,5	2

Ci-après, un diagramme d'isodoses, résultats en $\mu\text{Sv/h}$ (sans les 5 cm de Pb ici).

Plan (z, y), une semaine de décroissance.



À partir de la version 2011.2, il est possible de ne considérer qu'une partie de la géométrie lors du calcul de la décroissance radioactive. Cela peut s'avérer utile pour connaître l'évolution d'une pièce située dans une situation d'irradiation pendant le calcul, mais qui pourrait être déplacée pour une autre utilisation ensuite. Il est alors possible de déterminer quelle est la composante du débit de dose équivalente (par exemple) due à cette partie seule en s'affranchissant des expositions dues aux autres pièces (régions) de la géométrie modélisée. Pour s'affranchir d'une région, il faut lui attribuer le matériau vacuum ou blackhole au **WHAT(6)** de la carte **ASSIGNMA** de la région en question (cf. *chapitre 5* du présent document).

12. Utilisation d'un utilitaire de géométrie : SimpleGeo

By Karl Heinz BUCHEGGER & Christian THEIS (CERN)

Le projet **SimpleGeo** propose un outil graphique capable de composer/manipuler la géométrie d'entrée de **FLUKA** (y compris l'aspect matériel). Il permet également une visualisation 3 D.

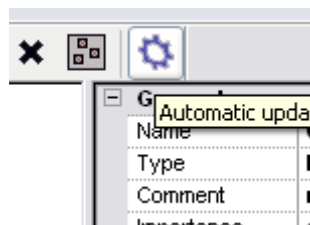
- Téléchargement : site web <http://theis.web.cern.ch/theis/simplegeo/>
- Installation pour Windows : exécuter le programme d'installation.

12.1. Manipuler un fichier d'entrée de FLUKA dont la géométrie existe déjà

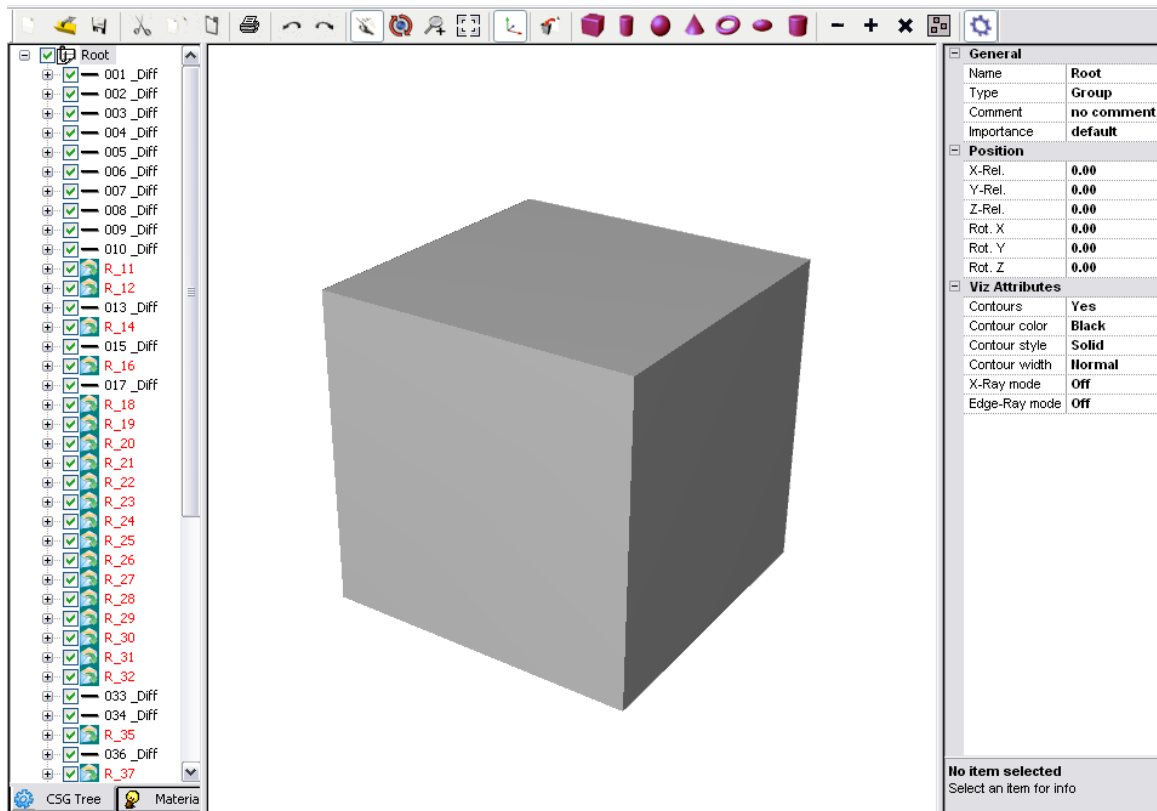
Il faut sélectionner : **File/Import**.

Une fenêtre de recherche de fichier s'ouvre, il suffit de la sélectionner après avoir suivi la bonne arborescence.

Ensuite, pour des raisons de performances de votre machine, la géométrie ne s'affichera que lorsqu'on aura sélectionné la petite icône permettant la mise à jour, complètement à droite de la barre d'outils.



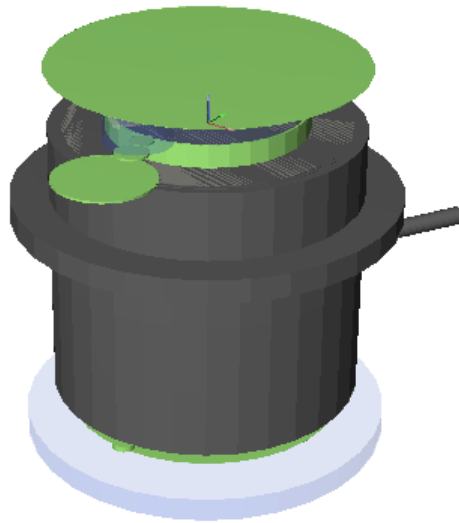
Après un délai, plus ou moins long selon la complexité de la géométrie, nous avons l'affichage suivant :



Vous noterez à gauche :

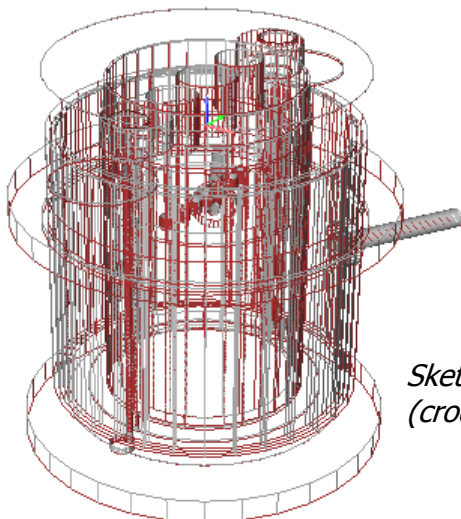
- L'onglet **CSG Tree** décrivant toutes les régions composant la géométrie ainsi que le détail des formes composant ces régions en appuyant sur la croix pour développer l'arborescence.
- L'onglet **Materials** où apparaissent tous les matériaux utilisés dans cette géométrie. Attention, les numéros de matériaux que vous avez définis dans votre fichier d'entrée ne correspondront pas à la liste des matériaux préenregistrés dans **SimpleGeo**. Vous pouvez modifier le matériau assigné à chaque région ou modifier la liste des matériaux pour la faire correspondre à celle de votre fichier d'entrée.

Pour l'instant, la vue est « bouchée », car vous ne voyez que la forme qui englobe toute la géométrie, le « Blackhole ». Pour visualiser, ce qu'il y a « à l'intérieur », vous devez désactiver le matériau « Blackhole » en décochant la case idoine, soit la région n° 1 ici. Ensuite, vous verrez la forme « Vacuum » englobant toute la géométrie. Il faut donc procéder de même avec les régions n° 2 et n° 17 dans notre exemple. À noter que vous pouvez obtenir le même résultat en cliquant sur le matériau dans l'onglet **Materials**, mais attention dans ce cas aux régions ayant le même matériau inclus dans votre forme globale. Deux autres possibilités : dans la fiche consacrée à chaque forme définie à droite, dans la partie **Viz attributes** sélectionnez **X-Ray mode** ou **Edge-Ray mode**. Pour notre exemple, nous obtenons la figure suivante :



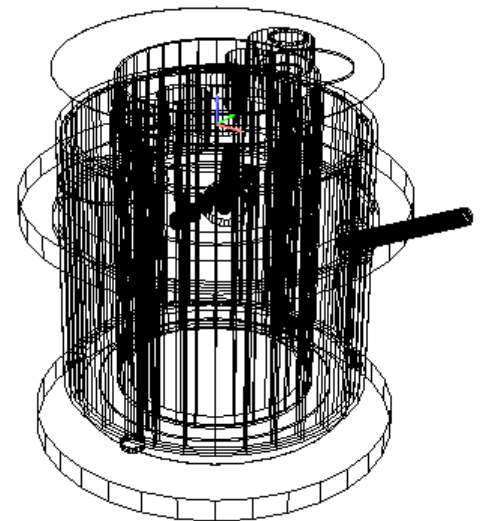
Plusieurs types de vues sont possibles, autant en termes de plans que de styles, la version par défaut est **Flat Shading**, qui montre les matériaux avec des effets d'ombres. Voyez l'onglet **View** pour tester d'autres styles.

Exemples :

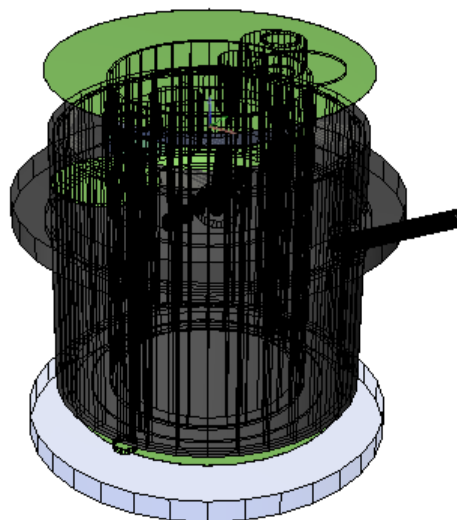


*Sketch rendering
(croquis)*

*Skeleton
rendering
(matériaux)*



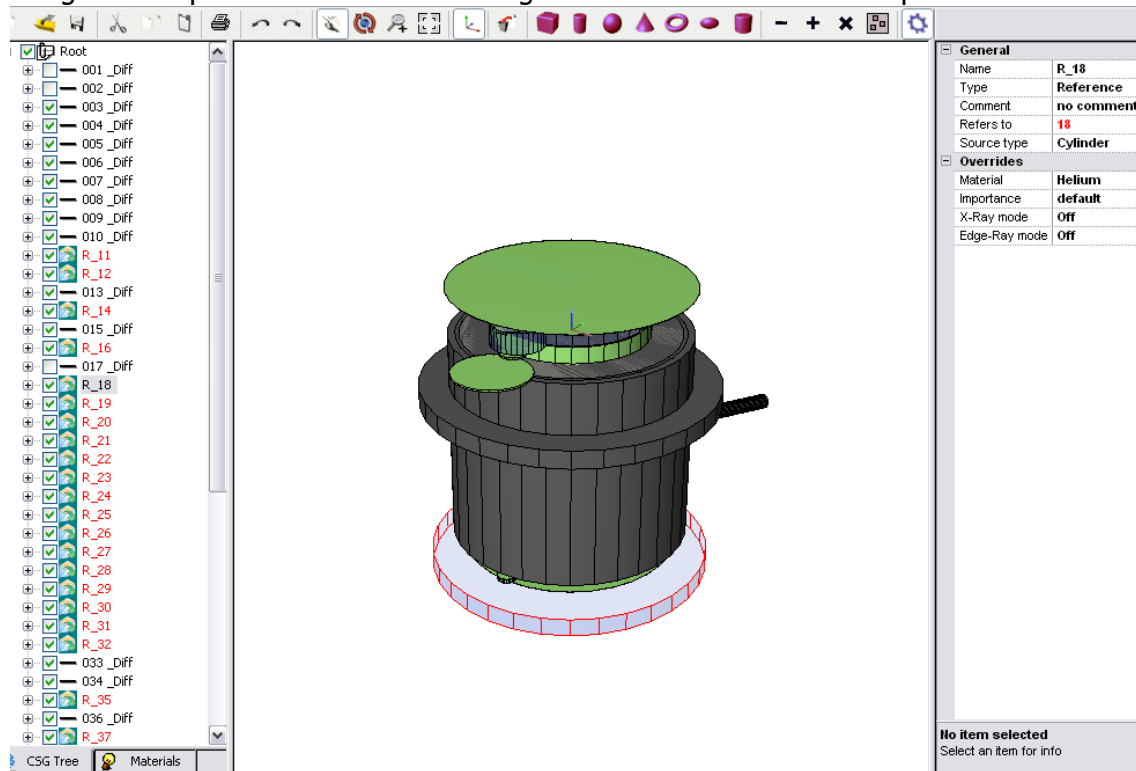
*Flat shading/overlay
sketch (superposition)*



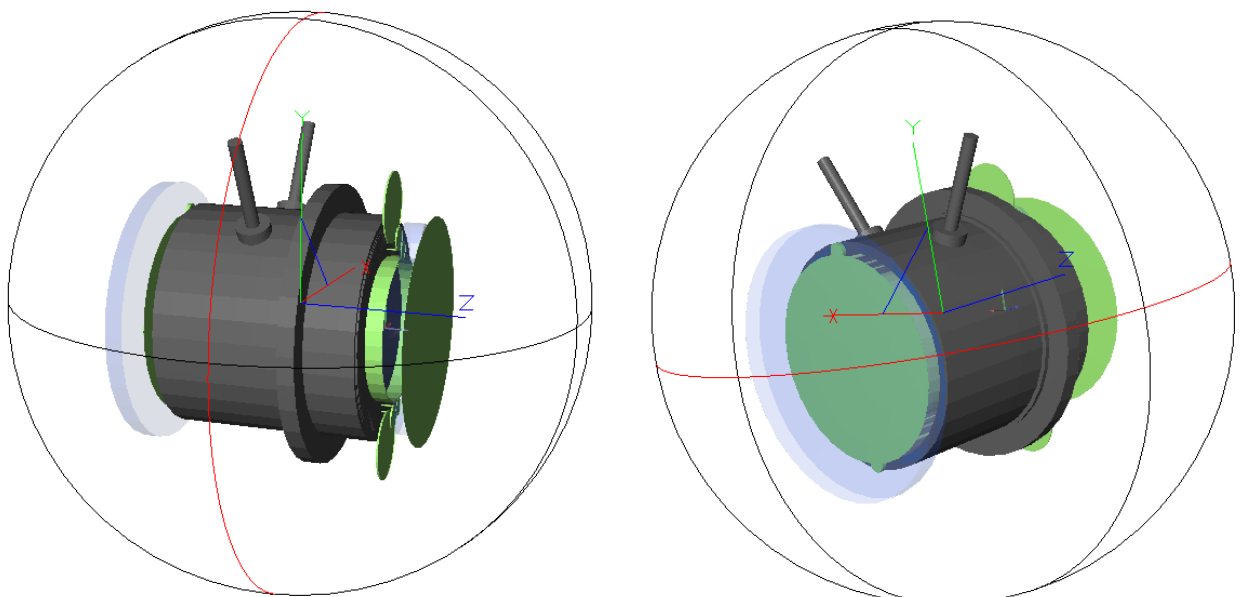


Le bouton ci-dessus permet de pointer sur n'importe quelle région de l'image pour obtenir les informations la définissant. La région concernée est alors cerclée de rouge. On peut obtenir le même effet en cliquant sur la région dans l'onglet **CSG Tree**.

Il est également possible d'effectuer des agrandissements à l'aide la loupe.



Le bouton ci-dessous permet de visualiser la forme dans toutes les directions voulues. Il suffit de cliquer sur la sphère enveloppe pour changer d'orientation selon ce que l'on veut afficher.

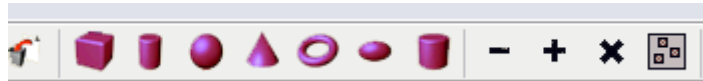


L'image peut être sauvée sous forme d'un fichier *.bmp portant le nom choisi par l'utilisateur à l'emplacement souhaité, en sélectionnant **File/Save Image**.

12.2. Créer une géométrie pour FLUKA avec SimpleGeo

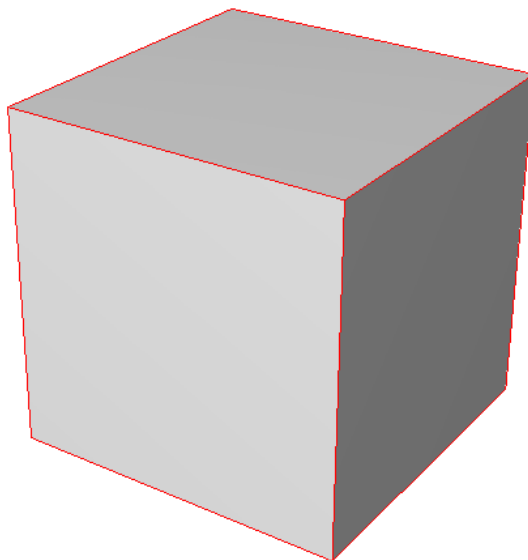
Sélectionnez **File/New**.

Choisissez la forme voulue en cliquant sur la barre d'outils :



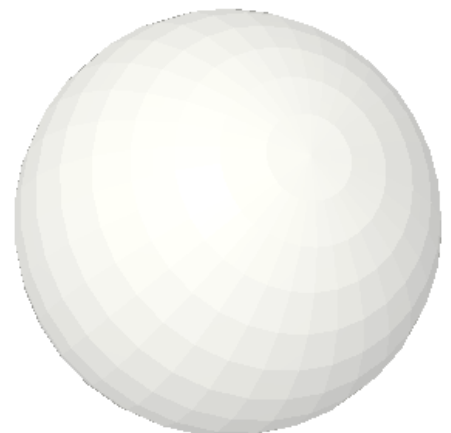
Dans notre exemple, nous choisissons **BOX**. L'origine (0, 0, 0) est définie au coin inférieur gauche et les longueurs dans les sens positifs.

Notre forme globale est donc définie comme suit : Origine x = -50 000, y = -50 000, z = -50 000 et tailles en x = 100 000, en y = 100 000, et en z = 100 000.

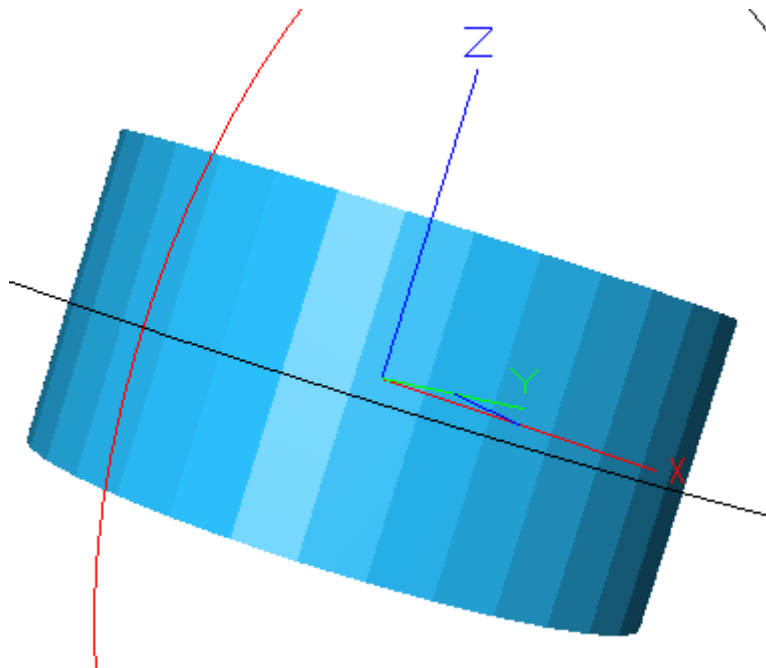


General	
Name	Box
Type	Box
Comment	no comment
Importance	default
Position	
X-Rel.	-50000.00
Y-Rel.	-50000.00
Z-Rel.	-50000.00
Rot. X	0.00
Rot. Y	0.00
Rot. Z	0.00
Viz Attributes	
Contours	Yes
Contour color	Black
Contour style	Solid
Contour width	Normal
X-Ray mode	Off
Edge-Ray mode	Off
Properties	
Material	Blackhole
Size-X	100000.00
Size-Y	100000.00
Size-Z	100000.00

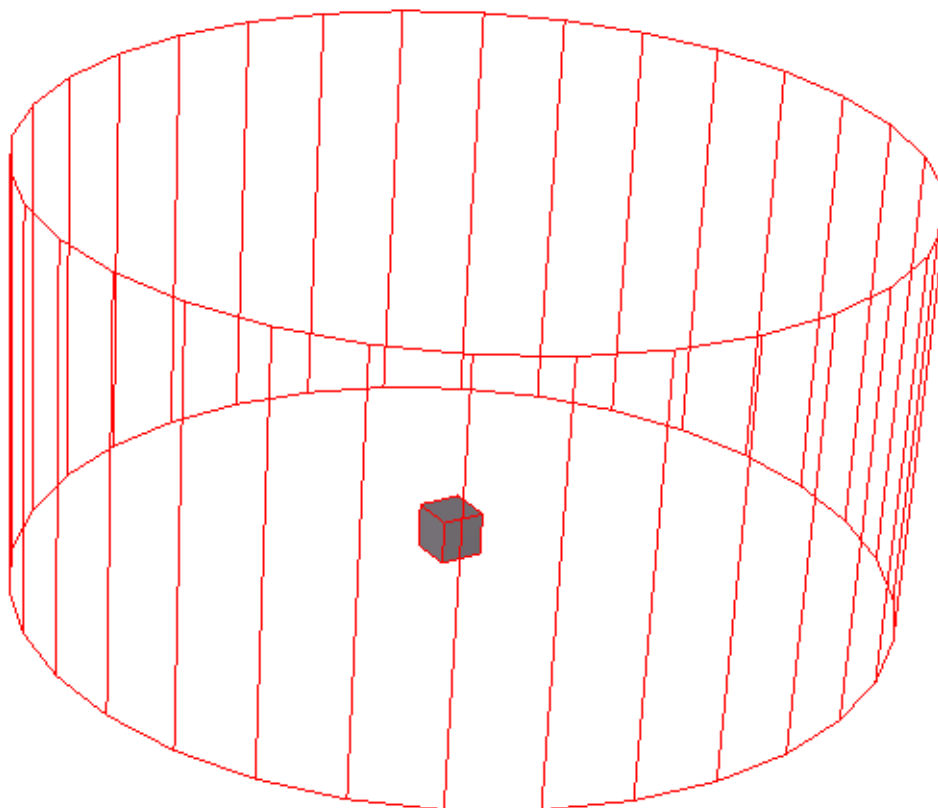
Ensuite, nous définissons une sphère pour le **vacuum**. Par défaut, l'origine (0,0,0) est définie au centre de la sphère. Ici, le rayon est de 10 000.



Nous définissons ensuite un cylindre. Par défaut, l'origine (0,0,0) est le centre de la face inférieure, la hauteur est dirigée dans le sens des z positifs. Il est possible de définir des cylindres infinis. L'orientation dans d'autres plans se fait par les options **ROT X, Y, Z**. Ici le cylindre a pour origine (0,0,0), 1 000 de rayon et 1 000 de hauteur.



Notre 4e forme est à nouveau une boîte, son origine (coin inférieur gauche) se situe en $X = 0$, $Y = 0$ et $Z = 100$, ses dimensions sont de 100 dans toutes les directions.



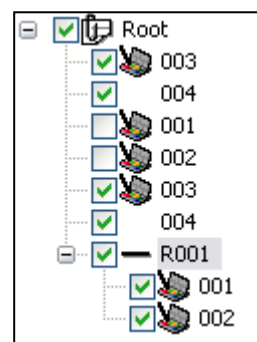
Si l'on veut ensuite se servir de cette géométrie dans un fichier d'entrée **FLUKA** il faut remplacer tous les noms par des chiffres. Cela s'effectue dans la fenêtre de droite dans la case **name** de la partie **general**.

Nous avons défini les formes, les « bodies », il faut à présent définir les régions.
Pour les régions à un seul corps, il suffit d'assigner le matériau adéquat au corps et la région est définie. Pour les régions plus complexes, il convient d'utiliser l'un des boutons **-**, **+** ou **X** dans la barre d'outils qui représentent les principaux opérateurs booléens.
- représente l'opérateur de soustraction, **+** celui d'union et **X** celui d'intersection.

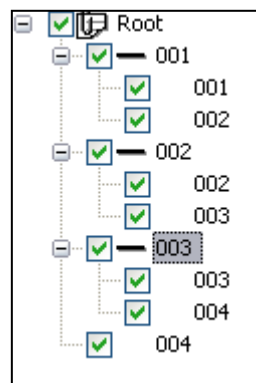
Dans notre cas, la forme 1 englobe la forme 2 pour constituer la région 1, le trou noir.
Il en va de même avec les formes 2 et 3 constituant la région 2, le vide.
Il en va de même avec les formes 3 et 4 constituant la région 3, l'eau.
La région 4 est entièrement contenue dans la forme 4, l'argent.

En pratique, on clique droit sur la forme et on sélectionne **clone primitive** qui crée une copie de celle-ci, il faut créer autant de copies que la forme sera utilisée dans la description des régions. On crée une **différence** et avec la souris on y glisse et dépose (**drag & drop**) les formes constituant la différence dans l'ordre : d'abord la forme à laquelle on va soustraire la ou les autres.

Ci-contre, la région **R001** est constituée de par la différence entre les corps 1 et 2 définis précédemment. Par défaut, à la création, la différence s'appelle **différence** et est précédée du signe **-**.
On procède de même pour définir les régions 2 et 3, en veillant avant chaque création de **différence** de revenir à la racine (**root**) sinon la **différence** sera créée à l'intérieur de la première.



Ci-contre, le résultat final en mode développé.



L'exportation de cette géométrie dans un format compatible avec **FLUKA** s'effectue à l'aide de **File/Export/Fluka**. Si à ce moment-là, le format ne correspond pas (i.e. les régions et corps ont un nom non compatible), l'exportation ne se fait pas.

Ci-dessous, un exemple.

```
* D:\FLUKA\FLUKA\memo\version6\Simple245.dat
* Created: 16.3.2007
* At: 8:27:22
TITLE
MC-CAD Test
GEOBEGIN
0 0 MC-CAD
```

```

* 001
RPP 001 -50000.00 50000.00 -50000.00 50000.00 -50000.00 50000.00
* 002
SPH 002 0.00 0.00 0.00 10000.00
* 003
RCC 003 0.00 0.00 0.00 0.00 0.00 1000.00
1000.00
* 004
RPP 004 0.00 100.00 0.00 100.00 100.00 200.00
END
* 001; assigned material: Blackhole; mat # (1)
0001 +1 -2
* 002; assigned material: Vacuum; mat # (2)
0002 +2 -3
* 003; assigned material: Water; mat # (26)
0003 +3 -4
* 004; assigned material: Silver; mat # (13)
0004 +4

END
GEOEND
ASSIGNMAT 1.0 1.0
ASSIGNMAT 2.0 2.0
ASSIGNMAT 26.0 3.0
ASSIGNMAT 13.0 4.0
    
```

Évidemment, les index des matériaux sont ceux définis dans **SimpleGeo**. Ils ne correspondront pas forcément à ceux figurant dans votre liste de matériaux.

12.3. Utilisation de la macro DaVis3D

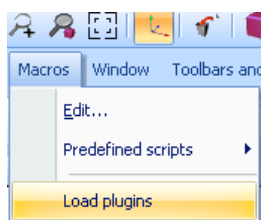
Les plugins sont à décompresser dans le répertoire où est installé **SimpleGeo** (*cf. procédure d'installation livrée avec l'archive*).

Le plugin **DaVis3D** permet entre autres de superposer les valeurs d'un estimateur **USRBIN** sur la zone choisie de sa géométrie.

Nous ne revenons pas sur la compilation des résultats (de plusieurs cycles) issus de la simulation (utilisation de la routine **usbsuw** pour des fichiers binaires, cf. paragraphe 8.2.)

Le plugin n'accepte que les fichiers de type **ASCII**, si l'on a un fichier de résultats binaires, il faudra le convertir à l'aide de la routine **usbrea** livrée avec **FLUKA**.

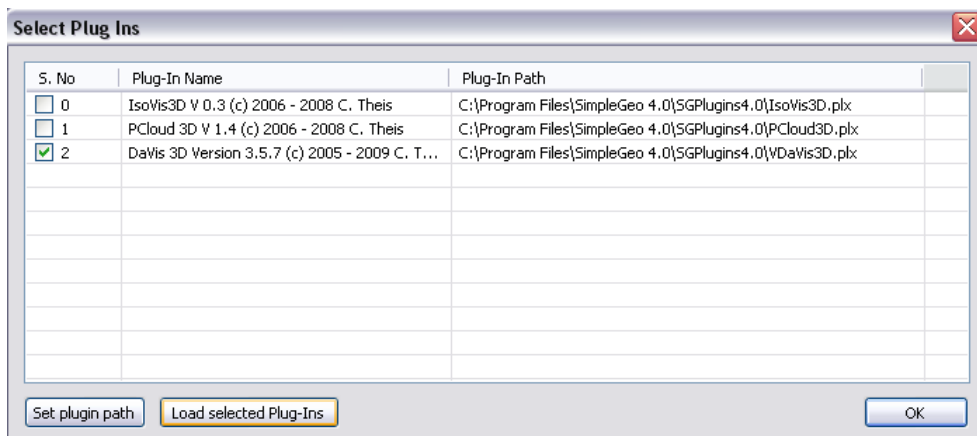
Par défaut, les macros ne sont pas activées dans le logiciel, il faut les charger.



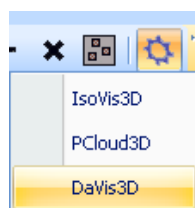
Sélectionnez :

Si les plugins sont dans le répertoire d'installation de **SimpleGeo**, le chemin pour y accéder sera trouvé automatiquement, l'indiquer sinon.

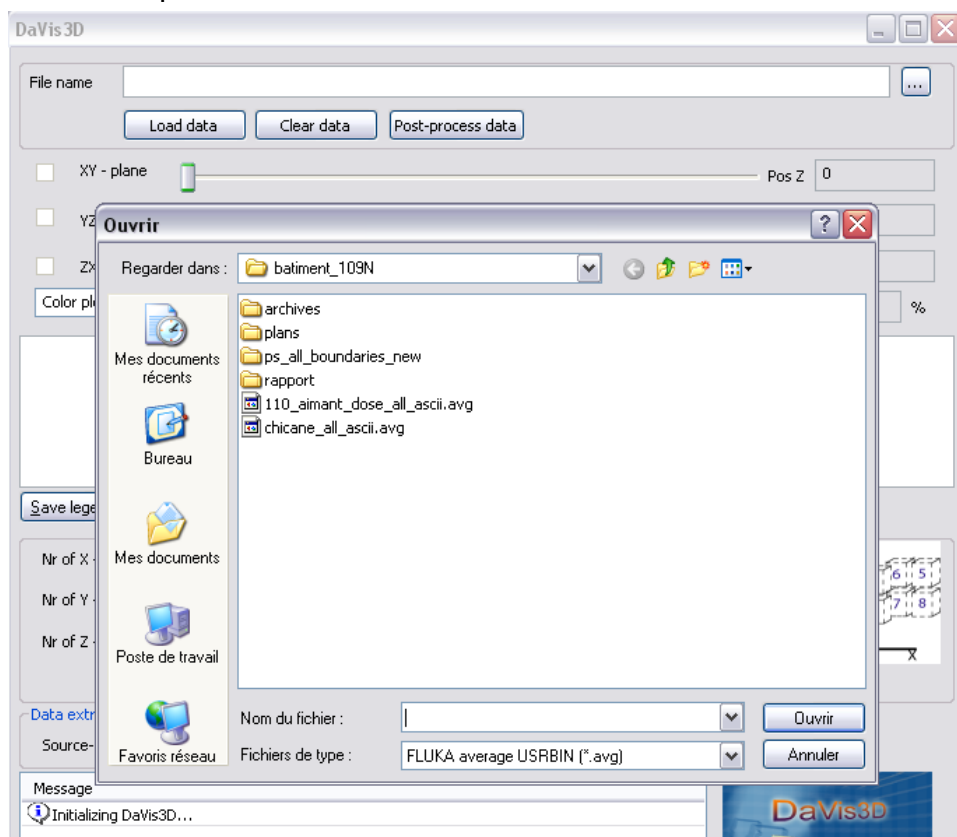
Ne pas oublier « Load selected Plug-Ins » :



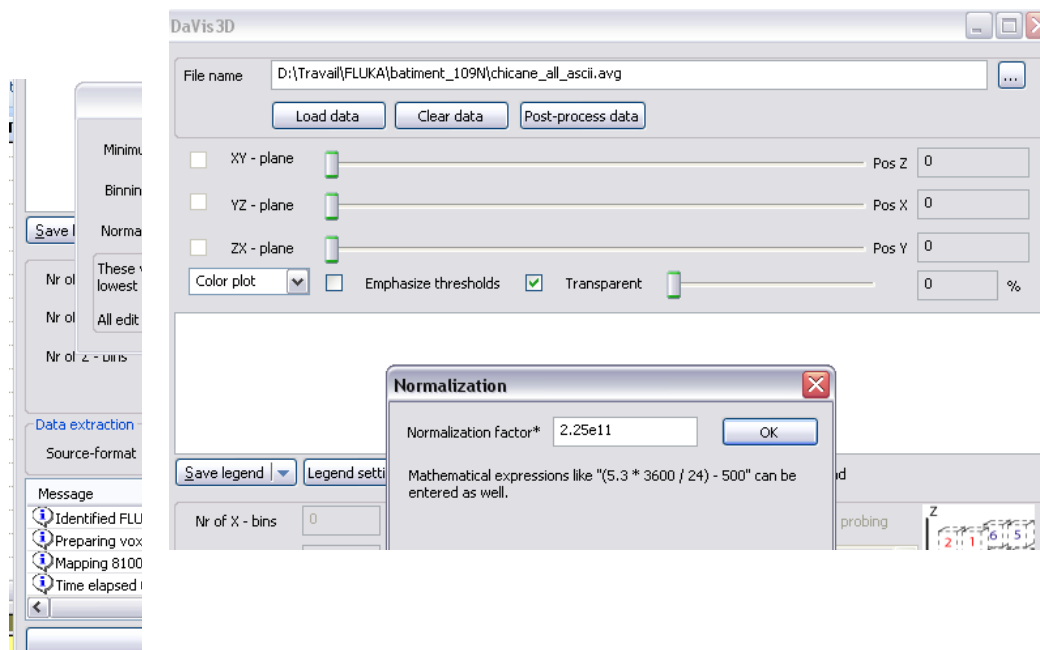
Pour activer **DaVis3D**, sélectionner l'onglet idoine :



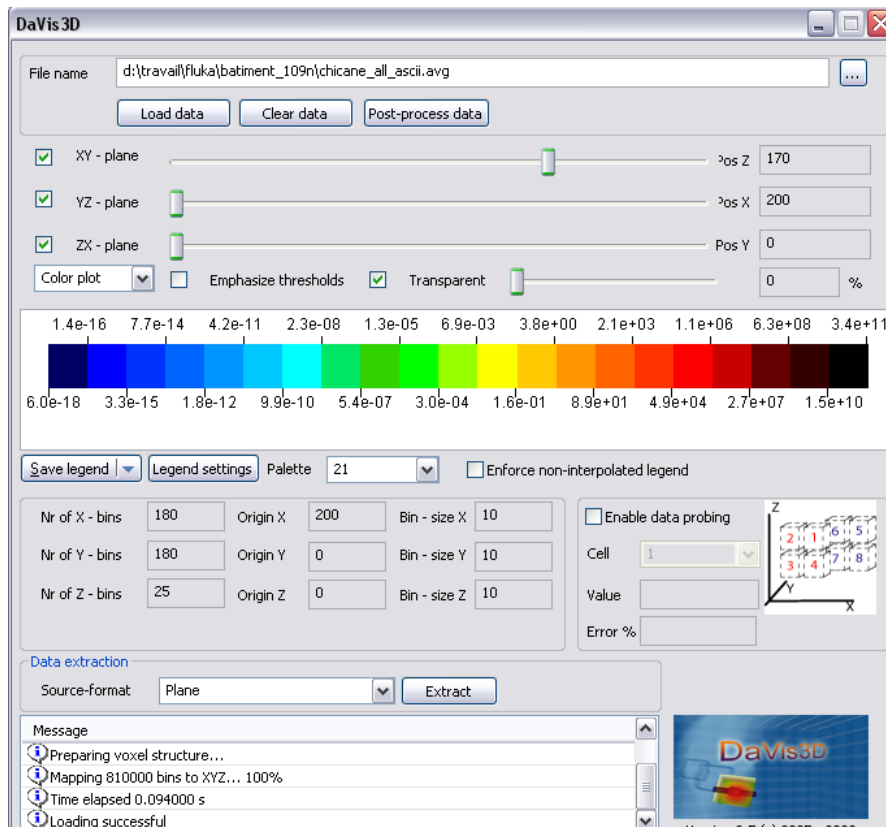
Une fenêtre s'ouvre, en activant « Load data » une fenêtre classique d'explorateur de fichier se superpose à cette première fenêtre :



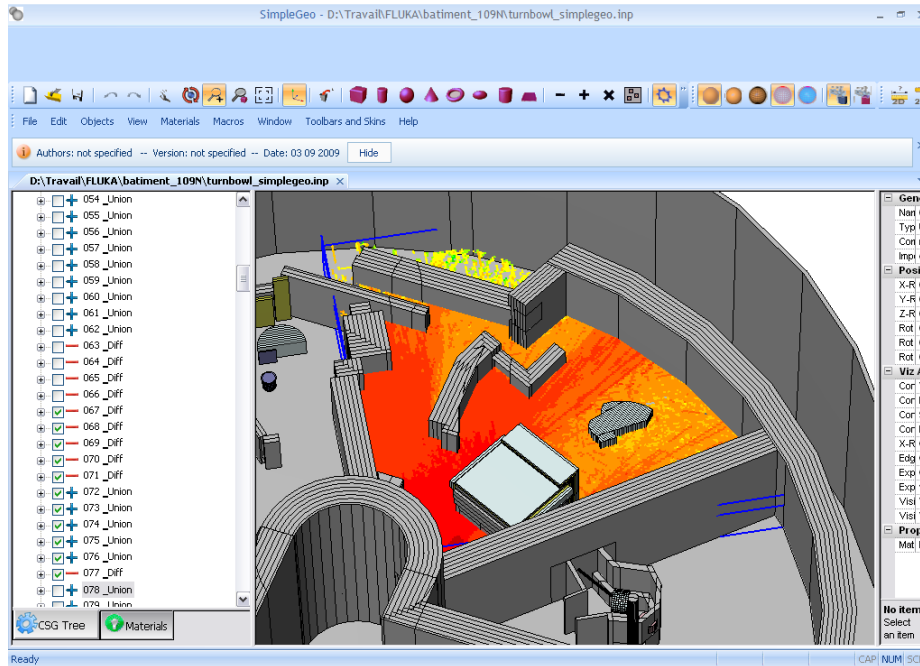
On choisit un fichier avec l'extension ***.avg** de format **ASCII** regroupant les valeurs moyennes issues de plusieurs cycles d'un estimateur **USBIN**. On choisit le facteur de normalisation (la valeur par défaut est 1), on clique **OK**.



On peut ensuite sélectionner son type de projection (quel plan, à quelle position, etc.), le type de graphe (color plot, smooth color plot, contour plot, surface plot) qui offrent diverses nuances à tester pour savoir laquelle convient le mieux au graphe que l'on veut tracer.



Note : vous voyez dans la fenêtre de message de DaVis3D « loading successful » qui signifie que le fichier de données a été bien chargé par le programme.
Enfin, un exemple de ce que cela peut donner à l'écran :



13. Utilisation de l'interface FLAIR : initiation

By Vasilis Vlachoudis & Tony Empl (CERN)

13.1. Généralités

Référence obligatoire et beaucoup plus complet que ce simple tuto, le manuel **flair.pdf** et le lien vers le site web : <http://www.fluka.org/flair/index.html>

Il s'agit d'une interface avancée pour **FLUKA** afin de faciliter l'édition de fichiers d'entrée, la construction de l'exécutable, l'exécution du code, la visualisation des fichiers de sortie, la fusion des fichiers de données, et la génération de graphes de qualité à l'aide de **GNU PLOT** ainsi que des graphes 3D photo-réalistes de la géométrie avec les données superposées avec **POVRAY**. Une plate-forme entièrement basée sur **python** et **Tkinter**.

Il faut une configuration minimale pour faire fonctionner **flair** sous **linux** :

- L'interprète **Python** version 2.3 ou ultérieures. A priori, il est inclus dans votre distribution linux : <http://www.python.org>
- La boîte à outils **Tkinter**, c'est la boîte à outils graphiques par défaut pour python et est normalement incluse dans **python**. Ce n'était pas le cas pour ma machine, il faut alors demander à un gentil **root** de vous l'installer, si vous êtes un simple **user**.
- **Tcl/Tk** version 8.4 ou ultérieures. Habituellement installé sur tous les systèmes **linux**, et sur **MS-Windows** inclus dans la distribution **python**.
- Éventuellement, pour le tracé des graphes :
 - **gnuplot** version 4.0 ou ultérieure : <http://www.gnuplot.info>
 - **PovRay** version 3.6 ou ultérieure : <http://www.povray.org>

Installation : nous ne décrivons que la méthode proposée aux utilisateurs **linux** n'ayant pas de pouvoirs étendus sur leur machine.

Pour une installation manuelle, téléchargez l'archive (*.tar) de la dernière version sur le site web de flair et décompressez-la dans le répertoire voulu à l'aide des commandes suivantes :

```
$ tar xzvf flair-X-XX.tgz ou $ gunzip -c flair-X-XX.tgz | tar xvf -
```

Notez que ceci ne créera aucune association avec les extensions ***.flair** et ***.inp** et ça ne créera pas davantage les liens dans le menu "démarrer". Ceci doit être fait à la main : prenez le raccourci **flair.desktop** figurant dans le répertoire d'installation et collez-le sur le bureau, clic droit, modifiez les propriétés en mettant le véritable chemin d'accès vers l'exécutable de **flair**.

Il convient également d'ajouter un alias dans votre fichier de configuration :

```
alias flair=/.../flair
```

avec ... = **chemin-où-vous-avez-installé-flair**

et tant que vous y êtes :

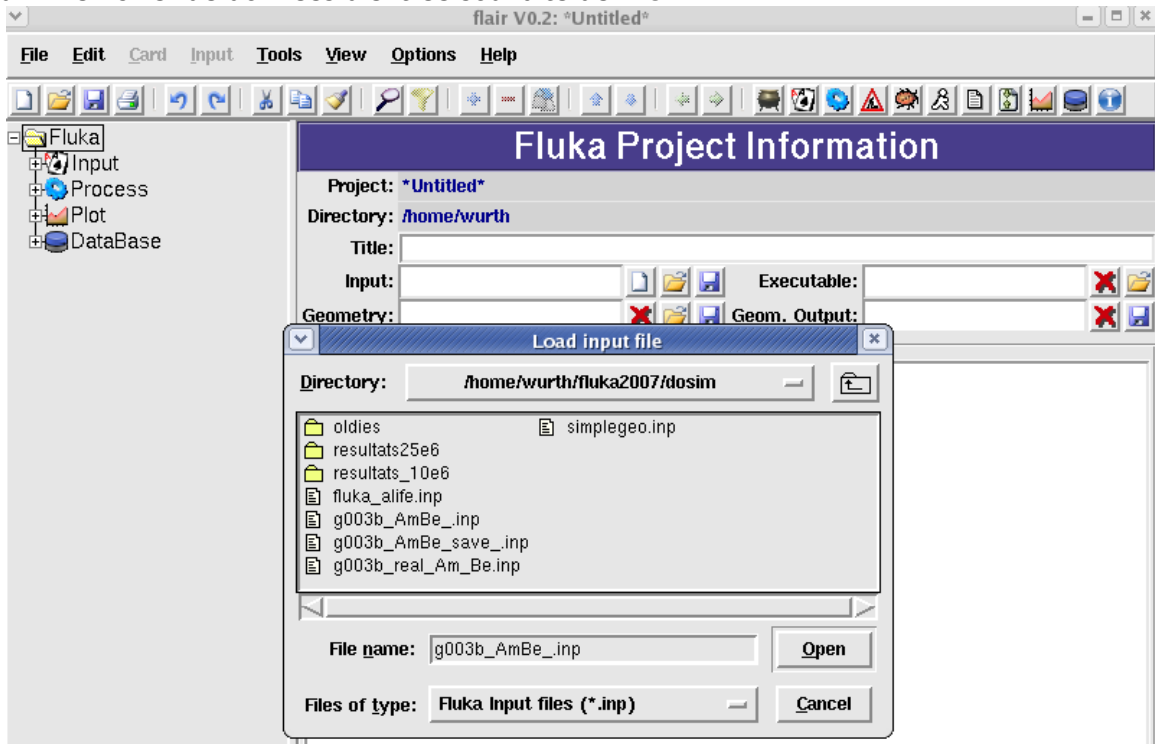
```
alias pt='PYTHONPATH=/.../lib python /.../PeriodicTable.py (chemin vers les tables d'isotopes)
```

```
alias fm='PYTHONPATH=/.../lib python /.../Manual.py (chemin vers le manuel)
```

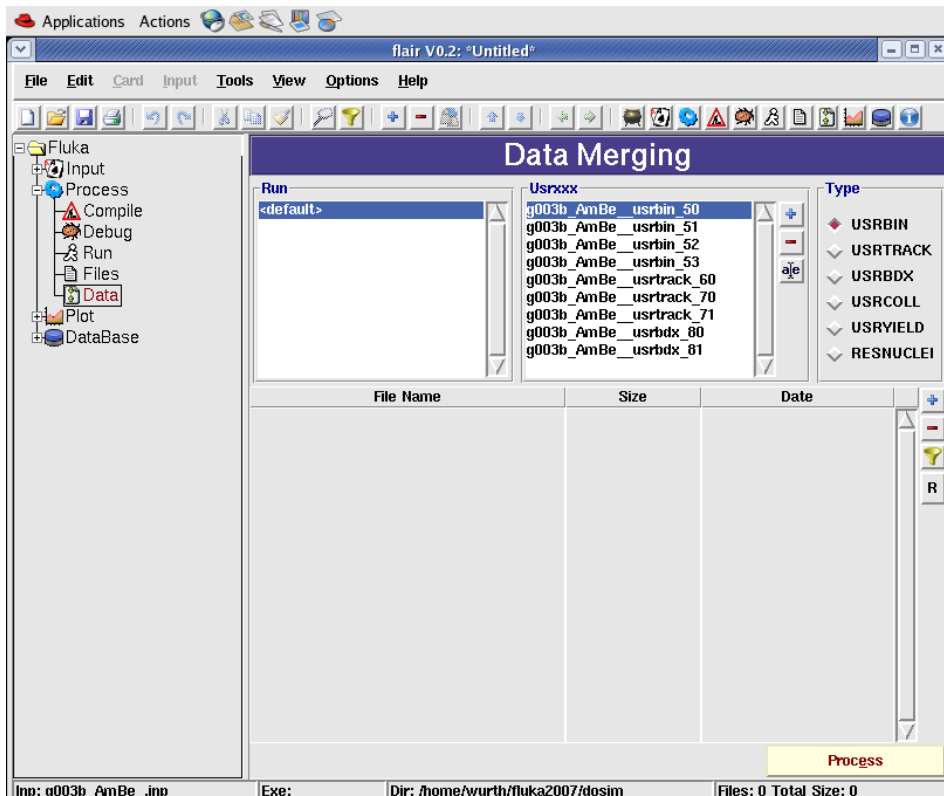
Nous allons décrire principalement l'utilisation de **FLAIR** dans le cadre de l'exploitation des données issues d'une simulation.

13.2. Charger un fichier et inspecter les données

Ouvrir le fichier de données d'entrée souhaité de **FLUKA** :



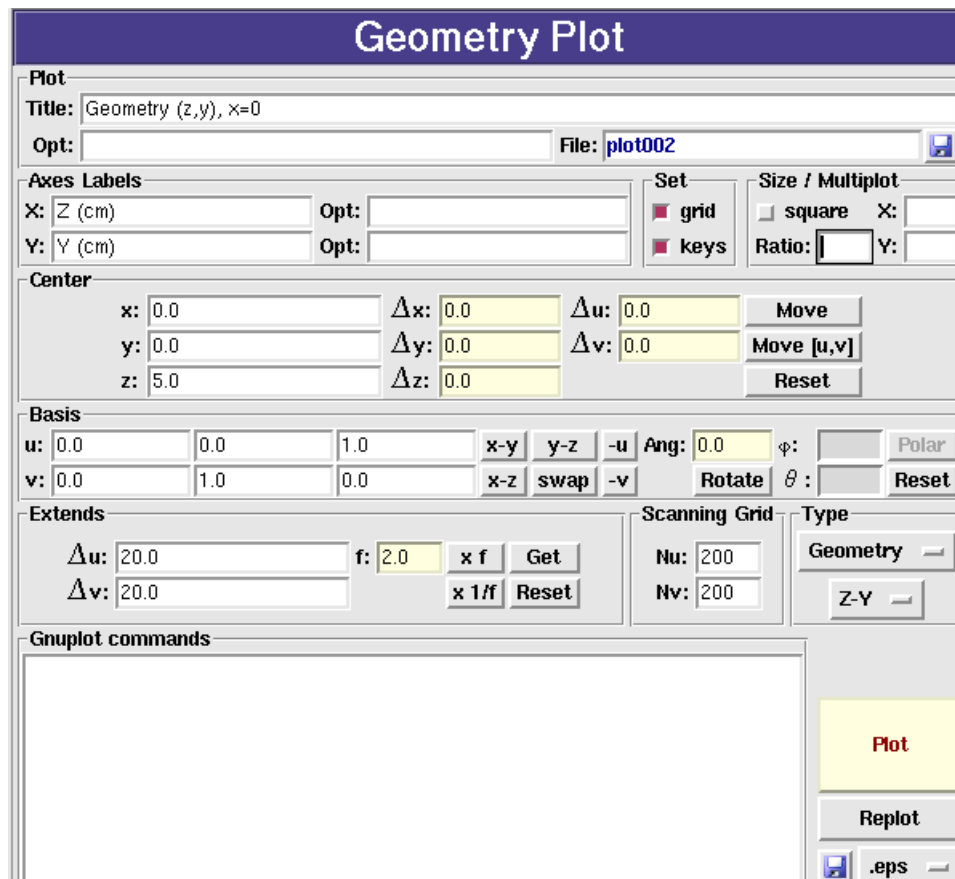
À ce moment-là, toutes les données sont lisibles dans **FLAIR** : onglet **Input**
On peut également visionner tous les fichiers de sortie de chaque cycle : onglet **Process/Files**
Afin de fusionner les données de plusieurs cycles : onglet **Process/Data**



Vous sélectionnez le type souhaité ainsi que la série de fichiers à compiler.
 Dans notre exemple : **g003b_AmBe_usrbin_50** va regrouper tous les fichiers **g003b_AmBe_xxx_fort.50** qui sont les résultats de l'estimateur **USRBIN** défini dans le fichier d'entrée, avec **xxx** = numéro de cycle.
 Vous pressez **Process : Flair** va créer le fichier regroupant les résultats sur les **xxx** cycles du même estimateur.
 Cette opération est strictement la même que celles effectuées par les programmes **usbsuw** pour **USRBIN**, **ustsuw** pour **USRTRACK** et **USRCOLL**, **usxsuw** pour **USRBDX**, **usysuw** pour **USRYIELD** et **usrsuw** pour **RESNUCLEi**. Ces programmes sont appelés par **FLAIR** au lieu d'être appelés directement par l'utilisateur.

13.3. Généralités sur les graphes et représentation de la géométrie

Général : onglet **Plot**, sélectionnez le type : **Geometry, USR-1D, USRBIN, USERDUMP, RESNUCLE**.
 Dans la **plot list**, allez sur le graphe concerné et double cliquez, ceci met le type de graphe à jour et vous ouvre une fenêtre. Nous allons détailler les différents types des graphes que l'on peut obtenir.



Plot : titre et police d'affichage au choix (cette dernière option est valable uniquement pour la sortie au format ***.eps**)

Axes Labels : titres et polices d'affichage.

Set : on peut choisir d'afficher **grid** ou non pour une meilleure lecture des coordonnées à l'aide d'une grille.

Size / Multiplot : options pour définir la taille du graphe dans la fenêtre, afin d'en tracer plusieurs dans un même fichier.

Center : coordonnées de l'origine du graphe (0,0,5 ici) et coordonnée absolue du plan sur lequel on choisit de projeter. Dans notre exemple, nous projetons sur le plan (z,y), le graphe est donc donné à $x = 0.0$. Les autres cases permettent de décaler cette origine.

Basis : coefficients directeurs $u =$ l'abscisse du graphe, $v =$ l'ordonnée du graphe.

C'est dans cette partie que l'on choisit le type de projection, pour tracer une projection dans le plan (z,y), choisissez (y,z) et **swap** ou renseignez les cases à la main.

Vous pouvez également choisir un angle de rotation. Choisissez la valeur et appuyer sur

Rotate.

Extends : donne le domaine du tracé ici 20.0 sur l'axe des abscisses et l'axe des ordonnées. Pour changer automatiquement ces valeurs, vous pouvez choisir un facteur f et multiplier ou diviser par ce facteur en appuyant sur **x f** ou **x 1/f**.

Type : plusieurs choix possibles, **geometry** (notre exemple) qui correspond à un graphe **all boundaries** obtenus avec **ALIFE** ou **pawlevbin** (devenu **gplevbin** lorsqu'il est lié à **flair**), **regions**, **material** (au point depuis la version 0.4), **lattice** et **magnetic field**.

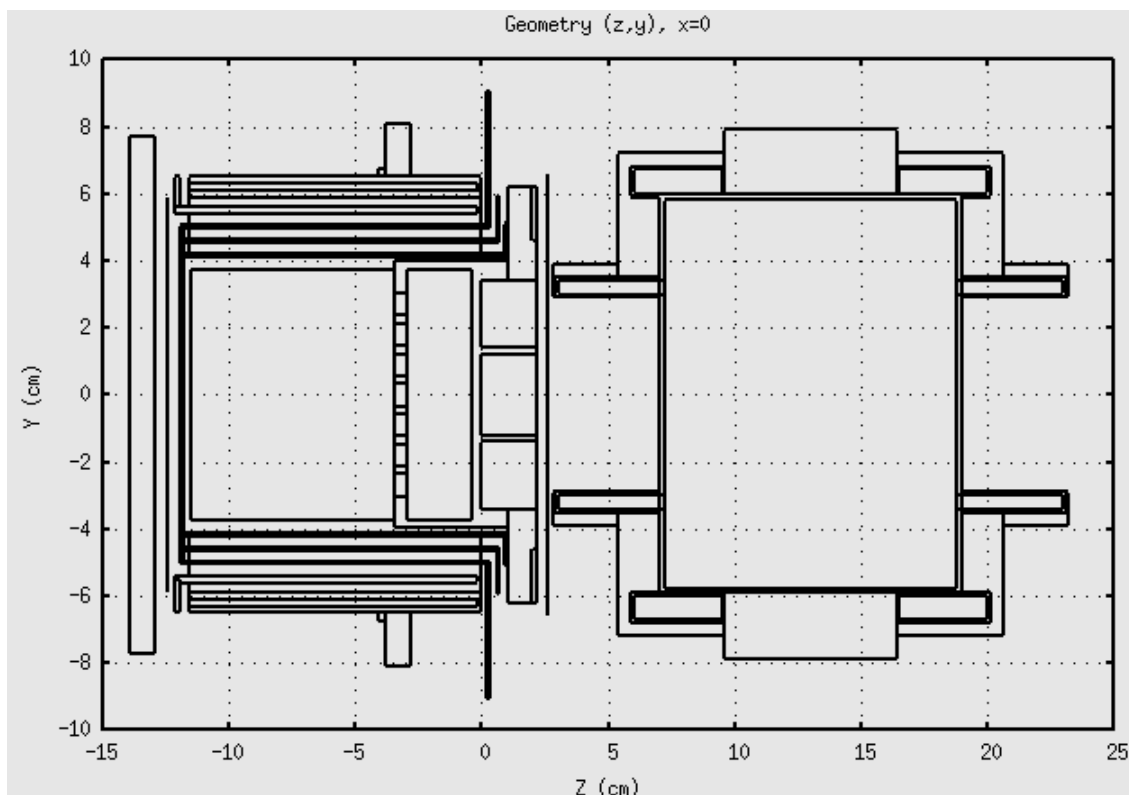
Gnuplots commands : l'utilisateur peut ajouter toutes les commandes qu'il souhaite voir exécutées par **gnuplot** à condition de respecter la syntaxe.

Nous verrons d'autres exemples de commandes gnuplot dans les paragraphes suivants, retenons par exemple *set xrange [-50 :65]* qui force le domaine des abscisses du graphe quelque soit la commande indiquée précédemment.

Pour tracer le graphe : presser **Plot**.

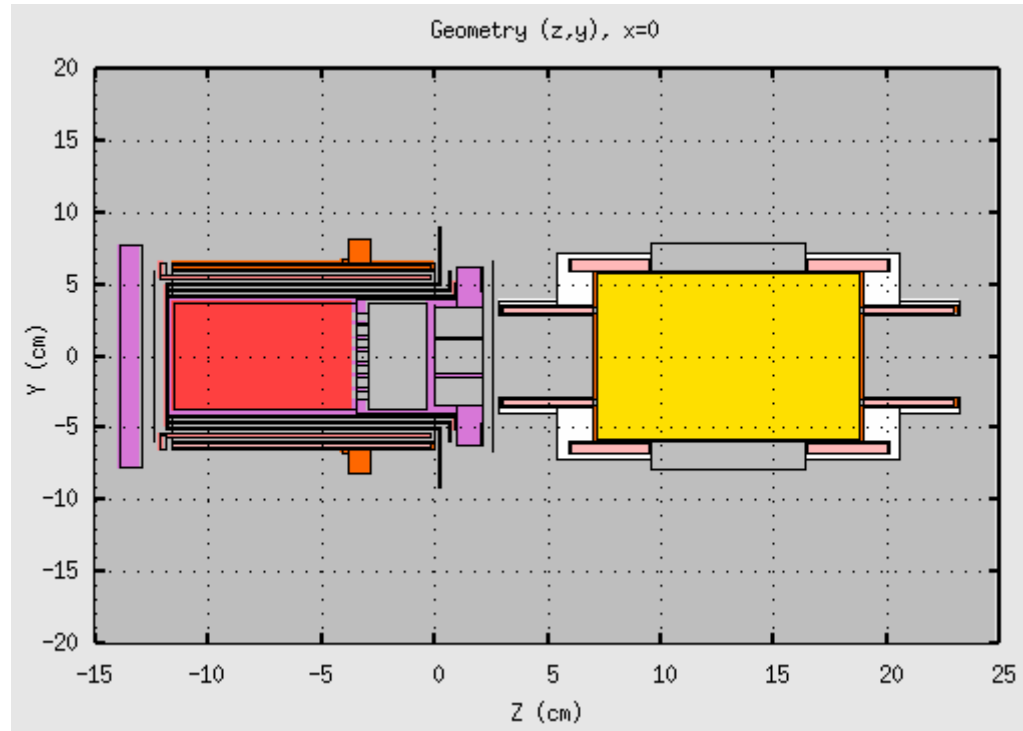
Quelques instants peuvent être nécessaires à l'obtention d'un graphe, **Flair** exécute **FLUKA** avec une carte **PLOTGEOM** ayant les caractéristiques voulues.

Exemple : ensemble cible-source de Spiral2.



Le graphe peut être sauvé en format ***.eps** (ou ***.png** ou ***.svg**), en appuyant sur la petite disquette à côté du choix du format de sortie.

Exemple : ensemble cible-source de Spiral2 avec l'option **material** : *les paramètres extends et center sont importants ici, ce sont les mêmes que précédemment, seul le type a été modifié.*



13.4. Représentation d'un graphe USRBIN

Dans la partie **Axes Labels**, nous trouvons un champ supplémentaire par rapport au tracé de géométrie, **CB = color bar**, la légende de la barre de couleurs.

Detector : il s'agit de charger un fichier généré par l'option **Data merging** ou un fichier obtenu manuellement à l'aide du programme **usbsuw**.

Binning info par détecteur présent dans le fichier (un seul souvent), le **type** (10 : X-Y-Z = binning cartésien), le type de particule mesuré (neutron ici), les limites du **binning** et le nombre de **bins**. Les valeurs minimale et maximale et de l'intégrale calculées par **gplevbin** (**pawlevbin**) exécuté directement par **Flair**.

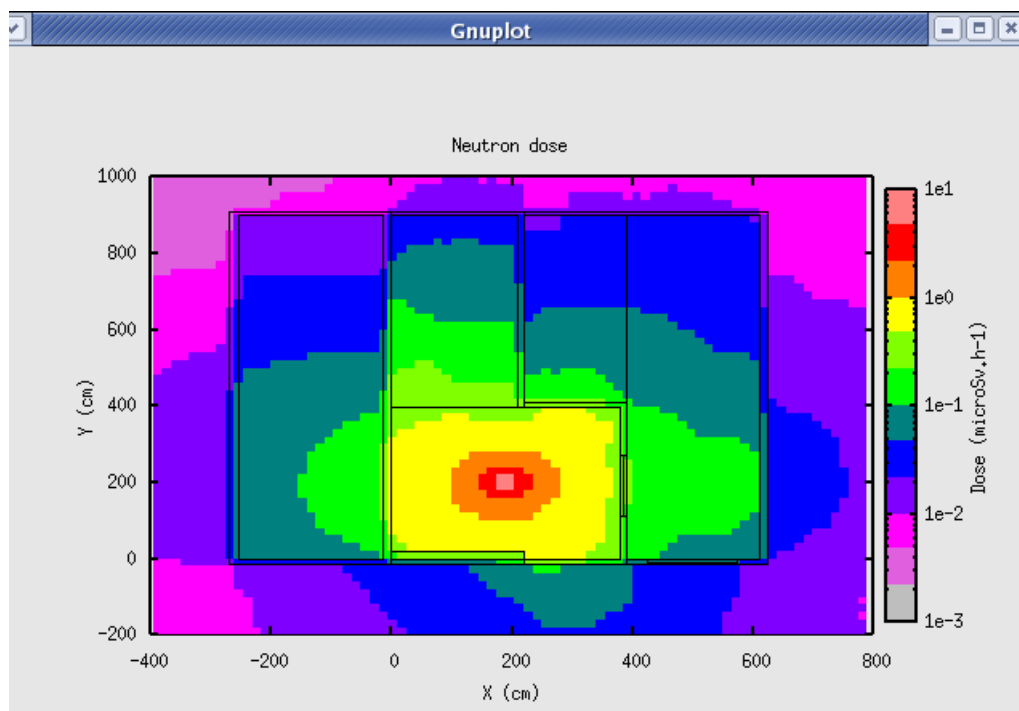
Color Band : le facteur de normalisation, ici il s'agit de la valeur exacte **f** (et non pas **-1/f** comme pour **pawlevbin**), la valeur minimale, le nombre de couleurs par décade (**CPD**) et le nombre de couleurs utilisées **colors** (valeur maximale = 30).

Il est possible de superposer la géométrie (si l'on sélectionne **Auto**, les axes et la projection seront les mêmes que pour le détecteur).

Ici aussi, j'ai utilisé **Gnuplot Commands** pour choisir mes limites de graphe ainsi que le format de l'affichage de la bande de couleurs.

Le graphe est tracé à l'aide de la commande **Plot** (cf. page suivante).

Ici aussi, il est possible de sauver le graphe dans l'un des trois formats cités précédemment. Mais à l'heure où j'écris ces lignes, je vous conseille la capture d'écran car les conversions ne sont pas terribles, le format ***.eps** propose des couleurs assez mal rendues et un décalage de l'échelle, le format ***.png** propose un damier assez immonde en noir et blanc et le format ***.svg**, une synthèse de deux autres.



Sous **linux**, la touche **Impr écran** propose de sauver une image en format ***.png** que l'on peut retravailler ensuite en la rognant. La taille du fichier obtenu est tout à fait raisonnable (une dizaine de ko).

Remarque : les résultats habituels d'un estimateur **USRBIN** sont donnés en $\text{part/cm}^2/\text{particule incidente}$. Or ici, nous utilisons la routine **deq99c** qui convertit la fluence en dose équivalente (unité des facteurs : pSv.cm^2), les résultats sont donc proposés en pSv/neutrons . Pour passer en $\mu\text{Sv/h}$, il faut multiplier par $4.8\text{e}5$ neutrons/s (émission de notre source) * 3600 (s en 1 h) et $1\text{e-}6$ (conversion pSv - μSv) = 1728.

13.5. Représentation d'un graphe USR-1D

Il s'agit de tracer des graphes issus des détecteurs, **USRBDX**, **USRTRACK**, **USRCOLL** et **USRYIELD**.

Axes Range : on peut choisir l'échelle logarithmique, donner les valeurs limites pour deux séries d'axes.

Pour charger les résultats d'un détecteur, il faut appuyer sur le **+**, charger le fichier contenant les résultats (ici un fichier obtenu avec l'option **data merging** regroupant les résultats d'un détecteur **USRTRACK**).

Ne pas oublier de sélectionner le détecteur par son nom dans le **SDUM** de la carte correspondante du fichier d'entrée **FLUKA** (ici **neureg38**) même s'il n'y en a qu'un seul ! Choisissez le type de coordonnées portées en x : la racine du produit des bornes de l'intervalle (**GeoMean**), la valeur moyenne des valeurs limites des intervalles (**Mean**), les bornes inférieures des intervalles (**Low**) ou les bornes supérieures des intervalles (**High**).

Choisissez le type de coordonnées portées en y : La valeur donnée dans le fichier de résultats (**Y**), le produit de Y par la valeur moyenne de l'intervalle d'énergie correspondant (**Y x <X>**), le produit de Y par la borne inférieure de l'intervalle correspondant (**Y x Xl**), le produit de Y par la borne supérieure de l'intervalle correspondant (**Y x Xh**) ou le produit de Y par la taille de l'intervalle (**Y x DX**).

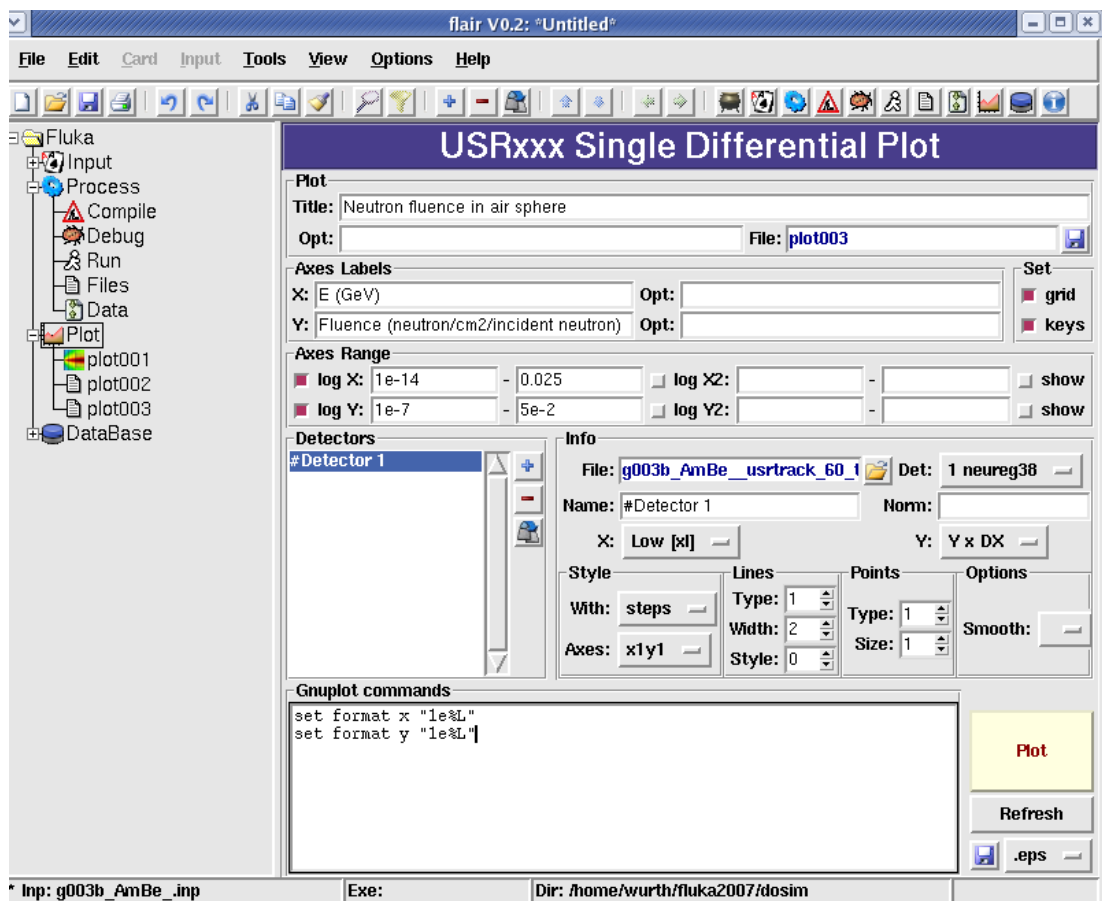
La dernière option est celle retenue.

Y dans notre cas est la fluence par cm^2 par particule primaire et par GeV.

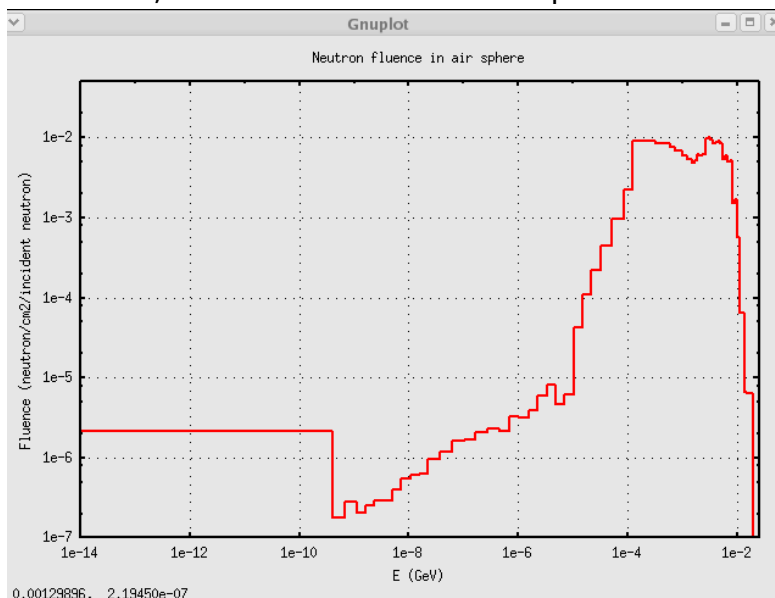
Pour obtenir la fluence par cm^2 par particule primaire, il s'agit donc de multiplier **Y** par la largeur de l'intervalle d'énergie correspondant, CQFD.

Style : permet de « customiser » son graphe, nous avons choisi **steps** pour figurer qu'il s'agit d'intervalles ici, une largeur de 2.

Dans la partie **Gnuplot Commands**, nous avons fixé le format des axes.



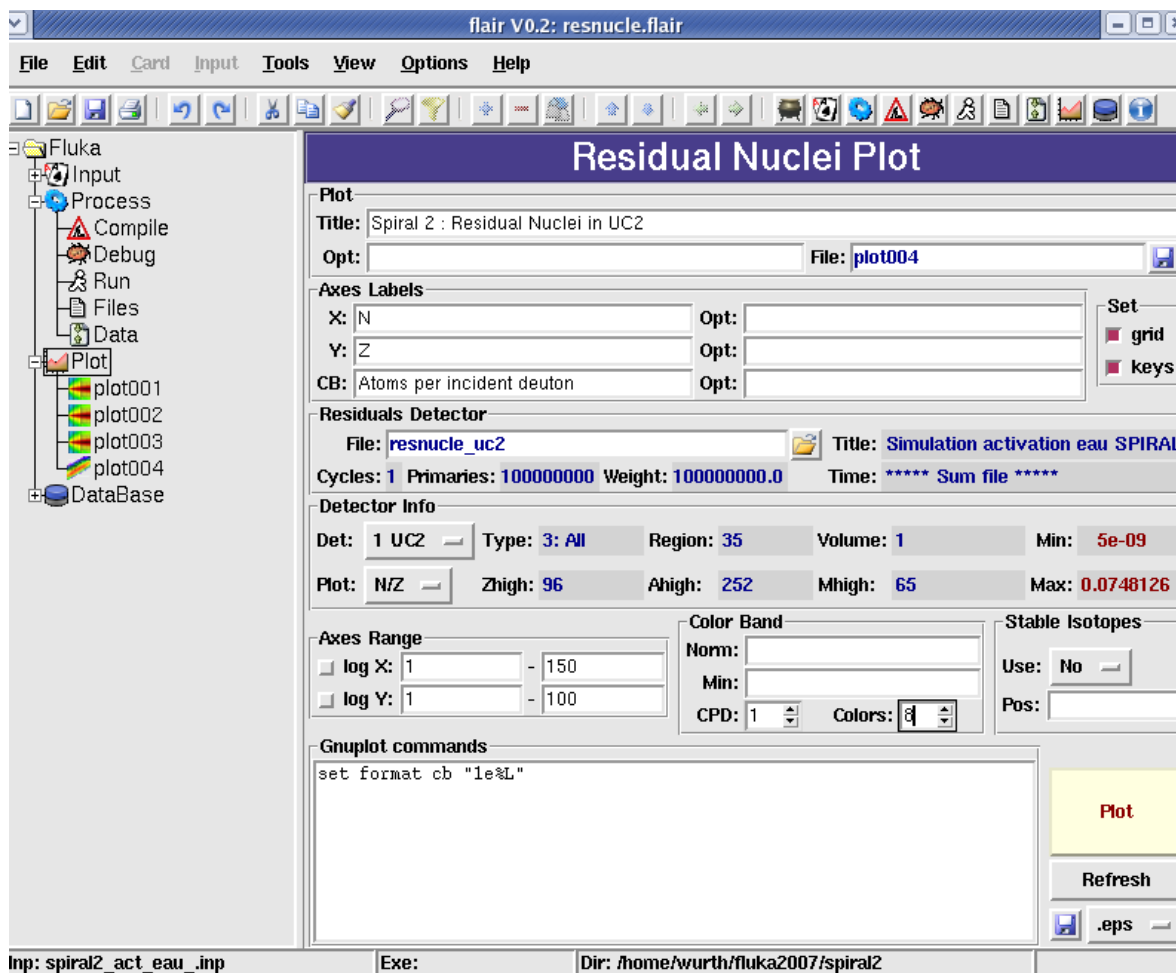
Nous obtenons la figure suivante, que l'on peut sauver dans l'un des trois formats précités. Ici, le résultat est tout à fait acceptable contrairement à la partie **USRBIN PLOT**.



Il s'agit de résultats obtenus lors de la validation du spectre de la source Am-Be (cf. *validation_Am_Be_SD_v1*). Nous avons représenté la fluence des neutrons dans une petite sphère de 0,25 cm de rayon centrée autour du point d'émission de notre source ponctuelle isotrope. Le plateau représente la part des neutrons thermiques.

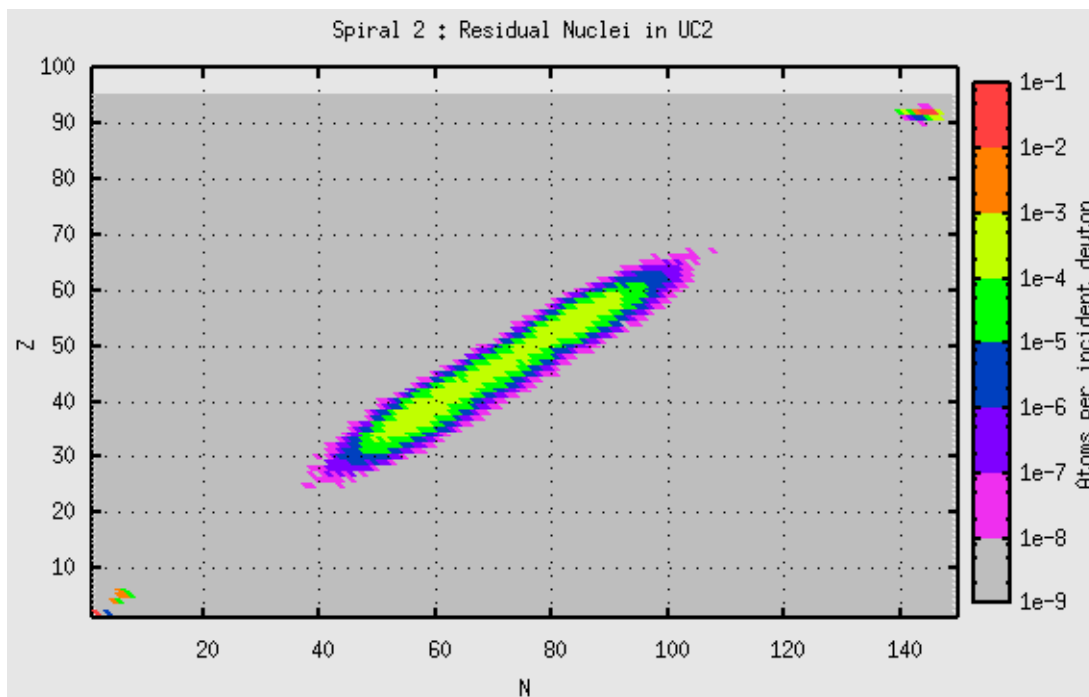
13.6. Représentation d'un graphe RESNUCLEI

Pour tracer un tel graphe, représentant la distribution des nucléides résiduels, nous devons charger un autre fichier d'entrée, celui consacré à l'étude de l'activation de l'eau de refroidissement de la cible de SPIRAL2 (cf. *Spiral2_activation_eau_refroidissement_v3*).

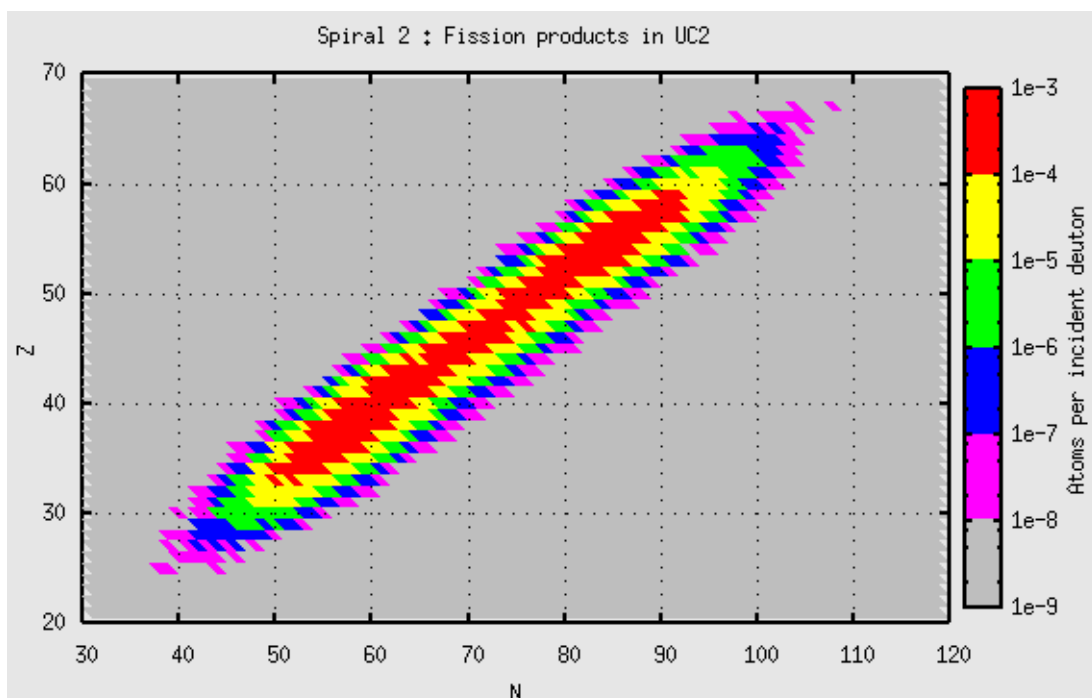


Dans **Detector Info**, on sélectionne le détecteur désiré s'il y en a plusieurs dans un même fichier, et on peut également choisir sa projection, **Z, A, Z vs A, Z vs N, A vs Z** et **N vs Z**. Pour le reste, les options sont les mêmes que pour les tracés précédents. Ci-après la page de commande du graphe **Residual Nuclei** et le graphique correspondant.

Nous avons donc représenté l'ensemble des noyaux produits et présents dans la région de carbure d'uranium de la cible de Spiral2.



Si l'on désire représenter uniquement les produits de fissions : on limite le domaine de N de 30 à 120 et celui de Z de 20 à 70, on assigne 6 couleurs et 1 couleur par décade à la barre de couleurs.



Toutes ces opérations peuvent être sauvegardées par **File/Save** ou **Save As**.
Le fichier aura une extension ***.flair**.

13.7 Utilisation de l'éditeur de géométrie

À partir de la version 0.9.1, il est possible d'utiliser **flair** en éditeur de géométrie, du fait de l'installation d'un module complémentaire.

13.7.1 Intallation

En plus de la procédure décrite au 13.1, il faut également télécharger l'archive : **flair-geoviewer-X.X-X.tgz** (ou celle qui convient pour votre distribution, cf. la page de téléchargement de **flair** sur www.fluka.org).

Ce module **flair-geoviewer** doit être compilé avant d'être installé :

```
$ tar xzvf flair-geoviewer-X.X-X.tgz
$ cd flair-geoviewer-X.X-X
$ make
$ make install
```

Le répertoire d'installation par défaut est : `/usr/local/flair`, si vous désirez l'installer dans un autre répertoire (dans l'exemple : `/my/directory`), la dernière commande ci-dessus doit être modifiée comme suit :

```
$ make DESTDIR=/my/directory install
```

Dans ce cas, il faudra copier manuellement le fichier `geoviewer.so` ainsi créé dans le répertoire d'installation de **flair**. Cette opération est obligatoire, l'éditeur de géométrie ne sera pas détecté au démarrage de **flair** (le message sera le suivant dans l'output window) :

```
"Geometry viewer not found"
"Error: no module named geoviewer"
```

Il se peut également que des bibliothèques nécessaires ne soient pas installées. L'éditeur de géométrie permet d'exporter des images de la géométrie aux formats habituels (bitmap) des fichiers images. Il faut pour cela que les bibliothèques permettant de créer des images sous **python** (l'environnement de **flair** et **flair-geoviewer**) soient installées.

Un message d'erreur apparaît au démarrage de **flair** lorsque ces bibliothèques sont absentes (dans l'output window) :

```
"PIL.Image and PIL.ImageTk not found"
```

PIL is the python-imaging and python-imaging-tk libraries.

13.7.2 Utilisation

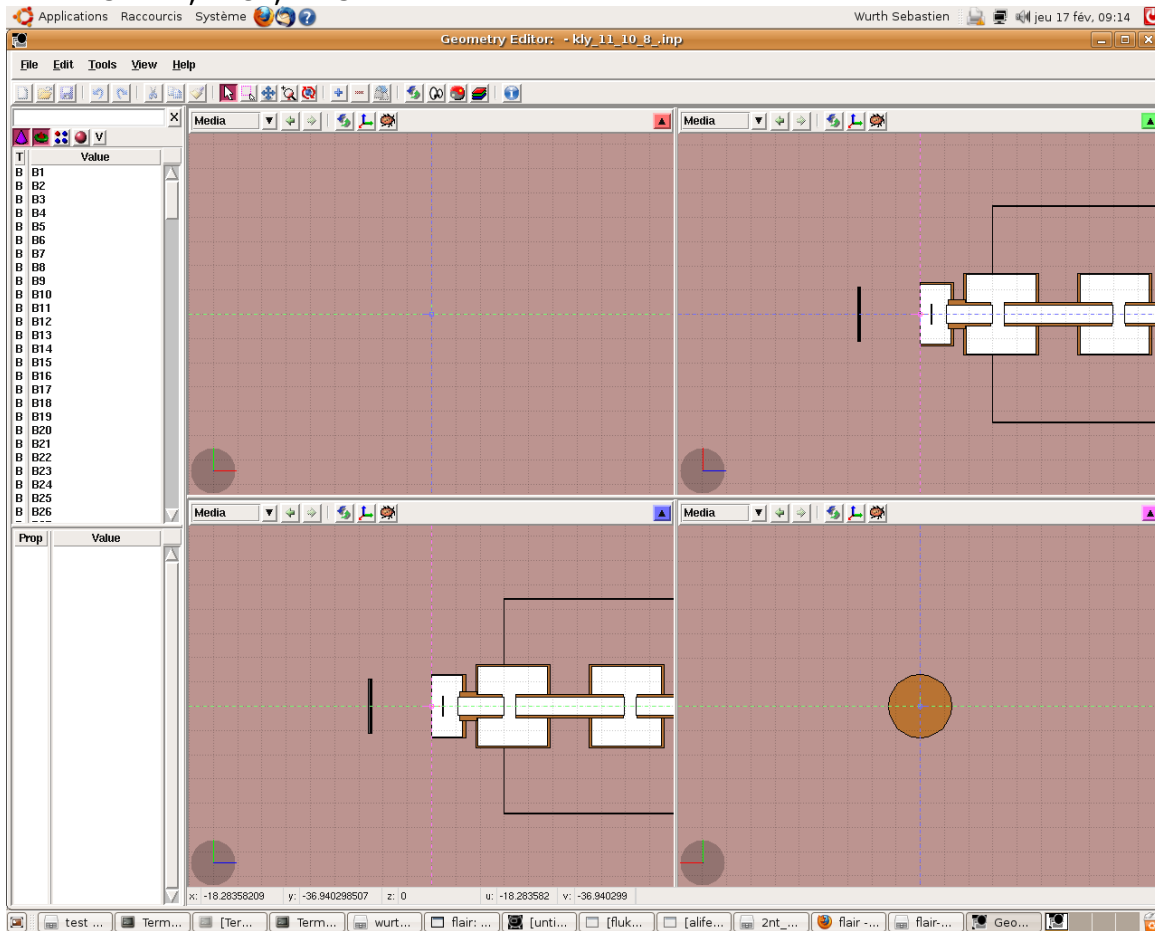
Toutes les possibilités offertes par ce module sont décrites dans l'aide accessible depuis **flair**, section « geometry editor ». Nous en présentons quelques-unes.

Pour qu'une géométrie soit détectée par le module, il faut bien sûr qu'elle soit définie dans **flair** soit par l'ouverture d'un fichier édité par ailleurs, soit par une géométrie créée dans **flair**.

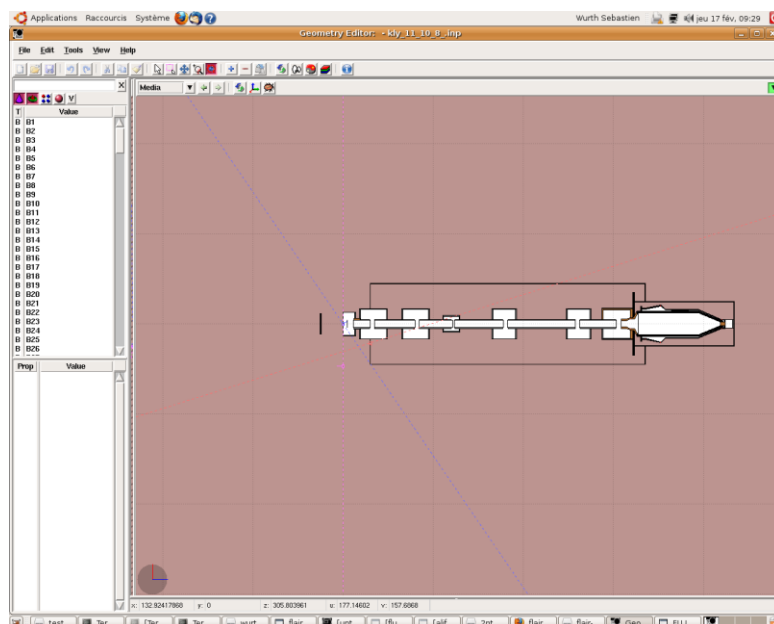
Lancer **geoviewer** : F4 dans **flair** ou petite icône :



- Visualisation : à l'ouverture, 4 fenêtres apparaissent respectivement nommées RED, GREEN, BLUE, MAGENTA.



Elles représentent diverses projections de votre géométrie. En appuyant sur le bouton de couleur en haut à droite, la fenêtre sélectionnée prend alors toute la place (exemple pour GREEN).

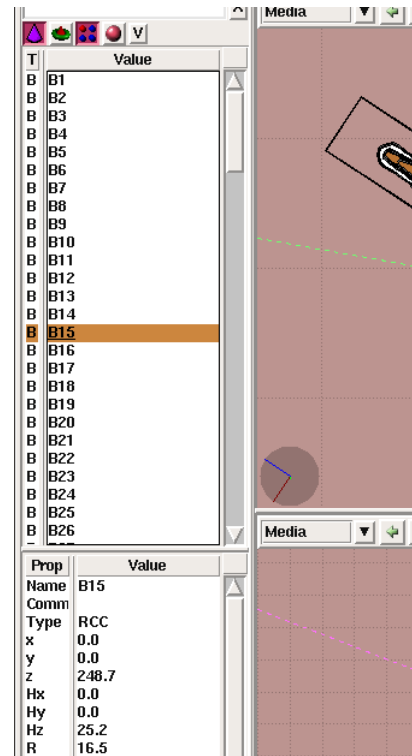


La partie gauche inventorie les « bodies », les « regions », les « lattice », les « materials » avec possibilité de filtrer telle ou telle catégorie.

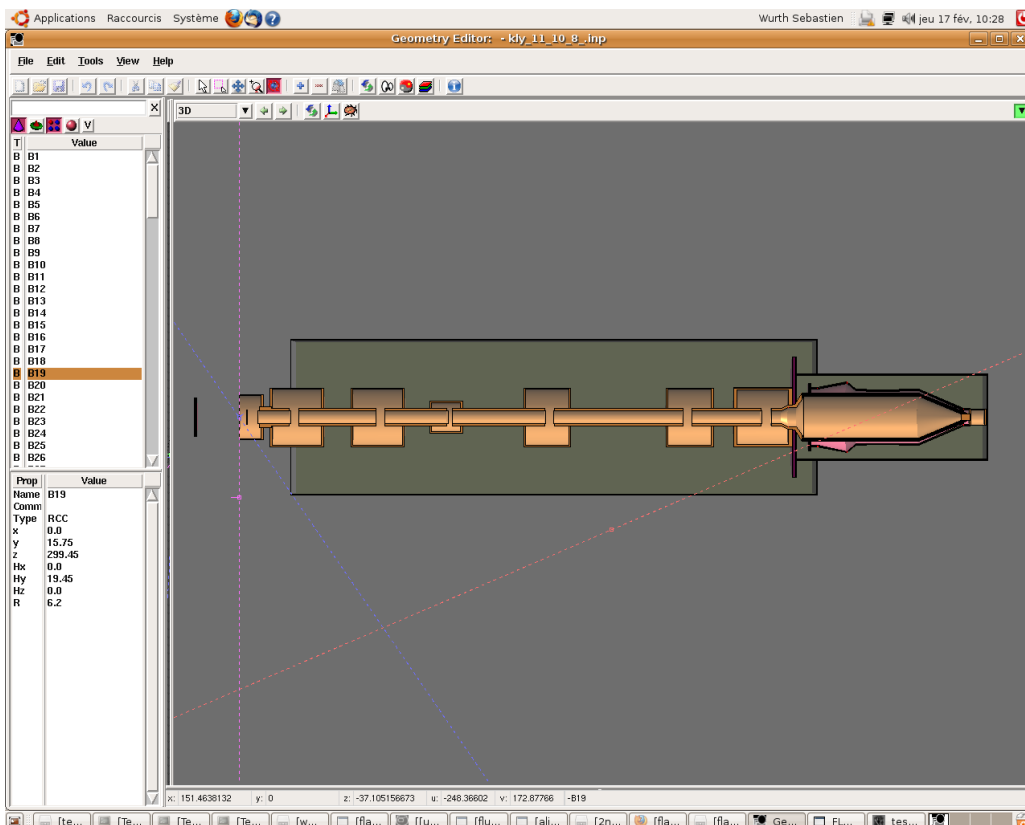
En haut, la liste, et en bas, les détails de l'item sélectionné : par exemple, le « body » 15.

Un clic droit sur un « body » ouvre une fenêtre comportant une série d'options d'édition.

L'item sélectionné apparaît dans la visualisation si les projections sélectionnées le permettent.



Il est possible de sélectionner diverses options de visualisations (frontières « boundaries », matériaux « media », régions, 3D).

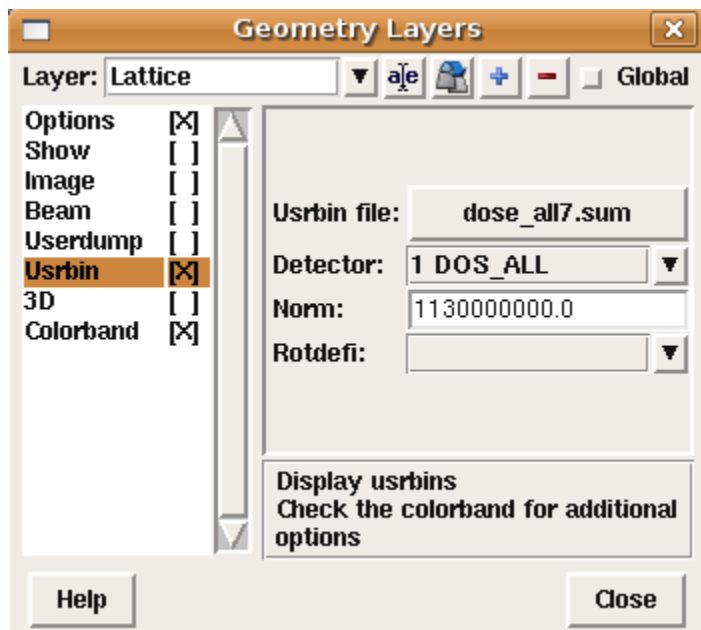




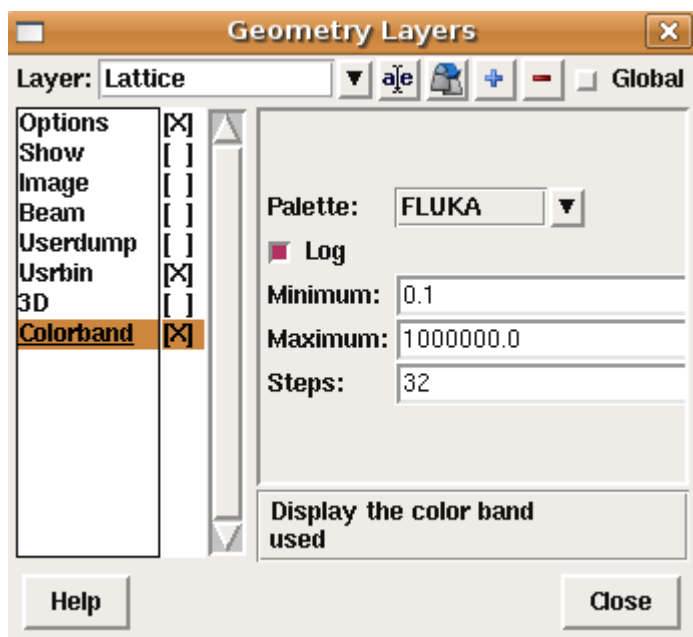
Par l'onglet « view / layers » ou l'icône ci-contre, il est possible de superposer une donnée sur la géométrie : une image, le résultat d'un détecteur (qui soit en rapport avec la géométrie, bien entendu).

Nous décrivons la démarche à suivre pour superposer un fichier de données issues d'une carte **USRBIN**.

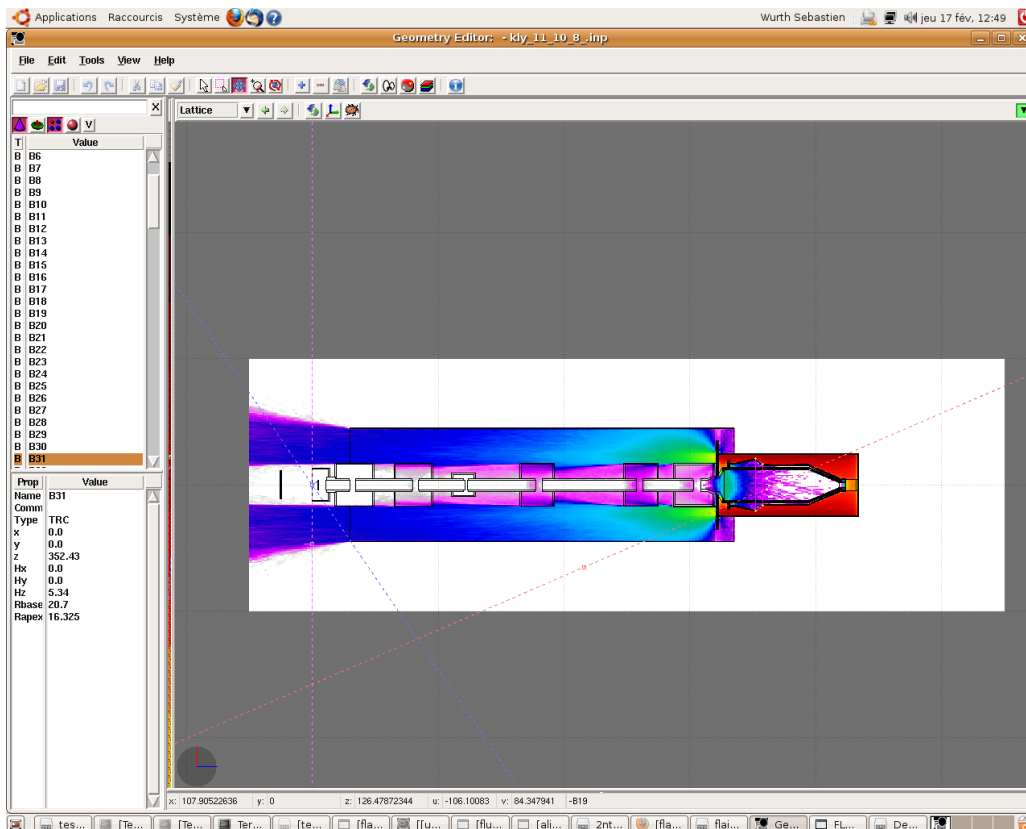
Cocher la case **Usrbin**, chercher un fichier à charger, sélectionner le détecteur, indiquer l'éventuel facteur de normalisation à appliquer.



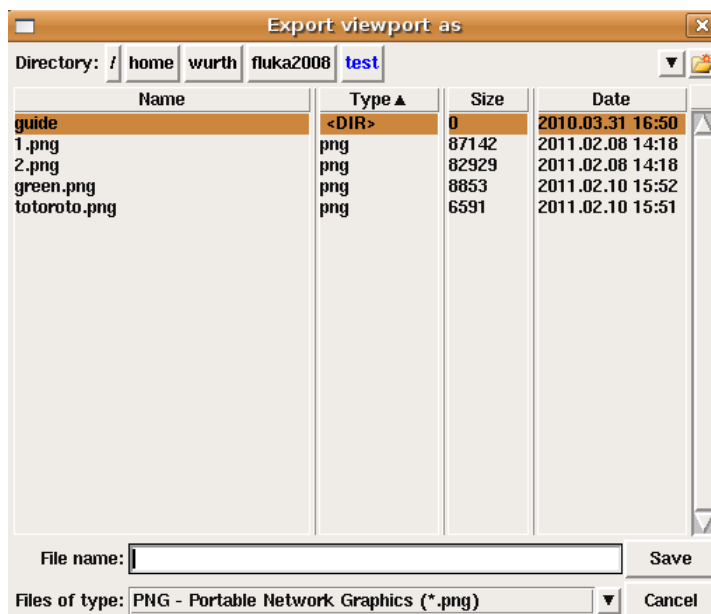
Il ne faut pas oublier de cocher la case **Colorband** et de la définir selon ses besoins, nature de la palette, échelle log ou non, les valeurs minimales et maximales et le nombre de couleurs de cette palette.



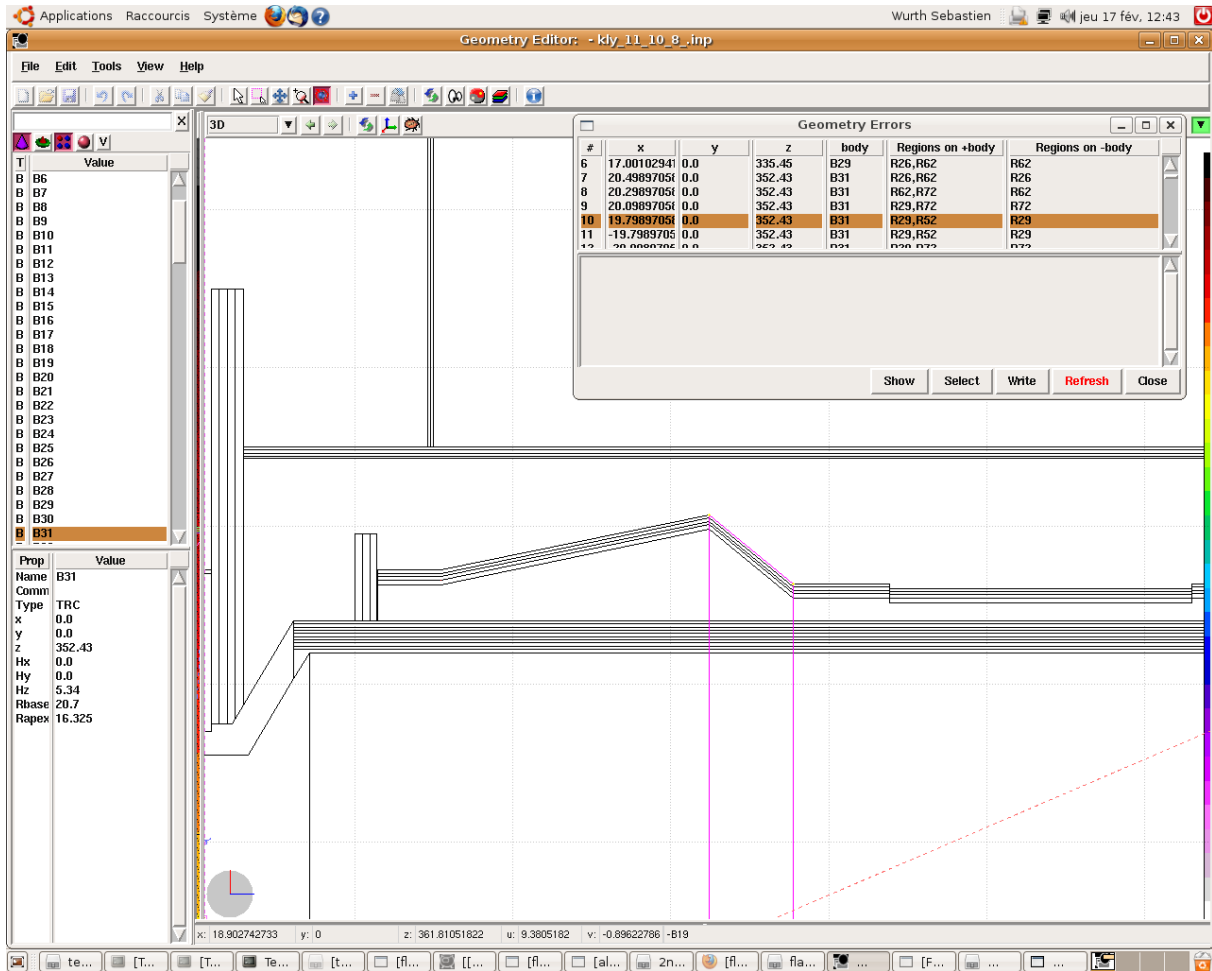
Après en temps de chargement et en sélectionnant la bonne visualisation les données du détecteur se superposent à la géométrie.



- Exporter un fichier image sous le format bitmap : « file/export ».



- Erreurs dans la géométrie : clic droit sur la fenêtre, « show errors » : une fenêtre s'ouvre avec la liste des erreurs, en cliquant sur une ligne, l'emplacement est indiqué sur la figure.



14. Techniques de réduction de la variance, biaisage

Afin de trouver le meilleur compromis entre un temps de calcul (noté t) raisonnable et des résultats statistiquement acceptables (variance, notée σ , pas trop importante), diverses techniques de biaisage peuvent être combinées. Les méthodes employées s'inspirent de celles utilisées dans l'étude décrite dans [3].

14.1 Biaisage d'importance

Ce type de biaisage combine deux techniques. Le « surface splitting » et la « russian roulette ».

La première réduit σ mais augmente t , alors que la seconde fait l'inverse.

L'utilisateur assigne une importance relative à chaque région en tenant compte de :

- L'atténuation de la fluence attendue par rapport aux autres régions.
- La probabilité de contribution au résultat final (calcul) des particules pénétrant dans la région.

Le « surface splitting »

Si une particule passe d'une région d'importance I_1 à une région d'importance plus grande, $I_2 > I_1$:

- La particule est remplacée par $n = I_2/I_1$ particules identiques dotées des mêmes caractéristiques ;
- Le poids statistique de chacune de ses « filles » est multiplié par I_1/I_2 .

La « russian roulette »

Si une particule passe d'une région d'importance I_1 à une région d'importance moindre, $I_2 < I_1$:

- La particule est soumise à un test aléatoire de survie :
 - Elle a une probabilité égale à I_2/I_1 de survivre et son poids statistique est augmenté d'un facteur I_1/I_2 ;
 - Elle a une probabilité $1-I_2/I_1$ d'être « tuée ».

Cette technique est intéressante pour maintenir une population de particules constante, en compensation de l'atténuation due à l'absorption ou la distance. Dans **FLUKA**, elle peut être définie par type de particule.

D'après [3], nous adoptons les facteurs suivants de compensation, en fonction du matériau, de son épaisseur et du type de particule considéré, photon et neutron ici.

Matériau	ρ (g/cm^3)	TVL		l (cm)	F_{comp}	
		(g/cm ²)			sans unité	
		photon	neutron		photon	neutron
beton	2.35	107	85	10	1.658	1.890
				20	2.749	3.572
Fe	7.877	85	305	5	2.906	1.346
				10	8.447	1.812
Pb	11.35	60	985	5	8.827	1.142
				10	1.066	3.201
Polyéthylène (PE)	0.95	171	9.4	5	1.066	3.201
				10	1.136	10.248

Avec : ρ : masse volumique du matériau, TVL = épaisseur de déci-atténuation, l = épaisseur de la région et F_{comp} , facteur de compensation directement lié au facteur de biaisage.

$$F_{comp} = \exp(\ln(10) * \frac{\rho * l}{TVL})$$

Concrètement, plus l'épaisseur et la distance par rapport à la source augmentent, plus l'importance de la région considérée augmente.

Carte **BIASING**

Cette carte permet de biaiser la multiplication des particules secondaires par région et de définir les paramètres de biaisage d'importance aux frontières des régions par région et par type de particule.

Signification des paramètres à renseigner :

➤ Si **WHAT(1)** > ou = à 0.0 :

- **WHAT(1)** spécifie la nature des particules à biaiser :
 - = 0.0 : toutes les particules
 - = 1.0 : hadrons et muons
 - = 2.0 : électrons, positrons et photons
 - = 3.0 : neutrons de basse énergie (< 20 MeV)
 - **WHAT(2)** = facteur de roulette russe (ou splitting) par lequel le nombre moyen de particules secondaires produites dans une collision est réduit (ou augmenté). N'a de sens que pour les hadrons, muons et photons issus de réactions photonucléaires. Valeur par défaut = 1.0.
 - **WHAT(3)** = importance de la région (l'intervalle des valeurs acceptées est de 0.0001 à 100000.0). Valeur par défaut = 1.0.
 - **WHAT(4)** = indice le plus bas des régions (ou nom correspondant) d'importance égale à **WHAT(3)** et/ou de facteur de multiplicité égal à **WHAT(2)**. ("De la région **WHAT(4)**..."). Valeur par défaut = 2.0.
 - **WHAT(5)** = indice supérieur (ou nom correspondant) des régions d'importance égale à **WHAT(3)** et/ou de facteur de multiplicité égal à **WHAT(2)**. ("... à la region **WHAT(5)**..."). Valeur par défaut = **WHAT(4)**.
 - **WHAT(6)** = pas dans l'assignement des indices ("... par pas de **WHAT(6)**"). Valeur par défaut = 1.0
 - **SDUM = PRINT** : les compteurs de biaisage d'importance sont imprimés (dans le *.out, ceci est très utile pour ajuster les importances et les fenêtres de poids statistique, cf. §14.3.).
- = **NOPRINT** : les compteurs ne sont pas imprimés (annule toutes les requêtes de **PRINT** précédemment définies).
- = **USER** : le biaisage d'importance est défini dans la routine **USIMBS** (cf. manuel).
- = **NOUSER** : remise aux valeurs par défaut (annule toutes les requêtes **USER** précédemment définies).
- = **RRPRONLY** : biaisage de multiplicité pour les particules primaires uniquement.
- = blank : ignoré

Valeurs par défaut = **NOPRINT**, **NOUSER**, biaisage de multiplicité pour toutes les générations (si la requête en a été faite).

Si **WHAT(1)** < 0.0 : cf. manuel.

Exemple :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
BIASING          2.0          0.0          10.0          7.0          11.0          2.0
```

Ici, nous avons assigné une importance 10 pour les photons, électrons et positrons dans les régions d'indice 7, 9 et 11.

14.2 Biaisage de la cascade électromagnétique

Le « leading particle biasing » (LPB) est utilisé pour éviter l'augmentation exponentielle du nombre de particules de la cascade électromagnétique.

Dans la technique de Monte-Carlo, 2 particules sont présentes au stade final lors des interactions électromagnétiques.

Si l'option LPB est activée, seule l'une de ces particules est retenue de façon aléatoire et son poids statistique est ajusté de façon à conserver le produit poids statistique * probabilité.

La plus énergétique des deux particules est retenue avec la plus grande probabilité (elle est considérée comme la plus à même de propager la cascade).

L'option LPB est très efficace pour réduire t , mais augmente σ en introduisant de larges fluctuations de poids statistiques. En effet, quelques particules de basse énergie ont un poids statistique important, alors que de nombreuses particules de plus haute énergie ont un poids statistique moins important.

Par conséquent, cette option doit presque toujours s'accompagner de celle décrite dans le paragraphe suivant, les fenêtres d'énergie et de poids statistiques afin de réduire à nouveau la variance.

Carte **EMF-BIAS**

Cette carte définit les paramètres de biaisage spécial pour électrons et photons incluant le LPB région par région et le libre parcours moyen matériau par matériau.

Pour **SDUM** = **LPBEMF** (valeur par défaut) :

- **WHAT(1)** > 0.0 : le Leading Particle Biasing (LPB) est activé.
La combinaison des effets du biaisage est définie par le "bit pattern" de **WHAT(1)**.

WHAT(1) se présente sous la forme suivante :

$$2^0 b_0 + 2^1 b_1 + 2^2 b_2 + 2^3 b_3 + 2^4 b_4 + 2^5 b_5 + 2^6 b_6 + 2^7 b_7 + 2^8 b_8 + 2^9 b_9$$

Où la signification des 10 bits est la suivante :

b0 = 1: LPB activated for bremsstrahlung and pair production (old default)

b1 = 1: LPB activated for bremsstrahlung

b2 = 1: LPB activated for pair production

b3 = 1: LPB activated for positron annihilation at rest

b4 = 1: LPB activated for Compton scattering

b5 = 1: LPB activated for Bhabha & Moller scattering

b6 = 1: LPB activated for photoelectric effect

b7 = 1: LPB activated for positron annihilation in flight

b8 = 1: not used

b9 = 1: not used

Notons que **WHAT(1)** = 1022 active LPB pour tous les effets physiques (des valeurs supérieures à 1022 sont considérées comme étant égales à 1022).

< 0.0 : le leading particle biasing est inactivé.

= 0.0 : ignoré

- **WHAT(2)** > 0.0 : seuil d'énergie en dessous duquel LPB est active pour les électrons et positrons (pour les électrons de tels seuils se réfèrent à l'énergie cinétique, pour les positrons, à l'énergie totale + l'énergie de masse)
 < 0.0 : efface toute définition antérieure et attribue une valeur infinie (i.e., LPB est active pour toutes les énergies).
 = 0.0 : ignoré.

Valeur par défaut : leading particle biasing est active à toutes les énergies pour les électrons et positrons.

- **WHAT(3)** > 0.0 : seuil d'énergie en dessous duquel LPB est active pour les photons.
 < 0.0 : efface toute définition antérieure et attribue une valeur infinie (i.e., LPB est active pour toutes les énergies).
 = 0.0 : ignoré.

Valeur par défaut : leading particle biasing est active à toutes les énergies pour les photons.

- **WHAT(4)** = indice inférieur (ou nom correspondant) des régions où le biaisage défini précédemment est activé (« De la région **WHAT(4)**... »)

Valeur par défaut = 2.0.

- **WHAT(5)** = indice supérieur (ou nom correspondant) des régions où le biaisage défini précédemment est activé (« ... à la région **WHAT(5)** ... »).

Valeur par défaut = **WHAT(4)**.

- **WHAT(6)** = pas dans l'assignation des indices (« ... par pas de **WHAT(6)** »).

Default = 1.0

- **SDUM** = LPBEMF (Leading Particle Biasing for EMF). Ceci est la valeur par défaut.

Pour des **SDUM** différents de **LPBEMF**, cf. manuel, carte **EMF-BIAS**.

Exemple 1 :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
EMF-BIAS      152.0      0.0      5.E-4      16.0      20.0      2.0LPBEMF
```

LPB est appliqué aux régions 16, 18 et 20 concernant la diffusion Compton en dessous de 0,5 MeV et l'annihilation des positrons en vol et au repos.

Le code 152 = 2^3 (annihilation au repos) + 2^4 (Compton) + 2^7 (annihilation en vol).

Exemple 2 :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
EMF-BIAS      1022.0      0.0      0.0      3.0      8.0
```

LPB est appliqué aux régions 3, 4, 5, 6, 7 et 8 pour tous les électrons et photons, à toutes les interactions et à toutes les énergies.

14.3 Biaisage par fenêtres d'énergie et de poids statistique

Il s'agit d'une combinaison de « surface splitting » et de « russian roulette », mais basée sur la valeur absolue du poids statistique de chaque particule.

L'utilisateur définit un intervalle de poids, en fonction de l'énergie, de la région, de la particule.

Les particules ayant un poids statistique plus important que la limite supérieure subissent le « surface splitting », celles ayant un poids moins important que la limite inférieure subissent la « russian roulette », tuées ou remises dans la « fenêtre ».

Il s'agit de l'outil le plus puissant, mais également le plus compliqué à mettre en œuvre.

Cette technique est pourtant indispensable lorsque d'autres techniques de biaisage génèrent de larges fluctuations de poids statistiques dans une région de l'espace des phases.

- « Tuer » une particule de poids statistique faible permet de décroître t , mais n'a que très peu d'impact sur le résultat du calcul (et donc sur σ).
- Le « splitting » d'une particule ayant un poids statistique très important :
 - Augmente t ;
 - Mais réduit également σ , en évitant les larges fluctuations pour la contribution au calcul.
- Le but est de réduire le produit $\sigma^2 \cdot t$.

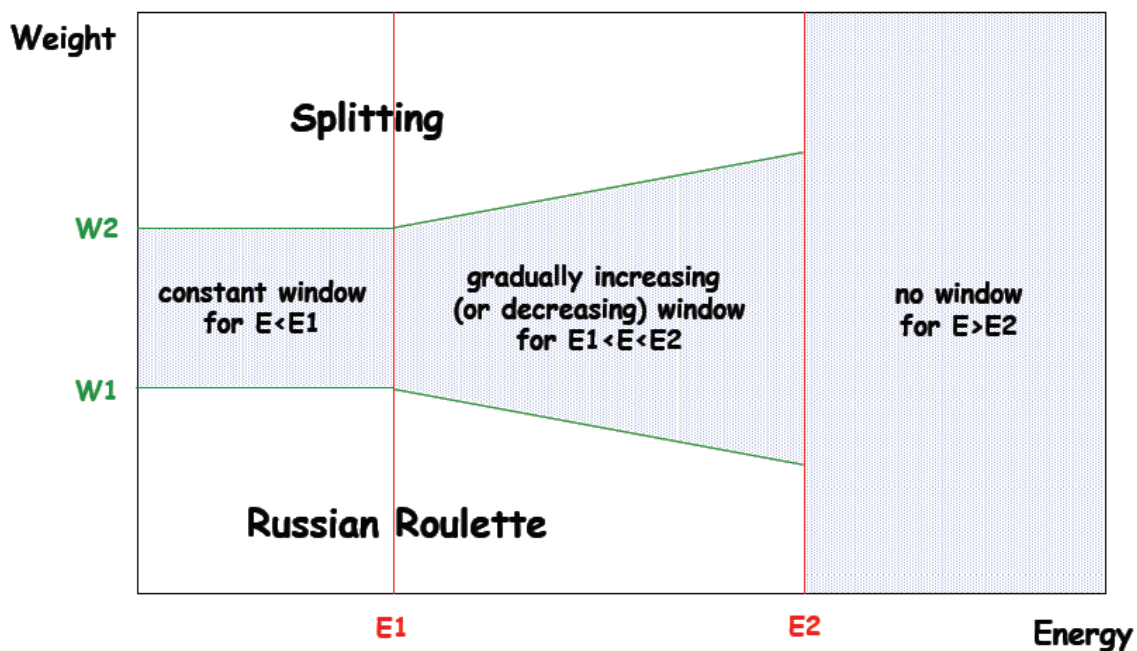
Une fenêtre trop large est inefficace, mais elle ne doit pas être trop étroite non plus.

Un facteur typique entre les deux limites de la fenêtre est 10.

(Il est possible de ne garder qu'une technique : RR seulement, avec la limite supérieure infinie ou SS seulement avec une limite inférieure égale à 0).

En pratique, ce type de biaisage est activé par deux options dans **FLUKA**. La première définit les valeurs limites des énergies ainsi que le facteur d'amplification de la fenêtre (la taille de la fenêtre au seuil d'énergie le plus haut) par type de particule. La seconde définit les valeurs limites des poids statistiques par région.

Schématiquement, nous avons ceci avec $E1$ et $E2$ les limites inférieures et supérieures en énergie et $w1$ et $w2$ les limites inférieures et supérieures en poids statistique.



Concrètement, il est conseillé (et nous avons procédé de la sorte) :

- D'effectuer une simulation de test sans ces options, mais avec toutes les autres d'intérêt pour le problème étudié ;
- D'analyser les compteurs qui permettent de connaître le nombre de « surface splitting » et de « russian roulette » ainsi que les poids statistiques correspondants, par région, et par type de particule ;

- D'ajuster les fenêtres de poids dans un intervalle dont les limites sont séparées d'un facteur 10 autour de la valeur moyenne du poids de chaque type de particule dans chaque région considérée.

Cartes **WW-THRES** et **WW-FACTO**

14.3.1 Carte **WW-THRES**

Cette carte définit les limites en énergie pour la fenêtre de poids Russian Roulette/splitting et applique à cette fenêtre les facteurs de modification dépendant du type de particule définis par **WW-FACTO**. (Nous mentionnons également la carte **WW-PROFIL** destinée aux neutrons). Cette carte n'a de sens que si au moins une carte **WW-FACTO** est également définie.

- **WHAT(1)** > 0.0 : seuil supérieur d'énergie cinétique en GeV pour Russian Roulette (RR)/Splitting pour la fenêtre de poids. Pour les neutrons de basse énergie, il faut indiquer le plus petit groupe correspondant.
 = 0.0 : ignoré
 < 0.0 : tout seuil précédemment défini est annulé.
- **WHAT(2)** >= 0.0 et < **WHAT(1)** : seuil inférieur d'énergie cinétique en GeV pour Russian Roulette (RR)/Splitting pour la fenêtre de poids. Pour les neutrons de basse énergie, il faut indiquer le plus grand groupe correspondant.
 < 0.0 or > **WHAT(1)** : **WHAT(2)** est égal à **WHAT(1)**
- **WHAT(3)** > 0.0 : facteur d'amplification utilisé pour définir la largeur de la fenêtre de poids à l'énergie la plus élevée représentée par **WHAT(1)**.

Cette valeur est obtenue en multipliant la partie supérieure de la fenêtre (domaine du splitting) par **WHAT(3)** et en divisant la partie inférieure de la fenêtre (domaine de la roulette russe) par **WHAT(3)**.

< 0.0 : **WHAT(3)** est utilisé comme facteur de multiplication pour les niveaux inférieurs et supérieurs de chaque région pour les particules sélectionnées par **WHAT(4-6)**. Autrement dit, pour ces particules, les parties inférieure et supérieure sont multipliées par **WHAT(3)**.

Valeur par défaut = 10.0 (facteur d'amplification pour splitting/RR au seuil haut).

Le facteur de multiplication dépendant du type de particule est fixé à 1.0 par défaut.

- **WHAT(4)** = indice inférieur des particules (ou nom correspondant de la particule) auquel les limites indiquées précédemment s'appliquent. Notons ici que de façon exceptionnelle, le code 40.0 indique des neutrons de basse énergie.

Le code 8.0 indique les neutrons d'énergie > 20 MeV. (« De la particule de code **WHAT(4)**... »). Valeur par défaut : 1.0.

- **WHAT(5)** = indice inférieur des particules (ou nom correspondant de la particule) auquel les limites indiquées précédemment s'appliquent. (« ... à la particule de code **WHAT(5)** ... »). Valeur par défaut = **WHAT(4)** si **WHAT(4)** > 0.0, toutes les particules sinon.
- **WHAT(6)** = pas dans l'assignation des codes de particules. (« ... par pas de **WHAT(6)** »).
- **SDUM** = **PRIMARY** : la fenêtre de poids s'applique également aux particules primaries.
 = **NONPRIMARY** : la fenêtre de poids ne s'applique pas aux particules primaries.

Valeur par défaut = **PRIMARY**

Par défaut : (les options **WW-THRES** ou **WW-FACTO** non définies): pas de définition de fenêtre de poids.

Exemple 1 :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
WW-THRESh      2.0      0.05      2.4      3.0      7.0      0.0
```

Les limites de la fenêtre de poids pour les particules de codes 3 à 7 (electrons, positrons et photons, les neutrinos étant désactivés) auxquels on a assigné des facteurs de poids statistique via la carte **WW-FACTO**r s'appliquent à partir de 50 MeV. Au-delà de cette énergie, la largeur de la fenêtre est progressivement augmentée jusqu'à 2 GeV. À 2 GeV, la valeur supérieure de poids est augmentée d'un facteur 2.4 et la valeur inférieure diminuée d'un facteur 2.4.

Exemple 2 :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
WW-THRES      50.0E-3      1.0E-4      2.0      7.0      7.0
```

Les facteurs de la fenêtre de poids statistique s'appliquent à partir de 100 keV pour les photons. Au-delà de cette énergie, la largeur de la fenêtre est progressivement augmentée jusqu'à 50 MeV. À cette énergie-là, la valeur supérieure de poids est augmentée d'un facteur 2.0 et la valeur inférieure diminuée d'un facteur 2.0.

14.3.2 Carte WW-FACTOr

Cette carte définit les fenêtres de poids dans les régions sélectionnées.

Attention : l'option **WW-FACTO**r seule n'est pas suffisante pour définir correctement une fenêtre de poids statistique. Une ou plusieurs cartes **WW-THRESh** sont également nécessaires Pour activer la fenêtre.

- **WHAT(1)** ≥ 0.0 : paramètre de roulette russe (RR) (poids limite inférieur de la fenêtre au seuil bas défini dans **WW-THRESh**).
 < 0.0 : remet à -1.0 (pas de RR) une valeur positive précédemment

définie dans une carte **WW-FACTO**r précédente.

Cette valeur peut être modifiée par **WHAT(4)** de l'option **WW-THRESh** ou par **WHAT(2)** de l'option **WW-PROFI**e. Valeur par défaut = -1.0 (pas de RR)

- **WHAT(2)** $> 1.7 \cdot \text{WHAT(1)}$: paramètre de splitting (poids limite supérieur de la fenêtre au seuil bas défini dans **WW-THRESh**).
 $= 0.0$: ignoré
 $< 1.7 \cdot \text{WHAT(1)}$: remet à 1 (pas de splitting) une valeur précédemment

définie dans une carte **WW-FACTO**r précédente.

Cette valeur peut être modifiée par **WHAT(4)** de l'option **WW-THRESh** ou par **WHAT(2)** de l'option **WW-PROFI**e. Valeur par défaut = 1.0 (pas de splitting).

- **WHAT(3)** > 0.0 : facteur multiplicatif à appliquer aux deux seuils d'énergie pour RR/splitting (définis par l'option **WW-THRESh**) dans la région d'intérêt.
 $= 0.0$: ignored
 < 0.0 : remet à 1.0 (les seuils ne sont pas modifiés) une valeur

précédemment définie dans une carte **WW-FACTO**r antérieure.

Par défaut = 1.0 (les seuils RR/splitting ne sont pas modifiés).

- **WHAT(4)** = indice inférieur des régions (ou nom correspondant) dans lesquelles les paramètres définis précédemment s'appliquent. (« De la région **WHAT(4)**... »)

Valeur par défaut = 2.0.

- **WHAT(5)** = indice supérieur des régions (ou nom correspondant) dans lesquelles les paramètres définis précédemment s'appliquent. (« ... à la région **WHAT(5)** ... »)

Valeur par défaut = **WHAT(4)**.

- **WHAT(6)** = pas dans l'assignation des indices. (« ... par pas de **WHAT(6)** ... »).

Valeur par défaut = 1.0.

SDUM = un nombre de 1.0 à 5.0 dans n'importe quelle position, indiquant le profil de la fenêtre de biaisage à appliquer pour les neutrons de basse énergie pour les régions sélectionnées (cf. **WW-PROFI**e dans le manuel). Exceptionnellement, ici **SDUM** doit être un nombre, en format libre plutôt qu'une chaîne de caractères.

= laissé vide, zéro or valeur non-numérique : ignoré.

< 0.0 : remet à 1.0 une valeur précédemment définie.

Valeur par défaut (si aucune carte **WW-PROFI**e n'est présente) : profil numéro 1.

Valeur par défaut : (option **WW-FACTO**r ou **WW-THRES**h non présentes) : aucune fenêtre de poids n'est définie.

Exemple 1 :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
WW-FACTOR      76.0    1200.0    1.0    5.0    6.0    0.0
```

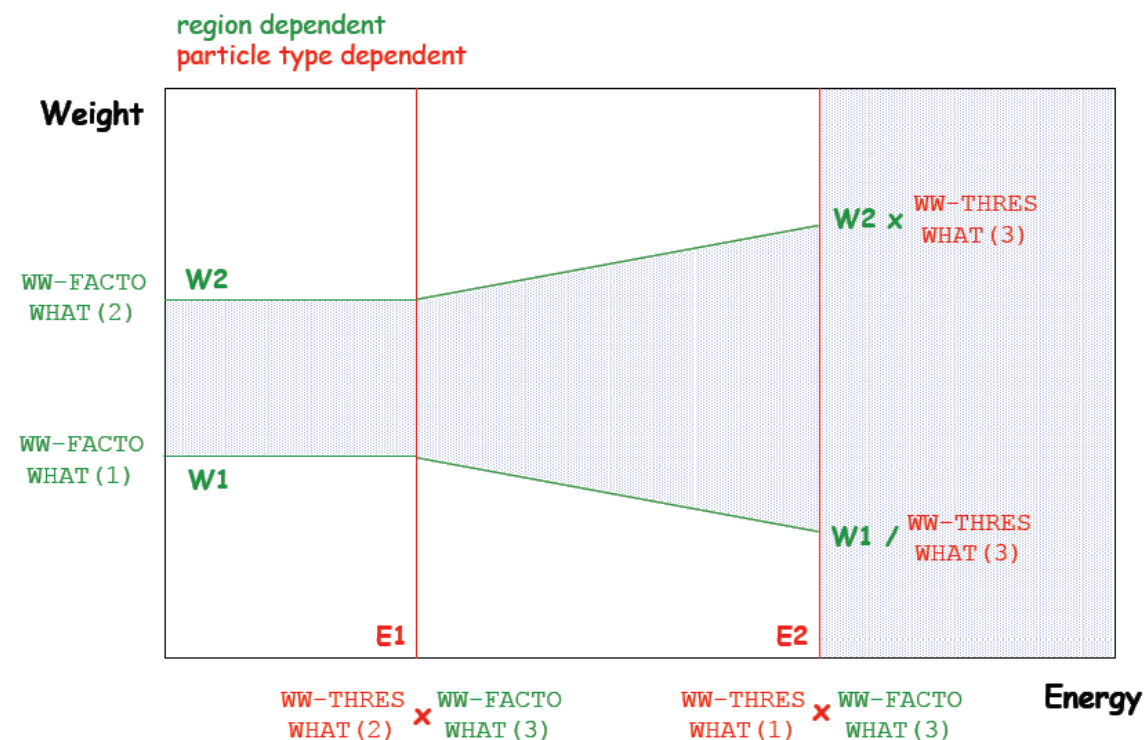
Dans les régions 5 et 6, le domaine de poids statistique auquel s'applique la fenêtre est 76.0 à 1200. Il n'y a pas de modification des deux seuils d'énergie.

Exemple 2 :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
WW-FACTOR      13.0    120.0    1.5    27.0    31.0    2.0    3.
```

Dans les régions 27, 29 et 31, la fenêtre de poids est définie de 13 à 120. Les deux seuils d'énergie définis par **WW-THRES**h sont multipliés par 1,5. Le profil pour les neutrons de basse énergie à appliquer est le n° 3.

Schématiquement, les deux cartes combinées donnent la figure suivante :



14.4 Biaisage de la longueur d'interaction

Réduction du libre parcours moyen.

Ici, il s'agit de réduire artificiellement la longueur d'interaction des photons, afin d'obtenir davantage d'interactions pour un même nombre de particules incidentes.

Carte **LAM-BIAS**

Cette carte sert à biaiser la durée de décroissance de particules instables, la longueur des interactions inélastiques des hadrons, photons et muons et la direction de la décroissance des particules secondaires.

Cette carte permet de définir une grande variété de paramètres différents, nous nous contentons de décrire uniquement le cas qui nous intéresse plus particulièrement. Dans ce cas, **SDUM** = blank (laissé vide). Pour tous les autres cas, se référer au manuel.

- **WHAT(1)** :

-1.0 < **WHAT(1)** < 0.0 : la durée de vie moyenne d'une particule dans son état de repos est réduite d'un facteur = $|\mathbf{WHAT(1)}|$. Au point de décroissance ainsi défini, la roulette russe (i.e. choix aléatoire) décide si la particule survit ou non après création des produits de décroissance. Ceux-ci sont créés dans tous les cas et leur poids statistique est ajusté en tenant compte du rapport entre les probabilités de survie biaisée et physique.

0.0 < **WHAT(1)** < 1.0 : la durée de vie moyenne d'une particule dans son état de repos est réduite d'un facteur = $|\mathbf{WHAT(1)}|$. Au point de décroissance ainsi défini, la particule survit toujours avec un poids réduit. Ses filles héritent du même poids.

$|\mathbf{WHAT(1)}| > 1.0$: une possible valeur définie précédemment est remise à la valeur par défaut (pas de biaisage).

WHAT(1) = 0.0 : ignore.

- **WHAT(2)** :

WHAT(2) : facteur de biaisage pour les interactions inélastiques hadroniques.

-1.0 < **WHAT(2)** < 0.0 : la longueur des interactions inélastiques pour les hadrons est réduite d'un facteur $|\mathbf{WHAT(2)}|$. Au point de décroissance ainsi défini, la roulette russe (i.e. choix aléatoire) décide si la particule survit ou non après création des produits de décroissance. Ceux-ci sont créés dans tous les cas et leur poids statistique est ajusté en tenant compte du rapport entre les probabilités de survie biaisée et physique.

0.0 < **WHAT(2)** < 1.0 : la longueur des interactions inélastiques pour les hadrons est réduite d'un facteur $|\mathbf{WHAT(2)}|$. Au point de décroissance ainsi défini, la particule survit toujours avec un poids réduit. Les particules secondaires sont créées dans tous les cas et leur poids statistique est ajusté en tenant compte du rapport entre les probabilités de survie biaisée et physique.

WHAT(2) = 0.0 : ignoré.

$|\mathbf{WHAT(2)}| \geq 1.0$: une possible valeur définie précédemment est remise à la valeur par défaut (pas de biaisage).

- **WHAT(3)** :

WHAT(3) > 2.0 : numéro ou nom du matériau auquel le facteur défini précédemment doit s'appliquer.

< 0.0 : remet la valeur par défaut

= 0.0 : ignore si une valeur a déjà été assignée à un matériau spécifique ; sinon :

tous les matériaux (valeur par défaut).

0.0 < **WHAT(3)** <= 2.0 : tous les matériaux.

- **WHAT(4)** = indice inférieur des codes de particules (ou nom correspondant) pour lesquelles ces options sont appliquées. (« De la particule de code **WHAT(4)**... »)
 Valeur par défaut = 1.0
- **WHAT(5)** = indice supérieur des codes de particules (ou nom correspondant) pour lesquelles ces options sont appliqués. (« ... à la particule de code **WHAT(5)**... »)
 Valeur par défaut = **WHAT(4)** si **WHAT(4)** > 0.0, 46.0 sinon.
- **WHAT(6)** = pas dans l'assignation des codes de particules. (« ... par pas de **WHAT(6)** »).
 Valeur par défaut = 1.0.

Exemple :

```
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7...+...8
LAM-BIAS      0.      0.02      1.0      7.
```

La longueur des interactions inélastiques des photons pour tous les matériaux est réduite d'un facteur 50.

14.5 Combinaison de divers types de biaisage dans une application concrète

L'analyse complète figure dans le document intitulé *VALIDATION DES PAROIS DE CONFINEMENT DES INSTALLATIONS D'ACCÉLÉRATEURS DU BÂTIMENT 109N DE L'INSTITUT DE PHYSIQUE NUCLÉAIRE D'ORSAY À L'AIDE DU CODE FLUKA* de référence *IPNO|SCR|FLUKA|IPN_Orsay_bâtiment_109N_validation_blindages_v1|SW_23-11-2009*

15. Utilisation de flupix : un cd image de flair et fluka

FLUPIX = FLUKa in KnopPIX by Vasilis.Vlachoudis@cern.ch 2008

FLUPIX est un live CD Linux sur lesquels sont préinstallés **FLUKA**, **flair** et tous les outils nécessaires pour effectuer des simulations.

Afin d'économiser de l'espace disque, **FLUPIX** ne fonctionne qu'à travers une « machine virtuelle » libre et « open source », VirtualBox, développée par Sun (www.virtualbox.org).

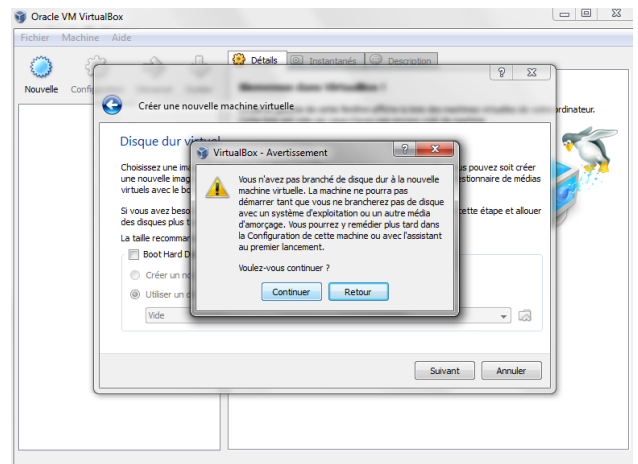
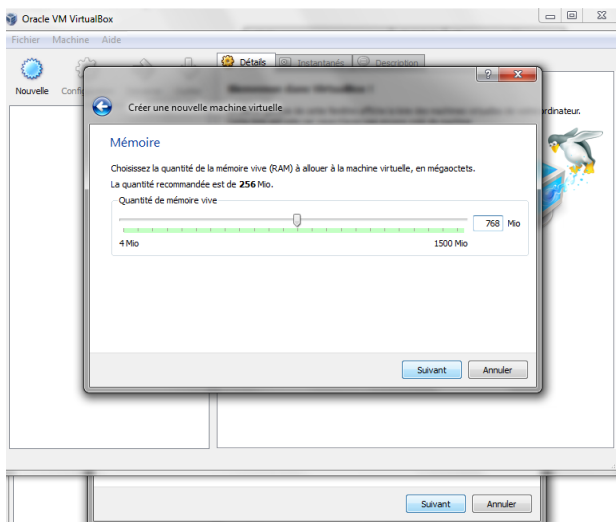
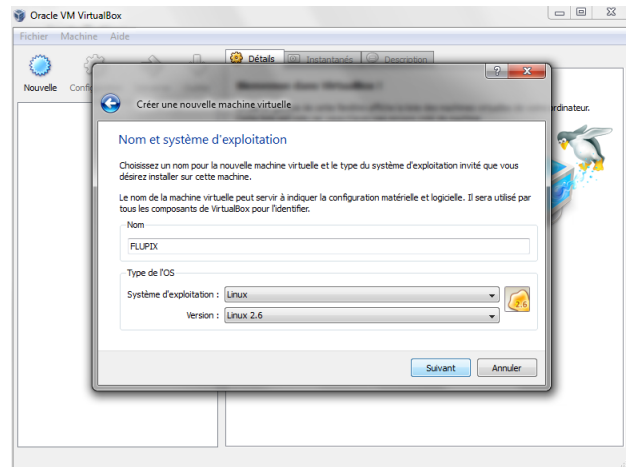
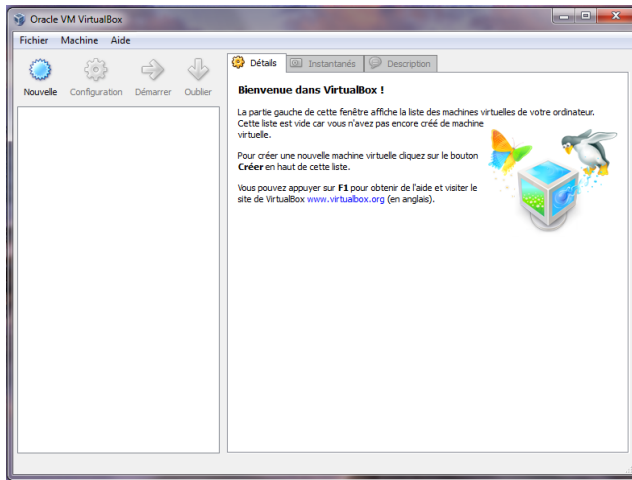
Conditions requises :

OS	RAM	Espace disque
Windows XP	1 Go	> 5 Go
Windows Vista / 7	2 Go	> 5 Go
Linux	1 Go	> 5 Go

15.1. Installation

- Télécharger et installer à partir de VirtualBox les fichiers adéquats pour votre plateforme (site www.virtualbox.org) ~ 70 Mo.
- Télécharger flupix-XXX.iso et les fichiers auxiliaires flupix-vdi.zip à partir de la section de téléchargement du site de fluka (www.fluka.org) respectivement ~ 330 Mo et < 1 Mo (il faut s'inscrire au préalable, gratuitement bien sûr).

- Démarrer **VirtualBox** et créer une machine virtuelle utilisant les paramètres suivants :



Nom : **FLUPIX**

Type de l'OS : **Linux 2.6**

Mémoire vive : **512 Mo** (minimum, 768 Mo recommandé)

Disque dur de démarrage : <pas de disque dur> (cliquer "continuer" au message d'avertissement)

15.2. Configuration

15.2.1 Configuration générale

Fichier/Gestionnaire des medias (CTRL+D) : onglet images cd/dvd : chercher flupix-XXX.iso et sélectionner.

Réseau : configuration/réseau, cocher activer la carte réseau :

Les types PCnet-FASTIII ou INTEL PRO 1000/MT sont compatibles.

Mode d'accès réseau : NAT

15.2.2 Configurations optionnelles : son, usb et disque dur image

De façon optionnelle, on peut changer :

L'audio, en sélectionnant configuration/son : activer le son et en sélectionnant le pilote audio hôte adéquat : Windows DirectSound ou Audio Controller: ICH AC97

L'USB : c'est un peu délicat, les instructions suivantes sont valables pour windows, concernant les hôtes linux, il convient de se référer au manuel de VirtualBox.

- Cocher « activer le contrôleur USB »
- Cocher « activer le contrôleur USB 2.0 (EHCI) »
- Insérer votre clé USB
- Cliquer sur ajouter un nouveau filtre depuis un périphérique (ALT+ins)
- Ôter la clé USB

Pour utiliser la clé USB, il faut l'insérer uniquement au démarrage de la machine virtuelle. Sinon, l'hôte, en l'occurrence Windows la reconnaîtra et la machine virtuelle ne pourra pas y accéder.

15.2.3 Configuration des disques durs image

On peut créer un disque pour échanger et sauver les données ou utiliser ceux existants à partir des fichiers flupix-vdi.zip. Localiser le répertoire d'installation .VirtualBox. Considérons que <user> est le nom de l'utilisateur. Décompresser le contenu de flupix-vdi.zip dans l'emplacement suivant selon le système.

- XP : C:\Documents And Settings\\.VirtualBox\HardDisks
 - Vista / 7 : C:\Users\\.VirtualBox\HardDisks
 - Mac : /Users/<user>/Library/VirtualBox/HardDisks
 - Linux : /home/<user>/.VirtualBox/HardDisks
- Dans le gestionnaire des medias, onglet disque durs, ajouter les deux disques « flupix.vdi » et « swap.vdi ».
- Puis dans paramètres/stockage, cliquer sur l'icône « ajouter un disque dur » ou « ins », a priori, les disques apparaîtront directement si l'étape précédente a été effectuée.

À présent « flupix.vdi » apparaît en tant que maître primaire, faire de même avec « swap.vdi » qui apparaîtra en esclave primaire.

15.3. Exécuter FLUPIX

Sélectionner l'icône de **FLUPIX** dans la liste de machines virtuelles dans VirtualBox et cliquer sur « démarrer ». Ensuite, vous pouvez démarrer **flair** et l'utiliser comme sous l'environnement linux.

FLUKA est installé dans */usr/local/fluka*.

Le script rfluka se trouve dans */usr/local/fluka/flutil*

Si d'aventure vous n'utilisez pas **flair**, mais la méthode par exécution dans un terminal :
Placez-vous dans le répertoire de votre fichier toto.inp, mettons qu'il s'appelle */home/flupix*
et que vous utilisez l'exécutable par défaut flukahp.

[home/flupix] > /usr/local/fluka/flutil/rfluka -N0 -M5 toto (enter)

15.4. Trucs et astuces

15.4.1 Configuration clavier et souris

Par défaut, le clavier est configuré en QWERTY, pour passer en AZERTY, dans un terminal :

> sudo setxkbmap fr (enter) ou > sudo loadkeys fr (enter)

La touche pour activer la souris dans VirtualBox est la touche Ctrl de droite. **FLUPIX** a les additions clients de VirtualBox, ce qui signifie que la souris de Windows peut être utilisée par **FLUPIX**.

15.4.2 Transfert de données

Le transfert de données entre l'hôte et l'invité peut être effectué facilement en créant un dossier partagé et en l'installant à partir de linux. Il existe un utilitaire au chemin suivant qui permet cela : */usr/local/bin/vboxmount*

Il va créer un répertoire dans VirtualBox baptisé "vbox" (par exemple) et pointant vers :
C:\Documents And Settings\user (pour windows XP) ou C:\Users\<user> (windows vista / 7), cf. 15.2.3 pour toutes les configurations.

Dans un terminal de type **FLUPIX** :

> cd

> vboxmount vbox windows

L'utilitaire vboxmount va créer un répertoire dans $\${HOME}/windows$ et va le lier au répertoire partagé "vbox". Si aucune option particulière n'est donnée, le répertoire est nommé « vbox ».

Note : selon la nature de la machine hôte, cette opération pourra poser ou non un problème.

L'opération peut aboutir à un message d'erreur de type « protocol error » montrant que l'opération a échoué parce qu'il manque un module à votre machine virtuelle.

Il faut alors :

1) Préparer le système à une installation supplémentaire, la commande peut être la suivante (consulter l'aide de VirtualBox sinon) :

> sudo yum install binutils gcc make patch libgomp glibc-headers glibc-devel kernel-headers kernel-devel

2) Installer les additions invité de VirtualBox : dans la fenêtre de **FLUPIX**, périphérique / installer les additions invité. Ceci fait apparaître un nouveau cd virtuel qu'il faut lancer pour que l'installation puisse se faire.



- ⇒ Toutes les procédures sont détaillées dans l'aide de VirtualBox et sur divers sites (notamment celui de VirtualBox).
- ⇒ Un copier/coller du message d'erreur dans un moteur de recherche peut également fournir quelques liens utiles, en français notamment.

Pour migrer d'une version antérieure de **FLUPIX**, ouvrir un terminal et taper la commande :

> **migrate.sh**

Copyright Vasilis.Vlachoudis@cern.ch for the European Organization for Nuclear Research (CERN)

Conclusion

Conformément à l'objectif défini en introduction, nous nous sommes attachés à initier à l'utilisation du code de calcul **FLUKA** pour des simulations simples (et un petit peu moins). En s'appuyant sur quelques cas, de plus en plus nombreux au fil de l'incrémement du numéro de version, nous avons tenté de constituer une somme de petites informations. Nous avons décrit les premiers pas à effectuer pour installer et exécuter le code, et avons essayé de répertorier le plus d'informations possibles destinées à simplifier la vie de l'utilisateur débutant en cas de message d'erreur inédit et dont l'explication ne semble figurer nulle part.

Nous avons découpé le document en séparant les différentes étapes de la découverte du code, les différentes tâches à effectuer avant d'obtenir des résultats acceptables.

D'inévitables modifications de ce document sont appelées à survenir avec les futures versions de **FLUKA**, le retour d'expérience accru, les connaissances du rédacteur, des utilisateurs.

Sébastien WURTH, SPR IPN d'Orsay.

Références pour la rédaction de ce document

[1] Code de calcul :

- Ferrari, P.R. Sala, A. Fasso`, and J. Ranft, "FLUKA: a multi-particle transport code", CERN 2005-10 (2005), INFN/TC_05/11, SLAC-R-773
- G. Battistoni, S. Muraro, P.R. Sala, F. Cerutti, A. Ferrari, S. Roesler, A. Fasso`, J. Ranft, "The FLUKA code: Description and benchmarking", Proceedings of the Hadronic Shower Simulation Workshop 2006, Fermilab 6--8 September 2006, M. Albrow, R. Raja eds., AIP Conference Proceeding 896, 31-49, (2007)

[2] Routine deq99f.c :

- Stefan Roesler and Graham R. Stevenson, "deq99.f - A FLUKA user-routine converting fluence into effective dose and ambient dose equivalent", Technical Note CERN-SC-2006-070-RP-TN, EDMS No. 809389 (2006).
- M. Pelliccioni, "Overview of fluence-to-effective dose and fluence-to-ambient dose equivalent conversion coefficients for high energy radiation calculated using the FLUKA code", Radiation Protection Dosimetry 88 (2000) 279-297.

[3] Thèse présentée par Maher Cheikh MHAMED, Production de noyaux exotiques par photofission, Le projet ALTO : Premiers résultats, réf. IPNO-T-06-05.

[4] Site web : <http://www.fluka.org> (plus particulièrement <http://www.fluka.org/course/WebCourse/index.html>)

[5] Andreas Morsch, ALIFE : a geometry editor and Parser for FLUKA

[6] Pour le chapitre I., document technique de Alfredo Ferrari et Joachim Vollaire (CERN), « Présentation générale du code de calcul FLUKA », 03-03-2006.

[7] SimpleGeo : Theis C., Buchegger K.H., Brugger M., Forkel-Wirth D., Roesler S., Vincke H., "Interactive three dimensional visualization and creation of geometries for Monte Carlo calculations", Nuclear Instruments and Methods in Physics Research A 562, pp. 827-829 (2006).

Annexes

A. Quelques messages d'erreurs typiques

Cf. le site web de FLUKA, www.fluka.org, plus précisément le lien suivant : <http://www.fluka.org/course/WebCourse/crash/>

- Si le programme stoppe immédiatement, avec un message « **segmentation fault** », en ne produisant aucun fichier, il peut s'agir d'un problème de compilation. Vous n'avez peut-être pas la bonne version de **FLUKA** pour la machine que vous utilisez. Il faut alors télécharger la bonne version, la compiler et la lier, cf. **2. du présent document**.
- En admettant que l'on utilise **mu.inp** et que l'on obtienne le message suivant après le premier cycle :

Saving output and random number seed
No ranmu002 generated!
No mu001.err generated!

Il s'est produit une erreur : vérifiez vos fichiers **.log* et **.out*

Quelque soit le nombre de cycles demandés, ce message apparaîtra chaque fois, il se produit au premier cycle, il se reproduit donc...

- Vérifier les **.log* et **.out* est la première chose à faire et permet le plus souvent de trouver l'origine de l'erreur : ***** Unknown Input card !!!!!!!! *****

Une erreur est intervenue à la lecture d'une carte figurant dans le **.inp*

Typiquement, la carte incriminée sera identifiée grâce aux **.out* et **.log*

Le nom de l'option a été mal écrit et n'est donc pas reconnu, ou est mal placé.

Attention également aux caractères invisibles pouvant s'insérer par mail ou avec des éditeurs de texte de type windows (notepad, wordpad) : **il est très fortement conseillé d'utiliser des éditeurs de texte UNIX (nedit, emacs, VI, ...).**

- Vous avez le message suivant :

Abort called from CDWPRS reason LNUMRC = .TRUE. Run stopped ! STOP LNUMRC = .TRUE.

Dans les versions les plus récentes de **FLUKA**, l'exécution est interrompue si un nombre est écrit dans les champs réservés aux « **WHAT** » sans le point décimal, « . ».

Le même message peut apparaître si les nombres ne sont pas correctement alignés dans le format d'entrée « fixé ».

- Une erreur lors de la lecture de l'**Unit 14** renvoie au fichier **nuclear.bin**, ce fichier est peut-être corrompu ou manquant, vérifiez vos fichiers binaires dans **\$FLUPRO**.
- Une erreur lors de la lecture de l'**Unit 1** révèle un problème dans la génération des nombres aléatoires. Cela peut se produire si vous commencez avec un numéro de cycle n , avec $n > 0$: vérifiez l'existence de **ran***n** dans votre répertoire de lancement (*** représente le nom de votre fichier **.inp*).
- Une erreur dans la lecture de l'**Unit 12** est liée au fichier **PEMF**.

Vous avez oublié de donner le fichier **PEMF** à **FLUKA**, ou le fichier ne contient pas certains matériaux, le plomb dans l'exemple ci-après (erreur ne survenant plus avec la version 2005 et les suivantes).

Message :

End of file on unit 12

Program stopped In Emfret because the

Following names were not recognized:

'LEAD

- D'autres erreurs liées au processus électromagnétique :

***** Photon : have you got the right mat. data set?**

Généralement, ce message signifie que vous essayez de transporter un photon d'une énergie plus élevée que la limite supérieure dans le tableau des sections efficaces du fichier **PEMF**.

Un message similaire est imprimé pour une erreur concernant les électrons.

- Un message du type : ****** Low energy neutron xsec not found for some media**

Une correspondance de noms n'a pas été trouvée entre les matériaux de **FLUKA** et les sections efficaces pour les neutrons de basse énergie. Avez-vous besoin de neutrons de basse énergie ? (Voyez les cartes **DEFAULTS** et **LOW-BIAS**, cf. manuel).

- Vérifiez les noms des matériaux.

- Vérifiez la carte **LOW-MAT** (si utilisée) : cette carte est-elle absolument nécessaire, et l'utilisez-vous correctement ? Cf. Manuel.

- Votre géométrie est-elle déboguée ?

Il peut s'agir d'une erreur triviale : fausses cartes d'entrée, mauvaise syntaxe ou un peu moins facile à détecter : voyez le rapport de « debug » dans **ALIFE**, si vous l'utilisez, ou directement dans **FLUKA**, **DEBUG** à la fin de **GEOEND**.

Si vous avez un message du type : **fmt : read unexpected character**
apparent state : internal I/O
last format: (2X,A3,I5,6D10.3)
lately reading sequential formatted internal IO

Une erreur est survenue à la lecture des données concernant les corps (**body data**), pour le format simple précision.

Si **last format : (2X,A3,I5,9(A2,I5))**, l'erreur s'est produite à la lecture des données de régions (**region data**).

Ce message peut également apparaître si un caractère invisible s'est inséré dans une ligne de la déclaration de la géométrie : une fois encore, utilisez un éditeur de texte UNIX.

Si votre message d'erreur ne correspond à aucun de ces cas, voyez le fichier **core**. Lors de la simulation, un dossier temporaire noté **fluka_XXXXX** (où **XXXXX** est un nombre de 5 chiffres choisi par rapport à la date et l'heure du lancement du programme) est ouvert et tous les fichiers de données nécessaires au calcul ainsi que les fichiers de sortie en formation : avec, **core**, ***.out**, ***.err**, ***.log**, les comptes rendus des calculs des estimateurs, etc.

Notez qu'à la fin de la simulation, seuls certains fichiers sont transférés automatiquement dans le répertoire sur lequel pointe **\$FLUPRO**, le répertoire temporaire est effacé avec tous les fichiers non importants.

Vous pouvez utiliser le débogueur de linux : **gdb**, appliqué au fichier **core** et à l'exécutable qui l'a généré.

Tapez la commande suivante : **gdb /home/path/executable core.nnnnn**

Utilisez **where**, **bt**, **frame x**, **list** pour obtenir quelques infos sur ce qui a provoqué le crash. Remarque : Le programme **gdb** n'est pas dénué de bugs.

B. Exemple de fichier d'entrée

Nous allons reproduire un fichier d'entrée d'une simulation qui donne des résultats, toujours dans le cadre de l'étude de l'irradiateur.

Le fichier est agrémenté de **commentaires** explicitant les différentes options (ils sont placés **avant** la carte qu'ils explicitent).

Certaines parties ont déjà été expliquées en détail, notamment la géométrie (*cf. 4. du présent document*) et les matériaux (*cf. 5.*), nous n'y reviendrons que très brièvement.

Fichier *irradie19.inp*

***** La carte de titre, une seule par fichier (cf. manuel).**

TITLE

Test irradiateur curie

***** Valeurs par défaut pour la simulation, à choisir selon le type d'étude effectuée, le type d'interactions mises en jeu, le type de particule, etc. Cf. manuel. Notre problème ne met en jeu que des interactions des photons avec la matière, donc le choix EM-CASCA est rapidement justifié.**

DEFAULTS

EM-CASCA

*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789

*** Caractéristiques du faisceau : source ponctuelle isotrope de césium 137, le poids statistique des photons est 1.0, cf. 3. 2) du présent document.

BEAM -661.7E-6 0.0 1.E4 0.0 0.0 1.0 PHOTON

*** Cette source est placée au point de coordonnées (40.0, 40.0, 0.1)

BEAMPOS 40.0 40.0 0.1

*** Début de la déclaration de la géométrie, cf. 4.

GEOBEGIN COMBINAT

Irradiateur cylindrique

RPP	1	-9999.0	+9999.0	-9999.0	+9999.0	-9999.0	+9999.0
RPP	2	-900.0	+900.0	-900.0	+900.0	-900.0	+900.0
RPP	3	+7.3	+72.7	+7.3	+72.7	-12.7	+77.7
RPP	4	+7.5	+72.5	+7.5	+72.5	-12.5	+77.5
RPP	5	+8.0	+72.0	+8.0	+72.0	-12.0	+65.0
ZCC	6	40.0	40.0	20.0			
XYP	7	0.0					
TRC	8	40.0	40.0	0.0	0.0	0.0	25.0
		8.25	20.0				
XYP	9	25.0					

END

* black hole

1	5	1	-2				
---	---	---	----	--	--	--	--

* vacuum

2	5	2	-3				
---	---	---	----	--	--	--	--

* inox

3	5	3	-4				
---	---	---	----	--	--	--	--

* plomb1

4	5	4	-5				
---	---	---	----	--	--	--	--

* plomb2

5	5	5	-6	-8			
---	---	---	----	----	--	--	--

* cylindre air

6	5	5	6	-8	-9		
---	---	---	---	----	----	--	--

* cône air

7	5	5	8				
---	---	---	---	--	--	--	--

* plomb3 : partie non définie précédemment

8	5	6	-8	9	-7		
---	---	---	----	---	----	--	--

* plomb4 : partie non définie précédemment

9	5	6	-8	7			
---	---	---	----	---	--	--	--

END
GEOEND

*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789

*** Début de la déclaration des matériaux, cf. 5.

MATERIAL 18.0 39.948 0.0016629 26.0 ARGON
 MATERIAL 24.0 51.9961 7.140 27.0 CHROME
 MATERIAL 28.0 58.6934 8.908 28.0 NICKEL

*
 *23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789

MATERIAL 7.84 29.0 INOX
 COMPOUND -0.78000 11.0 -0.15000 27.0 -0.07000 28.0 INOX

*
 MATERIAL 0.001205 30.0 AIR
 COMPOUND -0.75500 7.0 -0.23200 8.0 -0.01300 26.0 AIR

*
 * External Black Hole
 ASSIGNMAT 1.0 1.0
 * Vacuum
 ASSIGNMAT 2.0 2.0
 * inox
 ASSIGNMAT 29.0 3.0
 * plomb
 ASSIGNMAT 17.0 4.0
 ASSIGNMAT 17.0 5.0
 ASSIGNMAT 17.0 8.0
 ASSIGNMAT 17.0 9.0
 * air
 ASSIGNMAT 30.0 6.0

```

ASSIGNMAT 30.0      7.0
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
*** Demande de tracé d'une section de la géométrie : nom du fichier de sortie :
irradie19XXX_PLOTGEOM.STORE avec XXX = numéro du cycle, cf. 8. 1)
PLOTGEOM          0.0      1.0      0.0      0.0      1.0      0.0
      irradiateur curie modèle cylindrique plan z,y
      40.0      0.0      -15.0      40.0      80.0      80.0
      0.0      1.0      0.0      0.0      0.0      1.0
      1.0      1.0      0.0      0.0
* **** Scoring ****
*** Début de la section des estimateurs
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
*** Activation de la routine FLUSCW pour le calcul des doses, cf. 7. 2)
USERWEIGH          1.              0.
*** Définition de 7 estimateurs USRBIN qui calculeront la fluence particulière dans des mailles
cartésiennes différentes selon ce que l'on désire calculer, cf. 6.
*** Le 1er, nommé « Z0 », calcule la fluence dans une boîte dont x et y sont compris entre 0 et 80,
et z entre 0 et 1, avec 100 mailles dans toutes les directions
*
USRBIN          10.      7.      -76.      80.      80.      1.20
USRBIN          0.      0.      0.      100.      100.      100.&
*
*** Le 2ème, nommé « Z25 », calcule la fluence dans une boîte dont x et y sont compris entre 0 et
80, et z entre 24.5 et 25.5, avec 100 mailles dans toutes les directions
*
USRBIN          10.      7.      -77.      80.      80.      25.5 Z25
USRBIN          0.      0.      24.5      100.      100.      100.&
*
*** Le 3ème, nommé « Z35 », calcule la fluence dans une boîte dont x et y sont compris entre 0 et
80, et z entre 34.5 et 35.5, avec 100 mailles dans toutes les directions
*
USRBIN          10.      7.      -78.      80.      80.      35.5 Z35
USRBIN          0.      0.      34.5      100.      100.      100.&
*
*** Le 4ème, nommé « Z45 », calcule la fluence dans une boîte dont x et y sont compris entre 0 et
80, et z entre 34.5 et 35.5, avec 100 mailles dans toutes les directions
*
USRBIN          10.      7.      -79.      80.      80.      45.5 Z45
USRBIN          0.      0.      44.5      100.      100.      100.&
*
*** Le 5ème, nommé « Z55 », calcule la fluence dans une boîte dont x et y sont compris entre 0 et
80, et z entre 54.5 et 55.5, avec 100 mailles dans toutes les directions
*
USRBIN          10.      7.      -80.      80.      80.      55.5 Z55
USRBIN          0.      0.      54.5      100.      100.      100.&
*
*** Le 6ème, nommé « X0 », calcule la fluence dans une boîte dont x compris entre 39 et 41, y
compris entre 20 et 60, et z entre 0 et 65, avec 100 mailles dans toutes les directions
*
USRBIN          10.      7.      -81.      41.      60.      65.X0
USRBIN          39.      20.      0.      100.      100.      100.&
*
*** Le 7ème, nommé « total », calcule la fluence dans une boîte dont x et y sont compris entre 20 et
60, et z entre 0 et 65, avec 100 mailles dans toutes les directions
*
USRBIN          10.      7.      -82.      60.      60.      65.total
USRBIN          20.      20.      0.      100.      100.      100.&
*
*23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789
*** La carte qui définit les conditions d'arrêt et lance le transport, cf. manuel
*** Retenons WHAT(1) : nombre maximal d'histoires primaires simulées durant le « run »
    
```

*** Ici, 4 millions d'histoires primaires seront simulées

START 4000000.

*** Stoppe l'exécution du programme, cf. manuel.

STOP

Il existe de très nombreuses options, notamment en ce qui concerne les estimateurs, le générateur de nombres aléatoires, etc.

Les fichiers de sortie principaux en fin de simulation seront :

irradie19001.out : il retrace l'historique de l'intégralité du cycle de simulation

irradie19001.log : il contient tous les problèmes liés à l'exécution du cycle, la cause de l'erreur, s'il s'en est produit, se trouve généralement dans ce fichier ou dans le ***.out**

irradie19001_fort.76 : il contient les résultats du binning nommé Z0

irradie19001_fort.77 : il contient les résultats du binning nommé Z25

irradie19001_fort.78 : il contient les résultats du binning nommé Z35

irradie19001_fort.79 : il contient les résultats du binning nommé Z45

irradie19001_fort.80 : il contient les résultats du binning nommé Z55

irradie19001_fort.81 : il contient les résultats du binning nommé X0

irradie19001_fort.82 : il contient les résultats du binning nommé total

C. Routine *source.f* pour simuler une source isotrope de cobalt 60

Nous reproduisons ici l'intégralité du fichier **source.f** utilisé pour simuler une source isotrope de cobalt 60 dont la procédure est décrite dans le chap. 9.

```

*$ CREATE SOURCE.FOR
*COPY SOURCE
*
*==== source =====*
*
  SUBROUTINE SOURCE (NOMORE)

    INCLUDE '(DBLPRC)'
    INCLUDE '(DIMPARG)'
    INCLUDE '(IOUNIT)'

*
*-----*
*
  Created on 07 january 1990   by   Alfredo Ferrari & Paola Sala
*
*
*
*
  Last change on 21-jun-98   by   Alfredo Ferrari
*
*
* This is just an example of a possible user written source routine.
* note that the beam card still has some meaning - in the scoring the
* maximum momentum used in deciding the binning is taken from the
* beam momentum. Other beam card parameters are obsolete.
*
*-----*
*
  INCLUDE '(AACOLL)'
  INCLUDE '(BEAM)'
  INCLUDE '(EPISOR)'
  INCLUDE '(FHEAVY)'
  INCLUDE '(PAPROP)'
  INCLUDE '(LTCLCM)'
  INCLUDE '(STACK)'
  INCLUDE '(STARS)'
  LOGICAL LFIRST
  SAVE LFIRST
  DATA LFIRST / .TRUE. /
*====*

```

```

*
*           BASIC VERSION
*
*=====
*           NOMORE = 0
* +-----
* | First call initializations:
* | IF ( LFIRST ) THEN
* | *** The following 3 cards are mandatory ***
* |           TKESUM = ZERZER
* |           LFIRST = .FALSE.
* |           LUSSRC = .TRUE.
* | *** User initialization ***
* |           END IF
* |
* +-----
* | Push one source particle to the stack. Note that you could as well
* | push many but this way we reserve a maximum amount of space in the
* | stack for the secondaries to be generated
* | Lstack is the stack counter: of course any time source is called it
* | must be =0
* |           LSTACK = LSTACK + 1
* | Wt is the weight of the particle
* |           WT (LSTACK) = ONEONE
* |           WEIPRI = WEIPRI + WT (LSTACK)
* | Particle type (1=proton.....). Ijbeam is the type set by the BEAM
* |           IJBEAM = WHASOU(5)
* | card
* +-----
* | Heavy ion:
* | IF ( IJBEAM .EQ. -2 ) THEN
* |           IJHION = IPROZ * 1000 + IPROA
* |           IJHION = IJHION * 100 + KXHEAV
* |           IONID = IJHION
* |           CALL DCDION ( IONID )
* |           CALL SETION ( IONID )
* |           ILO (LSTACK) = IJHION
* +-----
* | Normal hadron:
* | ELSE
* |           ILO (LSTACK) = IJBEAM
* |           END IF
* +-----
* | From this point .....
* | Particle generation (1 for primaries)
* |           LO (LSTACK) = 1
* | User dependent flag:
* |           LOUSE (LSTACK) = 0
* | User dependent spare variables:
* |           DO 100 ISPR = 1, MKBMX1
* |               SPAREK (ISPR,LSTACK) = ZERZER
* |           100 CONTINUE
* | User dependent spare flags:
* |           DO 200 ISPR = 1, MKBMX2
* |               ISPARK (ISPR,LSTACK) = 0
* |           200 CONTINUE
* | Save the track number of the stack particle:
* |           ISPARK (MKBMX2,LSTACK) = LSTACK
* |           NPARMA = NPARMA + 1
* |           NUMPAR (LSTACK) = NPARMA
* |           NEVENT (LSTACK) = 0
* |           DFNEAR (LSTACK) = +ZERZER
* | ... to this point: don't change anything
* | Particle age (s)
* |           AGECHK (LSTACK) = +ZERZER
* |           AKNSHR (LSTACK) = -TWOTWO
* | Group number for "low" energy neutrons, set to 0 anyway
* |           IGROUP (LSTACK) = 0
* | Kinetic energy of the particle (GeV)
* | Stratagème de G. Battistoni pour simuler une source non monoénergétique

```

```

* FLRDM(XDUMMY) : nombre aléatoire compris entre 0 et 1
* WHASOU(n) : les champs de la carte SOURCE du fichier d'entrée (cf. A).
* Cela permet de modifier les paramètres sans avoir à recompiler.
* WHASOU(1) est la valeur test pour le nombre aléatoire.
* WHASOU(2) est la première valeur d'énergie
* WHASOU(3) est la seconde valeur d'énergie
      IF (FLRNDM(XDUMMY) .LE. WHASOU(1)) THEN
          PBEAM = WHASOU(2)
      ELSE
          PBEAM = WHASOU(3)
      ENDIF
      TKE (LSTACK) = SQRT ( PBEAM**2 + AM (IJBEAM)**2 ) - AM (IJBEAM)
* Particle momentum
      PMOM (LSTACK) = PBEAM
*      PMOM (LSTACK) = SQRT ( TKE (LSTACK) * ( TKE (LSTACK) + TWOTWO
*      & * AM (ILO(LSTACK)) ) )
* Cosines (tx,ty,tz)
*** John Clem's method to define an isotropic source ***
* ONEONE = 1, TWOPIP = 2π
* FLRNDM(nDUMMY) avec n = Y,Z : nombres aléatoires compris entre 0 et 1
      COSTH = ONEONE - 2.D0*FLRNDM(YDUMMY)
      IF (COSTH.GT.ONEONE) THEN
          COSTH = ONEONE
      ELSEIF (COSTH.LT.-ONEONE) THEN
          COSTH = -ONEONE
      ENDIF
      ANG = TWOPIP*FLRNDM(ZDUMMY)
      UBEAM = SQRT(ONEONE - COSTH*COSTH)*COS(ANG)
      VBEAM = SQRT(ONEONE - COSTH*COSTH)*SIN(ANG)
      WBEAM = COSTH
*** End of John's code ****
      TX (LSTACK) = UBEAM
      TY (LSTACK) = VBEAM
      TZ (LSTACK) = WBEAM
*      TZ (LSTACK) = SQRT ( ONEONE - TX(LSTACK)**2 - TY(LSTACK)**2 )
* Polarization cosines:
      TXPOL (LSTACK) = -TWOTWO
*      TYPOL (LSTACK) = -TWOTWO
*      TZPOL (LSTACK) = -TWOTWO
      TYPOL (LSTACK) = +ZERZER
      TZPOL (LSTACK) = +ZERZER
* Particle coordinates
      XA (LSTACK) = XBEAM
      YA (LSTACK) = YBEAM
      ZA (LSTACK) = ZBEAM
* Calculate the total kinetic energy of the primaries: don't change
      IF ( ILO(LSTACK) .EQ. -2 .OR. ILO(LSTACK) .GT. 100000 ) THEN
          TKESUM = TKESUM + TKE (LSTACK) * WT (LSTACK)
      ELSE IF ( ILO(LSTACK) .NE. 0 ) THEN
          TKESUM = TKESUM + ( TKE (LSTACK) + AMDISC (ILO(LSTACK)) )
          & * WT (LSTACK)
      ELSE
          TKESUM = TKESUM + TKE (LSTACK) * WT (LSTACK)
      END IF
* Here we ask for the region number of the hitting point.
*      NREG (LSTACK) =...
* The following line makes the starting region search much more
* robust if particles are starting very close to a boundary:
      CALL GEOCRS ( TX (LSTACK), TY (LSTACK), TZ (LSTACK) )
      CALL GEOREG ( XA (LSTACK), YA (LSTACK), ZA (LSTACK),
          & NREG (LSTACK), IDISC )
* Do not change these cards:
      CALL GEOHSM ( NHSPNT (LSTACK), 1, -11, MLATTC )
      NLATTC (LSTACK) = MLATTC
      CMPATH (LSTACK) = ZERZER
      CALL SOEVSV
      RETURN
=====
*==== End of subroutine Source =====*
      END
    
```

D. Tracer un profil de géométrie avec ALIFE

Ce programme permet de créer la géométrie d'un fichier d'entrée de **FLUKA**, nous ne l'utiliserons ici que pour tracer la géométrie d'un fichier déjà existant.

ALIFE est gratuit et peut s'obtenir par FTP (cf. le site de **FLUKA**, fluka.org pour la marche à suivre). Tutorial ici : <http://pcfluka.mi.infn.it/course/WebCourse/alife/P001.html>

On obtient une archive nommée par exemple **ALIFE.tgz**

D.1. Installation d'ALIFE

Il s'agit tout d'abord de décompresser l'archive dans un dossier appelé ALIFE :

tar - xvzf ALIFE.tgz

Il faut ensuite copier le fichier **alifeDefaults** à la racine :

cp ALIFE/alifeDefaults ~/.alifeDefaults

Le fichier **.alifeDefaults** contient le texte suivant :

```
set my_tag_length 8
set myfont 7x13
set highprec no
set btype XYP
set blue "#0000CA"
set Executable "~/AliRoot/fluka/fluka_tpc.exe"
set standardOut fort.11
```

Éditez ce fichier pour indiquer le chemin vers votre propre exécutable **FLUKA** (ligne bleue).
Exemple : mon exécutable nommé **flukahp** se trouve dans **/home/wurth/fluka2006**, le fichier **.alifeDefaults** devient alors :

```
set my_tag_length 8
set myfont 7x13
set highprec no
set btype XYP
set blue "#0000CA"
set Executable "~/fluka2006/flukahp"
set standardOut fort.11
```

Ceci est absolument nécessaire, **ALIFE** utilise **FLUKA** et doit donc pouvoir y accéder.

D.2. Tracé d'un profil de géométrie

Lancez **ALIFE** à partir d'un terminal, soit vous avez créé un alias, soit vous vous placez dans le répertoire où se trouve l'exécutable, par exemple, si je suis à la racine et que mon exécutable se trouve dans **~/fluka2006/ALIFE** :

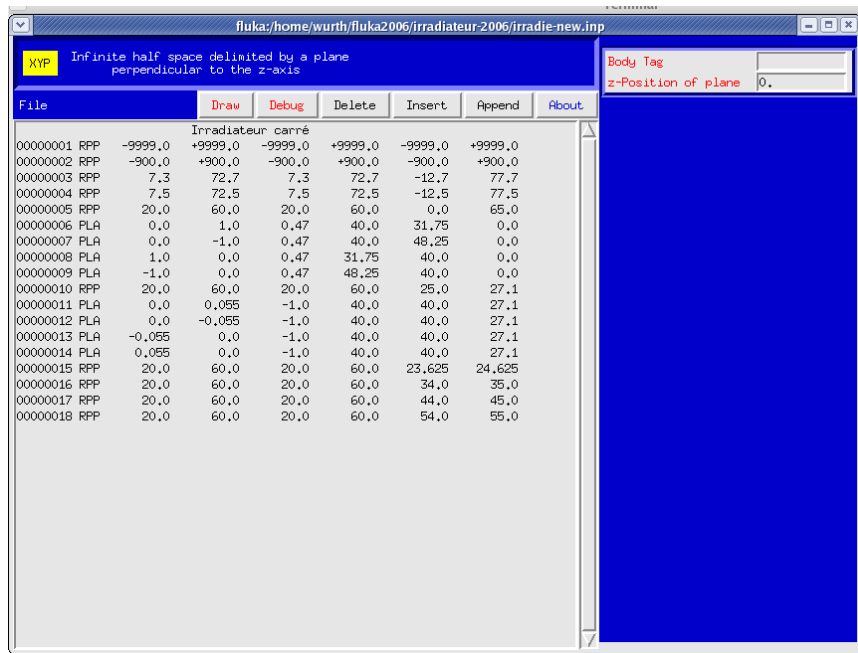
[wurth@ipnscr01 ~] \$

[wurth@ipnscr01 ~] \$ cd fluka2006/ALIFE

[wurth@ipnscr01 ALIFE] \$ alife&

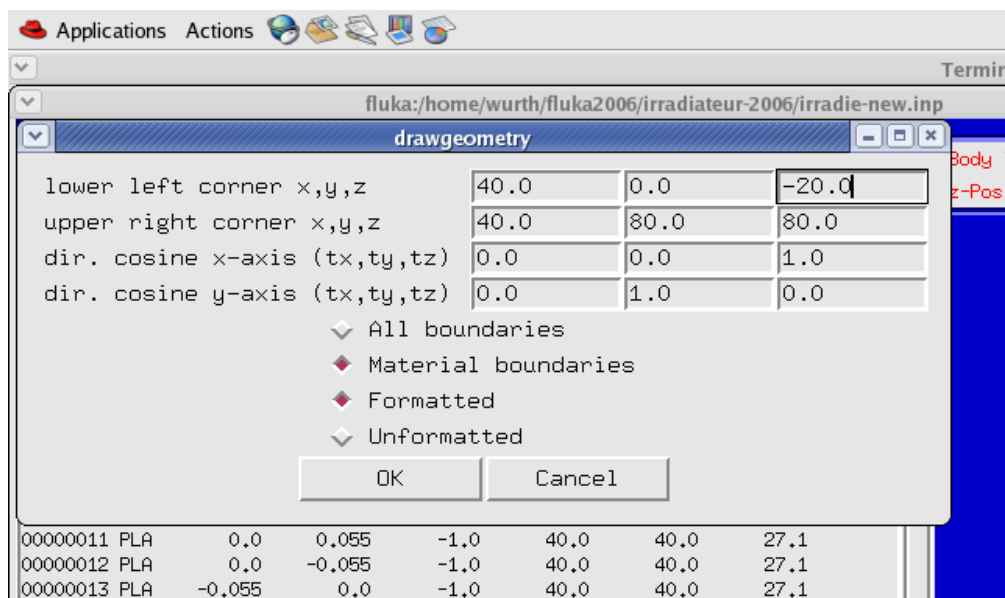
Une fenêtre s'ouvre, utilisez l'onglet **File**, puis **Open** et l'explorateur pour ouvrir le fichier de votre choix. Ici, **irradie-new.inp**

Cf. figure à la page suivante : La liste des formes de géométrie telle qu'elle est décrite dans le fichier apparaît.



Remarque : Notez l'onglet **Debug**, qui permet de chercher les éventuelles erreurs de votre géométrie dans un domaine de coordonnées choisi par l'utilisateur. Ceci peut être également effectué directement dans **FLUKA**, en complétant la carte **GEOEND** (cf. le manuel à ce sujet).

Le tracé s'effectue en sélectionnant l'onglet **Draw**. Une autre fenêtre s'ouvre, dans laquelle l'utilisateur est invité à saisir les coordonnées des limites de son graphe.



- La première ligne indique les limites inférieures en x, y et z.
- La seconde ligne indique les limites supérieures en x, y et z.
- La troisième ligne indique les cosinus directeurs en x.
- La quatrième ligne indique les cosinus directeurs en y.

Il est important d'entrer des nombres munis de la virgule.

Lorsque la géométrie ne comporte pas d'erreur, un fichier **PLOTGEOM.STORE** est généré dans le répertoire **ALIFE**.

Les deux options suivantes concernent la nature du graphe :

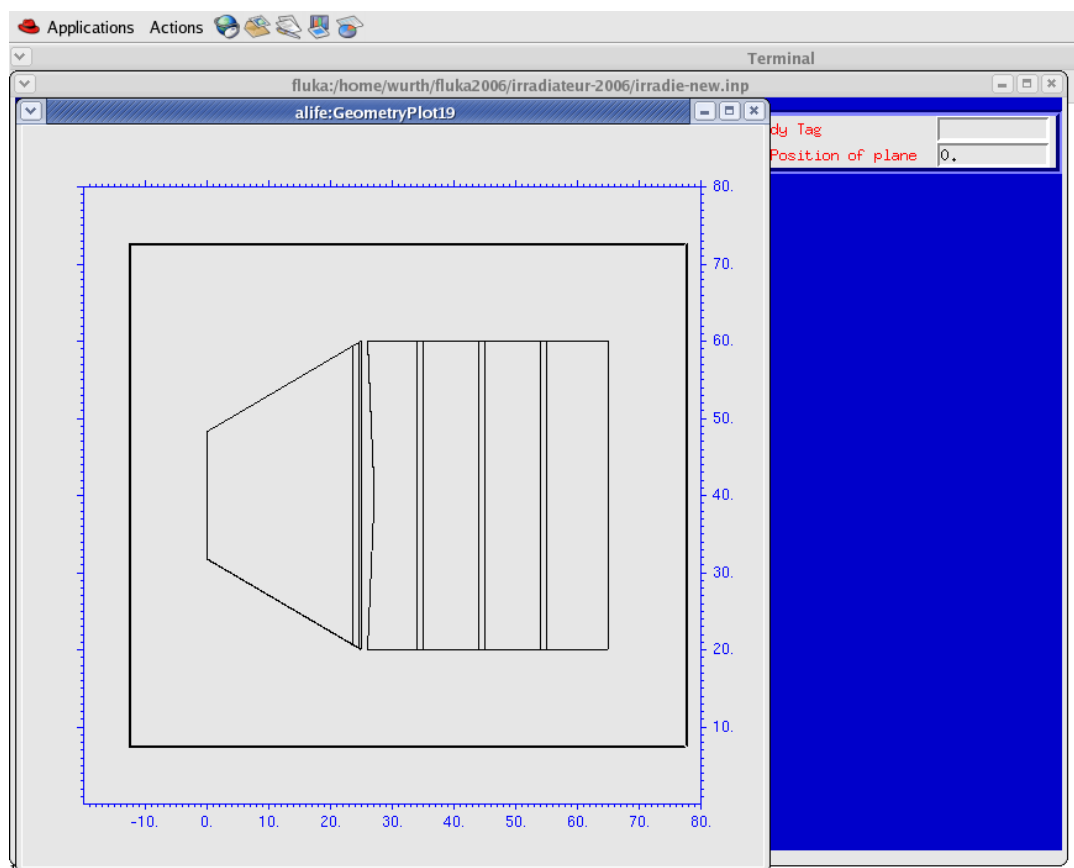
- On peut tracer toutes les limites géométriques dues à la construction, « All boundaries ».
- On peut tracer les limites dues aux différents matériaux (définis dans le fichier d'entrée), « Material boundaries ».

Les deux derniers onglets précisent le format du graphique tracé,

- « Formatted » pour voir le graphique à l'écran et pour obtenir un fichier **PLOTGEOM.STORE** « formaté ».

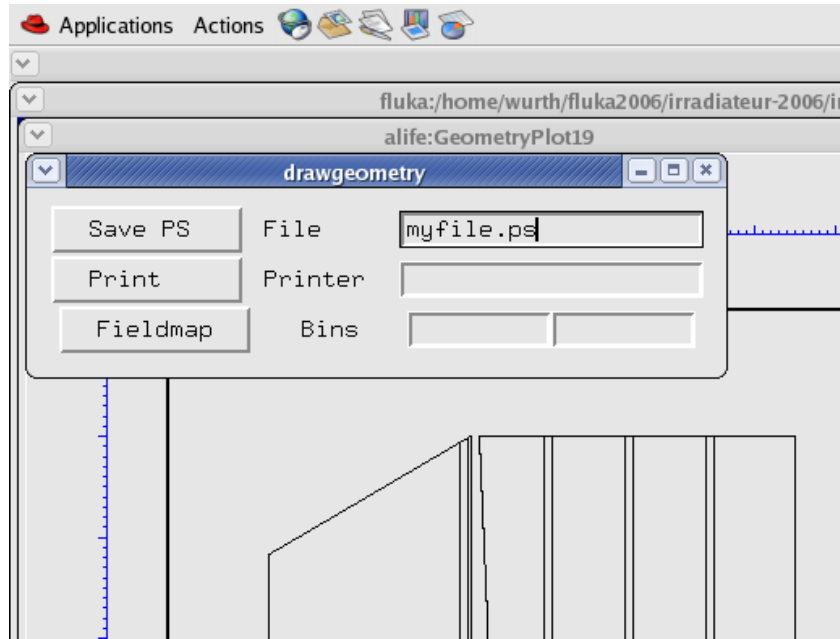
- « Unformatted », le graphe n'apparaîtra pas à l'écran et le fichier **PLOTGEOM.STORE** sera « non formaté » et utilisable dans **pawlevbin** (Cf. chap. 8).

Dans l'exemple, nous avons choisi de tracer une section de la géométrie avec les limites dues aux matériaux dans le plan (z, y), x étant fixé à 40. Les limites en z (qui sera donc l'abscisse) sont -20, 80 et celles en y (l'ordonnée) sont 0, 80. En sélectionnant **Draw**, une fenêtre s'ouvre et un fichier **PLOTGEOM.STORE** est généré dans le répertoire **ALIFE** (rappel).



Un double clic sur cette fenêtre de graphique ouvre un onglet permettant de sauver ce graphe sous un fichier au format **postscript** ou d'effectuer d'autres opérations (impression, ...).

Saisissez le nom de votre fichier, par exemple, **myfile.ps** et sélectionnez **Save PS**.



Un fichier **myfile.ps** est créé dans le répertoire d'**ALIFE**.

Pour superposer ce profil de géométrie à une courbe d'isodoses obtenue avec **USRBIN** et le traitement adéquat (**usbsuw** + **pawlevbin**, cf. chap. 8), il faut donc utiliser le fichier **PLOTGEOM.STORE** non formaté ainsi obtenu.

E. Utilisation de l'interface graphique flukaGUI

E.1. Installation

Version 6 d'octobre 2006. GUI = Graphical User Interface.

Cette interface développée par l'Université de Houston (collaboration NASA/CERN) a pour but de traiter en une fois les différentes étapes décrites dans le chapitre 8. le tout dans un confort d'utilisation assez intéressant.

Évidemment, il faut que FLUKA2006.3 soit installé sur votre machine.

Il faut également qu'une version de **ROOT** compatible avec **flukaGUI** soit installée.

Plus d'infos au lien suivant : <http://www.fluka.org/tools/FlukaGUI.html>

Les versions 5.11/06, 5.12/00, 5.13/02 et 5.13/04 sont supportées.

Installation et configuration de ROOT (exemple) :

Je télécharge une version compatible avec ma machine sur le site <http://root.cern.ch>

Exemple : **root_v5.12.00.Linux.FedoraCore3.gcc3.4.3.tar.gz**

J'extrais cette archive dans le répertoire de mon choix : **/vol0**

```
$ gunzip root_v5.12.00.Linux.FedoraCore3.gcc3.4.3.tar.gz
```

```
$ tar -zxvf root_v5.12.00.Linux.FedoraCore3.gcc3.4.3.tar
```

Cette dernière commande va me créer un répertoire **root** dans lequel l'archive est extraite. Il faut à présent configurer les variables d'environnement (comme pour FLUKA)

```
$ setenv ROOTSYS /vol0/root
```

```
$ setenv LD_LIBRARY_PATH /vol0/root/lib :./
```

```
$ set path = ( /vol0/root/bin $path )
```

ROOT est à présent configurée et on peut installer l'interface qui y fera appel.

On télécharge la version 6 de flukaGUI au lien suivant :

<http://www.fluka.org/tools/flukagui/aFlukaGUI-v06.tgz>

```
$ tar -zxf aFlukaGui-v06.tgz
```

Ceci décompresse l'archive dans le répertoire **aFlukaGUI-v06**.

```
$ cd aFlukaGUI-v06
```

```
$ ./configure (ceci teste la configuration avec la version de ROOT et de FLUKA)
```

```
$ make (ceci exécute le makefile, toute l'installation s'effectue normalement avec cette seule commande)
```

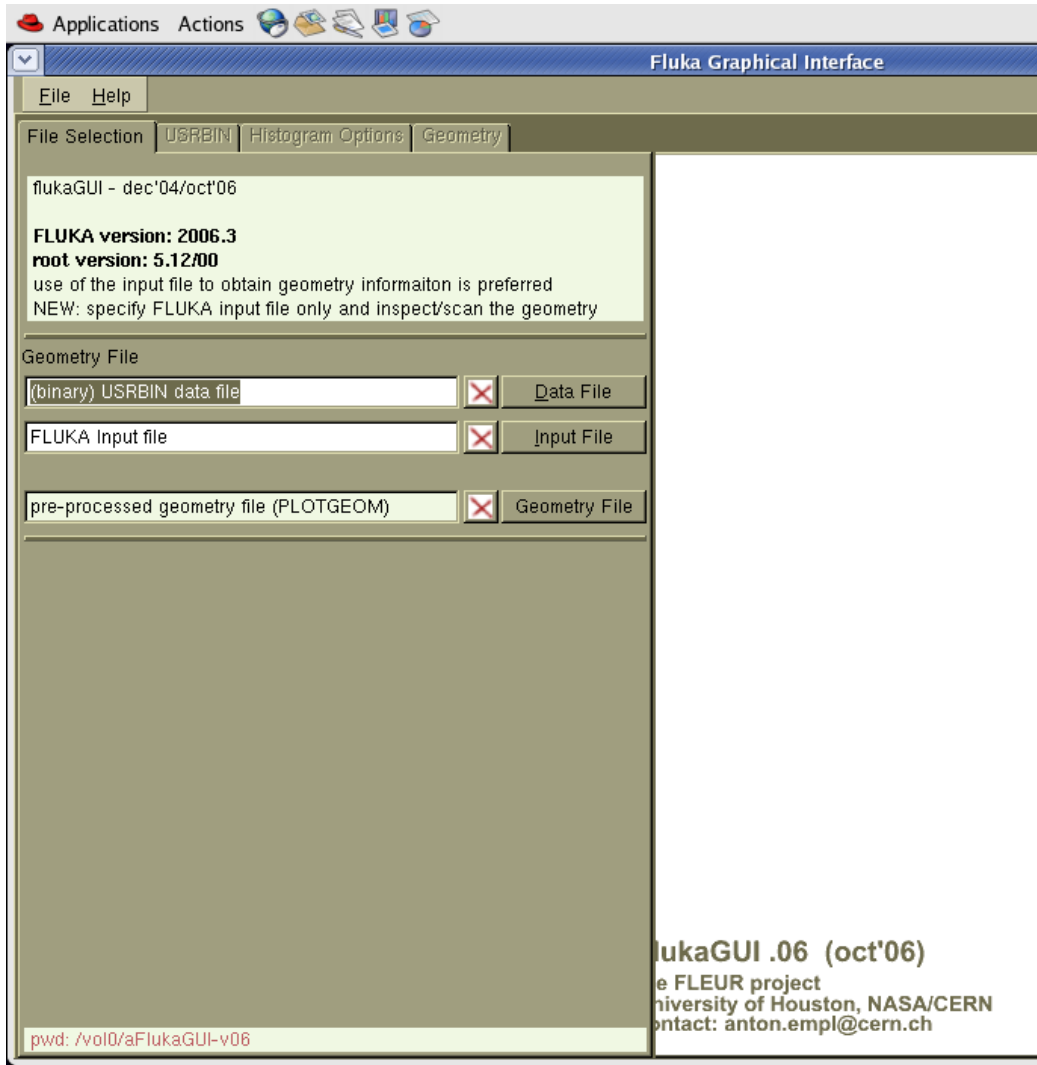
```
$ ./src/flukaGUI (exécute le programme)
```

E.2. Petit aperçu des possibilités

Après la dernière commande, si tout se passe bien, la fenêtre de l'interface s'ouvre (cf. page suivante).

On peut alors choisir un fichier de données issu d'un estimateur **USRBIN**, un fichier d'entrée de **FLUKA**, ou un fichier de géométrie.

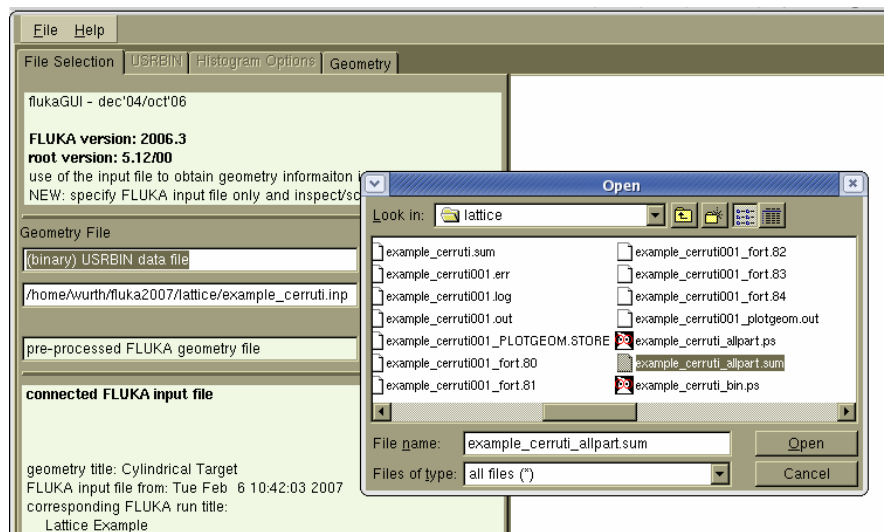
À noter que la géométrie peut être affichée dans l'interface à l'aide du fichier contenant les données d'entrée (c'est écrit sur l'image ci-dessous : « specify FLUKA input only and inspect/scan the geometry »).

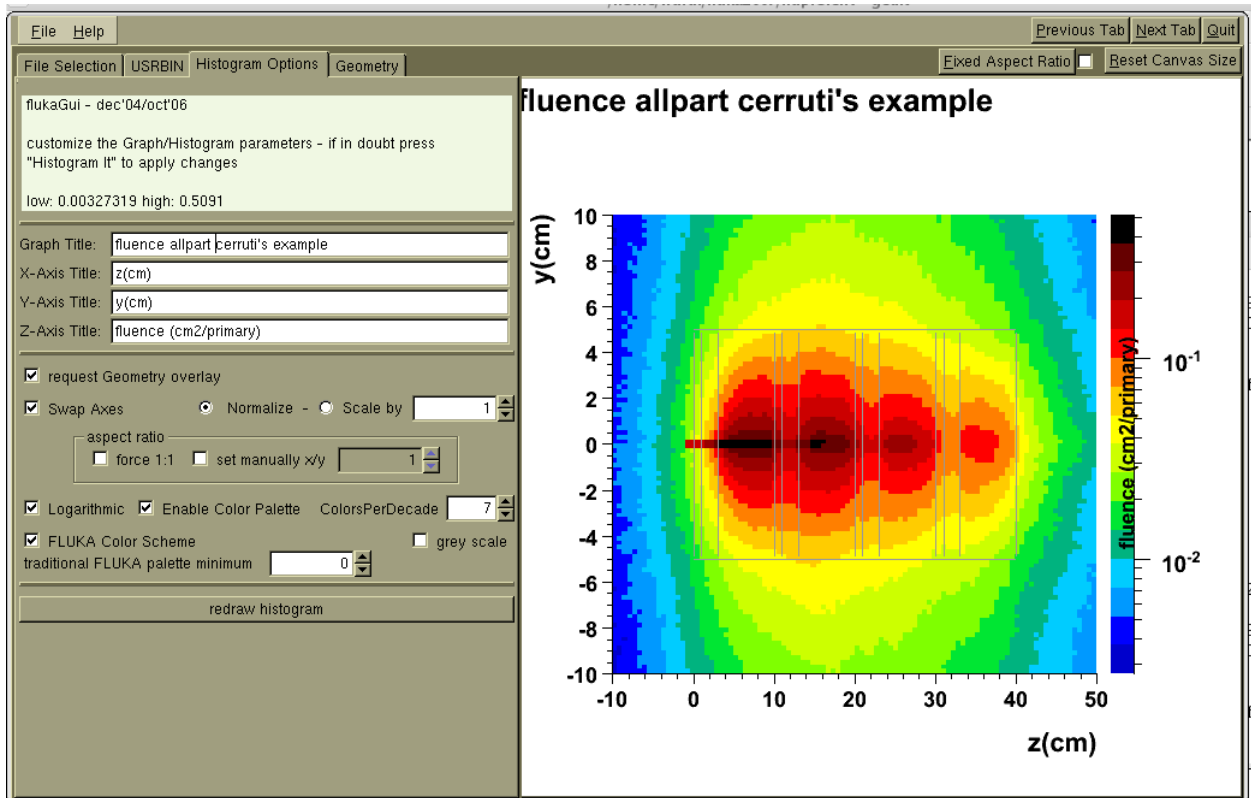


Ensuite, il suffit de se laisser guider, pour passer d'un onglet à l'autre **File Selection**, **USRBIN**, **Histogram Options**, **Geometry** et utiliser les diverses options disponibles.

Exemple.

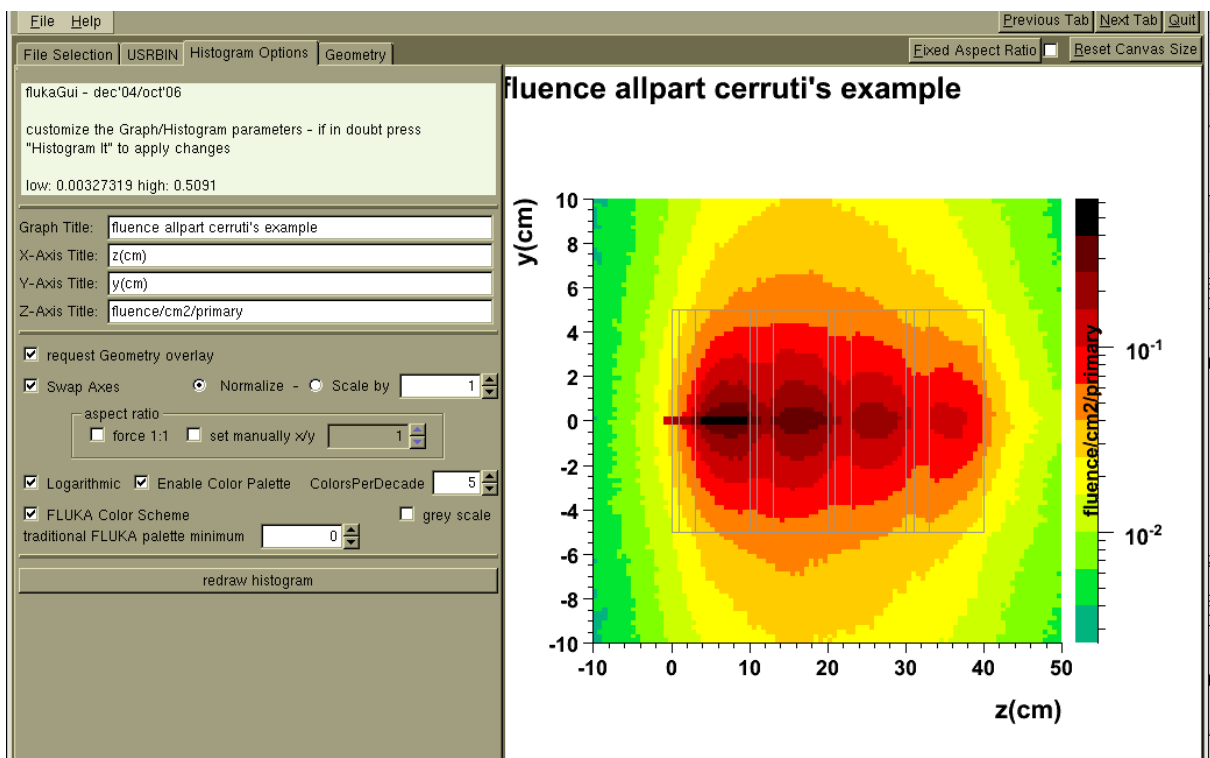
Sélectionnons un fichier de données **USRBIN** (fichier ***.sum**), un fichier de géométrie et un fichier de données d'entrée **FLUKA**.



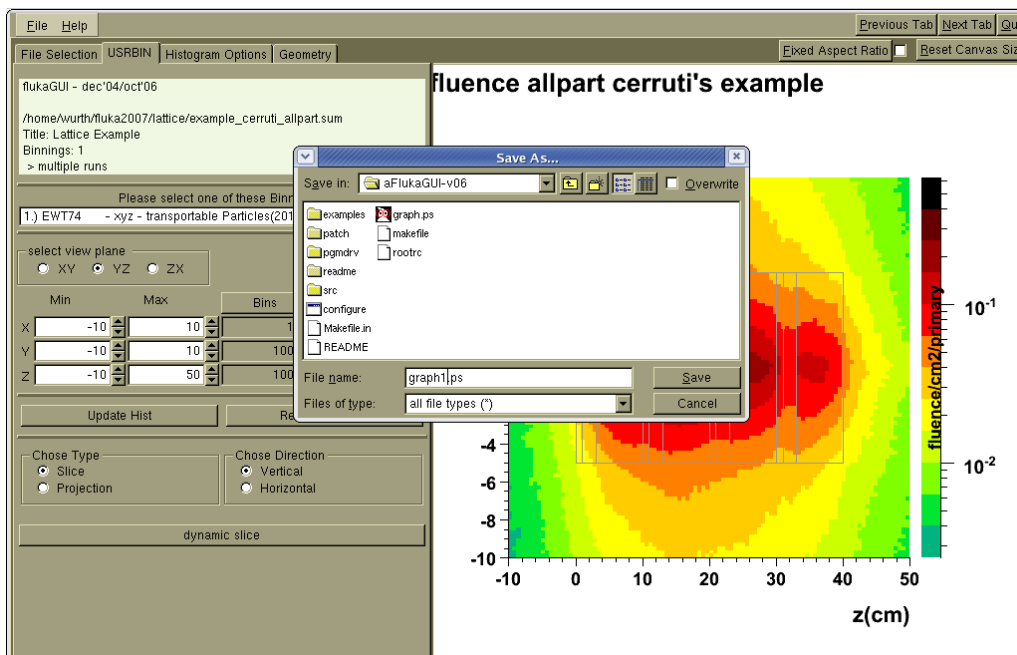


Selon vos choix et options, jouez avec les divers onglets pour obtenir le graphe voulu, selon la projection souhaitée, etc. Ici, un graphe dans le plan (z, y) avec le profil de géométrie superposé.

Modification du nombre de couleurs par décade :



Sauvegarde du graphique affiché à l'écran (ce peut être la géométrie, ou l'histogramme) dans un fichier **postscript** dont le nom peut-être choisi par l'utilisateur ainsi que l'emplacement auquel le fichier sera créé.



Remarques : Bien que convivial et intuitif, ce programme comporte encore un certain nombre de bugs, notamment l'affichage de messages d'erreurs intempestifs lorsque l'on désire charger un fichier (de données d'entrée, d'estimateur **USRBIN** ou de géométrie), ainsi que la quasi-invisibilité de la géométrie superposée à l'histogramme lors de la sauvegarde du graphe dans un fichier **postscript**. Ces éléments concourent à me faire préférer la méthode à l'ancienne qui consiste à répéter les étapes en passant par les programmes **usbsuw**, **pawlevbin** et **bpawlev**. C'est tout aussi rapide, il n'y a pas de bugs, et la géométrie est bien visible ! Gageons que les versions futures d'**aFlukaGUI** seront dénuées de ces inconvénients. Dans le même genre **Flair** semble déjà plus performant même s'il y reste également quelques bugs (cf. partie consacrée à cette interface, chap. **13**).

F. Utilisation de l'option LATTICE pour la construction de la géométrie

F.1. Généralités

Cette option peut s'avérer utile si la géométrie à représenter est composée de plusieurs cellules devant se reproduire un certain nombre de fois. Nous ne décrivons ici qu'un cas simple qui depuis la version 2006.3 permet de pouvoir tout déclarer dans le seul fichier d'entrée. Les cas plus complexes doivent faire appel à la routine **lattic.f**, cf. *le manuel aux sections géométrie, user routines pour de plus amples informations*.

Cette opération permet de gagner en temps de calcul, **FLUKA** applique des symétries par rapport à la région originelle pour transporter les particules dans les régions « clones ». Il faut cependant noter que les estimateurs basés sur des calculs par régions ne seront plus significatifs ici (**SCORE**, **USRBDX**, **USRTRACK**, etc.). Il faudra utiliser des estimateurs indépendants des régions, **USRBIN**, par exemple.

Rappel : L'architecture de la section de géométrie se compose de la façon suivante :

Carte **GEOBEGIN**

Titre de la géométrie

Déclaration des corps

Carte **END**

Déclaration des régions

Carte **END**

Cartes **LATTICE** (option)

Volumes des régions volumes (option, cf. titre de la géométrie).

Carte **GEOEND**

Il faut donc définir proprement la géométrie de base dans une forme « conteneur » qui sera répliquée autant de fois que requise. Les régions additionnelles doivent également être décrites, mais sans leur assigner de matériaux. Ces régions seront définies par une carte **LATTICE**.

La transformation géométrique doit être définie à l'aide de la carte **ROT-DEFI** et associée à chaque cellule en tout point et en direction de chaque cellule afin de permettre de la relier à tout point et direction de la cellule de base.

La carte **LATTICE** identifie la cellule et établit la correspondance entre le numéro de région et le numéro de cellule. Le numéro de région est celui figurant dans la déclaration des régions et le numéro de cellule est celui choisi par l'utilisateur.

➤ Dans la carte **LATTICE**, les significations sont les suivantes :

- **WHAT(1)** : numéro de la "Région-conteneur" de la première cellule (« De la région WHAT(1)... »). Par défaut : pas de défaut
- **WHAT(2)** : numéro de la "Région conteneur" de la dernière cellule (« ... à la région WHAT(2)... »). Par défaut = **WHAT(1)**.
- **WHAT(3)** : longueur du pas dans l'assignation des numéros de "régions-conteneurs" (« ... par pas de **WHAT(3)** »). Par défaut = 1.0
- **WHAT(4)** : numéro de la première cellule assignée à la région **WHAT(1)**. Par défaut : pas de défaut.
- **WHAT(5)** = numéro de la dernière cellule assignée à la région **WHAT(2)**.

Par défaut : pas de défaut.

- **WHAT(6)** = longueur du pas assignant les numéros de cellules (« par pas de **WHAT(6)** »). Par défaut = 1.0
- **SDUM** = index de la transformation associée à cette cellule.

Exceptionnellement, **SDUM** peut contenir un nombre entier, suivant les caractères suivants : "ROT#", "Rot#", "rot#", "RO#", "Ro#", "ro#". Si l'une de ces expressions est présente, la transformation comportant le numéro x sera associée à cette carte **LATTICE**.

Une géométrie peut être constituée de zones composées de cellules et de zones "normales" décrites par des régions standard. De nombreuses cartes **LATTICE** peuvent être décrites dans une même géométrie, avec des symétries diverses dans des zones diverses. En principe, toute transformation analytique est possible (rotation, translation, réflexion, ou une combinaison de ces opérations).

Il faut veiller à ce que toute région de la cellule de base (cellule 0) est entièrement contenue après transformation des coordonnées dans toute cellule copiée appartenant à la transformation. Les régions « prises » entre deux cellules différentes peuvent provoquer des comportements imprévisibles. La cellule de base ne doit pas nécessairement décrire une région réelle, mais peut constituer un prototype à reproduire un certain nombre de fois.

Remarque : Les régions constituant les cellules copiées n'ont pas à être incluses dans d'autres cartes du fichier d'entrée. Les matériaux, seuils, etc., doivent uniquement être assignés aux régions constituant la cellule de base. Bien entendu, ceci implique que toutes les copies d'une même cellule de base partageront le même matériau, les mêmes propriétés physiques....

➤ La carte **ROT-DEFI** :

- **WHAT(1)** : assigne un index de transformation et l'axe de rotation correspondant.

< ou = 0.0: la carte est ignorée.

> 0.0 : interprété comme suit : $i+j*100$

où i = index de la rotation

j = 1 rotation par rapport à l'axe x

= 2 rotation par rapport à l'axe y

= 3 rotation par rapport à l'axe z

Par défaut = 0.0 (pas de transformation définie)

- **WHAT(2)** : angle polaire de la rotation ($\theta = 0 \dots 180$ degrés). Par défaut : pas de valeur par défaut.
- **WHAT(3)** : angle azimutal de la rotation ($\varphi = -180 \dots 180$ degrés). Par défaut : pas de valeur par défaut.
- **WHAT(4)** : décalage en X de la translation. Par défaut : pas de valeur par défaut.
- **WHAT(5)** : décalage en Y de la translation. Par défaut : pas de valeur par défaut.
- **WHAT(6)** : décalage en Z de la translation. Par défaut : pas de valeur par défaut.
- **SDUM** : non utilisé.

Par défaut, si l'option **ROT-DEFI** n'est pas donnée : aucune transformation n'est définie.

Remarques :

1. Cette option a d'abord été créée pour définir la transformation géométrique appliquée à un maillage (binning) constituant un estimateur (**USRBIN**, par exemple). Elle a ensuite été étendue pour la définition de géométrie (je vous conseille vivement la lecture de toutes les sections consacrées à ces problèmes de « lattice » dans le manuel de **FLUKA**. Cf. la carte **ROT-DEFIni**, la section lattice du chapitre consacré à la géométrie et le paragraphe consacré à la routine **lattic** dans le chapitre « user routines »).

2. Important : les paramètres **WHAT**(4 à 6) de la carte **ROT-DEFI** définissent les transformations à effectuer pour passer des coordonnées de la cellule copiée à la cellule de base et non le contraire, je suis personnellement tombé dans le panneau. Autrement dit, si le plan d'origine se situe en $z = 20.0$, et la copie en $z = 50.0$, la transformation définie en **dz (WHAT(6))** sera -30.0 et non pas $+30.0$! **/ ! **

Illustrons ceci avec des exemples.

F.2. Exemple 1

Nous désirons décrire un réseau de cylindres identiques séparés d'une distance identique dans le plan (x,y), représentant éventuellement des barres de combustibles.

```

GEOBEGIN
                                LATTICE UC2 bars
                                COMBINAT
RPP  1  -99999.0  99999.0  -99999.0  99999.0  -99999.0  99999.0
RPP  2  -10000.0 10000.0  -10000.0 10000.0  -10000.0 10000.0
RCC  3      0.0      0.0      0.0      0.0      0.0      4.0
      2.0
RCC  4      0.0      0.0      1.0      0.0      0.0      2.0
      1.0
RCC  5     -5.0      0.0      0.0      0.0      0.0      4.0
      2.0
RCC  6     -5.0      0.0      1.0      0.0      0.0      2.0
      1.0
RCC  7      0.0     -5.0      0.0      0.0      0.0      4.0
      2.0
RCC  8      0.0     -5.0      1.0      0.0      0.0      2.0
      1.0
RCC  9     -5.0     -5.0      0.0      0.0      0.0      4.0
      2.0
RCC 10     -5.0     -5.0      1.0      0.0      0.0      2.0
      1.0
RCC 11      5.0      0.0      0.0      0.0      0.0      4.0
      2.0
RCC 12      5.0      0.0      1.0      0.0      0.0      2.0
      1.0
RCC 13      5.0     -5.0      0.0      0.0      0.0      4.0
      2.0
RCC 14      5.0     -5.0      1.0      0.0      0.0      2.0
      1.0
RCC 15      5.0      5.0      0.0      0.0      0.0      4.0
      2.0
RCC 16      5.0      5.0      1.0      0.0      0.0      2.0
      1.0
RCC 17      0.0      5.0      0.0      0.0      0.0      4.0
      2.0
RCC 18      0.0      5.0      1.0      0.0      0.0      2.0
      1.0
RCC 19     -5.0      5.0      0.0      0.0      0.0      4.0
      2.0
RCC 20     -5.0      5.0      1.0      0.0      0.0      2.0
      1.0
RPP 21    -10.0     +10.0     -10.0     +10.0     -10.0     +10.0
END
*XAAA.....OR.....OR.....OR.....OR.....OR.....OR.....OR.....OR.....OR.....
* blackhole
  001      +1      -2
* vide
  002      +2     -21
* original
  003      +3      -4
* intérieur

```

```

004          +4
* replique1
005          +5      -6
006          +6
* replique2
007          +7      -8
008          +8
* replique3
009          +9      -10
010         +10
* replique4
011         +11      -12
012         +12
* replique5
013         +13      -14
014         +14
* replique6
015         +15      -16
016         +16
* replique7
017         +17      -18
018         +18
* replique8
019         +19      -20
020         +20
* conteneur
021         +21      -3      -5      -7      -9      -11      -13      -15      -17
          -19
END
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
LATTICE      005.      006.      1.      2.      Rot#1
LATTICE      007.      008.      3.      4.      Rot#2
LATTICE      009.      010.      5.      6.      Rot#3
LATTICE      011.      012.      7.      8.      Rot#4
LATTICE      013.      014.      9.      10.     Rot#5
LATTICE      015.      016.     11.     12.     Rot#6
LATTICE      017.      018.     13.     14.     Rot#7
LATTICE      019.      020.     15.     16.     Rot#8
GEOEND
* define the transformation for lattice cell
*
rot. numb.   theta   phi      dx      dy      dz
ROT-DEFI    1.      0.      0.      +5.     0.     0.
ROT-DEFI    2.      0.      0.      0.      +5.     0.
ROT-DEFI    3.      0.      0.      +5.     +5.     0.
ROT-DEFI    4.      0.      0.      -5.     0.     0.
ROT-DEFI    5.      0.      0.      -5.     +5.     0.
ROT-DEFI    6.      0.      0.      -5.     -5.     0.
ROT-DEFI    7.      0.      0.      0.      -5.     0.
ROT-DEFI    8.      0.      0.      +5.     -5.     0.
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
MATERIAL    1.0      1.0079  0.0000899  3.0      HYDROGEN
MATERIAL    6.0      12.011  2.0      4.0      CARBON
MATERIAL    7.0      14.0067 0.0012506  5.0      NITROGEN
MATERIAL    8.0      15.9994 0.001429  6.0      OXYGEN
MATERIAL    11.0     22.9898 0.971     7.0      SODIUM
MATERIAL    12.0     24.305  1.738    8.0      MAGNESIU
MATERIAL    13.0     26.982  2.6989  9.0      ALUMINUM
MATERIAL    14.0     28.0855 2.330    10.0     SILICON
MATERIAL    16.0     32.0650 1.960    11.0     SULFUR
MATERIAL    18.0     39.948  0.0016629 12.0     ARGON
MATERIAL    19.0     39.0983 0.862    13.0     POTASSIU
MATERIAL    20.0     40.0780 1.550    14.0     CALCIUM
MATERIAL    24.0     51.9961 7.140    15.0     CHROMIUM
MATERIAL    26.0     55.847  7.877    16.0     IRON
MATERIAL    28.0     58.6934 8.908    17.0     NICKEL
MATERIAL    29.0     63.546  8.92     22.0     COPPER

```

```

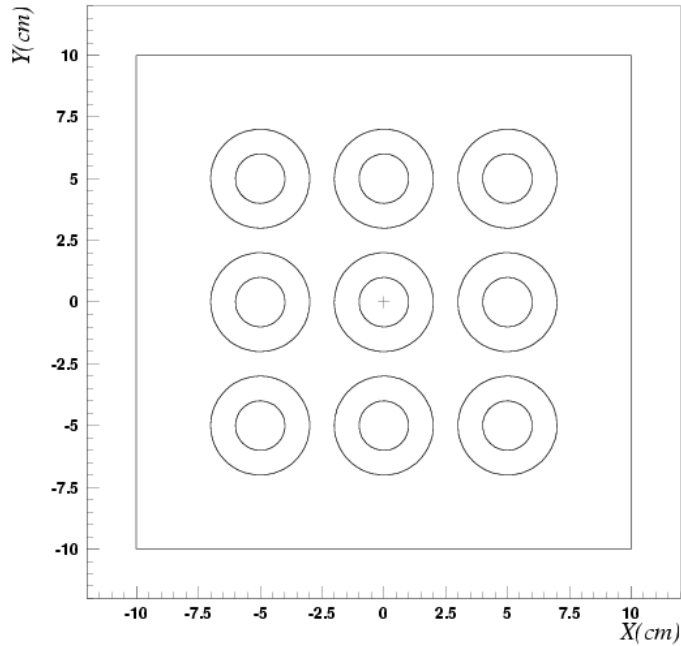
MATERIAL 82.0      207.2      11.35      18.0
MATERIAL 92.0      238.02891  19.05      24.0
MATERIAL 73.0      180.9479    16.654     26.0
MATERIAL 32.0      72.64       5.32       27.0
*23456789 123456789 123456789 123456789 123456789 123456789 123456789
MATERIAL 7.84      19.0
COMPOUND -0.78000    16.0        -0.15000   15.0        -0.07000   17.0
*
MATERIAL 0.001205  20.0
COMPOUND -0.75500    5.0         -0.23200   6.0         -0.01300   12.0
*
MATERIAL 2.35      21.0
COMPOUND -0.00560    3.0         -0.49830   6.0         -0.01710   7.0
COMPOUND -0.00240    8.0         -0.04560   9.0         -0.31580   10.0
COMPOUND -0.00120    11.0        -0.01920   13.0        -0.08260   14.0
COMPOUND -0.01220    16.0
*
MATERIAL 2.32      23.0
COMPOUND 1.0      10.0        2.0      6.0
*
MATERIAL 2.20      25.0
COMPOUND 1.0      24.0        2.0      4.0
*
MATERIAL 1.0      28.0
COMPOUND 2.0      3.0         1.0      1.0      6.0
*
*...+...1....+...2....+...3....+...4....+...5....+...6....+...7...
* External Black Hole
ASSIGNMAT 1.0      1.0
* Vacuum
ASSIGNMAT 2.0      2.0
* air
ASSIGNMAT 20.0     21.0
* carbone
ASSIGNMAT 4.0      3.0
*ASSIGNMAT 4.0     3.0        19.0      2.0

* uc2
ASSIGNMAT 25.0     4.0
*ASSIGNMAT 25.0     4.0        20.0      2.0
***
    
```

La région 21 est le « conteneur ».

Les régions 3 et 4 constituent la cellule de base, les autres régions constituent les 8 répliques (cartes « lattice ») dont les centres sont respectivement décalés (cartes « rot-defi » en (x,y) de (-5,0), (0,-5), (-5,-5),(+5,0),(+5,-5),(+5,+5),(0,+5),(-5,+5) par rapport au centre (0,0) de la cellule de base. Et on voit bien dans les valeurs WHAT(4,5) des cartes **ROT-DEFI** qu'il s'agit des décalages à effectuer à partir des centres des copies vers le centre de la cellule de base.

Ceci donne la géométrie suivante, plan (x,y), z=2.0 :



F.3. Exemple 2

Cet exemple a été proposé sur le fil de discussion FLUKA par Francesco Cerruti, du comité scientifique de **FLUKA**.

Il s'agit de répliquer 3 fois une cible constituée de 3 parties en carbone, aluminium et plomb. Les répliques sont situées à 10, 20, 30 cm de la cellule de base.

```

TITLE
Lattice Example
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7..
DEFAULTS                                     NEW-DEFA
BEAM           -3.5 -0.082425   -1.7    0.0    0.0    1.0PROTON
BEAMPOS        0.0    0.0    -1.0    0.0    0.0    0.0
*...+...1...+...2...+...3...+...4...+...5...+...6...+...7..
GEOBEGIN                                           COMBNAME
  0  0          Cylindrical Target
**AAA*IIII + + + + +
SPH BLK      0.0 0.0 0.0 10000.
* vacuum box
RPP VOI      -1000. 1000. -1000. 1000. -1000. 1000.
* target
ZCC TARG     0.0 0.0 5.0
* planes limiting the target
XYP ZTlow    0.0
XYP ZThigh   10.
**
* planes segmenting the target
XYP T1seg    1.
XYP T2seg    3.
**
* planes for lattice
XYP Zlat1    20.
XYP Zlat2    30.
XYP Zlat3    40.
  END
* Regions
* Black Hole
BLKHOLE      5 +BLK -VOI
* Vacuum around
VAC          5 +VOI - ( +TARG -ZTlow +Zlat3 )

```

```

* Target seg. 1
TARGS1      5  +TARG -ZTlow +T1seg
* Target seg. 2
TARGS2      5  +TARG -T1seg +T2seg
* Target seg. 3
TARGS3      5  +TARG -T2seg +ZThigh
* Target replicas
TARGRP1     5  +TARG -ZThigh +Zlat1
TARGRP2     5  +TARG -Zlat1 +Zlat2
TARGRP3     5  +TARG -Zlat2 +Zlat3
      END
* define region TARGRP1 as a replica, with name Target2, associated
* to roto-traslation number 1
LATTICE      TARGRP1                Target2                Rot# 1
LATTICE      TARGRP2                Target3                Rot# 2
LATTICE      TARGRP3                Target4                Rot# 3
*AAAAAAAx_+_____+_____+_____+_____+_____+_____+
GEOEND
*...+....1....+....2....+....3....+....4....+....5....+....6....+....7....+....8
ASSIGNMA     BLCKHOLE      BLKHOLE
ASSIGNMA     CARBON        TARGS1
ASSIGNMA     ALUMINUM     TARGS2
ASSIGNMA     LEAD          TARGS3
ASSIGNMA     VACUUM       VAC
*AAAAAAAx_+_____+_____+_____+_____+_____+_____+
*define the transformation for lattice cell Target2
*
      rot.numb.   theta   phi     dx     dy     dz
ROT-DEFI      1.0     0.0    0.0    0.0    0.0   -10.0
ROT-DEFI      2.0     0.0    0.0    0.0    0.0   -20.0
ROT-DEFI      3.0     0.0    0.0    0.0    0.0   -30.0
*
*
PLOTGEOM      0.0     0.0    0.0    0.0    1.0    0.0
              Cible plan (z,y)
              0.0   -10.0   -10.0    0.0   10.0   50.0
              0.0    1.0    0.0    0.0    0.0    1.0
              1.0    1.0    0.0    0.0
*
USRBIN        10.  ALL-PART   -80.    10.    10.    50.EWT74
USRBIN        -10. -10.    -10.   100.  100.  100.&
*
RANDOMIZ       1.0
START         10000.0
STOP
    
```

Le conteneur est le cylindre infini en z coupé en $z = 0.0$ et $z = 40.0$, TARG.

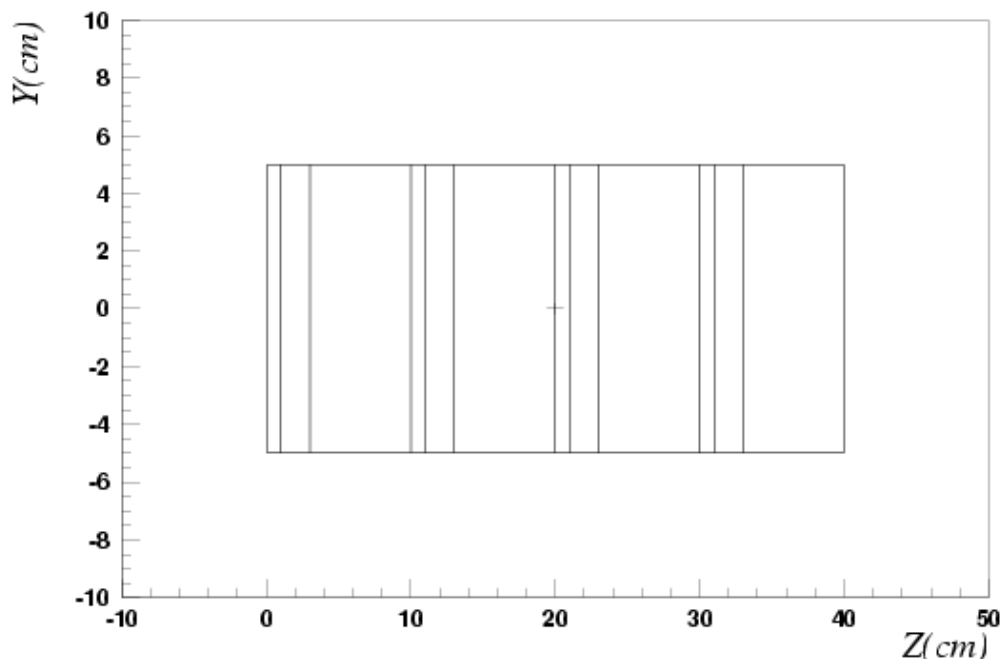
Les constituants de la cellule de base (comprise entre $z = 0.0$ et $z = 10.0$) sont TARGS1, TARGS2, et TARGS3.

Les répliques sont respectivement comprises entre $z = 10.0$ et 20.0 , $z = 20.0$ et 30.0 , $z = 30.0$ et 40.0 .

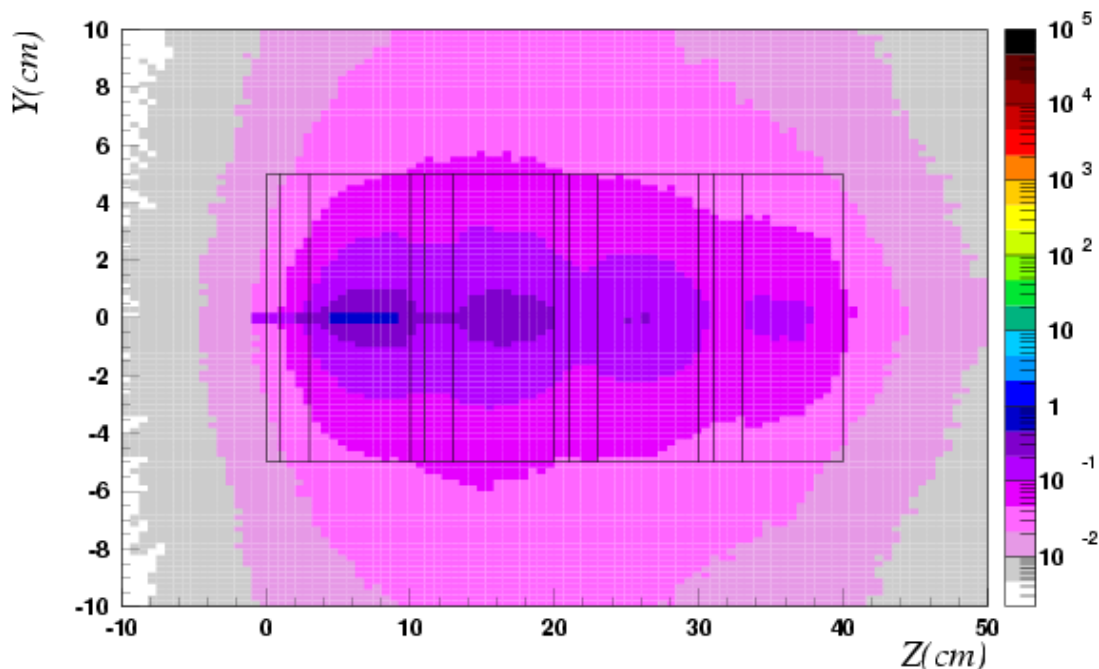
Les cartes **LATTICE** indiquent que la première réplique aura pour nom de cellule Target2 et est que la transformation 1 lui est appliquée. La seconde réplique est nommée Target3, elle est associée à la transformation 2 et la troisième réplique, Target4 est transformée selon Rot# 3.

Les cartes **ROT-DEFI** décrivent ces transformations, il s'agit de translations selon l'axe z, de respectivement $+10.0$, $+20.0$, $+30.0$ par rapport à la cellule de base (d'où, encore une fois, le signe -, cf. **G.1. remarque n°2**).

La géométrie dans le plan (z,y) avec $x = 0.0$ est la suivante :



Pour des raisons de tests, j'ai ajouté un estimateur **USRBIN**, calculant la fluence des toutes les particules, totale donc. Les résultats (exprimés en $part/cm^2/proton\ incident$) pour une simulation d'un cycle de 10 000 protons figurent sur le graphe suivant :



G. Commandes UNIX et routines fortran utiles

- Décompresser :

archive.tar : > **tar xf archive.tar**

archive.tgz : > **tar xzvf archive.tgz**

archive.tar.bz2 : > **bunzip2 -k archive.tar.bz2 => archive.tar**

archive.bz2 : > **tar xvfz archive.bz2**

- Compresser :

archive en archive.tar.gz : > **tar -czf archive.tar.gz archive**

archive en archive.tar.bz2 : > **tar cvjf archive.tar.bz2 archive**

- Compilation :

Routine fluka, exemple source.f : > **fff source.f** donne **source.o**

Lier cette routine à un exécutable nommé myfluka : > **lfluka -o myfluka -m fluka source.o**

Compiler un fichier fortran ordinaire : > **g77 -o test test.f** donne un exécutable **test**

> **g77 -c test.f** donne un fichier objet **test.o**

- Exécution du fichier toto.inp avec l'exécutable myfluka sur cinq cycles :

> **rfluka -e myfluka -N0 -M5 toto**

- Arrêter une simulation non terminée de **FLUKA** en terminant proprement le cycle en cours : se placer dans le répertoire temporaire **fluka_xxxx**, taper **touch rfluka.stop**

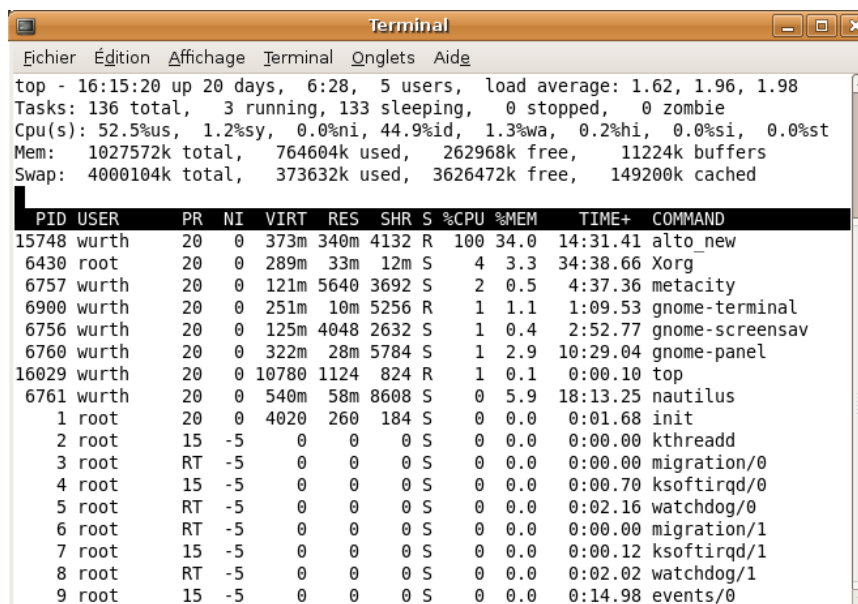
- Tuer un processus :

Taper **top** dans un terminal, cela affiche les il est très facile de voir quel processus semble prendre le plus de puissance CPU surtout mais parfois de RAM.

Pour tuer le processus, tapez sur la touche "k" pour kill, il vous sera demandé le PID du programme facilement visible dans la liste, tapez le, validez et le tour est joué. Si c'est un programme système alors il est possible que les droits vous soit refusés.

Exemple : je veux « tuer » le processus dont le PID est 7102, je tape :

> **kill 7102** (enter)



```

Terminal
-----
Fichier  Édition  Affichage  Terminal  Onglets  Aide

top - 16:15:20 up 20 days,  6:28,  5 users,  load average: 1.62, 1.96, 1.98
Tasks: 136 total,  3 running, 133 sleeping,  0 stopped,  0 zombie
Cpu(s): 52.5%us,  1.2%sy,  0.0%ni, 44.9%id,  1.3%wa,  0.2%hi,  0.0%si,  0.0%st
Mem:   1027572k total,  764604k used,  262968k free,  11224k buffers
Swap:  4000104k total,  373632k used,  3626472k free,  149200k cached

  PID USER   PR   NI  VIRT  RES  SHR  S  %CPU  %MEM    TIME+  COMMAND
 15748 wurth   20    0  373m 340m 4132  R  100  34.0   14:31.41 alto_new
  6430 root    20    0  289m  33m  12m  S   4   3.3   34:38.66 Xorg
  6757 wurth   20    0  121m 5640 3692  S   2   0.5   4:37.36 metacity
  6900 wurth   20    0  251m  10m 5256  R   1   1.1   1:09.53 gnome-terminal
  6756 wurth   20    0  125m 4048 2632  S   1   0.4   2:52.77 gnome-screensav
  6760 wurth   20    0  322m  28m 5784  S   1   2.9  10:29.04 gnome-panel
16029 wurth   20    0 10780 1124  824  R   1   0.1   0:00.10 top
  6761 wurth   20    0  540m  58m 8608  S   0   5.9  18:13.25 nautilus
    1 root    20    0  4020  260  184  S   0   0.0   0:01.68 init
    2 root    15  -5     0     0     0  S   0   0.0   0:00.00 kthreadd
    3 root    RT  -5     0     0     0  S   0   0.0   0:00.00 migration/0
    4 root    15  -5     0     0     0  S   0   0.0   0:00.70 ksoftirqd/0
    5 root    RT  -5     0     0     0  S   0   0.0   0:02.16 watchdog/0
    6 root    RT  -5     0     0     0  S   0   0.0   0:00.00 migration/1
    7 root    15  -5     0     0     0  S   0   0.0   0:00.12 ksoftirqd/1
    8 root    RT  -5     0     0     0  S   0   0.0   0:02.02 watchdog/1
    9 root    15  -5     0     0     0  S   0   0.0   0:14.98 events/0
    
```


H. Un peu de physique

La définition de la fluence définie par l'ICRU (International Commission on Radiation Units) : la fluence ϕ est le quotient de dN , le nombre de particules incidentes par da , la section

efficace d'une sphère : $\phi = \frac{dN}{da}$

Mais attention, cette définition ne doit pas induire en erreur, le sens physique de la fluence n'est pas celui d'un flux de particules par unité de surface !

La signification physique de la fluence est celle de la densité de parcours de la particule dans un volume infinitésimal.

$$\phi = \lim_{\Delta V \rightarrow 0} \frac{\sum_i L_i}{\Delta V} \quad \text{Unité : [cm X cm}^{-3} = \text{cm}^{-2}\text{]}$$

Si dans la première définition dA est la surface de la sphère de l'ICRU de section efficace da , alors $dA = 4 * da$.

Pour un corps convexe, la longueur moyenne de corde est égale à : $\bar{L} = \frac{4V}{A}$

Et par conséquent, d'après la définition de l'ICRU, $\phi = \frac{dN}{da} = \frac{\bar{L}dN}{dV}$

$\bar{L}dN$ est la longueur de corde totale pour les N particules traversant la sphère.

Nous avons vu que les deux définitions sont équivalentes, celle de l'ICRU cache le fait que la fluence est la mesure de la concentration de parcours des particules dans un volume infinitésimal autour d'un point de l'espace. Plus la distance parcourue dans ce volume est importante, plus le nombre d'interactions l'est !

Pour estimer la fluence à une limite (interface), il ne faut pas compter le nombre de particules la traversant (définition du courant), mais introduire un facteur $\frac{\omega}{\cos \theta}$ pour chaque particule, où ω est le poids statistique (*weight*) de la particule considérée et θ l'angle entre la direction de la particule et la normale à la surface au point d'interaction. Généralement, dans nos études simples sans biaisage, le poids statistique sera considéré comme valant 1.

Le facteur à introduire sera donc : $\frac{1}{\cos \theta}$

Considérons un volume ΔV de base A égale à la surface traversée et d'épaisseur infinitésimale Δt .

Le parcours de la particule dans le volume ΔV est $\frac{\Delta t}{\cos \theta}$: trigonométrie classique, dans le triangle rectangle ainsi formé, l'hypoténuse est le parcours de la particule dans le volume, le côté adjacent est l'épaisseur infinitésimale.

La contribution de la particule à la densité de trajectoires est :

$$\frac{\Delta t}{\cos \theta * \Delta V} = \frac{\Delta t}{\cos \theta * A * \Delta t} = \frac{1}{\cos \theta * A}$$

Remarques :

- Le courant est égal à la fluence seulement si toutes les particules sont parallèles et perpendiculaires à la surface. En cas de distribution angulaire isotrope, la fluence vaut le double du courant.
- La fluence est indépendante de l'orientation de la surface, mais pas le courant !

Les estimateurs et la quantité calculée :

USRTRACK, USRCOLL : dérivée de la fluence $\frac{d\phi}{dE}$ moyenne d'un type donné de particule ou famille de particules dans une région donnée.

USRBDX : dérivée seconde de la fluence $\frac{d\phi}{dEd\Omega}$ moyenne d'un type donné de particule ou famille de particule sur une surface donnée.

USRBIN : distribution spatiale de l'énergie déposée (dose) ou fluence totale ou densité d'interactions inélastiques ou transfert de quantité de mouvement dans une forme géométrique régulière (cylindrique ou cartésienne) définie par l'utilisateur.

USRYIELD : dérivée seconde d'une quantité de particules s'échappant d'une surface. La distribution peut être en fonction de l'énergie ou de l'angle d'incidence, mais également en fonction de quantités plus exotiques.

SCORE : énergie déposée et densité d'interactions inélastiques dans chaque région.

RESNUCLEi : nucléides résiduels dans une région donnée.

I. Carte DETECT : fichiers utiles

I.1. Récupération de données de sortie : DETOUTPUT

```

*$ CREATE DETOUTPUT.FOR
*$ COPY DETOUTPUT
*
PROGRAM DETOUTPUT

CHARACTER FILE*80, RUNTIT*80, RUNTIM*32, CHNAME*10

*===== detect =====*
*=====*
*      S. Wurth v1 12/07/2007 inspired by      *
*      routine usbrea.f by A. Ferrari & A. Fasso' *
*
*      RUNTIT: title of the job                *
*      RUNTIM: time of the job                 *
*      WEIPRI: total weight of the primary particles *
*      NCASE: number of primary particles      *
*      NEDT: detector number                  *
*      CHNAME: detector name                  *
*      NBIN: number of energy bins            *
*      EMIN : minimum total energy            *
*      EBIN : width of energy bin             *
*      ECUT : cut-off energy for the signal   *
*      NBIN values IV(I) : they are the spectrum channels, or energy bins *
*
*=====*
INTEGER*4 NCASE, NDET, NBIN, IV(1024)
REAL EMIN, EBIN, ECUT
CHARACTER CHFORM*20
PARAMETER (NRGNMX = 10)
PARAMETER (NDTCMX = 10)
PARAMETER (NSCRMX = 10)
PARAMETER (NDTCM2 = 10)
PARAMETER (NDTBIN = 1024)
PARAMETER (MXXRGN = 10)
CHARACTER*10 TITDET, TITSCO
LOGICAL LDTCTR
COMMON /DETCT/ EDTMIN(NDTCMX), EDTBIN(NDTCMX), EDTCUT(NDTCMX),
&      KDTREG(NRGNMX,NDTCMX), KDTDET(NDTCMX,NSCRMX),
&      NDTSCO, NDTDET, LDTCTR, IDTREG(MXXRGN),
&      DTSCD(NSCRMX)
COMMON /DETC/ TITDET(NDTCMX), TITSCO(NSCRMX)
PARAMETER (NDTCM2 = 10)
COMMON /DETLOC/ ACCUMP(NDTCM2), ACCUMN(NDTCM2),
&      ICOINC(NDTCM2), NCLAS
DIMENSION JB(NRGNMX), LIO(NRGNMX)
LOGICAL LOPEN, LSTATI, LREFLX, LREFLY, LRFLZ
MIO = 0
NCTOT = 0
WCTOT = 0.E+00
LSTATI = .FALSE.
WRITE(*, '(' Type the input file name: ', $)')
READ (*, '(A)') FILE
LQ = LNNBLN(FILE)
IF (LQ .LE. 0) GO TO 500
OPEN (UNIT=1, FILE=FILE, STATUS='OLD', FORM='UNFORMATTED')
IRECRD = 0
READ (1) RUNTIT, RUNTIM, WEIRPI, NCASE
IRECRD = IRECRD + 1
WRITE(*, *) RUNTIT, RUNTIM, WEIPRI, NCASE
READ (1) NDET, CHNAME, NBIN, EMIN, EBIN, ECUT
IRECRD = IRECRD + 1
IRECRD = 0
READ(1) (IV(I), I=1, NBIN)
IRECRD = IRECRD + 1
DO 400 II = 1, NBIN
*      WRITE(*, *) IV(II)
400 CONTINUE
IRECRD = 0
500 CONTINUE
CLOSE (UNIT=1)
    
```

```

WRITE(*,'(' Type the output file name: ',,$)')
READ (*,'(A)')FILE
OPEN (UNIT=1,FILE=FILE,STATUS='UNKNOWN',FORM='FORMATTED')
WRITE (UNIT=1,
& FMT='(/,25X,'***** Title of the run: ',A80, ' ' *****')')
& RUNTIT
WRITE (UNIT=1,
& FMT='(/,15X, '***** Detector number: ',I5, ' ' *****'
& /,15X, '***** Detector name: ',A10, ' ' *****'
& /,15X, '***** Number of energy bins: ',I6, ' ' *****'
& /,15X, '***** Minimum total energy (GeV): ',1P,E11.4,
& ' ' *****'/,15X, '***** Width of each energy bin (GeV): ',
& 1P,E11.4, ' ' *****'
& /,15X, '***** Cutoff energy for the signal (GeV): '
& ,1P,E11.4, ' ' *****')')
& NDET,CHNAME,NBIN,EMIN,EBIN,ECUT
900 CONTINUE
WRITE (UNIT=1,
& FMT='(/15X, '***** The spectrum follows: *****',/)'
DO 1000 I = 1, NBIN
WRITE (UNIT=1,
& FMT='(I6 ':'',I8)')
& I,IV(I)
1000 CONTINUE
STOP
END

*$ CREATE LNNBLN.FOR
*$ COPY LNNBLN
*
*==== Lnnbln =====*
*
INTEGER FUNCTION LNNBLN (CARD)
*
* CHARACTER CARD*(*)
*
LENGTH = LEN (CARD)
DO 100 LQ = LENGTH, 1, -1
IF ( CARD (LQ:LQ).NE.' ') GO TO 200
100 CONTINUE
LQ = 0
200 CONTINUE
LNNBLN = LQ
RETURN
*==== End of function Lnnbln =====*
END

```

I.2. Fichier d'entrée Ge-CANBERRA et source de type SG500

```

TITLE
Sampling Ge detector
*GLOBAL 4.0
DEFAULTS PRECISIO
*23456789 123456789 123456789 123456789 123456789 123456789 123456789
*BEAM -59.54E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -88.03E-6 1.E5 4.7 0.0 -1.0PHOTON
BEAM -122.06E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -136.47E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -165.86E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -255.13E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -320.08E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -391.70E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -514.00E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -661.66E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -898.04E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -1173.23E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -1332.49E-6 1.E5 4.7 0.0 -1.0PHOTON
*BEAM -1836.05E-6 1.E5 4.7 0.0 -1.0PHOTON
BEAMPOS 2.0 2.0 -6.0
*****Energy(keV)***Zmin*****Zmax*****Radius*****
*23456789 123456789 123456789 123456789 123456789 123456789 123456789
SOURCE

```

GEOBEGIN

COMBINAT

		Ge detector					
RPP	1	-99999.0	+99999.0	-99999.0	+99999.0	-99999.0	+99999.0
RPP	2	-9000.0	+9000.0	-9000.0	+9000.0	-9000.0	+9000.0
RPP	3	-200.0	+200.0	-200.0	+200.0	-200.0	+200.0
RCC	4	0.0	0.0	-8.2	0.0	0.0	8.35
		4.9					
RCC	5	0.0	0.0	-8.025	0.0	0.0	8.0
		4.7					
RCC	6	0.0	0.0	-7.225	0.0	0.0	7.2
		4.7					
RCC	7	0.0	0.0	-10.46	0.0	0.0	2.26
		2.7					
RCC	8	0.0	0.0	-10.26	0.0	0.0	2.75
		2.5					
RCC	9	0.0	0.0	-10.71	0.0	0.0	1.74
		3.0					
RCC	10	0.0	0.0	-10.56	0.0	0.0	1.72
		2.75					
RCC	11	0.0	0.0	0.15	0.0	0.0	4.25
		4.3					
RCC	12	0.0	0.0	0.35	0.0	0.0	3.85
		4.0					
RCC	13	0.0	0.0	4.4	0.0	0.0	23.6
		3.75					
RCC	14	0.0	0.0	15.3	0.0	0.0	12.7
		4.0					
RCC	15	0.0	0.0	4.5	0.0	0.0	23.4
		3.65					
RCC	16	0.0	0.0	5.0	0.0	0.0	6.4
		3.1					
RCC	17	0.0	0.0	5.115	0.0	0.0	6.13
		2.86					
RCC	18	0.0	0.0	6.65	0.0	0.0	4.75
		0.8					
RCC	19	0.0	0.0	6.655	0.0	0.0	5.00
		0.795					
RCC	20	0.0	0.0	6.65	0.0	0.0	6.35
		0.15					
RCC	21	0.0	0.0	12.5	0.0	0.0	0.4
		0.8					
RCC	22	0.0	0.0	12.8	0.0	0.0	0.7
		1.2					
RCC	23	0.0	0.0	13.4	0.0	0.0	0.3
		3.0					
RCC	24	0.0	0.0	13.7	0.0	0.0	0.6
		2.0					
RCC	25	0.0	0.0	13.7	0.0	0.0	0.6
		1.8					
RCC	26	0.0	0.0	14.3	0.0	0.0	0.4
		2.3					

END
 * black hole
 1 5 1 -2
 * vide
 2 5 2 -3
 * air
 3 5 3 -4 -7 -9 -11 -13 -14
 * Corps SG500 polystyr_polyeth
 4 5 4 -5 -6 -8
 * Interieur air
 5 5 5 -6 -8
 * Interieur résine
 6 5 6
 * Colerette exterieur
 7 5 7 -8
 * Air colerette
 8 5 8
 * Bouchon exterieur
 9 5 9 -10
 * Air bouchon
 10 5 10 -7 -8
 * Support plexi
 11 5 11 -12
 * Air

```

    12 5 12
* Enveloppe ALU
    13 5 13 -15
* Surplus
    14 5 14 -13 -15
* Vide
    15 5 15 -16 -19 -20 -21 -22 -23 -24 -26
* Bloc Ge
    16 5 16 -17 -20
* Bloc actif
    17 5 17 -18 -20
* Trou interne avec dead layer
    18 5 18 -19 -20
* Trou pour conducteur vide
    19 5 19 -20
* Conducteur interne alu
    20 5 20
* Petit socle alu
    21 5 21 -20
* Socle moyen alu
    22 5 22 -21
* Grand socle alu
    23 5 23 -22
* Piliers
    24 5 24 -25
* Vide
    25 5 25
* Dernier socle
    26 5 26
    END
GEOEND
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
* List of FLUKA pre-defined materials
* 1.0 = BLCKHOLE
* 2.0 = VACCUUM
* 3.0 = HYDROGEN
* 4.0 = HELIUM
* 5.0 = BERYLLIU
* 6.0 = CARBON
* 7.0 = NITROGEN
* 8.0 = OXYGEN
* 9.0 = MAGNESIU
* 10.0 = ALUMINUM
* 11.0 = IRON
* 12.0 = COPPER
* 13.0 = SILVER
* 14.0 = SILICON
* 15.0 = GOLD
* 16.0 = MERCURY
* 17.0 = LEAD
* 18.0 = TANTALUM
* 19.0 = SODIUM
* 20.0 = ARGON
* 21.0 = CALCIUM
* 22.0 = TIN
* 23.0 = TUNGSTEN
* 24.0 = TITANUM
* 25.0 = NICKEL
*
MATERIAL 16.0 32.0650 1.960 26.0 SULFUR
MATERIAL 19.0 39.0983 0.862 27.0 POTASSIU
MATERIAL 24.0 51.9961 7.140 28.0 CHROMIUM
MATERIAL 53.0 126.90447 4.94 29.0 IODINE
MATERIAL 32.0 72.64 5.323 30.0 GERMANIU
MATERIAL 3.0 6.941 0.534 41.0 LITHIUM
MATERIAL 5.0 10.811 2.370 42.0 BORON
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
MATERIAL 7.84 31.0 INOX
COMPOUND -0.78000 IRON -0.15000 CHROMIUM -0.07000 NICKEL INOX
*
MATERIAL 0.001205 32.0 AIR
COMPOUND -0.75500 NITROGEN -0.23200 OXYGEN -0.01300 ARGON AIR
*
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
MATERIAL 2.35 33.0 CONCRETE
    
```

```

COMPOUND -0.00560 HYDROGEN -0.49830 OXYGEN -0.01710 SODIUM CONCRETE
COMPOUND -0.00240 MAGNESIU -0.04560 ALUMINUM -0.31580 SILICON CONCRETE
COMPOUND -0.00120 SULFUR -0.01920 POTASSIU -0.08260 CALCIUM CONCRETE
COMPOUND -0.01220 IRON CONCRETE
*
MATERIAL 3.67 34.0 NaI
COMPOUND 1.0 SODIUM 1.0 IODINE NaI
*
MATERIAL 1.273 35.0 MYLAR
COMPOUND 8.0 HYDROGEN 10.0 CARBON 4.0 OXYGEN MYLAR
*
MATERIAL 0.94 36.0 POLYETH
COMPOUND 14.4 HYDROGEN 85.6 CARBON POLYETH
*
MATERIAL 1.19 37.0 PLEXIGLA
COMPOUND 6.0 HYDROGEN 4.0 CARBON 2.0 OXYGEN PLEXIGLA
*
MATERIAL 1.15 38.0 EPOXY
COMPOUND 19.0 HYDROGEN 18.0 CARBON 3.0 OXYGEN EPOXY
*
MATERIAL 0.90 39.0 POLYPROP
COMPOUND 12.3 HYDROGEN 87.7 CARBON POLYPROP
*
MATERIAL 1.06 40.0 POLYSTYR
COMPOUND 7.74 HYDROGEN 92.3 CARBON POLYSTYR
*
* External Black Hole
ASSIGNMAT 1.0 1.0
* Vacuum
ASSIGNMAT 2.0 2.0
ASSIGNMAT 2.0 15.0
ASSIGNMAT 2.0 19.0
ASSIGNMAT 2.0 25.0
* Aluminium
ASSIGNMAT 10.0 13.0
ASSIGNMAT 10.0 14.0
ASSIGNMAT 10.0 20.0
ASSIGNMAT 10.0 21.0
ASSIGNMAT 10.0 22.0
ASSIGNMAT 10.0 23.0
ASSIGNMAT 10.0 24.0
ASSIGNMAT 10.0 26.0
* Germanium
ASSIGNMAT 30.0 16.0
ASSIGNMAT 30.0 17.0
ASSIGNMAT 30.0 18.0
* Air
ASSIGNMAT 32.0 3.0
ASSIGNMAT 32.0 5.0
ASSIGNMAT 32.0 8.0
ASSIGNMAT 32.0 10.0
ASSIGNMAT 32.0 12.0
* Mylar
*ASSIGNMAT 35.0 4.0
* Polyeth
*ASSIGNMAT 36.0 9.0
ASSIGNMAT 36.0 4.0
ASSIGNMAT 36.0 7.0
* Plexiglas
ASSIGNMAT 37.0 11.0
* Epoxy
ASSIGNMAT 38.0 6.0
* Polyprop
ASSIGNMAT 39.0 9.0
* Polystyr
*ASSIGNMAT 40.0 4.0
*ASSIGNMAT 40.0 7.0
*
*23456789 123456789 123456789 123456789 123456789 123456789 123456789 123456789
*PLOTGEOM 0.0 1.0 0.0 0.0 1.0 0.0
*
* detectors plan x,y
* -10.0 -1.0 0.0 10.0 12.0 0.0
* 0.0 1.0 0.0 1.0 0.0 0.0
* 1.0 1.0 0.0 0.0
*

```

```

*USERWEIG                      1.                      0.
EMFCUT      512.E-6      1.E-6      0.      1.      40.      PROD-CUT
EVENTYPE
PHYSICS      2.0                      EVAP
          * **** Scoring ****                      EVAPORAT
*23456789 123456789 123456789 123456789 123456789 123456789 123456789
DETECT      0.      0.E-5 1.024E-3                      1.      17.0HPGe
*DETECT      0.      1.0E-3 2.024E-3                      1.      17.0HPGe
*
*USRBIN      10.      7.      -50.      7.      7.      30.TEST
*USRBIN      -7.      -7.      -15.      100.      100.      100.&
*
*USRBIN      10.      210.      -55.      7.      7.      30.TEST
*USRBIN      -7.      -7.      -15.      100.      100.      100.&
*
*23456789 123456789 123456789 123456789 123456789 123456789 123456789
* Volume actif = (pi * 2.86^2 * 6.13)- (pi * 0.8^2 * 4.75) = 147.97
*USRTRACK      1.      7.0      -60.      17.      147.97      100.TRACKTES
*USRTRACK      2000.                      &
*
*SCORE      7.
*
RANDOMIZE      1.0
*23456789 *23456789 *23456789 *23456789 *23456789 *23456789 *23456789
*START      200000.
START      6.0E6
STOP

```

I.3. Fichier source de type SG500

En rouge, les parties modifiées par rapport à la version standard.

```

*$ CREATE SOURCE.FOR
*COPY SOURCE
*
*=== source =====*
*
SUBROUTINE SOURCE ( NOMORE )

INCLUDE '(DBLPRC)'
INCLUDE '(DIMPAR)'
INCLUDE '(IOUNIT)'

*
*-----*
*
* Copyright (C) 1990-2006 by Alfredo Ferrari & Paola Sala *
* All Rights Reserved. *
*
*
* New source for FLUKA9x-FLUKA200x: *
*
* Created on 07 january 1990 by Alfredo Ferrari & Paola Sala *
*                               Infn - Milan *
*
* Last change on 03-mar-06 by Alfredo Ferrari *
*
* This is just an example of a possible user written source routine. *
* note that the beam card still has some meaning - in the scoring the *
* maximum momentum used in deciding the binning is taken from the *
* beam momentum. Other beam card parameters are obsolete. *
*-----*
*
INCLUDE '(BEAMCM)'
INCLUDE '(FHEAVY)'
INCLUDE '(FLKSTK)'
INCLUDE '(IOIOCM)'
INCLUDE '(LTCLCM)'
INCLUDE '(PAPROP)'
INCLUDE '(SOURCM)'
INCLUDE '(SUMCOU)'
LOGICAL LFIRST
DOUBLE PRECISION RANDY

```



```

DOUBLE PRECISION RADIUS
DOUBLE PRECISION MYRHO
DOUBLE PRECISION ZMIN
DOUBLE PRECISION ZMAX
SAVE LFIRST
DATA LFIRST / .TRUE. /
*-----*
*
*           BASIC VERSION
*-----*
      NOMORE = 0
* +-----*
* | First call initializations:
* | IF ( LFIRST ) THEN
* |   *** The following 3 cards are mandatory ***
* |     TKESUM = ZERZER
* |     LFIRST = .FALSE.
* |     LUSSRC = .TRUE.
* |   *** User initialization ***
* |   END IF
* |
* +-----*
* | Push one source particle to the stack. Note that you could as well
* | push many but this way we reserve a maximum amount of space in the
* | stack for the secondaries to be generated
* | Npflka is the stack counter: of course any time source is called it
* | must be =0
* |     NPFLKA = NPFLKA + 1
* | Wt is the weight of the particle
* |     WTFLK (NPFLKA) = ONEONE
* |     WEIPRI = WEIPRI + WTFLK (NPFLKA)
* | Particle type (1=proton.....). Ijbeam is the type set by the BEAM
* | card
* +-----*
* | (Radioactive) isotope:
* | IF ( IJBEAM .EQ. -2 .AND. LRDBEA ) THEN
* |   IARES = IPROA
* |   IZRES = IPROZ
* |   IISRES = IPROM
* |   CALL STISBM ( IARES, IZRES, IISRES )
* |   IJHION = IPROZ * 1000 + IPROA
* |   IJHION = IJHION * 100 + KXHEAV
* |   IONID = IJHION
* |   CALL DCDION ( IONID )
* |   CALL SETION ( IONID )
* |
* +-----*
* | Heavy ion:
* | ELSE IF ( IJBEAM .EQ. -2 ) THEN
* |   IJHION = IPROZ * 1000 + IPROA
* |   IJHION = IJHION * 100 + KXHEAV
* |   IONID = IJHION
* |   CALL DCDION ( IONID )
* |   CALL SETION ( IONID )
* |   ILOFLK (NPFLKA) = IJHION
* | Flag this is prompt radiation
* |     LRADDC (NPFLKA) = .FALSE.
* |
* +-----*
* | Normal hadron:
* | ELSE
* |   IONID = IJBEAM
* |   ILOFLK (NPFLKA) = IJBEAM
* | Flag this is prompt radiation
* |     LRADDC (NPFLKA) = .FALSE.
* |   END IF
* |
* +-----*
* | From this point .....
* | Particle generation (1 for primaries)
* |     LOFLK (NPFLKA) = 1
* | User dependent flag:
* |     LOUSE (NPFLKA) = 0
* | User dependent spare variables:

```

```

DO 100 ISPR = 1, MKBMX1
    SPAREK (ISPR,NPFLKA) = ZERZER
100 CONTINUE
* User dependent spare flags:
DO 200 ISPR = 1, MKBMX2
    ISPARK (ISPR,NPFLKA) = 0
200 CONTINUE
* Save the track number of the stack particle:
    ISPARK (MKBMX2,NPFLKA) = NPFLKA
    NPARMA = NPARMA + 1
    NUNPAR (NPFLKA) = NPARMA
    NEVENT (NPFLKA) = 0
    DFNEAR (NPFLKA) = +ZERZER
* ... to this point: don't change anything
* Particle age (s)
    AGESTK (NPFLKA) = +ZERZER
    AKNSHR (NPFLKA) = -TWOTWO
* Group number for "low" energy neutrons, set to 0 anyway
    IGROUP (NPFLKA) = 0
* Kinetic energy of the particle (GeV)
    TKEFLK (NPFLKA) = SQRT ( PBEAM**2 + AM (IONID)**2 ) - AM (IONID)
* Particle momentum
    PMOFLK (NPFLKA) = PBEAM
* PMOFLK (NPFLKA) = SQRT ( TKEFLK (NPFLKA) * ( TKEFLK (NPFLKA)
* & + TWOTWO * AM (IONID) ) )
* Cosines (tx,ty,tz)
* Ajout : isotropy
    CALL RACO(TXX,TTY,TZZ)
    UBEAM = TXX
    VBEAM = TYY
    WBEAM = TZZ
    TXFLK (NPFLKA) = UBEAM
    TYFLK (NPFLKA) = VBEAM
    TZFLK (NPFLKA) = WBEAM
* TZFLK (NPFLKA) = SQRT ( ONEONE - TXFLK (NPFLKA)**2
* & - TYFLK (NPFLKA)**2 )
* Polarization cosines:
    TXPOL (NPFLKA) = -TWOTWO
    TYPOL (NPFLKA) = +ZERZER
    TZPOL (NPFLKA) = +ZERZER
* Particle coordinates
    CALL SFECFE(SINT,COST)
    RADIUS = 4.699999D0
    ZMIN = -7.224999D0
    ZMAX = -0.0250001D0
    RANDY = FLRNDM(XDUMMY)
    MYRHO = RADIUS * SQRT(RANDY)
    XBEAM = MYRHO * COST
    YBEAM = MYRHO * SINT
    ZBEAM = ZMIN - ((ZMIN - ZMAX)*FLRNDM(ZDUMMY))
    XFLK (NPFLKA) = XBEAM
    YFLK (NPFLKA) = YBEAM
    ZFLK (NPFLKA) = ZBEAM
* WRITE(lunout,*) XBEAM, YBEAM, ZBEAM
* Calculate the total kinetic energy of the primaries: don't change
IF ( ILOFLK (NPFLKA) .EQ. -2 .OR. ILOFLK (NPFLKA) .GT. 10000 )
& THEN
    TKESUM = TKESUM + TKEFLK (NPFLKA) * WTFLK (NPFLKA)
ELSE IF ( ILOFLK (NPFLKA) .NE. 0 ) THEN
    TKESUM = TKESUM + ( TKEFLK (NPFLKA) + AMDISC (ILOFLK(NPFLKA)) )
& * WTFLK (NPFLKA)
ELSE
    TKESUM = TKESUM + TKEFLK (NPFLKA) * WTFLK (NPFLKA)
END IF
RADDLY (NPFLKA) = ZERZER
* Here we ask for the region number of the hitting point.
* NREG (NPFLKA) = ...
* The following line makes the starting region search much more
* robust if particles are starting very close to a boundary:
    CALL GEOCRS ( TXFLK (NPFLKA), TYFLK (NPFLKA), TZFLK (NPFLKA) )
    CALL GEOREG ( XFLK (NPFLKA), YFLK (NPFLKA), ZFLK (NPFLKA),
    & NRGFLK(NPFLKA), IDISC )
* Do not change these cards:
    CALL GEOHSM ( NHPNT (NPFLKA), 1, -11, MLATTC )
    NLATTC (NPFLKA) = MLATTC
    
```

```
CMPTH (NPFLKA) = ZERZER
CALL SOEVSV
RETURN
*==== End of subroutine Source =====*
END
```

J. Routine FLUSCW pour la conversion fluence/dose dans le cas des électrons

```
*$ CREATE FLUSCW.FOR
*COPY FLUSCW
*
*==== fluscw =====*
*
  DOUBLE PRECISION FUNCTION FLUSCW ( IJ      , PLA      , TXX      , TYY      ,
&                                     TZZ      , WEE      , XX       , YY       ,
&                                     ZZ       , NREG     , IOLREG , LLO      ,
&                                     NSURF )

  INCLUDE '(DBLPRC)'
  INCLUDE '(DIMPAR)'
  INCLUDE '(IOUNIT)'
  DIMENSION ENEDGE (1:25)
  DIMENSION FACTOR (1:25)
  DOUBLE PRECISION PLAMEV, FACTOR, ENEDGE
  INTEGER K

*-----*
*
* Copyright (C) 1989-2005      by      Alfredo Ferrari & Paola Sala
* All Rights Reserved.
*
* New version of Fluscw for FLUKA9x-FLUKA200x:
*
* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
* !!! This is a completely dummy routine for Fluka9x/200x. !!!
* !!! The name has been kept the same as for older Fluka !!!
* !!! versions for back-compatibility, even though Fluscw !!!
* !!! is applied only to estimators which didn't exist be- !!!
* !!! fore Fluka89. !!!
* !!! User developed versions can be used for weighting !!!
* !!! flux-like quantities at runtime !!!
* !!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!!
*
* Input variables:
*
*      Ij = (generalized) particle code (Paprop numbering)
*      Pla = particle laboratory momentum (GeV/c) (if > 0),
*           or kinetic energy (GeV) (if <0 )
*      Txx,yy,zz = particle direction cosines
*      Wee = particle weight
*      Xx,Yy,Zz = position
*      Nreg = (new) region number
*      Iolreg = (old) region number
*      Llo = particle generation
*      Nsurf = transport flag (ignore!)
*
* Output variables:
*
*      Fluscw = factor the scored amount will be multiplied by
*      Lsczer = logical flag, if true no amount will be scored
*              regardless of Fluscw
*
* Useful variables (common SCOHLP):
*
* Flux like binnings/estimators (Fluscw):
*      ISCRNG = 1 --> Boundary crossing estimator
*      ISCRNG = 2 --> Track length binning
*      ISCRNG = 3 --> Track length estimator
*      ISCRNG = 4 --> Collision density estimator
*      ISCRNG = 5 --> Yield estimator
*      JSCRNG = # of the binning/estimator
*-----*
*
  INCLUDE '(SCOHLP)'
```

```
*
* conversion fluence / H'(0,07,0°) pour electrons en pSv.cmq
* data de l'AIEA
  FLUSCW = ONEONE
  LSCZER = .FALSE.
  PLAMEV = -PLA * 1.d3
  DATA ENEDGE /
& 0.07d0,
& 0.08d0,
& 0.09d0,
& 0.10d0,
& 0.1125d0,
& 0.125d0,
& 0.15d0,
& 0.2d0,
& 0.3d0,
& 0.4d0,
& 0.5d0,
& 0.6d0,
& 0.7d0,
& 0.8d0,
& 1.0d0,
& 1.25d0,
& 1.5d0,
& 1.75d0,
& 2.0d0,
& 2.5d0,
& 3.0d0,
& 3.5d0,
& 4.0d0,
& 8.0d0,
& 1.0d1/
  DATA FACTOR /
& 221.0d0,
& 1056.0d0,
& 1527.0d0,
& 1661.0d0,
& 1627.0d0,
& 1513.0d0,
& 1229.0d0,
& 834.0d0,
& 542.0d0,
& 455.0d0,
& 403.0d0,
& 366.0d0,
& 344.0d0,
& 329.0d0,
& 312.0d0,
& 296.0d0,
& 287.0d0,
& 282.0d0,
& 279.0d0,
& 278.0d0,
& 276.0d0,
& 274.0d0,
& 272.0d0,
& 271.0d0,
& 275.0d0/
    DO 100 K = 1, 25
      IF(PLAMEV .LE. ENEDGE(K)) THEN
        FLUSCW = FACTOR(K)
      GOTO 200
      ELSE
        FLUSCW = ONEONE
      END IF
100  CONTINUE
    STOP
200  CONTINUE
    K=K+1
*    LSCZER = .FALSE.
*    OPEN (UNIT=48,FILE="plamevino",STATUS='UNKNOWN')
*    WRITE(48,*) PLA, PLAMEV, FLUSCW
    RETURN
*==== End of function Fluscw =====*
END
```