

Cours : Outil de Simulation Monte Carlo : Geant4

M. Gouighri¹

¹Faculté des Sciences
Université Ibn-Tofail, Kénitra

September 22, 2022



Objectif du module

Acquérir un savoir faire en modélisation informatique pour la physique corpusculaire et ses applications

- physique des particules
- physique nucléaire
- physique médicale
- astrophysique
- interface physique-biologie
- ...



Syntaxe

- En **rouge** : ce qui est important
- En **violet** : termes nouveaux
- En **jaune** : à noter
- En **jaune et courier** : programme
-  : attention !
-  : pour en savoir plus
-  : durée approximative

L'outil Geant4 : Introduction



25 minutes

1

Le rôle de la simulation

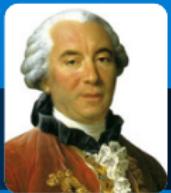
La simulation joue un **rôle fondamental** dans de nombreux domaines et dans de nombreuses phases d'un projet expérimental

- Concevoir le dispositif expérimental
- Évaluer et définir les **résultats de physique** du projet
- Estimer les **performances** de l'expérience
- Evaluer ses **risques** potentiels
- Développer, tester et optimiser les **logiciels de reconstruction et d'analyse des événements physiques**
- Contribuer aux **calculs** et à la validation des **résultats de physique**

Généralités sur la simulation d'un détecteur

- Caractéristiques générales d'une simulation
 - Spécifier la géométrie d'un détecteur de particules
 - Ensuite, le logiciel transporte automatiquement la particule que vous avez tirée dans le détecteur, en simulant les interactions de la particule en utilisant des méthodes Monte Carlo
- Le cœur de la simulation : la méthode Monte Carlo
 - une méthode pour chercher des solutions à un problème mathématique en utilisant un échantillonnage statistique à l'aide de nombres aléatoires





Comte de Buffon

Il y a plus de 200 ans...

- 1772 : les “aiguilles” du **Comte de Buffon**, estimation de π en jetant aléatoirement des aiguilles sur un parquet constitué de lattes de bois parallèles... (on démontre que la probabilité de croiser une ligne est proportionnelle à $2/\pi$)
- 1812 : reformulation mathématique par P.S. Laplace
- Années ~1940
 - modération neutronique par **E. Fermi**
 - résoudre des équations aux dérivées partielles par **J. Von Neumann, S. Ulam et N. Metropolis** pour le **projet Manhattan** à Los Alamos
 - première application au transport des particules
- 1947 : premier code **Monte Carlo (MC)** par **J. Von Neumann** sur un ordinateur
- 1963 : premier code de transport Monte Carlo en FORTRAN : **MCS**
- 1965 : **MCS** devient **MCN**, incluant les interactions neutroniques
- 1977 : **MCNP** en FORTRAN 77



Merci à J. P. Cussonneau, IN2P3

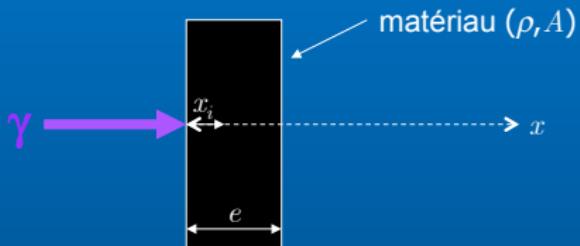
Monte Carlo

Pourquoi les méthodes MC sont-elles intéressantes ?

- En physique des particules élémentaires, il n'est pas possible de prédire où, quand et comment une particule va interagir
- Les interactions sont décrites en termes de sections efficaces, donc en termes de probabilités
- Avec une approche Monte Carlo, on peut
 - suivre une particule interaction après interaction
 - en tirant aléatoirement des paramètres physiques suivant des fonctions de distribution de probabilité qui décrivent les modèles théoriques des interactions particule-matière et reproduisent leur nature stochastique
 - pour un grand nombre d'événements

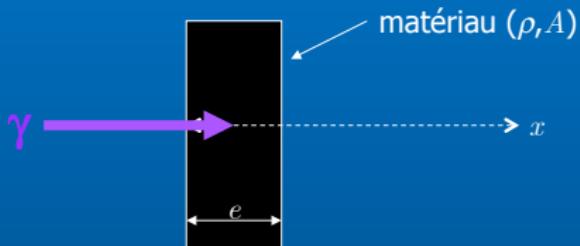
permettant le calcul de quantités physiques et d'effets macroscopiques avec un haut degré de confiance

Comment fonctionnent les méthodes MC pour le transport de particules ? (1/3)



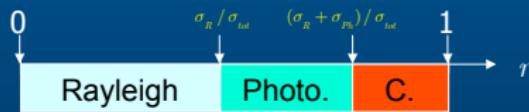
- ETAPE 1** : calcul du libre parcours moyen λ du photon : $\lambda = \frac{A}{\rho N_A \sigma_{tot}}$
- la longeur d'interaction du photon x_i est tirée suivant une loi exponentielle $e^{-x/\lambda}$
 - si $x_i < e$ alors le photon va interagir (sinon il n'interagit pas)

Comment fonctionnent les méthodes MC pour le transport de particules ? (2/3)



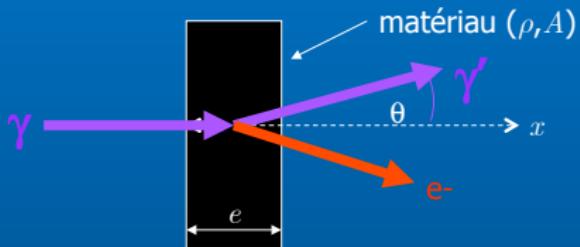
ETAPE 2 : détermination des processus physiques qui peuvent se produire:
diffusion Rayleigh (R), effet Photoélectrique (Ph) ou diffusion Compton (C) ?

- calcul de la section efficace totale : $\sigma_{tot} = \sigma_R + \sigma_{Ph} + \sigma_C$
- tirage aléatoire et uniforme de r entre 0 et 1



7

Comment fonctionnent les méthodes MC pour le transport de particules ? (3/3)



ETAPE 3 : détermination de l'état final : par ex. dans la diffusion Compton

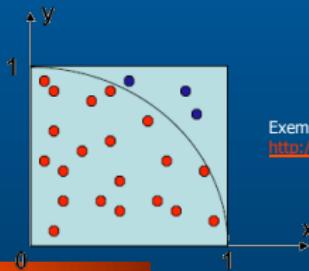
- calcul de l'angle de diffusion θ d'après la section efficace différentielle $d\sigma/d\Omega = f(\theta)$
- calcul cinématique de l'énergie du photon diffusé
- et le transport continue, pour le photon diffusé et l'électron éjecté...



P. S. Laplace

Un exemple célèbre : calcul Monte Carlo de π

- Proposé par Laplace en 1786
- Soit $M(x,y)$ un point de coordonnées $0 < x < 1$ et $0 < y < 1$
- x et y sont tirées aléatoirement (distribution uniforme)
- si $x^2+y^2 \leq 1$, alors M appartient au disque centré en $(0,0)$ de rayon 1
- la probabilité que M appartienne au disque vaut $\pi/4$
- cette probabilité peut être estimée par le rapport du nombre de points présents sur le disque au nombre total de points tirés
- plus on aura tiré de points, meilleure sera l'estimation, plus long sera le calcul...



Exemple en ligne :
<http://www-sop.inria.fr/mefisto/java/tutorial1/node15.html>

Geant4 ?

GEometry ANd Tracking

Geant4 : un ensemble de librairies pour simuler les interactions des particules avec la matière

- Initié par le CERN en 1994 pour la PHE (LHC), successeur de Geant3 (20 ans)
- R&D RD44, 1994-1998, 1^{ère} livrée en Décembre 1998
- Développé par une **collaboration internationale** (~100 membres)
- **Technologie Orienté-Objet (C++)**
- Librairies, pas un code utilisateur
- Mis à jour, entièrement ouvert et gratuit
- Deux livrées par an

<http://geant4.org>

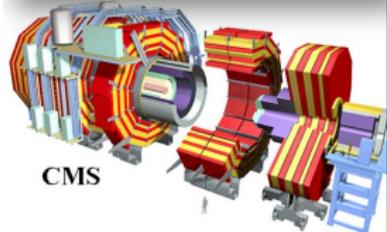
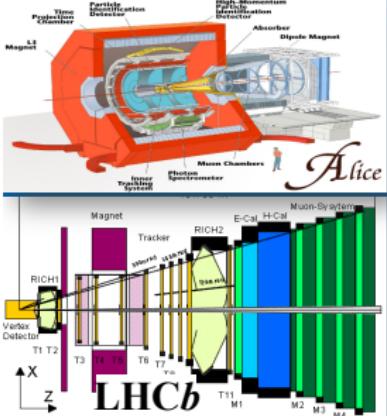
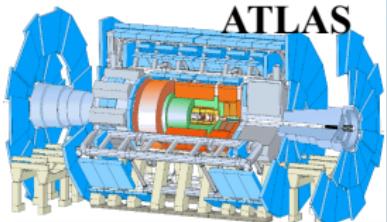
Geant4 : simuler une expérience de physique des particules

- Définir une géométrie
- Modéliser les processus d'interaction physique (électromagnétiques, hadroniques)
- Générer des particules primaires et les suivre dans la géométrie
- Sauvegarder la physique et l'analyser

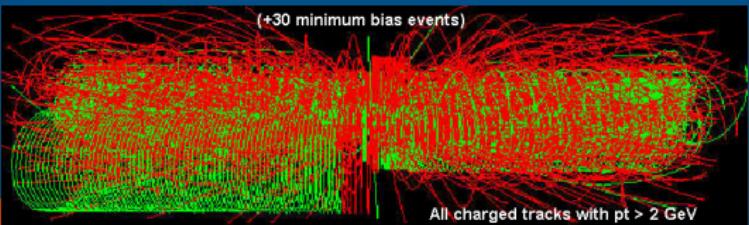
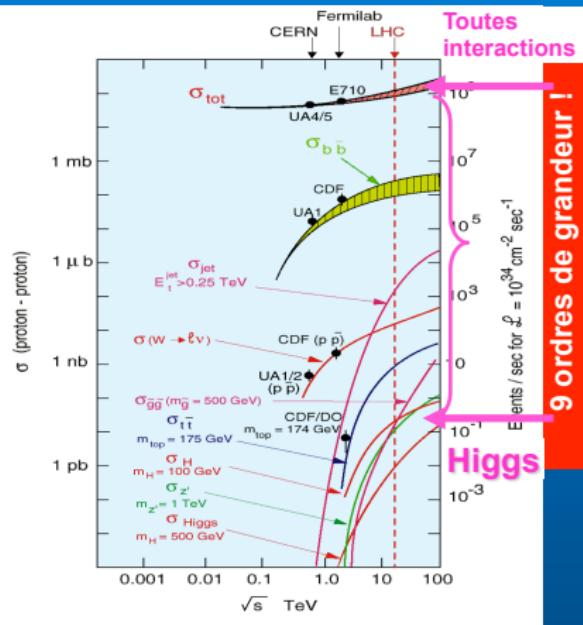
En plus

- Visualisation
- Interactivité
- Extensibilité





Né des exigences
des expériences
à grande échelle de physique
des particules :
LHC au CERN





PPARC



La collaboration Geant4



CERN, ESA, KEK, SLAC, TRIUMF,
TJNL, INFN, IN2P3, PPARC

Barcelona Univ., Budker Inst., Frankfurt
Univ., Karolinska Inst., Helsinki Univ.,
Lebedev Inst., LIP, Northeastern Univ. etc.



ATLAS



Le ZOO

EGS4, EGS5, EGSnrc

Geant3, Geant4

MARS

MCNP, MCNPX, A3MCNP, MCNP-DSP, MCNP4B

MVP, MVP-BURN

Penelope

Peregrine

Tripoli-3, Tripoli-3 A, Tripoli-4

De nombreux codes **ne sont pas distribués**
publiquement

Beaucoup d'activité autour des simulations MC !

DPM
EA-MC
FLUKA
GEM
HERMES
LAHET
MCBEND MCU
MF3D
NMTC
MONK
MORSE
RTS&T-2000
SCALE
TRAX
VMC++

Pourquoi Geant4 ?

- Un ensemble de librairies entièrement gratuit et accessible librement à tous
- Entièrement ouvert (ce n'est pas une « boîte noire »)
- Flexible et extensible par sa technologie Orienté-Objet
- De l'ADN à l'échelle d'une planète...
- En C++

open-source
la meilleure approche pour le calcul scientifique ?

Que peut faire Geant4 pour vous ?

- Transporter une particule pas-à-pas en prenant en compte les interactions avec des matériaux et des champs électromagnétiques externes jusqu'à ce que la particule
 - Perde totalement son énergie cinétique
 - Disparaîsse par interaction
 - Sorte d'un volume de simulation
- Fournit un moyen à l'utilisateur d'accéder au processus de transport et de récupérer les résultats de simulation
 - Au début et à la fin du transport
 - À la fin de chaque "pas" durant le transport
 - Lorsqu'une particule penètre dans la zone sensible d'un détecteur
 - etc.
 - On parle d'**actions utilisateur** ou **User Actions**

Ce que **vous** devez faire pour Geant4

- Il faut lui fournir **trois informations essentielles**
 - Géométrie du détecteur
 - Processus physiques & particules associées
 - Cinématique des particules qui pénètrent dans le détecteur
- Egalement, si nécessaire
 - Champs électromagnétiques
 - Des actions à prendre pendant le transport des particules
 - Des actions à prendre quand une particule pénètre dans une zone sensible d'un détecteur
 - etc.

Des outils pour vous aider

- Geant4 fournit des **outils** pour vous aider à préparer votre simulation
- De nombreuses possibilités pour décrire la **géométrie du détecteur**
 - Combiner des éléments géométriques de base (boîte, cylindre, trapèze, etc.)
 - Représentation par des surfaces planes
 - Représentation par des opérations booléennes, etc.
- Une manière standard de définir les **matériaux du détecteur**
 - Une grande variété d'exemples pour définir de nombreux matériaux
- Une grande variété de **particules**
 - Particules élémentaires standard (électron, muon, proton,...)
 - Particules instables (résonances, quarks, ...)
 - Ions
 - Particules exotiques (geantino, geantino chargé,... les vôtres !)

Choix des processus physiques

- Geant4 fournit une grande variété de **processus & modèles physiques** décrivant les interactions des particules avec la matière
- Principales catégories de processus physiques
 - Electromagnétiques “standard”
 - Electromagnétiques “de basse énergie”
 - Hadroniques
- Comment utiliser les processus physiques ?
 - Beaucoup d'exemples de “**Physics Lists**” dans le cadre d'applications
 - Des Physics Lists **recommandées** par la collaboration

Autres cours

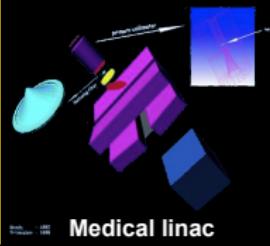
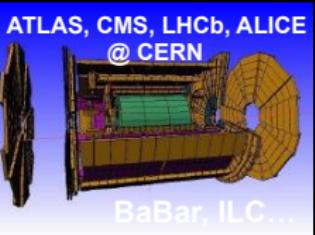
Documentation
et autres cours disponibles à l'adresse
unique

<http://geant4.org>

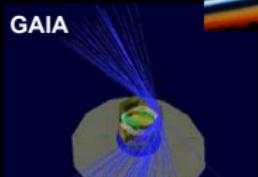
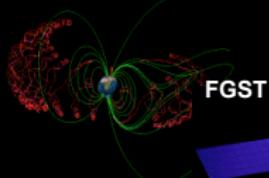
En résumé

- Ce cours est basé sur le programme d'apprentissage mis en place par la collaboration Geant4
 - une collaboration internationale de scientifiques et d'informaticiens, qui développent et mettent à jour l'outil Geant4
- Ce cours est une introduction rapide à l'architecture, à la modélisation, à la physique, aux fonctionnalités et aux applications de Geant4
 - il ne remplace pas l'étude de la documentation Geant4
- Des cours plus complets avec exercices sont organisés très régulièrement par la collaboration Geant4
 - plus d'information sur le site web de Geant4 : <http://geant4.org>

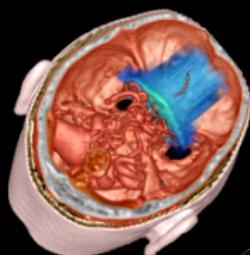
La langue officielle de Geant4 est
l'anglais...



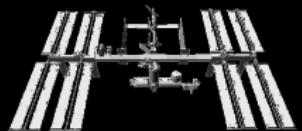
Earth magnetosphere



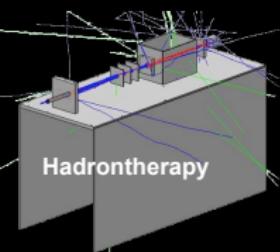
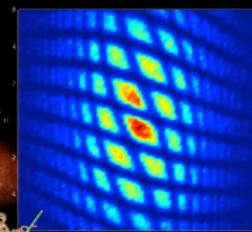
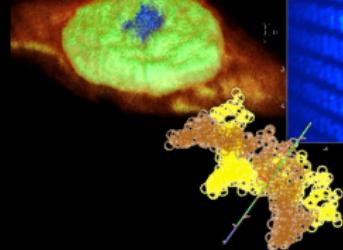
Geant 4



ISS



Physics-Biology



L'outil Geant4 : Structure de base



30 minutes

1

Le noyau de Geant4 ou kernel

- Fournit la **fonctionnalité centrale** de Geant4
 - Implémente Geant4 comme une **machine à états**
 - Traite les **runs, events, tracks, steps, hits, trajectories**
 - Fournit un cadre pour
 - Les processus physiques
 - La visualisation
 - Les (Graphical) User Interfaces ou (G)UI
 - La persistence (collection d'information)
 - La création d'histogrammes et l'analyse
 - Le code utilisateur

Run

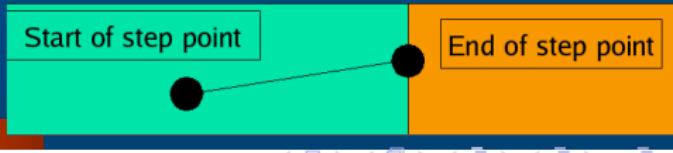
- C'est une collection d'événements produits dans des conditions identiques
- A l'intérieur d'un run, l'utilisateur ne peut pas changer
 - la géométrie des détecteurs
 - les réglages des processus physiques
- Par analogie avec la physique des hautes énergies, un run de Geant4 commence avec la commande “beamOn”
 - le détecteur est inaccessible dès que le faisceau est ON !
- Au démarrage du run
 - la géométrie est optimisée pour la navigation
 - les sections efficaces sont calculées en fonction des matériaux dans le dispositif simulé
 - les coupures de suivi à basse énergie sont définies
- La classe G4Run représente un run

Event (événement)

- Au début du traitement, un événement contient les particules primaires (émises par un générateur, par un “particle gun”,...). Elles sont poussées vers une pile (stack)
- Pendant le traitement, chaque particule est récupérée sur la pile et suivie
- Quand la pile est vide, le traitement de l'événement s'arrête
- Après le traitement, les objets suivants sont disponibles
 - la liste des particules primaires et des vertex
 - les collections de coups (“hits”)
 - les collections de trajectoires (en option)
 - les collections de chiffres (“digits”) (en option)
- La classe G4Event représente un événement

Step (pas)

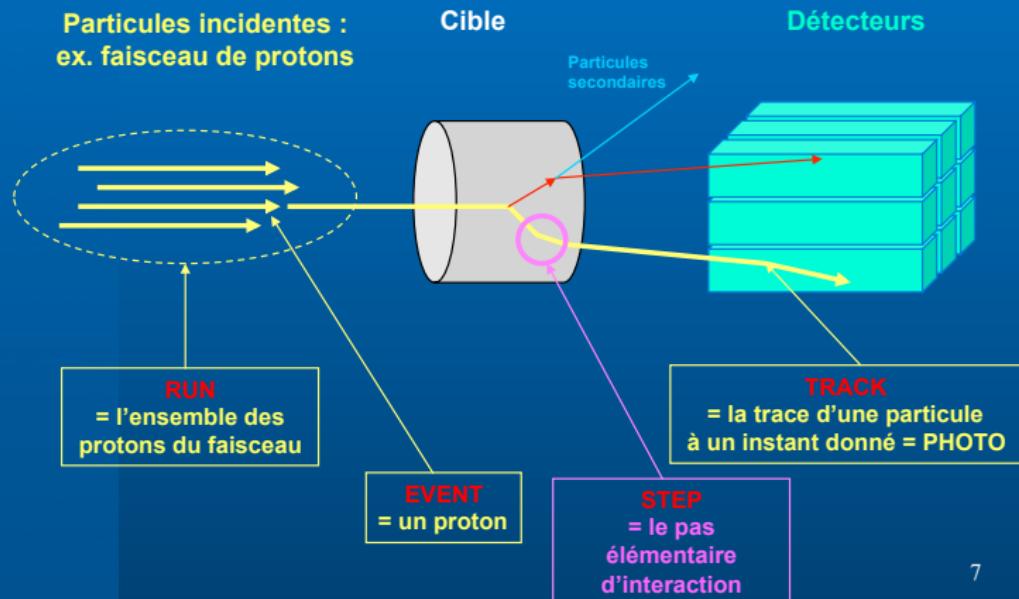
- C'est l'unité de base de la simulation
 - Classe associée **G4Step**
 - possède deux points : **pre-step** et **post-step**
 - contient les **incrémentations des quantités physiques** : énergie perdue, temps écoulé, etc...
 - chaque point contient une information sur le **volume** et sur le **matériau** traversés
 - si le step est limité par une frontière, alors le point final se situe exactement à la frontière mais **appartient (logiquement) au volume suivant**
 - ainsi des processus tels que la réfraction et la radiation de transition peuvent être simulés



Track (trace)

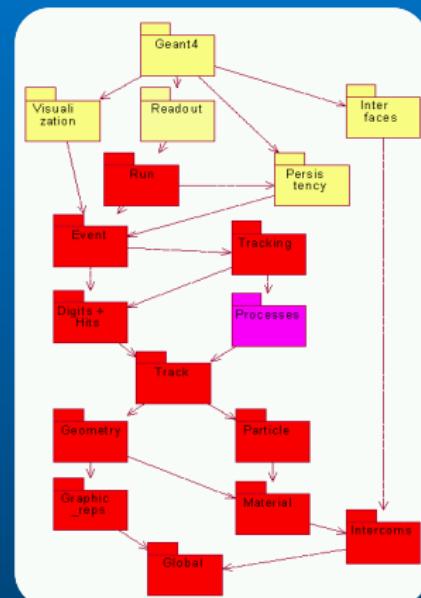
- C'est “la photo” d'une particule dynamique dans son environnement
 - quand la particule se déplace, les quantités sur la photo changent
 - à chaque instance, une trace (“track”) a une position, des quantités physiques...
 - ce n'est pas une collection de steps, ni une trajectoire !
- Durée de vie d'une trace
 - Créeé par un générateur ou un processus physique
 - par ex : décroissance radioactive
 - Détruite quand la particule
 - quitte le monde
 - disparaît (décroissance radioactive ou absorption)
 - son énergie devient nulle et aucun processus au repos n'est défini
 - l'utilisateur la tue
- Aucune trace ne survit après la fin d'un événement (pas de “persistance”)
 - l'utilisateur doit enregistrer la track dans une trajectoire (“trajectory”)
- La classe G4Track représente une track

Geant4 fonctionne comme une expérience de physique des particules



Les catégories de G4

- Contient 17 catégories
 - chacune est développée indépendamment et mise à jour par des groupes de travail
- Elles ont été étudiées pour minimiser les dépendances
 - le noyau est représenté par les catégories en rouge



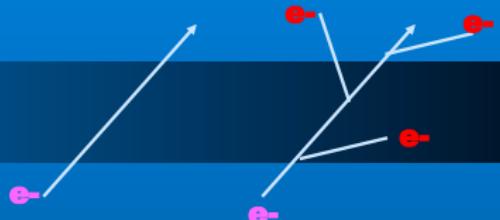
Le suivi des particules (tracking)

- Le suivi des particules est général
 - Il est indépendant
 - du type de particule
 - des processus physiques assignés à la particule
 - Il permet aux processus de
 - contribuer au calcul de la longueur du step
 - contribuer à tout changement dans les quantités physiques du track
 - générer les particules secondaires
 - suggérer des changements d'état du track
 - suspendre, remettre à plus tard, tuer

Les processus dans Geant4

- Toutes les interactions sont gérées par des processus
 - le transport des particules est aussi un processus ; la particule peut interagir avec des frontières géométriques et avec tout type de champ
 - il existe aussi un processus de paramétrisation de gerbe qui peut se substituer au transport
- Chaque particule possède sa propre liste de processus auxquels elle est sensible
 - au début de chaque step, tous ces processus sont interrogés pour obtenir une longueur d'interaction physique.
- Le processus qui propose la plus petite longueur d'interaction est celui qui se produit
 - le processus choisi est aussi celui qui limite la taille du step

Les coupures



- Une coupure (“cut, cutoff”) est un seuil de production des particules secondaires
 - ne s’applique qu’aux processus qui ont des divergences infrarouges
 - ce n’est pas une coupure sur le suivi
- Un seuil en énergie doit être défini en dessous duquel la perte d’énergie discrète devient continue
 - Spécifier un parcours (qui est converti en énergie pour chaque matériau) en dessous duquel la perte d’énergie continue commence, et la particule primaire est suivie jusqu’à une énergie nulle
 - Au-dessus du seuil spécifié, on crée explicitement les secondaires
 - En-dessous, on ne crée pas les secondaires, on ajoute la perte d’énergie correspondante à la perte d’énergie continue de la particule primaire

11

Les classes manager

- Les classes manager effectuent les transactions entre les objets d'une même catégorie et communiquent avec d'autres managers
- Ce sont des **singletons**
- L'utilisateur rencontrera essentiellement le **G4RunManager**
 - Il faut lui déclarer la géométrie du détecteur, la “physics list”, le générateur de particules primaires
- Autres classes manager
 - **G4EventManager** – gère les Events, les actions utilisateur
 - **G4SteppingManager** – gère les Steps, les processus physiques, le marquage des hits, les actions utilisateur
 - **G4TrackingManager** – gère les Tracks, le stockage des trajectoires, les actions utilisateur
 - **G4VisManager** – gère les drivers de visualisation

Les classes utilisateur

- Utiliser ces classes pour créer votre application à partir de l'outil Geant4 (les classes en violet sont obligatoires)
- Classes d'initialisation
 - G4VUserDetectorConstruction
 - G4VUserPhysicsList
- Classes d'action
 - G4VUserPrimaryGeneratorAction
 - G4UserRunAction
 - G4UserEventAction
 - G4UserSteppingAction
 - G4UserTrackingAction
 - G4UserStackingAction
- main() - non fourni par Geant4



13

Construire votre simulation en 7 étapes

1 - Décrire le détecteur



- Dériver votre propre classe concrète de la classe de base abstraite `G4VUserDetectorConstruction`
- Dans la méthode `Construct()`, il faut
 - Assembler tous les matériaux nécessaires
 - Construire les volumes de la géométrie du détecteur
- En option, on peut définir:
 - Des classes de sensibilité du détecteur et les assigner aux volumes du détecteur
 - Des régions dans n'importe quelle partie du détecteur
 - permet par ex. de fixer des coupures dans des zones du détecteur
 - Des attributs de visualisation des éléments de géométrie
 - Des champs électromagnétiques ou autres

2 - Sélectionner les processus physiques

- Geant4 n'a pas de particules ou processus définis par défaut
 - même le transport des particules doit être défini explicitement par l'utilisateur
- Dériver votre propre classe concrète de la classe de base abstraite `G4VUserPhysicsList`
 - Définir toutes les particules
 - Définir tous les processus nécessaires et les assigner aux bonnes particules
 - Définir les seuils de production (*cutoff*) en termes de longueur de parcours et les assigner au volume global (le "monde") et éventuellement à chaque région
- Geant4 fournit de nombreuses classes et méthodes pour y parvenir
- Des exemples de "physics lists" existent pour la physique électromagnétique et la physique hadronique

3 – Générer des événements primaires

- Pour chaque événement, l'utilisateur doit définir toutes les caractéristiques de la particule primaire
- Dériver une classe concrète de la classe de base abstraite `G4VUserPrimaryGeneratorAction`
- Générer les particules
 - Dériver votre propre générateur de `G4VPrimaryGenerator`
 - Ou utiliser les générateurs fournis :
 - `G4ParticleGun`
 - l'utilisateur fournit le nombre, l'énergie, la direction et le type de particules à générer
 - `G4GeneralParticleSource`
 - autres

4 - Classes d'action utilisateur

- **G4UserRunAction**
 - **BeginOfRunAction**
 - sélection du run
 - définir histogrammes
 - **EndOfRunAction**
 - analyse du run
 - remplir histogrammes
- **G4UserEventAction**
 - **BeginOfEventAction**
 - sélection de l'événement
 - **EndOfEventAction**
 - analyse de l'événement
- **G4UserSteppingAction**
 - **UserSteppingAction**
 - analyse du step
 - tuer, suspendre, remettre à plus tard une track



OBLIGATOIRE

5 - Le programme main()

- Geant4 ne fournit pas de `main()`
 - De nombreux exemples sont fournis dans le “User’s Guide: For Application Developers”
- Dans `main()`, vous devez :
 - Construire le `G4RunManager` (ou une classe qui en dérive)
 - Fournir au `G4RunManager` des pointeurs vers les classes utilisateur obligatoires dérivant de :
 - `G4VUserDetectorConstruction`
 - `G4VUserPhysicsList`
 - `G4VUserPrimaryGeneratorAction`
- D’autres classes qui peuvent être définies dans le `main()`
 - `VisManager`
 - `(G)UI session`
 - Classes utilisateur optionnelles

6 - Définir une session interface utilisateur

- Geant4 fournit plusieurs classes concrètes **G4UISession**
 - Sélectionner celle qui correspond à votre environnement de travail
 - Dans **main()**, construire l'une d'entre elles
 - Invoke sa méthode **sessionStart()**
- Sessions User Interface fournies pour vous:
 - **G4UIExecutive** – pour utiliser l'interface Qt
 - **G4UITerminal** – terminal à caractères de type C- et TC-shell
 - **G4GAG** – Tcl/Tk or Java PVM based GUI
 - **G4JAG** – interface vers JAS (Java Analysis Studio)
 - **G4UIBatch** – batch job avec fichier macro

7 - Visualisation

- Dériver votre propre classe concrète de `G4VisManager` d'après votre environnement de travail
- Geant4 fournit des interfaces vers plusieurs systèmes graphiques:
 - `G4VisExecutive` pour utiliser l'interface Qt
 - `ASCII`Tree (`ATree`)
 - `DAWNFILE` (`DAWNFILE`)
 - `G4HepRep` (`HepRepXML`)
 - `G4HepRepFile` (`HepRepFile`)
 - `OpenGLImmediateQt` (`OGLI`, `OGLIQt`)
 - `OpenGLImmediateX` (`OGLIX`)
 - `OpenGLImmediateXm` (`OGLIXm`, `OGLI_FALLBACK`, `OGLIQt_FALLBACK`)
 - `OpenGLStoredQt` (`OGL`, `OGLS`, `OGLSQt`)
 - `OpenGLStoredX` (`OGLSX`)
 - `OpenGLStoredXm` (`OGLSXm`, `OGL_FALLBACK`, `OGLS_FALLBACK`,
`OGLSQt_FALLBACK`)
 - `RayTracer` (`RayTracer`)
 - `VRML1FILE` (`VRML1FILE`)
 - `VRML2FILE` (`VRML2FILE`)
 - `gMocrenFile` (`gMocrenFile`)

Architecture d'une application

- Dans le répertoire **simulation**
 - **Simulation.cc**
 - CMakeList
 - Dans **simulation/src**
 - **DetectorConstruction.cc**
 - **PhysicsList.cc**
 - **PrimaryGeneratorAction.cc**
 - **RunAction.cc**
 - **EventAction.cc**
 - **SteppingAction.cc**
 - **HistoManager.cc**
 - Dans **simulation/include**
 - **DetectorConstruction.hh**
 - **PhysicsList.hh**
 - **PrimaryGeneratorAction.hh**
 - **RunAction.hh**
 - **EventAction.hh**
 - **SteppingAction.hh**
 - **HistoManager.hh**
- 
- **main()**
 - Instruction pour cmake
 - Matériaux et géométries
 - Particules et processus physiques
 - Particules primaires
 - Classe d'action utilisateur (run)
 - Classe d'action utilisateur (event)
 - Classe d'action utilisateur (step)
 - Classe manager de gestion d'histog.

En violet : fichiers obligatoires
En blanc : fichiers optionnels

Le préfixe G4

- Pour la portabilité, “G4” est ajouté aux types bruts de C++
 - `G4int`, `G4double`, ...
 - ainsi Geant4 implémente un type correct pour une architecture informatique donnée (les types de base comme `int`, `float`, `double`, etc. ont des plages de valeurs qui dépendent du compilateur et des plateformes)
- `G4cout` et `G4cerr` sont des objets `ostream` définis par Geant4
 - `G4endl` est aussi fourni
- Certains GUIs sont des buffers de sortie qui peuvent afficher des informations dans une autre fenêtre ou fournir des fonctionnalités de stockage / d'édition
 - Ne pas utiliser `std::cout`, etc. mais `G4cout`
- Ne pas utiliser `std::cin` pour la saisie. Utiliser à la place des commandes utilisateurs fournies par la catégorie `intercoms`
 - e.g. `G4UIcmdWithADouble`
 - Utiliser `G4cin`

23

Système d'unités



- **Geant4 n'a pas d'unité par défaut.** Tout nombre doit être multiplié par son unité :
 - par exemple :

```
G4double width = 12.5*m;  
G4double density = 2.7*g/cm3;
```
 - Si l'unité n'est pas spécifiée, Geant4 utilise une unité interne mais c'est fortement déconseillé !
 - La plupart des unités courantes sont disponibles
 - On peut définir de nouvelles unités
 - Se reporter à CLHEP: voir **SystemOfUnits.h**
- Diviser une variable par l'unité qu'on veut obtenir

```
G4cout << dE / MeV << " (MeV)" << G4endl;
```

Système d'unités

- Le système d'unité est défini dans les librairies **CLHEP**, à partir de
 - millimètre (**mm**), nanoseconde (**ns**), Mega eV (**MeV**), charge du positron (**eplus**), degré Kelvin (**kelvin**), quantité de matière (**mole**), intensité lumineuse (**candela**), radian (**radian**), steradian (**steradian**)
- Toutes les autres unités sont calculées à partir de ces unités de base
- En sortie, Geant4 peut choisir l'unité la plus appropriée à l'ordre de grandeur. Il suffit de spécifier la **catégorie** de données à afficher (**Length**, **Time**, **Energy**, etc...):

```
G4cout << G4BestUnit(StepSize, "Length");
```

StepSize sera affichée en **km**, **m**, **mm** ou ... **fermi**, en fonction de sa valeur

25

En résumé

- L'outil Geant4 contient 17 catégories qui dépendent le moins possible les unes des autres
- La plus grande unité de la simulation est le **run**, elle consiste en un ensemble d'**events**, **tracks** et **steps**
 - une “track” est l’arrêt sur image d’une particule dynamique, pas une trajectoire
- Le suivi et la physique sont gérés par des processus
- Les seuils de production (et l’empilement) permettent de gérer efficacement la simulation
 - Geant4 suit les particules jusqu’à une énergie nulle
- Les classes utilisateur permettent de personnaliser la simulation
 - il faut construire l’appareillage et définir la physique
 - des commandes permettent de dialoguer avec la simulation

L'outil Geant4 : Définition des matériaux



DetectorConstruction.hh (.cc)



15 minutes

1

Définition de matériaux

- Différents **types** de matériaux peuvent être définis

- isotopes → **G4Isotope**
- éléments chimiques → **G4Element**
- molécules → **G4Material**
- composés et mélanges → **G4Material**

- **Attributs** associés

- température, pression, état, **density** (= masse vol. en anglais)



2

Isotopes, éléments et matériaux

- **G4Isotope** et **G4Element** décrivent les propriétés atomistiques
 - numéro atomique,
 - nombre de nucléons,
 - masse molaire,
 - etc...
- **G4Material** décrit les propriétés macroscopiques de la matière
 - température,
 - pression,
 - état (=solide, liquide, gaz),
 - densité (= mass vol.),
 - etc...

Eléments

- Les éléments peuvent être définis par leur nom, leur symbole, leur numéro atomique et leur masse molaire en utilisant G4Element

```
a = 1.01*g/mole;  
G4Element* elH  
= new G4Element("Hydrogen",symbol="H",z=1.,a);  
  
a = 16.00*g/mole;  
G4Element* elO  
= new G4Element("Oxygen",symbol="O",z=8.,a);
```

Eléments et isotopes

Les éléments peuvent être assemblés à partir d'**isotopes** pré-définis par :

```
G4Isotope (const G4String& name,  
           G4int     z,      // number of atoms  
           G4int     n,      // number of nucleons  
           G4double a); // mass of mole
```

en précisant le nombre d'isotopes et en les “ajoutant” à l'élément en question:

```
G4Element (const G4String& name,  
            const G4String& symbol, // element symbol  
            G4int     nIso ); // # of isotopes  
  
G4Element::AddIsotope(G4Isotope* iso,      // isotope  
                      G4double relAbund); // fraction of atoms  
                                // per volume
```

Matériaux fait d'un seul élément (lAr)

- Matériaux constitués d'un seul élément

```
G4double density = 1.390*g/cm3;
```

```
G4double a = 39.95*g/mole;
```

```
G4Material* lAr =  
new G4Material("liquidArgon",18.,a,density);
```

Matériaux à partir d'une molécule (eau)

Une molécule est constituée de plusieurs éléments
(composition par nombre d'atomes)

```
a = 1.01*g/mole;
G4Element* elH = new G4Element("Hydrogen",symbol="H",z=1.,a);

a = 16.00*g/mole;
G4Element* elO = new G4Element("Oxygen",symbol="O",z=8.,a);

density = 1.000*g/cm3;
G4Material* H2O = new G4Material("Water",density,ncomp=2);

H2O->AddElement(elH, natoms=2);
H2O->AddElement(elO, natoms=1);
```

Matériaux composés d'éléments (air)

- Composé : composition par fraction massique

```
a = 14.01*g/mole;
G4Element* elN = new G4Element(name="Nitrogen",symbol="N",z= 7.,a);

a = 16.00*g/mole;
G4Element* elO = new G4Element(name="Oxygen",symbol="O",z= 8.,a);

density = 1.290*mg/cm3;

G4Material* Air = new G4Material(name="Air",density,ncomponents=2);

Air->AddElement(elN, 70.0*perCent);
Air->AddElement(elO, 30.0*perCent);
```

Matériaux fait d'un mélange (aérogel)

- Composition en matériaux

```
G4Element* elC = ...; // define "carbon" element  
G4Material* SiO2 = ...; // define "quartz" material  
G4Material* H2O = ...; // define "water" material  
  
density = 0.200*g/cm3;  
G4Material* Aerog =  
    new G4Material("Aerogel",density,ncomponents=3);  
Aerog->AddMaterial(SiO2,fractionmass=62.5*perCent);  
Aerog->AddMaterial(H2O ,fractionmass=37.4*perCent);  
Aerog->AddElement (elC ,fractionmass= 0.1*perCent);
```

Les gaz (CO₂)

- Il peut être nécessaire de préciser la température et la pression
 - par ex. calcul de dE/dx affecté

```
G4double density = 27.*mg/cm3;  
G4double temperature = 325.*kelvin;  
G4double pressure = 50.*atmosphere;  
  
G4Material* CO2 =  
    new G4Material("CarbonicGas", density, ncomponents=2  
                  kStateGas, temperature, pressure);  
CO2->AddElement(C,natoms = 1);  
CO2->AddElement(O,natoms = 2);
```

Exemple : le vide

- Le vide absolu n'existe pas : c'est un gaz à très faible masse volumique !
 - Ne pas définir de matériaux ayant $\rho = 0$



```
G4double atomicNumber = 1.;  
G4double massOfMole = 1.008*g/mole;  
G4double density = 1.e-25*g/cm3;  
G4double temperature = 2.73*kelvin;  
G4double pressure = 3.e-18*pascal;  
G4Material* Vacuum =  
    new G4Material("interGalactic", atomicNumber,  
                  massOfMole, density, kStateGas,  
                  temperature, pressure);
```

La base de données de matériaux NIST

- Base de données :
<http://physics.nist.gov/PhysRefData>
- Garantit précision sur les paramètres principaux :
 - masse volumique
 - potentiel d'ionisation moyen...
- Importée dans Geant4

La base de données de matériaux NIST

- Isotopes naturels
 - plus de 3000 isotopes disponibles
- Matériaux élémentaires
 - H → Cf (Z = 1 → 98)
- Composés
 - par ex. “G4ADIPOSE_TISSUE_ICRP”
 - Le **vide** “G4_Galactic”
- Matériaux HEP et nucléaire
 - par ex. Ar liquide, PbWO
- Il est même possible de construire des mélanges de matériaux NIST et de matériaux définis par l'utilisateur

La base de données de matériaux NIST

Eléments

Matériaux

===== ### Elementary Materials from the NIST Data Base =====			
Z	Name	ChFormula	density(g/cm^3) I(eV)
1	G4_H_H_2	8.3748e-05	19.2
2	G4_He	0.000166322	41.8
3	G4_Li	0.534	40
4	G4_Be	1.848	63.7
5	G4_B	2.37	76
6	G4_C	2	81
7	G4_N_N_2	0.0011652	82
8	G4_O_O_2	0.00133151	95
9	G4_F	0.00158029	115
10	G4_Ne	0.000838505	137
11	G4_Na	0.971	149
12	G4_Mg	1.74	156
13	G4_Al	2.6989	166
14	G4_Si	2.33	173

- Matériaux élémentaires
- Composés
- Physique nucléaire,
applications spatiales...

===== ### Compound Materials from the NIST Data Base =====			
N	Name	ChFormula	density(g/cm^3) I(eV)
13	G4_Adipose_Tissue		0.92 63.2
		1	0.119477
		6	0.63724
		7	0.00797
		8	0.232333
		11	0.0005
		12	2e-05
		15	0.00016
		16	0.00073
		17	0.00119
		19	0.00032
		20	2e-05
		26	2e-05
		30	2e-05
		4	G4_Air 0.00120479 85.7
		6	0.000124
		7	0.755268
		8	0.231781
		18	0.012827
		2	G4_Csl 4.51 553.1
		53	0.47692
		55	0.52308

Comment l'utiliser ?

- Plus besoin de prédéfinir éléments et matériaux !
- cf. Annexe du **Geant4 User's Guide for Application Developers**
- Interfaces utilisateur dédiées

```
G4NistManager* manager = G4NistManager::GetPointer();

G4Element* elm = manager->FindOrBuildElement("symb", G4bool iso);

G4Element* elm = manager->FindOrBuildElement(G4int Z, G4bool iso);

G4Material* mat = manager->FindOrBuildMaterial("name", G4bool iso);

G4Material* mat = manager->ConstructNewMaterial("name",
                                                 const std::vector<G4int>& Z,
                                                 const std::vector<G4double>& weight,
                                                 G4double density, G4bool iso);

G4double isotopeMass = manager->GetMass(G4int Z, G4int N);
```

En résumé

- De nombreuses possibilités pour créer vos matériaux
 - Isotopes, éléments chimiques, molécules, composés et mélanges...
 - Attention à la définition du vide
- Utiliser la base de données NIST

L'outil Geant4 : Construire la géométrie



DetectorConstruction.hh (.cc)



50 minutes

1

Géométrie :

1 - les bases

Décrire votre détecteur

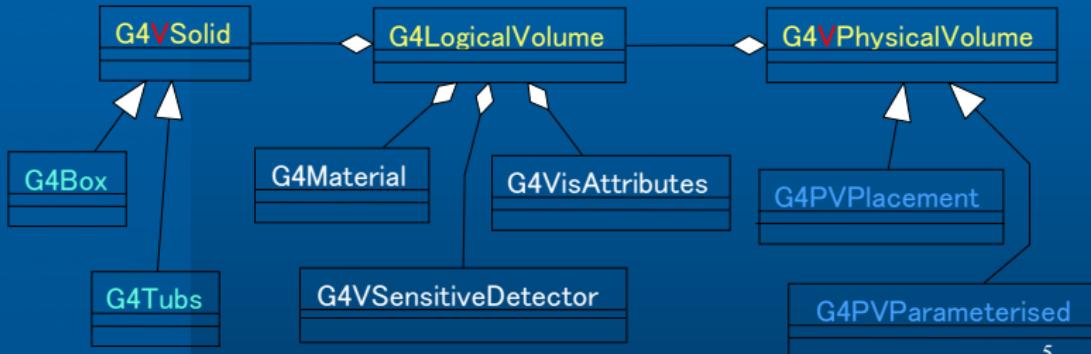
- Dériver votre propre classe concrète de la classe de base abstraite `G4VUserDetectorConstruction`
- Implémenter la méthode `Construct()`:
 - Construire les matériaux nécessaires
 - Définir les formes/solides nécessaires pour décrire la géométrie
 - Construire et placer les volumes de la géométrie
 - On peut aussi en option :
 - Définir les détecteurs sensibles et identifier les volumes du détecteur auxquels ils sont associés
 - Associer un champ électromagnétique à des régions du détecteur
 - Définir les attributs de visualisation des éléments de détecteur

Créer un volume de détecteur

- Commencer avec sa forme et sa taille
 - Boîte de 3x5x7 cm, sphère de R=8m
 - Ajouter des propriétés
 - matériaux,
 - champs E, B
 - le rendre sensible
 - le rendre visible
 - ...
 - Le placer dans un autre volume
 - à un seul endroit
 - de façon répétée en utilisant une fonction
- Volume SOLIDE
("solid")
 - Volume LOGIQUE
("logical")
 - Volume PHYSIQUE
("physical")

Définir la géométrie d'un détecteur

- Trois classes à utiliser
 - **G4VSolid** : forme, taille
 - **G4LogicalVolume** : volumes physiques filies, matériaux, sensibilité, coupures utilisateur, etc.
 - **G4VPhysicalVolume** : position, rotation



Géométrie

- Stratégie de base

```
G4VSolid* pBoxSolid =  
new G4Box("aBoxSolid",  
1.*m, 2.*m, 3.*m);  
  
G4LogicalVolume* pBoxLogical =  
new G4LogicalVolume( pBoxSolid,  
pBoxMaterial, "aBoxLogical");  
  
G4VPhysicalVolume* aBoxPhysical =  
new G4PVPlacement( pRotation,  
G4ThreeVector(posX, posY, posZ),  
"aBoxPhys", pBoxLogical, pMotherPhysical,  
many, copyNo, overlapCheck);
```

Volume logique :
+ volume solide : forme et taille
+ matériau, sensibilité, etc.



Volume physique :
+ rotation et position

Hiérarchie

- Un volume est toujours placé dans un volume **mère**



La position et la rotation du volume **fille** est décrite par rapport aux **système de coordonnées locales du volume mère**

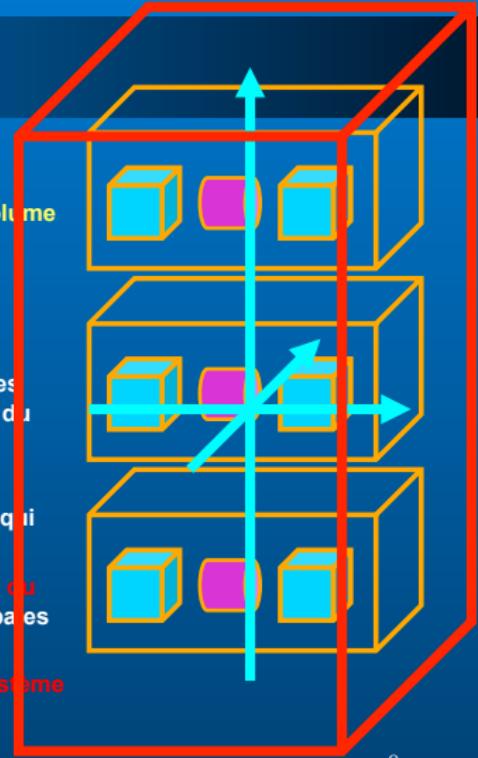
- L'origine du système de coordonnées locales du volume mère est **au centre** du volume mère

- **Attention**

- les volumes filles ne doivent pas dépasser du volume mère
 - Les volumes filles ne doivent pas se chevaucher

Hiérarchie

- Un ou plusieurs volumes peuvent être placés dans un volume mère
- Notez que la relation mère-fille est une information de G4LogicalVolume
 - Si le volume mère est placé plus d'une fois, toutes les filles apparaîtront dans tous les volumes physiques du volume mère
- Le WORLD volume doit être un volume physique unique qui contient entièrement tous les autres volumes
 - le WORLD volume définit les coordonnées globales du système. L'origine du système de coordonnées globales est au centre du WORLD volume
 - la position d'une track est donnée par rapport au système de coordonnées globales
 - il est obligatoire !

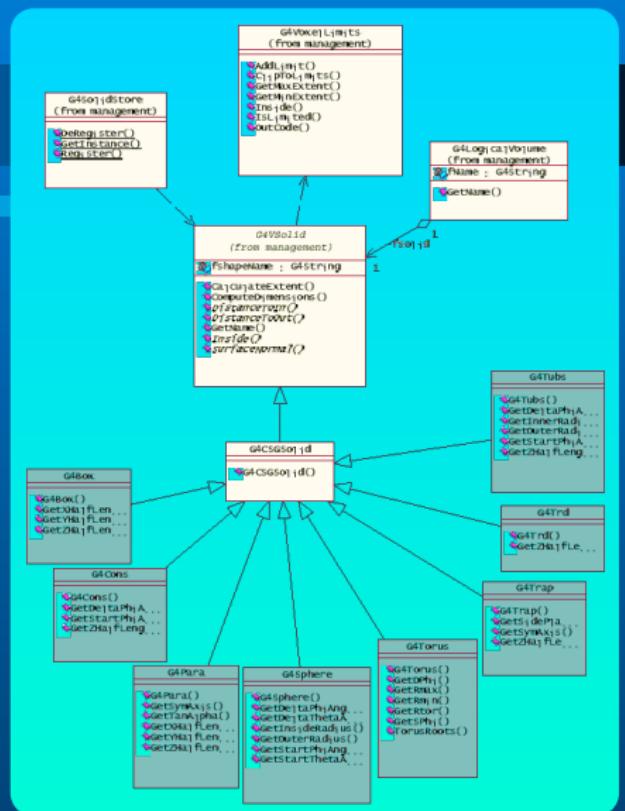


Géométrie :

2 – les volumes **solides**

G4VSolid

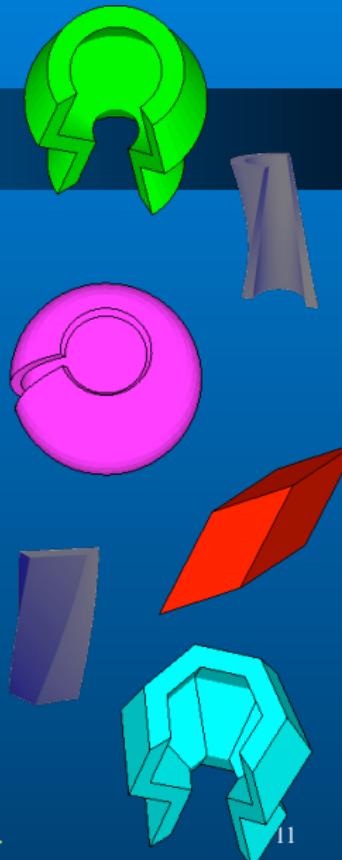
- C'est une **classe abstraite**.
- Tous les solides dans Geant4 en dérivent
 - définit mais n'implémente pas toutes les fonctions nécessaires pour :
 - Calculer les distances de/vers le solide
 - Vérifier si un point est dans le solide
 - Calculer l'étendue du volume
 - Calculer la normale à la surface en un point donné
- Une fois construit, chaque solide est automatiquement enregistré dans un lieu de stockage spécifique



Solides

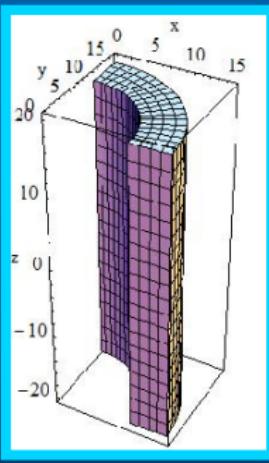
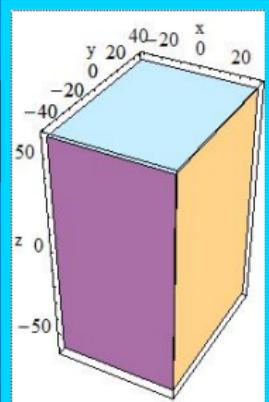
Solides définis dans Geant4 :

- Solides CSG (Constructed Solid Geometry)
 - G4Box, G4Tubs, G4Cons, G4Trd, ...
- Solides CSG spécifiques
 - G4Polycone, G4Polyhedra, G4Hype, ...
 - G4TwistedTubs, G4TwistedTrap, ...
- Solides BREP (Boundary REPresented)
 - G4BREPSolidPolycone,
G4BSplineSurface, ...
 - Tout ordre de surface
- Solides booléens
 - G4UnionSolid, G4SubtractionSolid, ...

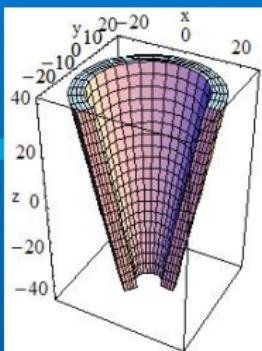


CSG: G4Box, G4Tubs

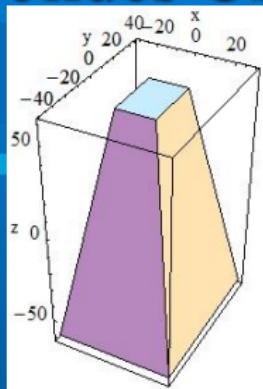
```
G4Box(const G4String &pname,    // name  
       G4double half_x,      // X half size  
       G4double half_y,      // Y half size  
       G4double half_z);    // Z half size  
  
G4Tubs(const G4String &pname,   // name  
        G4double pRmin,     // inner radius  
        G4double pRmax,     // outer radius  
        G4double pDz,       // Z half length  
        G4double pSphi,     // starting Phi  
        G4double pDphi);   // segment angle
```



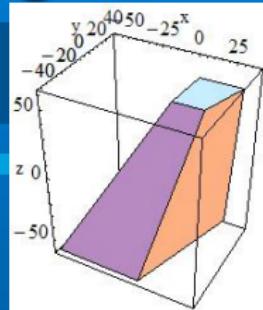
Autres solides CSG



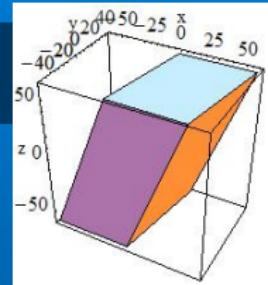
G4Cons



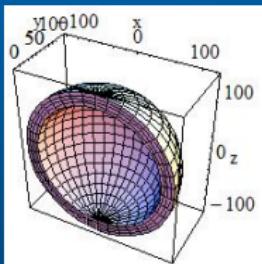
G4Trd



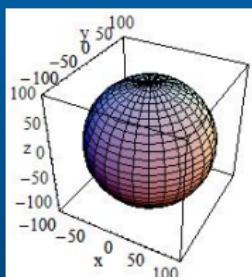
G4Trap



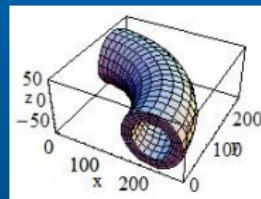
G4Para
(parallélépipède)



G4Sphere



G4Orb
(sphère pleine)

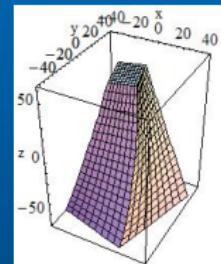
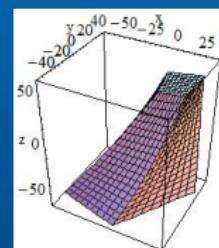
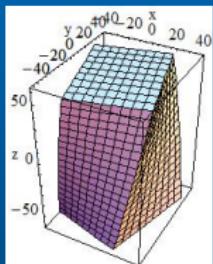
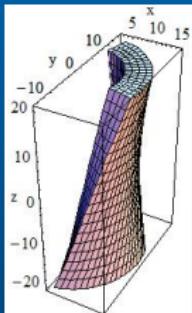
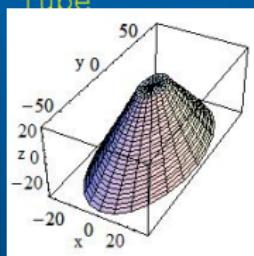
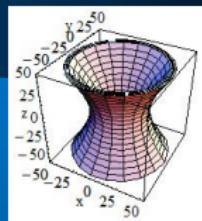
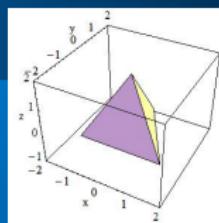
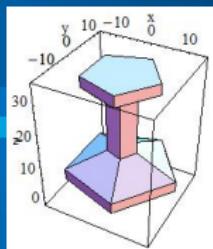
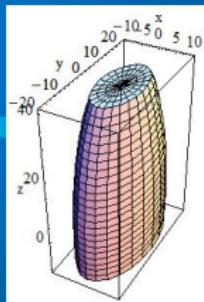
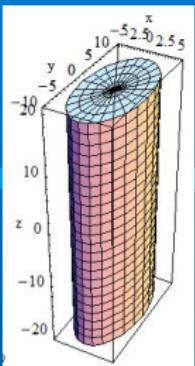


G4Torus

Consulter le User's Guide: For Application Developers pour toutes les formes disponibles

13

Autres solides CSG spécifiques

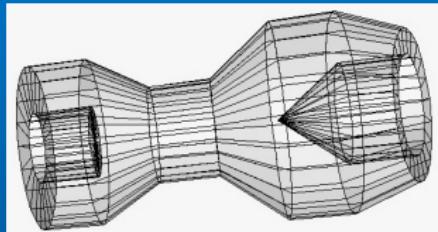


Consulter le User's Guide: For Application Developers pour toutes les formes disponibles

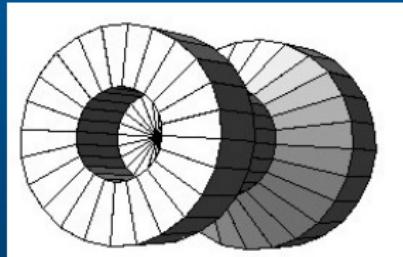
14

Solides CSG spécifiques: G4Polycone

```
G4Polycone(const G4String& pName,  
           G4double phiStart,  
           G4double phiTotal,  
           G4int    numRZ,  
           const G4double r[],  
           const G4double z[]);
```



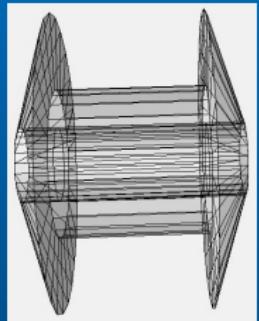
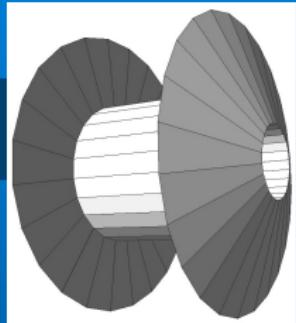
- **numRZ** : nombre de coins dans l'espace **r,z**
- **r, z** : coordonnées des coins
- Constructeur supplémentaire en utilisant des plans



Solides BREP



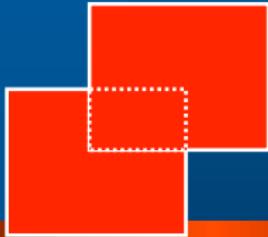
- **BREP = Boundary REPresented**
- **Un solide est défini par ses surfaces**
 - ex. 6 carrés pour un cube
- **Les surfaces peuvent être**
 - planes, 2nd ou ordre plus élevé
 - BREPs élémentaires
 - Splines, B-Splines,
NURBS (Non-Uniform B-Splines)
 - BREPs avancés
- **Quelques BREPs élémentaires pré-définis**
 - box, cons, tubs, sphere, torus, polycone, polyhedra
- **BREPs avancés créés par CAO**



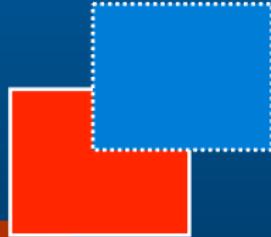
Solides booléens

- Des solides peuvent être combinés en utilisant des opérations booléennes
 - `G4UnionSolid`, `G4SubtractionSolid`, `G4IntersectionSolid`
 - Nécessite : 2 solides, 1 opération booléenne, et en option une transformation pour le 2nd solide
 - Le 2nd solide est positionné par rapport au système de coordonnées du premier solide
 - Le résultat est un solide
- Les solides que l'on peut combiner peuvent être CSG ou d'autres booléens
- Remarque : le temps de calcul pour le suivi des particules lors de la navigation dans un volume booléen complexe est proportionnel au nombre de solides CSG constituants

`G4UnionSolid`



`G4SubtractionSolid`

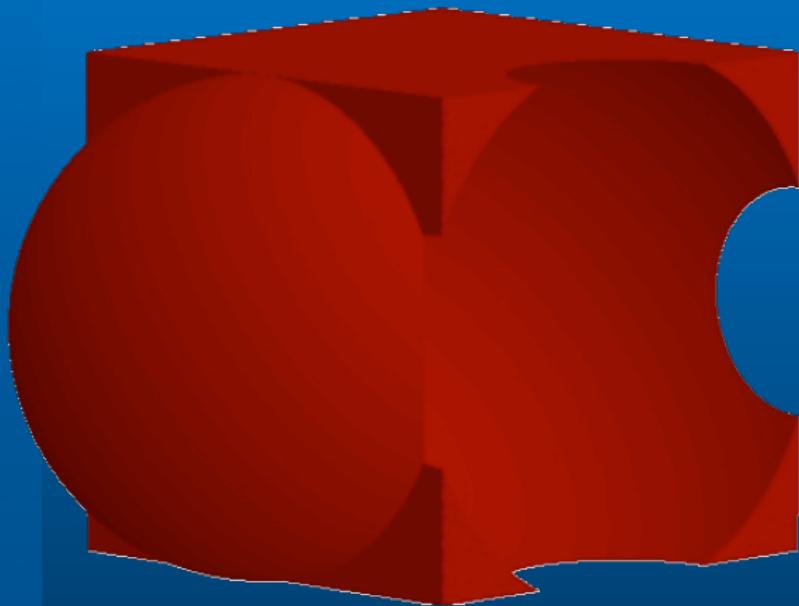


`G4IntersectionSolid`



17

Solides booléens



18

Solides booléens - exemple

```
G4VSolid* box = new G4Box("Box",50*cm,60*cm,40*cm);

G4VSolid* cylinder = new G4Tubs("Cylinder",0.,50.*cm,50.*cm,0.,2*M_PI*rad);

G4VSolid* union = new G4UnionSolid("Box+Cylinder", box, cylinder);

G4VSolid* subtract = new G4SubtractionSolid("Box-Cylinder", box, cylinder, 0, G4ThreeVector(30.*cm,0.,0.));

G4RotationMatrix* rm = new G4RotationMatrix();
rm->rotateX(30.*deg);

G4VSolid* intersect = new G4IntersectionSolid("Box&Cylinder", box, cylinder, rm, G4ThreeVector(0.,0.,0.));
```



L'origine et les coordonnées du solide combiné sont celles du premier solide

Géométrie : 3 – volumes logiques

G4LogicalVolume

- Contient toute l'information sur le volume sauf sa position
 - forme et dimension (**G4VSolid**)
 - matériau
- en option
 - champ magnétique
 - sensibilité
 - coupures utilisateur
 - attributs de visualisation
 - etc...
- Des objets physiques du même type peuvent partager un même volume logique

G4LogicalVolume

```
G4LogicalVolume(G4VSolid* pSolid, G4Material* pMaterial,  
                const G4String& name, G4FieldManager* pFieldMgr=0,  
                G4VSensitiveDetector* pSDetector=0,  
                G4UserLimits* pULimits=0,  
                G4bool optimise=true);
```



- Les pointeurs vers le solide et le matériau ne doivent pas être nuls
- Une fois créé, il est automatiquement placé dans le “LV store”
- N'est pas destiné à agir comme une classe de base

Géométrie : 4 – volumes physiques

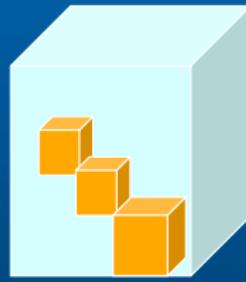
Volumes physiques

- “Placement” : un seul volume positionné



- “Repeated”: un volume répété plusieurs fois

- Peut représenter n’importe quel nombre de volumes *placement*
 - Réduit l’usage de mémoire
 - Volumes “Parameterised”
 - Répétition en fonction du numéro de copie
 - Volumes “Replica” et “Division”
 - Simple répétition selon un axe



repeated

- Un volume mère peut contenir soit

- Plusieurs volumes “placement” OU
 - Un volume “repeated”

G4VPhysicalVolume

- G4VPhysicalVolume est la classe de base des volumes physiques
 - Comme G4VSolid, c'est une classe abstraite
 - Il faut utiliser des classes dérivées pour placer les volumes logiques
- Implémentations de G4VPhysicalVolume
 - G4PVPlacement
 - G4PVParameterised
 - G4PVR replica
 - G4PVDivision

G4VPhysicalVolume

- Pour placer **un seul volume** : « **placement volume** »
 - Classe **G4PVPlacement**
 - Une instance de volume placé une fois dans un volume mère
- Pour placer **plusieurs volumes** : « **repeated volume** »
 - Classe **G4PVParameterised**
 - Paramétrisation par le **numéro de copie**
 - Forme, taille, matériau, position et rotation peuvent être paramétrisés en implémentant une classe concrète de **G4VPhysicalParameterisation**
 - Réduction de la consommation de mémoire
 - Une paramétrisation peut être utilisée uniquement pour des volumes qui
 - a) n'ont pas d'autres filles
 - ou b) sont identiques en taille et forme
 - Classes **G4PVReplica**, **G4PVDivision**
 - Découper un volume en petits morceaux (s'il existe une symétrie)₂₆

Implémentations de G4VPhysicalVolume

G4PVPlacement
G4PVParameterised
G4PVReplica
G4PVDivision

G4PVPlacement

```
G4PVPlacement(G4RotationMatrix* pRot,
               const G4ThreeVector& tlate,
               const G4String& pName,
               G4LogicalVolume* pCurrentLogical,
               G4VPhysicalVolume* pMotherPhysical,
               G4bool pMany,
               G4int pCopyNo,
               G4bool overlapCheck);
```

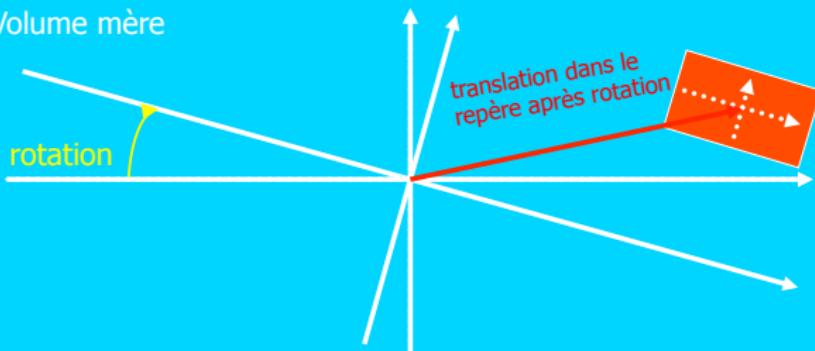
- **Volume unique positionné par rapport au volume mère**
 - dans un référentiel ayant subi une rotation et une translation par rapport au système de coordonnées du volume mère
- **Il existe trois constructeurs additionnels :**
 - Spécifier le volume mère comme un pointeur vers son volume logique au lieu de son volume physique
 - Utiliser `G4Transform3D` pour représenter la rotation et la translation directes du solide au lieu du référentiel
 - La combinaison des deux variantes ci-dessus

G4PVPlacement

```
G4ThreeVector boxPosition(100.*cm, 80.*cm, 0.*cm);
G4RotationMatrix* boxRotation = new G4RotationMatrix;
boxRotation->rotateZ(30.*deg); //Rotate around Z-axis
//Other Rotations : rotateX(angle), rotateY(angle)

G4VPhysicalVolume* boxPhys
= new G4PVPlacement(   boxRotation,
                      boxPosition,
                      "myBoxPhys"
                      boxLogical,
                      motherPhysical,
                      false,
                      0);
```

Volume mère



29

Géométrie :

5 - détecteur sensible

Pour quoi faire ?

Pour enregistrer des informations physiques (ex. perte d'énergie élémentaire) dans un élément de détecteur, on peut

- soit utiliser une classe d'action utilisateur de type **SteppingAction**
 - pratique pour les petites applications
 - à suivre... les « **hooks** »
- soit utiliser des détecteurs **sensibles** ou « **sensitive** »
 - la méthode du SteppingAction devient très « encombrée »
 - nécessite de subdiviser le problème, ce que propose le schéma des « **sensitive detectors** »

Principe

- S'applique à tout volume dans lequel on veut enregistrer des informations
 - volume de NaI où l'on enregistre l'énergie déposée pour ensuite estimer une production de lumière de scintillation
 - un plan de silicium où l'on enregistre aussi l'énergie déposée pour simuler la collection de charge
 - le corps d'un astronaute dans un vaisseau pour estimer les doses reçues
 - etc...
- Cette information à enregistrer est très dépendante des besoins de l'utilisateur
 - Geant4 propose donc un schéma général, abstrait
 - Mais aussi (depuis peu) des outils pour les besoins les plus courants

Sensibilité d'un détecteur

- Un volume logique devient **sensible** s'il possède un pointeur vers une classe concrète dérivée de `G4VSensitiveDetector`
- Un détecteur sensible
 - soit construit un ou plusieurs objets **coups (hits)**
 - soit accumule des valeurs à des coups existants
 - en utilisant la méthode `ProcessHit(...)` de cette classe dérivée et l'information venant d'un objet `G4Step`
- Un **coup (hit)** est un arrêt sur image de l'interaction physique d'une trace ou une accumulation d'interactions de traces dans la région sensible du détecteur

Remarque : il faut récupérer l'information sur le volume associé au step à partir du `PreStepPoint`

Géométrie :

6 - champ magnétique

Champ magnétique

- Créer une classe champ magnétique

- Champ uniforme: utiliser un objet de la classe `G4UniformMagField`

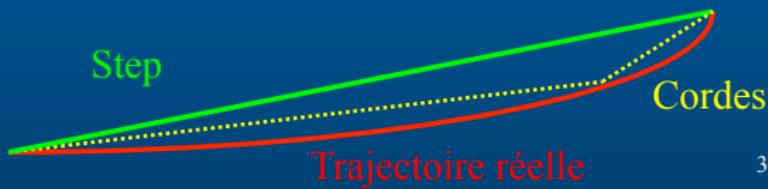
```
G4MagneticField* magField =  
  
new G4UniformMagField(G4ThreeVector(1.*Tesla,0.,0.));
```

- Champ non-uniforme: créer une classe concrète dérivée de `G4MagneticField` et implémenter la méthode `GetFieldValue`

```
void MyField::GetFieldValue(  
  
    const double Point[4], double *field) const  
  
    • Point[0..2] sont les positions dans le système de coordonnées global  
    • Point[3] est le temps  
    • field[0..2] renvoie les coordonnées du champ magnétique
```

Principe

- Pour propager une particule dans un champ, (ex. magnétique, électrique ou les deux), il faut résoudre l'équation du mouvement de la particule dans le champ
- On utilise une méthode de type Runge-Kutta pour l'intégration des équations différentielles du mouvement
 - Plusieurs méthodes Runge-Kutta sont disponibles dans Geant4
- Dans des cas spécifiques, d'autres steppers peuvent être utilisés :
 - Dans un champ uniforme, en utilisant la solution analytique
 - Dans un champ qui varie doucement, avec RK+approximation hélicoïdale
- Geant4 découpe le parcours en segments de corde linéaires, qui se rapprochent au plus possible du parcours



36

Exemple simple

- Dire à Geant4 d'utiliser notre champ

1. Récupérer le **Global Field Manager** à partir du **G4TransportationManager**

```
G4FieldManager* globalFieldMgr =  
    G4TransportationManager::GetTransportationManager()  
    ->GetFieldManager();
```

2. Déclarer le champ à ce Global Field Manager,

```
globalFieldMgr->SetDetectorField(magField);
```

3. Créer un **Chord Finder**.

```
globalFieldMgr->CreateChordFinder(magField);
```

- Optimiser les paramètres de la reconstruction de traces dans le champ de façon à ajuster la précision et la performance (ex. nanofaisceau VS planète).
 - Voir paramètres dans le **User's Guide For Application Developer**.

Géométrie : 7 - visualisation

Visualisation d'un détecteur

- Chaque volume logique peut avoir un objet **G4VisAttributes** associé
 - Visibilité, visibilité des volumes filles
 - Couleur, style de ligne, largeur de ligne
 - Modes “**wire-frame**” ou “**solid-style**”
- Pour les volumes paramétrisés, les attributs peuvent être assignés dynamiquement au volume logique
- Même durée de vie que les objets correspondants

Visualisation trajectoires et hits

- La classe **G4Trajectory** possède une méthode **Draw ()**
 - Bleu : particule chargée positivement
 - Vert : neutre
 - Rouge : négativement
 - On peut implémenter nos propres couleurs
- Dans le cas des détecteurs **sensibles**, chaque classe concrète dérivée de **G4VHit** doit fournir une implémentation de la méthode **Draw ()**
 - Marqueur coloré
 - Solide coloré
 - Change la couleur d'un élément de détecteur

Géométrie :

9 - fonctions avancées

Calculer volumes et masses

- Le **volume géométrique** d'un solide générique ou booléen peut être calculé à partir du **volume solide**

```
G4double GetCubicVolume();
```

- De même la **masse** peut être calculée à partir du **volume logique**

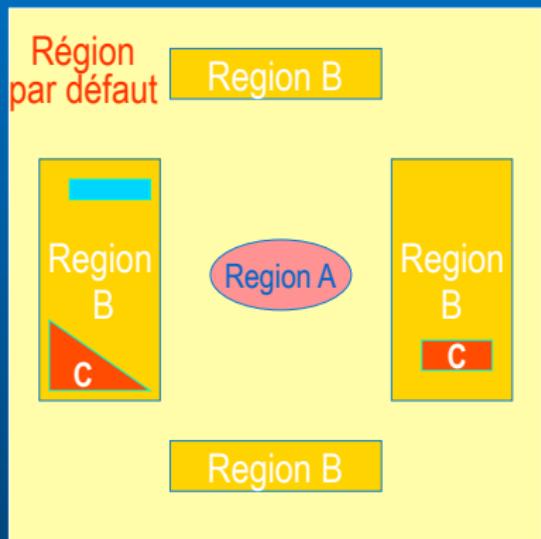
```
G4double GetMass(G4Bool forced=false,  
                  G4Material* parameterisedMaterial=0);
```

“Cuts” par région

- Geant4 possède un seuil (**‘cut’**) unique de production de particules secondaires exprimé en longueur (i.e. parcours minimum des secondaires)
 - Pour tous les volumes
 - Peuvent être différents pour chaque particule (e+, e-, gamma)
- On peut avoir des échelles de longueur très différentes au sein d'un même grand détecteur
 - ex. un détecteur de vertex (5 mm) et un détecteur à muons (2.5 cm)
 - avoir un cut faible unique peut créer des pénalités de performance
- Geant4 autorise plusieurs cuts
 - Globalement ou par particule
 - Au niveau d'un sous détecteur en utilisant des “régions”
 - Les cuts s'appliquent seulement aux gamma, électrons et positrons et seulement pour les processus qui ont une divergence infrarouge

Région d'un détecteur

- Concept de région
 - ensemble de volumes de la géométrie, typiquement un sous-système
 - barrique + “end-caps” d'un calorimètre
 - zones “profondes” de la structure
 - ou tout groupe de volumes
- Un ensemble de “cuts in range” unique est associé à une région
 - on peut utiliser un cut différent pour chaque particule gamma, e-, e+ dans une région



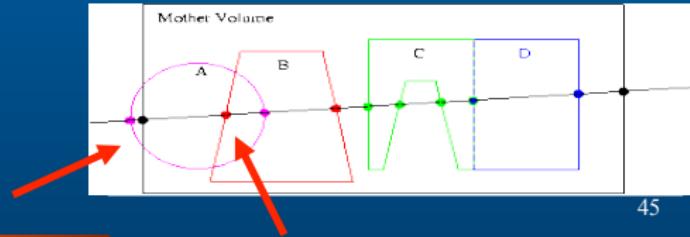
Debugger les géométries

- Un **volume overlapping** est un volume contenu qui **déborde** de son volume mère. Des volumes peuvent se **chevaucher** par erreur.



Geant4 refuse les géométries mal formées

- Le problème de détecter les chevauchements entre volumes est lié à la complexité de la description des modèles solides
- Des utilitaires sont fournis pour détecter les chevauchements
 - Graphiques (DAVID, OLAP)
 - Commandes du kernel (à la construction – cf. **G4PVPlacement**, à l'exécution)



45

Géométrie :

9 - exemples de géométries

Détecteur LHCb au CERN

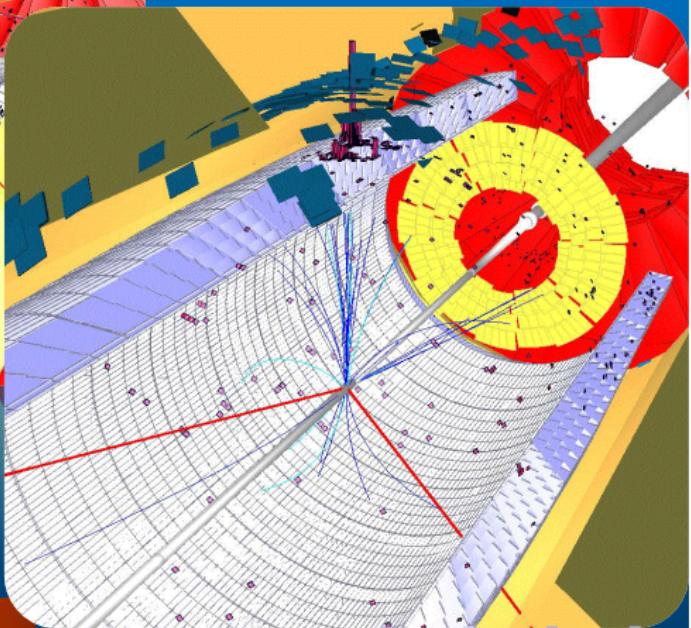
Courtesy of the LHCb Collaboration

Localisateur de vertex

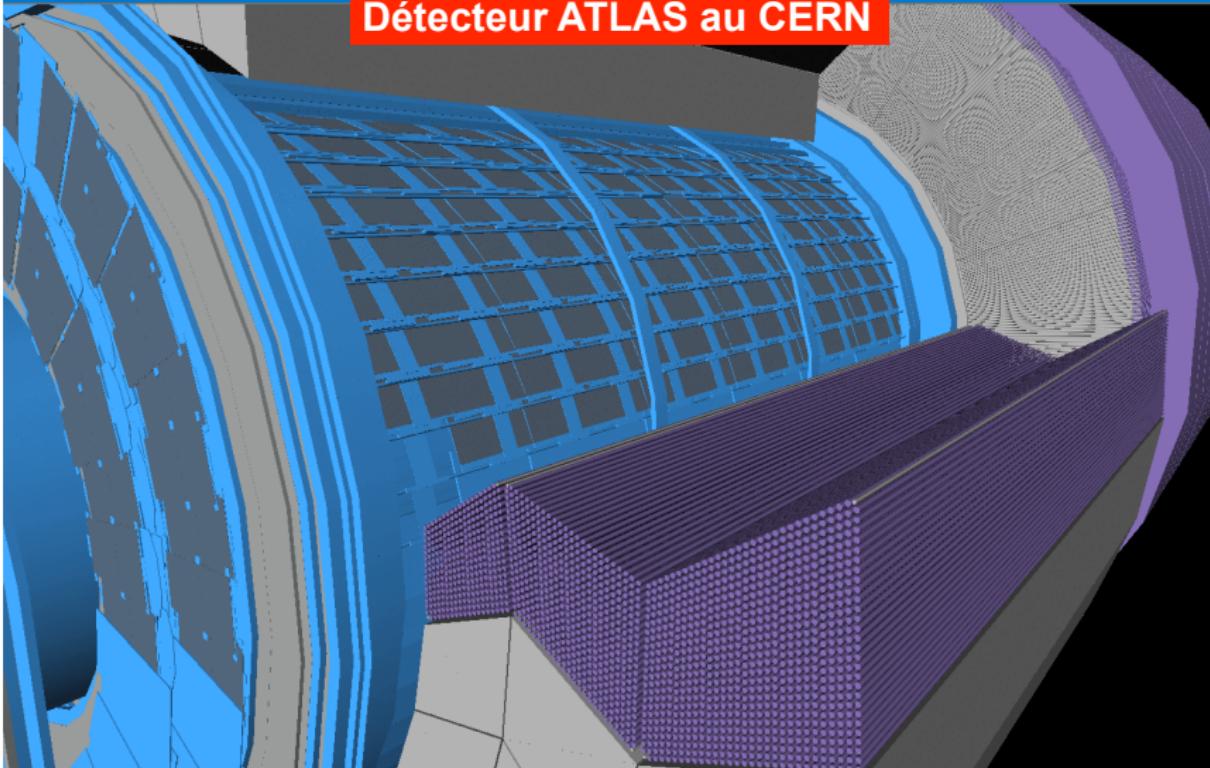
Détecteur CMS au CERN

Courtesy of the CMS Collaboration

Événements SUSY

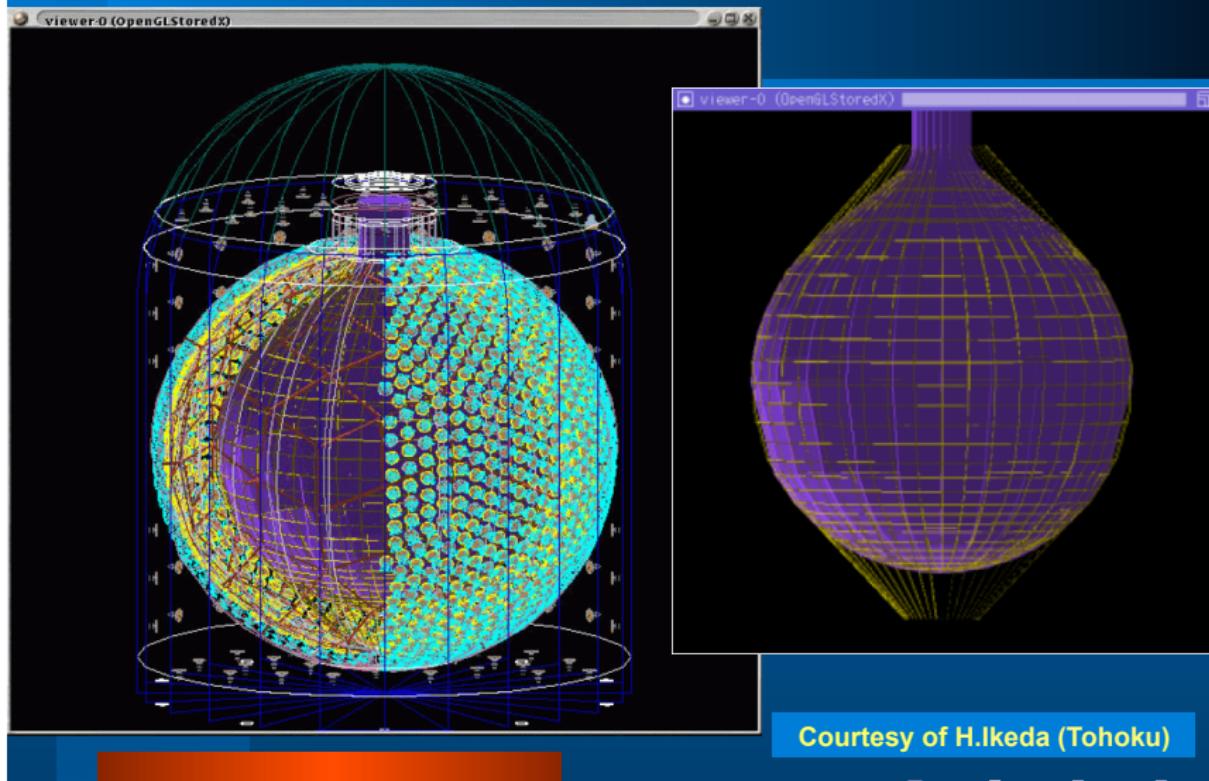


Détecteur ATLAS au CERN

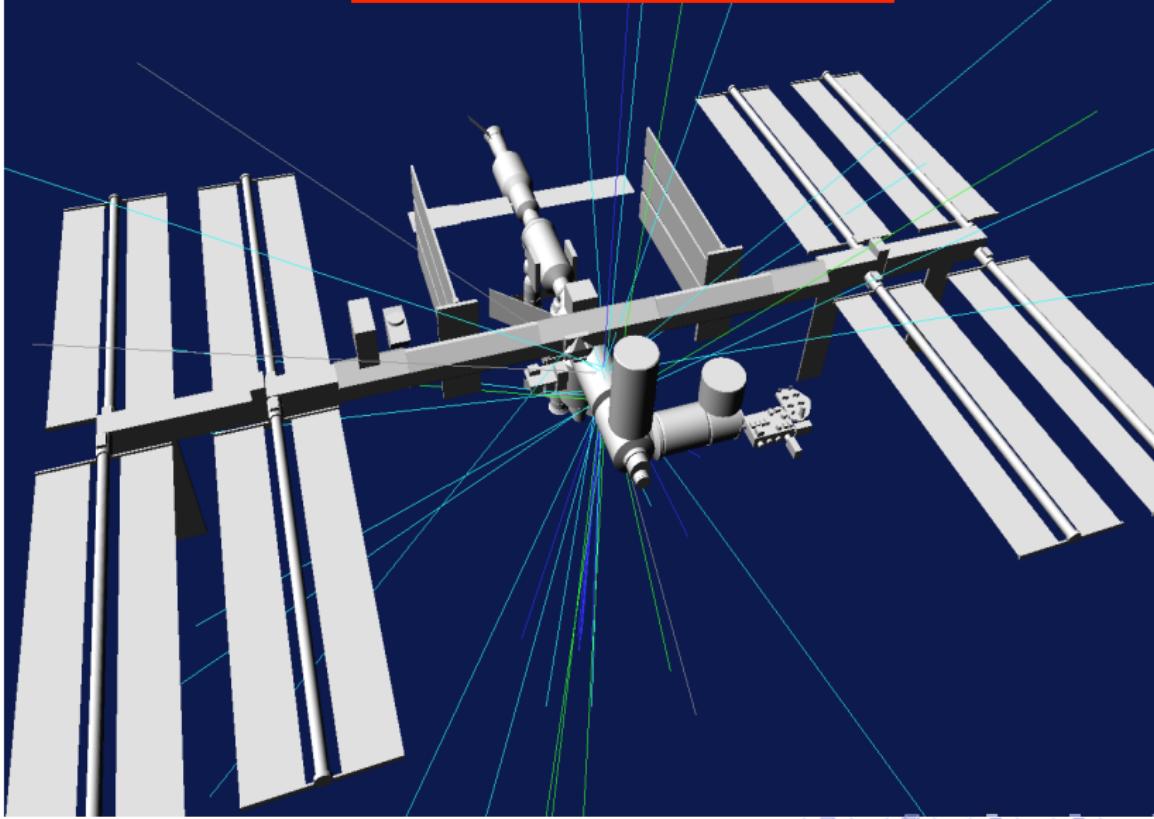


Courtesy of the ATLAS Collaboration

Détecteur anti-neutrino – Kamioka scintillateur liquide



International Space Station



Courtesy T. Ersmark, KTH Stockholm

Compléments

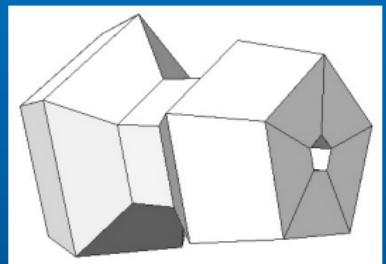


52

BREPS: G4BREPSSolidPolyhedra



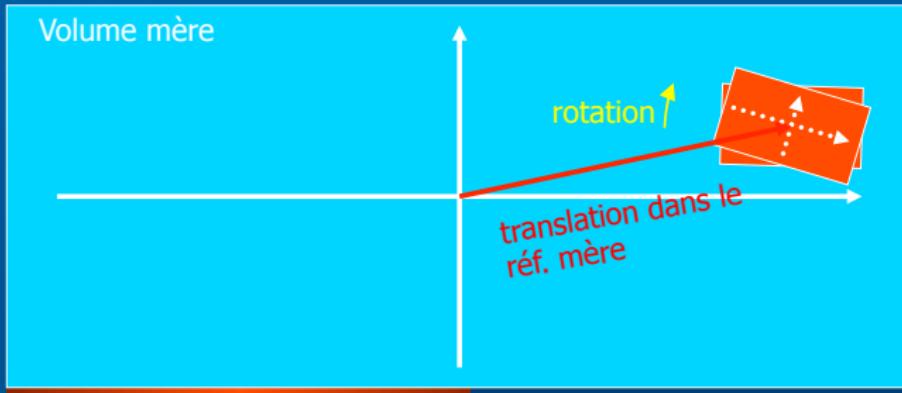
```
G4BREPSSolidPolyhedra(const G4String& pName,
                      G4double phiStart,
                      G4double phiTotal,
                      G4int sides,
                      G4int nZplanes,
                      G4double zStart,
                      const G4double zval[],
                      const G4double rmin[],
                      const G4double rmax[]);
```



- **sides** = nombres de côtés de chaque polygone dans le plan x-y
- **nZplanes** = nombre de plans perpendiculaires à l'axe z
- **zval[]** = coordonnées z de chaque plan
- **rmin[], rmax[]** = rayons des polygones interne et externe dans chaque plan

G4PVPlacement alternatif

```
G4PVPlacement(  
    G4Transform3D(G4RotationMatrix &pRot, // rotation du réf. fille  
                  const G4ThreeVector &tlate), // position dans le ref. mère  
    G4LogicalVolume *pDaughterLogical,  
    const G4String &pName,  
    G4LogicalVolume *pMotherLogical,  
    G4bool pMany,  
    G4int pCopyNo,  
    G4bool pSurfChk=false);
```



Détecteur sensible et hit

- Chaque volume logique peut avoir un pointeur vers un détecteur sensible
- Un coup (hit) est un arrêt sur image de l'interaction physique d'une trace ou une accumulation d'interactions de traces dans la région sensible du détecteur
- Un détecteur sensible crée des hits en utilisant l'information venant de G4Step. L'utilisateur doit fournir sa propre implémentation de la réponse du détecteur.
- Les objets hits, qui sont des objets de classe utilisateur, sont collectés dans un objet G4Event à la fin d'un événement
- Il ne faut pas le faire dans la classe UserSteppingAction

Définir un détecteur sensible

- Stratégie de base

```
G4LogicalVolume* myLogCalor = .....;

G4VSensitiveDetector* pSensitivePart =
    new MyCalorimeterSD("/mydet/calorimeter");

G4SDManager* SDMan = G4SDManager::GetSDMpointer();

SDMan->AddNewDetector(pSensitivePart);

myLogCalor->SetSensitiveDetector(pSensitivePart);
```

Classe hit - 1

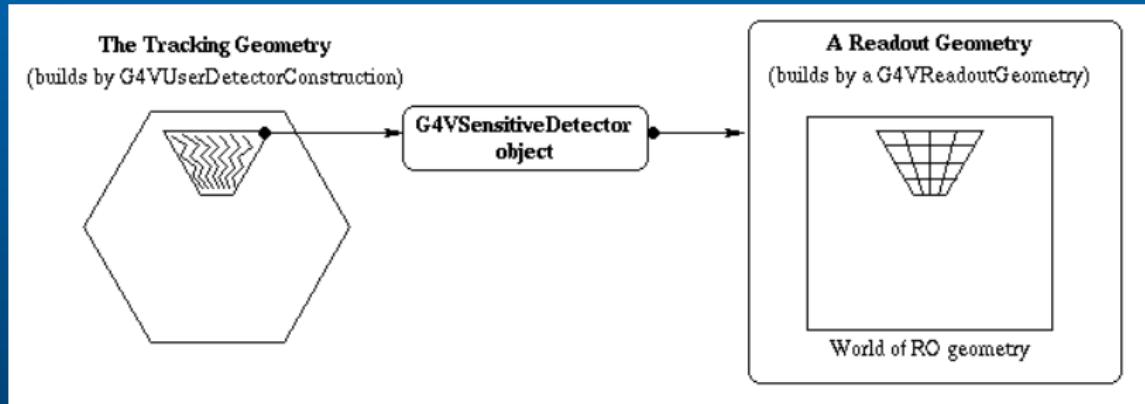
- Un **coup (hit)** est une classe utilisateur dérivée de **G4VHit**
- On peut stocker plusieurs types d'informations en implémentant notre propre classe concrète **classe hit**
- Par exemple:
 - Position et temps du step
 - Moment et énergie du track
 - Dépôt d'énergie du step
 - Information géométrique
 - ou toute combinaison des informations précédentes

Classe hit - 2

- Les objets hit d'une classe concrète hit doivent être stockés dans une collection dédiée qui est instanciée à partir de la classe patron `G4THitsCollection`
- La collection sera associée à un objet `G4Event` via `G4HCofThisEvent`
- Les collections de hits sont accessibles
 - à travers `G4Event` à la fin d'un event,
 - Utile pour analyser un événement
 - à travers `G4SDManager` pendant un event
 - Utile pour le filtrage d'event

Géométrie de lecture (readout)

- Une géométrie de lecture est une géométrie virtuelle et artificielle qui peut être définie parallèlement à la vraie géométrie du détecteur
- C'est une option
- Chacune est associée à un détecteur sensible



Digitisation

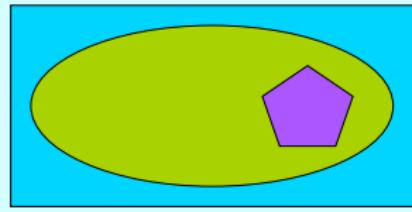
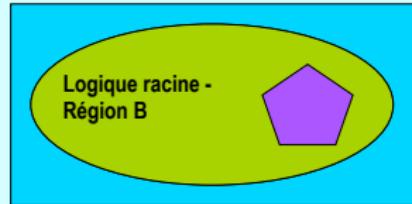
- Un **digit** représente la réponse d'un détecteur (ex. Un comptage ADC/TDC, un signal de déclenchement)
- Un digit est créé avec un ou plusieurs hits et/ou d'autres digits par une implémentation concrète dérivée de `G4VDigitizerModule`
- Au contraire du hit qui est généré automatiquement lors du suivi des particules, la méthode `digitize()` de chaque `G4VDigitizerModule` doit être explicitement invoquée par le code utilisateur (ex. `EventAction`)

Region et cut

- Chaque région a son unique ensemble de cuts.
- Le volume monde est reconnu comme la région par défaut. Les cuts par défaut sont définis dans la **Physics list**.
 - L'utilisateur ne peut pas définir une région du volume monde ou un cut dans la région par défaut
- Un **volume logique** devient un **volume logique racine** une fois qu'il est assigné à une région.
 - Tous les volumes filles appartenant au volume logique racine partagent la même région (et cuts) à moins qu'un volume fille appartienne à une autre volute logique racine
- **Restriction importante :**
 - Aucun volume logique ne peut être partagé par plus d'une région

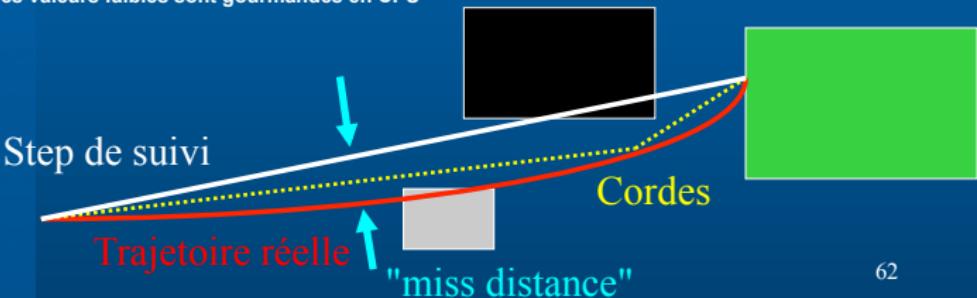
Volume monde – Région par défaut

Logique racine - Région A



Propagation

- On utilise les **cordes** pour interroger le **G4Navigator**, pour vérifier si le track a traversé une limite de volume.
- Un step de physique ou de tracking peut générer plusieurs **cordes**.
 - Dans certains cas, un step consiste en plusieurs tours
- L'utilisateur peut régler la précision de l'intersection avec un volume
 - En choisissant un paramètre appelé "**miss distance**"
 - C'est une mesure de l'erreur si le track approximé intercepte un volume
 - Des valeurs faibles sont gourmandes en CPU



Implémentations de G4VPhysicalVolume

G4PVPlacement
G4PVParameterised
G4PVReplica
G4PVDivision

Volumes physiques paramétrisés

- Des fonctions utilisateur définissent
 - les dimensions du solide
 - Fonction `ComputeDimensions`
 - où il est positionné
 - Fonction `ComputeTransformations`
- Options
 - le type de solide
 - Fonction `ComputeSolid`
 - le matériau
 - Fonction `ComputeMaterial`
- Limitations
 - S'applique aux solides simples CSG uniquement
 - Les volumes filles sont autorisés uniquement pour des cas spéciaux
- Très puissant
 - Considérer les volumes paramétrisés comme des volumes “feuilles”⁶⁴



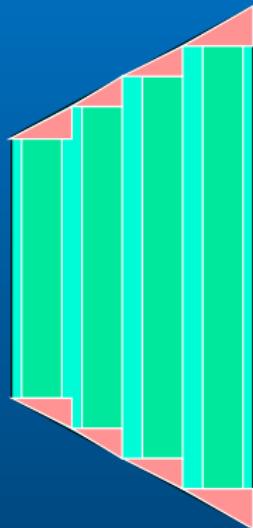
Utilisation

- **DéTECTEURS complexes**

- Avec de nombreuses répétitions de volumes
 - Régulièrement ou pas

- **APPLICATIONS médicaLES**

- Fantômes humains, cellulaires...
 - parallélépipèdes avec des matériaux différents



G4PVParameterised

```
G4PVParameterised(const G4String& pName,  
                   G4LogicalVolume* pCurrentLogical,  
                   G4LogicalVolume* pMotherLogical,  
                   const EAxis pAxis,  
                   const G4int nReplicas,  
                   G4VPVParameterisation* pParam);
```

- Réplique le volume `nReplicas` fois en utilisant la paramétrisation `pParam`, à l'intérieur du volume mère
- Le positionnement des répliques est dominant le long de l'axe cartésien spécifié
 - si `kUndefined` est spécifié comme axe, une voxélisation 3D est adoptée pour optimisation de la géométrie
- Représente plusieurs éléments de détecteur, qui diffèrent par leur position et leurs dimensions, calculées au moyen d'objets `G4VPVParameterisation`
- Constructeur alternatif en utilisant un pointeur vers le volume physique de la mère

Paramétrisation : exemple

```
G4VSolid* solidChamber = new G4Box("chamber", 100*cm, 100*cm, 10*cm);
G4LogicalVolume* logicChamber =
    new G4LogicalVolume(solidChamber, ChamberMater, "Chamber", 0, 0, 0);
G4double firstPosition = -trackerSize + 0.5*ChamberWidth;
G4double firstLength = fTrackerLength/10;
G4double lastLength = fTrackerLength;
G4VPVParameterisation* chamberParam =
    new ChamberParameterisation( NbOfChambers, firstPosition,
                                ChamberSpacing, ChamberWidth,
                                firstLength, lastLength);

G4VPhysicalVolume* physChamber =
    new G4PVParameterised( "Chamber", logicChamber, logicTracker,
```

kZAxis,

Utiliser **kUndefined** pour activer la voxélisation 3D pour optimisation

67

Paramétrisation : exemple - suite

```
class ChamberParameterisation : public G4VPVParameterisation
{
public:
    ChamberParameterisation( G4int NoChambers, G4double startZ,
                            G4double spacing, G4double widthChamber,
                            G4double lenInitial, G4double lenFinal );
    ~ChamberParameterisation();
    void ComputeTransformation (const G4int copyNo,
                               G4VPhysicalVolume* physVol) const;
    void ComputeDimensions (G4Box& trackerLayer, const G4int copyNo,
                           const G4VPhysicalVolume* physVol) const;
}
```

Paramétrisation : exemple - fin

```
void ChamberParameterisation::ComputeTransformation
(const G4int copyNo, G4VPhysicalVolume* physVol) const
{
    G4double Zposition= fStartZ + (copyNo+1) * fSpacing;
    G4ThreeVector origin(0, 0, Zposition);
    physVol->SetTranslation(origin);
    physVol->SetRotation(0);
}

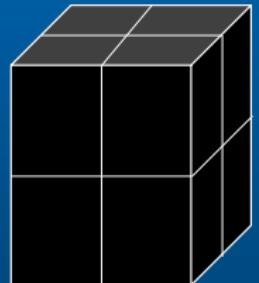
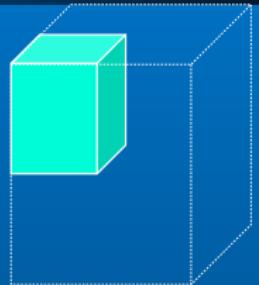
void ChamberParameterisation::ComputeDimensions
(G4Box& trackerChamber, const G4int copyNo,
 const G4VPhysicalVolume* physVol) const
{
    G4double halfLength= fHalfLengthFirst + copyNo * fHalfLengthIncr;
    trackerChamber.SetXHalfLength(halfLength);
    trackerChamber.SetYHalfLength(halfLength);
    trackerChamber.SetZHalfLength(fHalfWidth);
}
```

Implémentations de G4VPhysicalVolume

G4PVPlacement
G4PVParameterised
G4PVReplica
G4PVDivision

Volumes physiques répliqués

- Le volume mère est segmenté en répliques, toutes de la même taille et de mêmes dimensions
- Représente de nombreux éléments de détecteurs qui diffèrent seulement par leur position
- La réPLICATION peut avoir lieu le long :
 - Des axes cartésiens (X, Y, Z) : les tranches sont perpendiculaires à l'axe de réPLICATION
 - Système de coordonnées au centre de chaque réplique
 - Axe radial (Rho) – sections de cons/tubs centrées sur l'origine et orientation non changée
 - Même système de coordonnées que la mère
 - Axe Phi (Phi) – sections en phi ou cales de cons/tubs
 - Système de coordonnées choisi de façon à ce que l'axe X soit la bissectrice the l'angle créé par chaque cale



repeated

71



Un volume fille à répliquer



mother volume

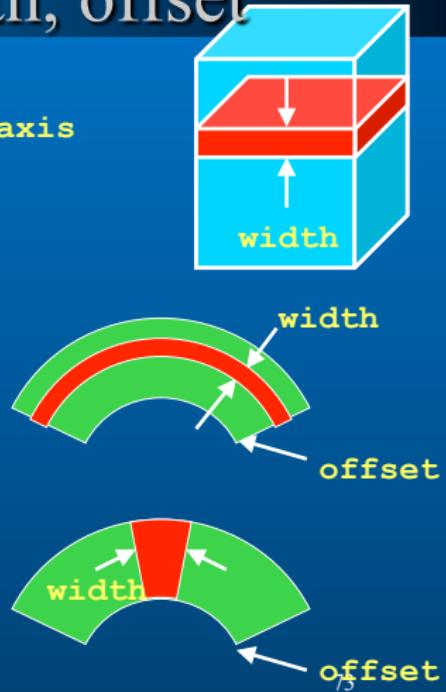
```
G4PVReplica(const G4String& pName,  
             G4LogicalVolume* pCurrentLogical,  
             G4LogicalVolume* pMotherLogical,  
             const EAxis pAxis,  
             const G4int nReplicas,  
             const G4double width,  
             const G4double offset=0);
```

- Constructeur alternatif : utiliser un pointeur vers le volume physique mère
- Un **offset** peut seulement être associé à un offset mère le long de l'axe de réPLICATION
- Possibilités et restrictions:
 - Des répliques peuvent être placées à l'intérieur d'autres répliques
 - Des volumes peuvent être placés (placement) à l'intérieur de répliques, en supposant qu'il n'y a pas d'intersection/chevauchement avec le volume mère ou d'autres répliques
 - Aucun volume ne doit être placé à l'intérieur d'une réplique radiale
 - Des volumes paramétrisés ne peuvent pas être placés dans une réplique

72

Répliques : axes, width, offset

- Axes cartésiens - **kXaxis**, **kYaxis**, **kZaxis**
 - Ne pas utiliser **offset**
 - Centre de la $n^{\text{ième}}$ fille donné par
$$-width * (nReplicas-1) * 0.5 + n * width$$
- Axe radial - **kRaxis**
 - Centre de la $n^{\text{ième}}$ fille donné par
$$width * (n+0.5) + offset$$
- Axe phi - **kPhi**
 - Centre de la $n^{\text{ième}}$ fille donné par
$$width * (n+0.5) + offset$$



Répliques : exemples

```
G4double tube_dPhi = 2.* M_PI;

G4VSolid* tube =
    new G4Tubs("tube", 20*cm, 50*cm, 30*cm, 0., tube_dPhi*rad);

G4LogicalVolume * tube_log =
    new G4LogicalVolume(tube, Ar, "tubeL", 0, 0, 0);

G4VPhysicalVolume* tube_phys =
    new G4PVPlacement(0,G4ThreeVector(-200.*cm, 0., 0.*cm),
                      "tubeP", tube_log, world_phys, false, 0);

G4double divided_tube_dPhi = tube_dPhi/6.;

G4VSolid* divided_tube =
    new G4Tubs("divided_tube", 20*cm, 50*cm, 30*cm,
               -divided_tube_dPhi/2.*rad, divided_tube_dPhi*rad);

G4LogicalVolume* divided_tube_log =
    new G4LogicalVolume(divided_tube, Ar, "div_tubeL", 0, 0, 0);

G4VPhysicalVolume* divided_tube_phys =
    new G4PVR replica("divided_tube_phys", divided_tube_log, tube_log,
                      kPhi, 6, divided_tube_dPhi);
```

Implémentations de G4VPhysicalVolume

G4PVPlacement
G4PVParameterised
G4PVReplica
G4PVDivision

G4PVDivision

- Type “spécial” des volumes paramétrisés
 - S’applique uniquement aux volumes CSG (box, tubs, cons, para, trd, polycone, polyhedra)
 - Divise un volume en des copies identiques le long de son axe (les copies ne sont pas strictement identiques)
 - ex. un tube divisé le long de son axe radial
 - on peut spécifier des offsets
- Les axes possibles de division varient selon le type de solide supporté
- G4PVDivision est la classe définissant cette division
 - La paramétrisation est calculée automatiquement en utilisant les valeurs fournies en entrée

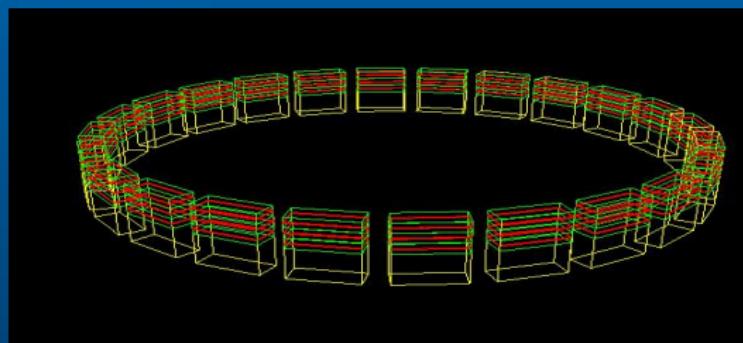
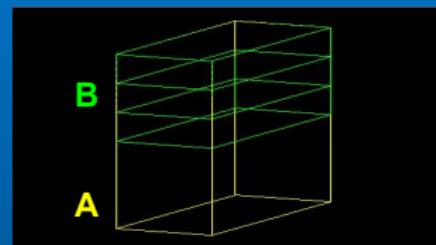
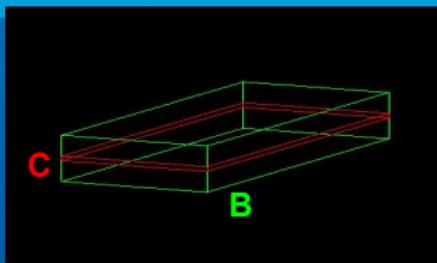
Objets en mouvement

- Dans certaines applications, il est nécessaire de simuler le mouvement de certains volumes
 - ex. Simulations en radiothérapie
- Geant4 peut traiter des volumes mobiles
 - Dans le cas où la vitesse du volume mobile est faible devant la vitesse des particules élémentaires, on suppose que la position du volume est fixe pendant un événement
- Deux astuces pour simuler les objets mobiles :
 1. Utiliser les volumes paramétrisés pour représenter le volume mobile
 2. Ne pas optimiser (voxelliser) le volume mère des volumes mobiles

Géométrie : touchables

Identifier un volume ?

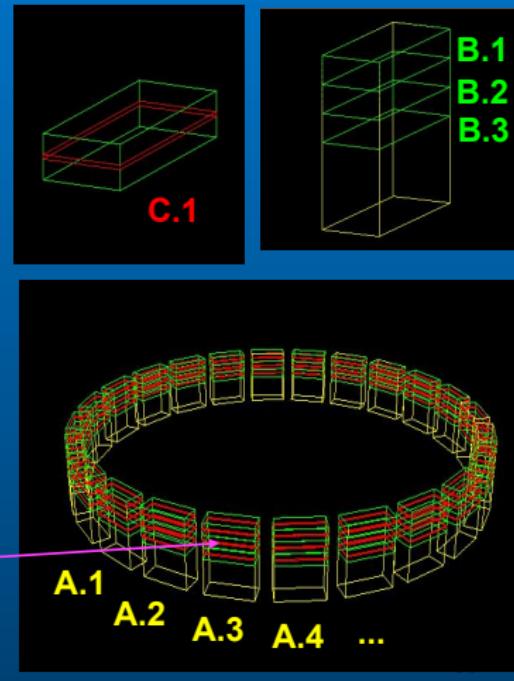
- Supposons qu'une géométrie contienne des couches sensibles C placées dans un volume B
- Le volume B est volume fille du volume divisé A



- le volume A a 24 positions dans le monde
- alors que dans "l'arbre logique" de la géométrie le volume C est représenté par un seul volume physique, dans la simulation il y aura plusieurs volumes C
- Alors comment identifier ces volumes C ?

Les touchables

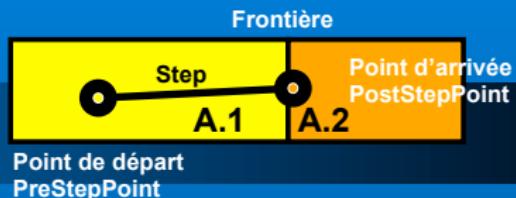
- Un **touchable** pour un volume permet de fournir un **unique identificateur** pour un élément de volume
- C'est une entité géométrique (volume ou solide) qui a un unique positionnement
 - Elle peut être identifiée en fournissant les numéros de copie de toutes les filles dans la hiérarchie géométrique
 - Dans notre cas
 - **CopyNo** de C dans B: 1
 - **CopyNo** de B dans A: 1,2,3
 - **CopyNo** de A dans le monde: 1, .., 24
 - Exemple d'identification de touchable:
 - A.3/B.2/C.1



Que peut faire un touchable ?

- **G4VTouchable** est la classe de base pour toutes les implémentations des touchables. Elle définit les méthodes suivantes auxquelles les touchables doivent répondre. **Depth** représente toujours le numéro du niveau à considérer dans l'arbre géométrique:
 - depth = 0 : niveau de **base** (volume C dans B)
 - depth = 1 : niveau du **volume mère** (volume B dans A)
 - depth = 2 : volume **grand-mère** (volume A dans le monde)
- **GetCopyNumber(G4int depth =0)**
 - Renvoie le numéro de copie au niveau spécifié
- **GetTranslation(G4int depth = 0)**
GetRotation(G4int depth=0)
 - Renvoie les composantes de transformation du volume
- **GetSolid(G4int depth =0)**
 - Renvoie le volume solide
- **GetVolume(G4int depth =0)**
 - Renvoie le volume physique

Touchables



- Comme déjà mentionné, G4Step possède deux objets G4StepPoint : un point de départ et un point d'arrivée.
- Toutes les informations géométriques du step doivent être prises à partir du PreStepPoint
- Chaque objet G4StepPoint possède
 - Une position dans les coordonnées du monde
 - Un temps global et un temps local
 - Un matériau
 - Un objet G4TouchableHistory qui contient l'information sur la hiérarchie de la géométrie
- G4TouchableHistory est un vecteur d'information pour chaque hiérarchie de géométrie
 - Numéro de copie
 - Transformation / rotation par rapport au volume mère

L'outil Geant4 :

Physique électromagnétique (QED) STANDARD



PhysicsList.hh (.cc)



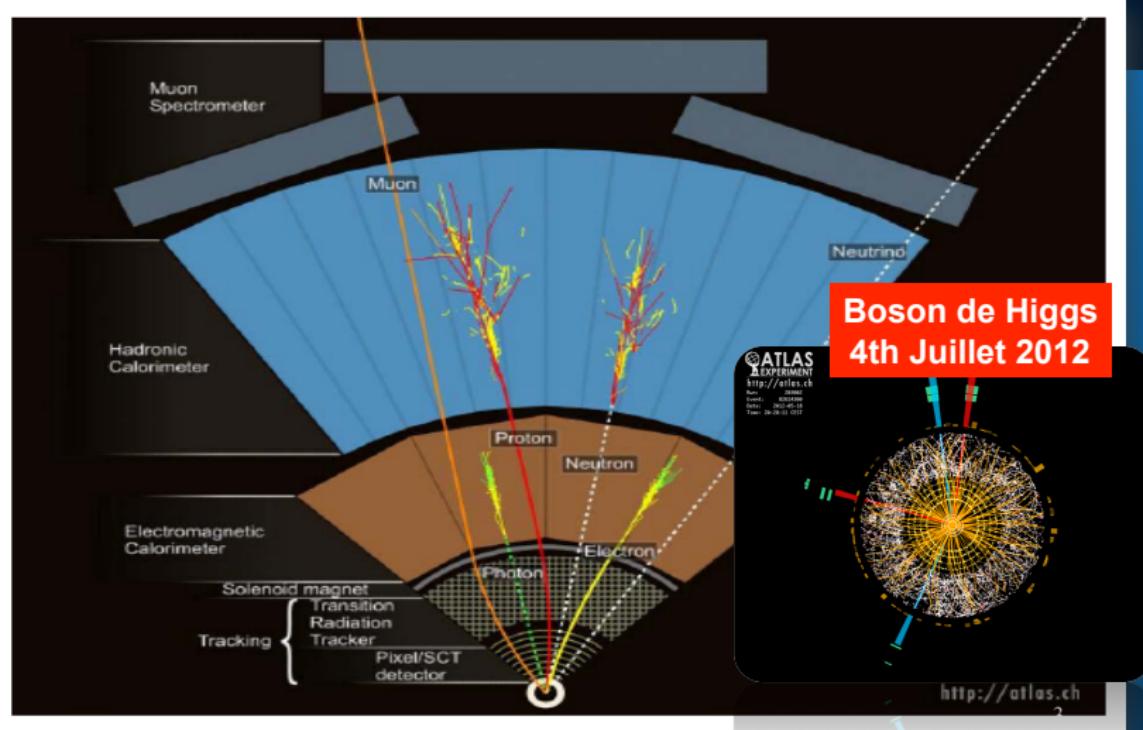
25 minutes

1

La physique EM de Geant4

- Apparue avec la première version de Geant4, basée sur l'expérience acquise avec Geant3 (1998)
- Pendant de nombreuses années, Physique des Hautes Energies
 - BaBar, SLAC (depuis 2000)
 - Expériences du LHC: ATLAS, CMS et LHCb (depuis 2004)
- De nombreux points communs avec le spatial, le médical et autres
- Une page web unique pour la physique EM
 - http://cern.ch/geant4/collaboration/working_groups/electromagnetic/index.shtml
 - Inclut une interface web vers un “repository” dédié à la validation

Simulation Geant4 d'ATLAS au LHC, CERN

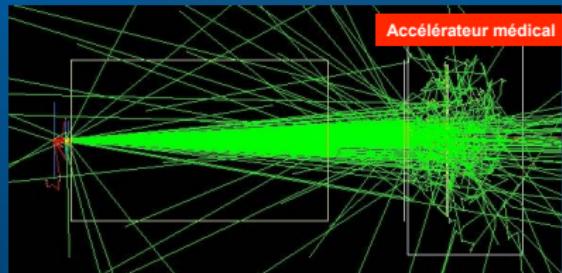
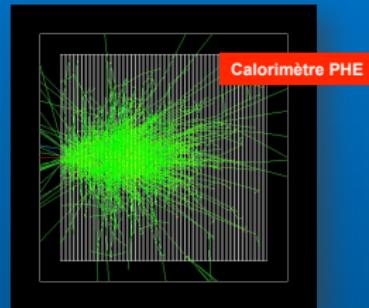


Sous-catégories électromagnétiques

- **Standard**
 - γ , e jusqu'à 100 TeV
 - hadrons jusqu'à 100 TeV
 - ions jusqu'à 100 TeV
- **Muons**
 - jusqu'à 1 PeV
- **X**
 - processus de productions de rayons X et de photons optiques
- **Haute énergie**
 - processus à haute énergie ($E > 10$ GeV)
 - physique des particules exotiques
- **Polarisation**
 - simulation de faisceaux polarisés
- **Optique**
 - interactions des photons optiques
- **Basse énergie**
 - librairie Livermore γ , e- de 250 eV à 1 GeV
 - librairie Livermore basée sur les processus polarisés
 - réécriture de PENELOPE, γ , e-, e+ de 100 eV à 1 GeV (version 2008)
 - hadrons et ions jusqu'à 1 GeV
 - microdosimétrie pour la radiobiologie (Geant4-DNA) de 7 eV à qq 100 MeV
 - désexcitation atomique (fluorescence + Auger)
- **Adjoint**
 - Monte Carlo à l'envers (du détecteur à la source)
- **Utils** : interfaces EM générales

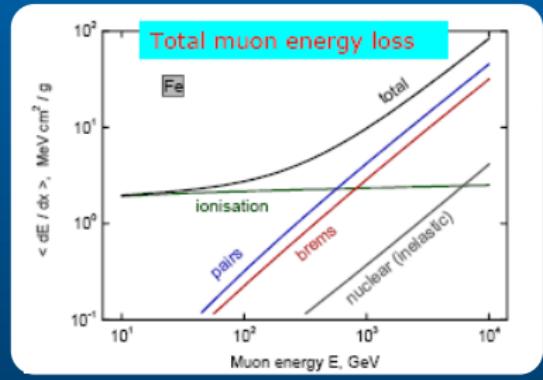
Transport des **gammas** et des **électrons**

- **Processus des gammas**
 - Création de paire $\gamma \rightarrow e^+e^-$
 - Diffusion Compton
 - Effet photoélectrique
 - Diffusion Rayleigh
 - (*Interactions gamma-noyau dans le package hadronique "CHIPS"*)
- **Processus des électrons et positrons**
 - Ionisation
 - Diffusion Coulombienne
 - Bremsstrahlung
 - Annihilation de positrons
 - (*Interactions nucléaires dans le package hadronique "CHIPS"*)
- **Adapté à la PHE & à d'autres domaines d'application utilisant électrons et gammas**



Muons

- Ionisation
- Rayonnement de freinage
- Production de paire e+e-
- (Interactions muon-noyau dans les packages hadroniques)



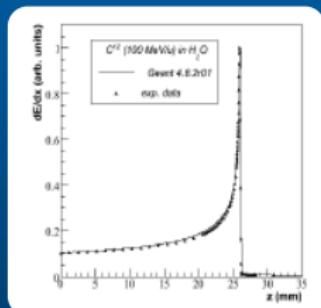
Hadrons et ions

- Diffusion coulombienne
- Ionisation
 - Formule de Bethe-Bloch corrigée pour E>2 MeV

$$-\frac{dE}{dx} = 4\pi N_e r_0^2 \frac{z^2}{\beta^2} \left(\ln \frac{2m_e c^2 \beta^2 \gamma^2}{I} - \frac{\beta^2}{2} \left(1 - \frac{T_c}{T_{\max}} \right) - \frac{C}{Z} + \frac{G - \delta - F}{2} + zL_1 + z^2 L_2 \right)$$

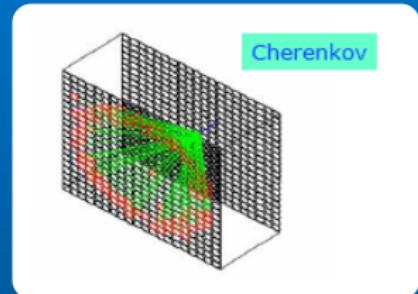
- C – shell correction
- G – Mott correction
- δ – density correction
- F – finite size correction
- L_1 – Barkas correction
- L_2 – Bloch correction
- Nuclear stopping
- Ion effective charge

- Paramétrisations du pic de Bragg pour E<2 MeV
(bases de données ICRU'49, 73 et NIST)



Rayons X et photons optiques

- Package standard
 - Radiation Cherenkov
 - Radiation synchrotron
 - Radiation de transition
 - Scintillation
- Physique EM de basse énergie
 - Relaxation atomique (fluorescence et transitions Auger)
- Optique
 - Réflexion
 - Réfraction
 - Absorption
 - Diffusion Rayleigh



Exemples de processus

La diffusion Compton

- Décrit la diffusion de photons sur des électrons atomiques quasi-libres



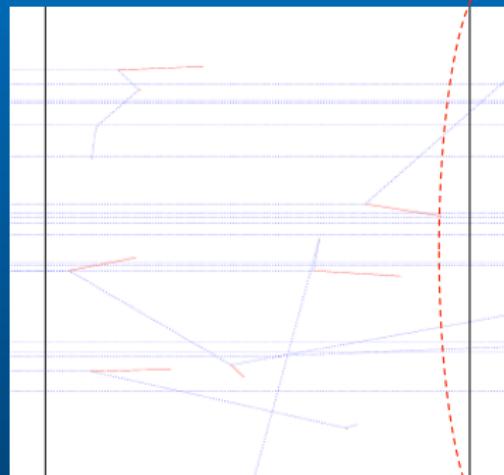
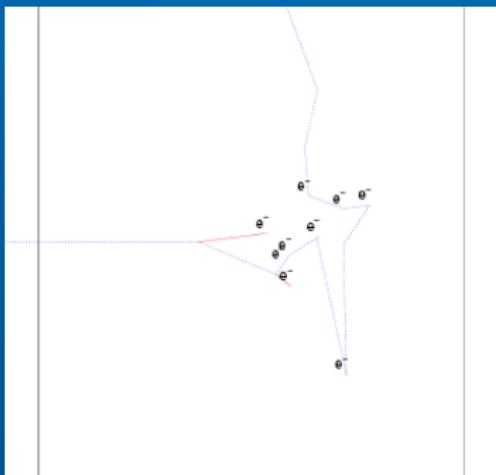
- Chaque électron se comporte comme une cible indépendante : c'est une diffusion incohérente

Section efficace par atome =
 $Z \times$ section efficace par électron

- La section efficace différentielle en énergie diffusée est décrite par la formule de Klein-Nishina, corrigée par un facteur prenant en compte l'énergie de liaison de l'électron atomique
- Relié à l'annihilation (e^+, e^-) par symétrie de croisement

Illustration de l'effet Compton

γ de 10 MeV dans 10 cm d'Al



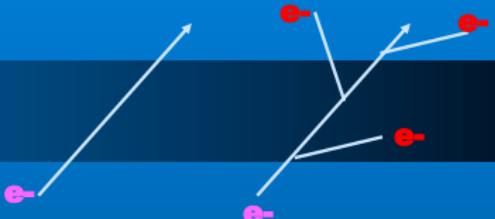
Ionisation

- Le mécanisme de base est une collision inélastique de la particule chargée en mouvement avec les électrons atomiques du matériau, éjectant un électron de l'atome



- Dans chaque collision individuelle, l'énergie transférée à l'électron est petite. Mais le nombre total de collisions est grand et on peut définir une perte d'énergie moyenne par unité macroscopique de longueur de parcours

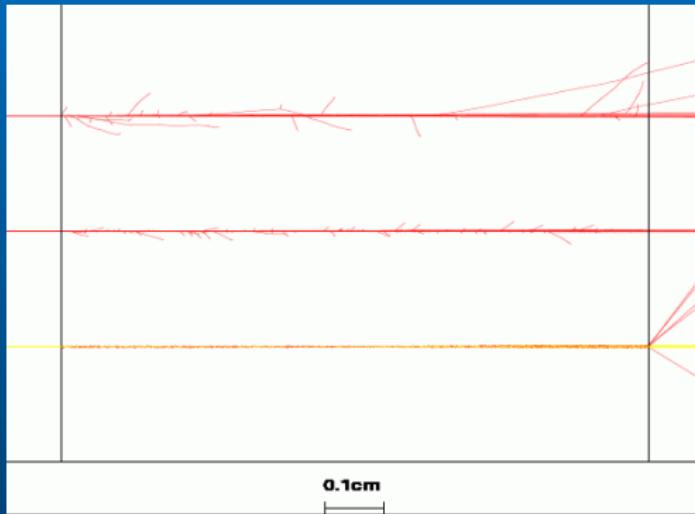
Génération des électrons secondaires (delta rays)



- Dans Geant4, on peut choisir de simuler ou non les électrons éjectés (ou « delta electrons ») d'énergies les plus hautes (et leurs interactions) au-dessus d'un seuil T_{cut}
 - **Sous ce seuil**, les électrons sont comptabilisés dans la perte d'énergie continue de la particule ionisante et ne sont pas explicitement simulés
 - **Au-dessus**, ils sont explicitement générés. Ils sont alors exclus du calcul de la perte d'énergie contenue de la particule ionisante.
- La perte d'énergie continue est calculée à partir de la formule de Bethe et Bloch « tronquée » (dE/dx pour $T < T_{cut}$)

Illustration des delta rays

e-, p, α de 200 MeV dans 1 cm d'Al



14

Fluctuations de perte d'énergie (straggling)

- La formule de Bethe et Bloch donne seulement la **perte d'énergie moyenne** par ionisation. Mais il existe des **fluctuations** (« straggling en énergie »): la distribution de la perte d'énergie ΔE dans une épaisseur Δx peut être très dissymétrique
 - distribution de Landau
- Les grandes fluctuations sont causées par des collisions peu fréquentes ayant de grands transferts d'énergie

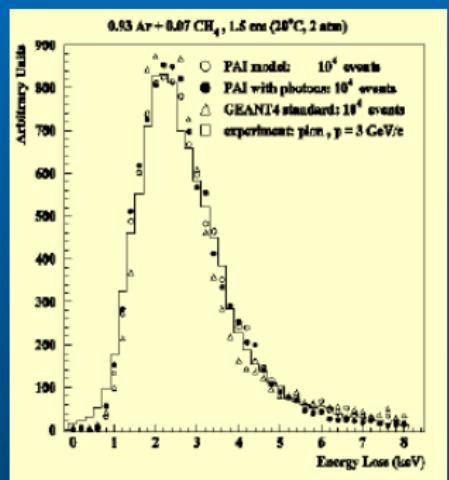
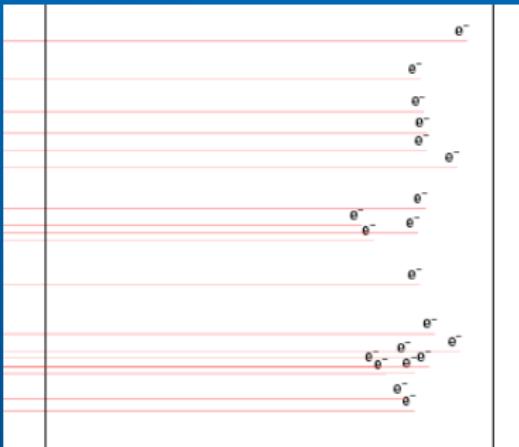
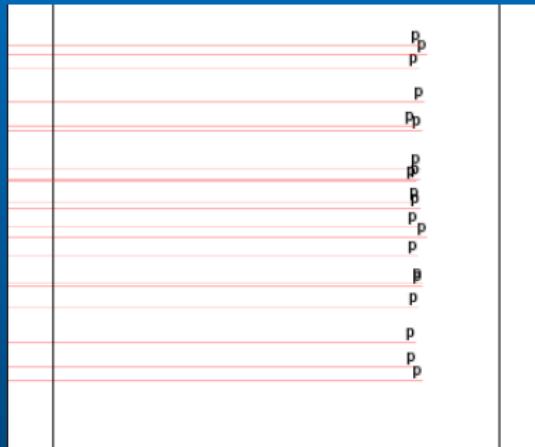


Illustration du straggling

e^- de 16 MeV dans 10 cm d'eau



p de 105 MeV dans 10 cm d'eau



Fluctuations sur $\Delta E \Rightarrow$ fluctuations sur le parcours

16

Multiple Coulomb Scattering (MSC)

- diffusion Coulombienne multiple -

- Les **particules chargées** qui traversent une épaisseur finie de matière subissent des **diffusions Coulombiennes élastiques multiples**
- L'effet cumulé de ces diffusions à petit angle entraîne une **déviation globale de la direction** de propagation de la particule
- Théorie de Lewis
- De nombreux modèles dans Geant4

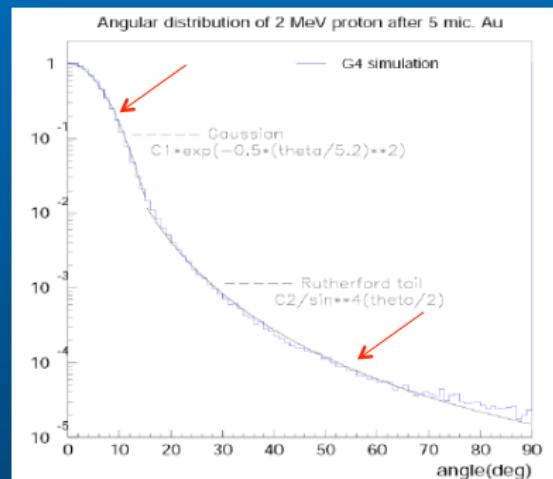
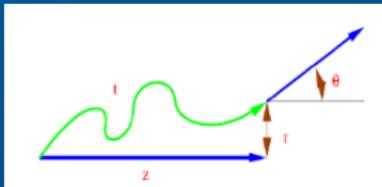
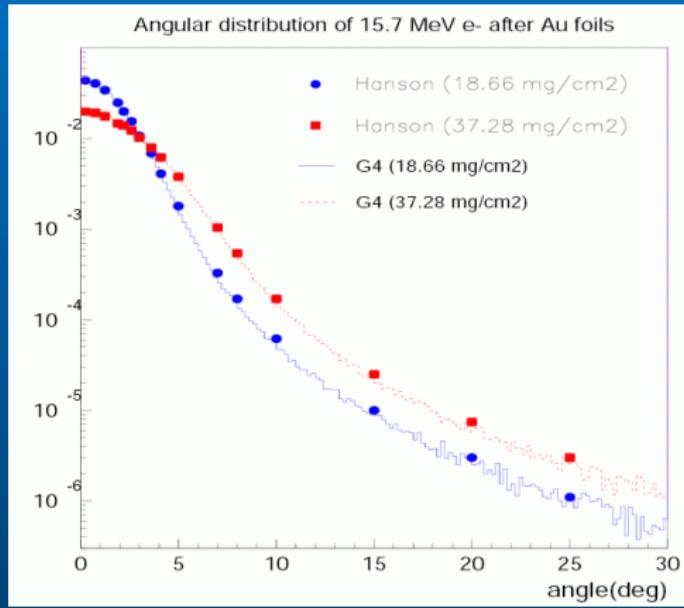


Illustration diffusion coulombienne



18

Comment utiliser la physique EM dans Geant4 ?

Architecture logicielle

- Depuis **Geant4 9.3** (2009) l'architecture logicielle est uniforme dans toutes les catégories électromagnétiques de Geant4
 - Approche cohérente pour des applications à haute et basse énergie
- Une interaction physique ou processus est décrite par une **classe de processus**
 - Nom: « **G4NomDuProcessus** »
 - Ex: **G4ComptonScattering** pour la diffusion Compton
 - Assignnée à des types de particules
 - Hérite de la classe de base **G4VEmProcess**
- Un processus physique peut être simulé selon plusieurs modèles, chacun étant décrit par une **classe de modèle**
 - Nom: « **G4NomDuModèleNomDuProcessusModel** »
 - Ex: **G4LivermoreComptonModel**
 - Les modèles peuvent être assignés à certains domaines en énergie et à des **G4Regions**
 - Hérite de la classe de base **G4VEmModel**
- Les **classes de modèle** calculent
 - Section efficace et pouvoir d'arrêt
 - Sélection de l'atome dans la cible
 - Etat final (cinématique, production des secondaires...)

Listes de physique

- Une liste de physique (“physics list”) est une classe utilisateur obligatoire qui fait l’interface entre la physique dont a besoin l’utilisateur et le noyau de Geant4
 - doit inclure toutes les particules nécessaires
 - chaque couple processus / particule est déclaré au `G4PhysicsListHelper`
- L’utilisateur peut écrire sa propre liste de physique (peut devenir fastidieux...)
- Geant4 fournit plusieurs configurations de listes de physique appelées “constructeurs” (`G4VPhysicsConstructor`) dans la librairie des “physics lists” électromagnétiques de Geant4, déjà écrites pour vous
- Ces constructeurs peuvent aussi être inclus dans une “liste de physique modulaire” (`G4VModularPhysicsList`) dans une application

Constructeurs de physique EM

- G4EmStandardPhysics – défaut
- G4EmStandardPhysics_option1 – PHE, rapide mais moins précis
- G4EmStandardPhysics_option2 – expérimental
- G4EmStandardPhysics_option3 – médical, spatial
- G4EmStandardPhysics_option4 – le + précis
- G4EmLivermorePhysics
- G4EmLivermorePolarizedPhysics
- G4EmPenelopePhysics
- G4EmDNAPhysics
- Situés dans \$G4SRC/source/physics_list/builders
- Avantages
 - Testés régulièrement et utilisés pour la validation

Combinent
- Standard
- Basse énergie

Exemple

```
G4PhysicsListHelper* ph = G4PhysicsListHelper::GetPhysicsListHelper();

theParticleIterator->reset();
while( (*theParticleIterator)() ){
    G4ParticleDefinition* particle = theParticleIterator->value();
    G4String particleName = particle->GetParticleName();

    if (particleName == "gamma") {

        ph->RegisterProcess(new G4PhotoElectricEffect(), particle);
        ph->RegisterProcess(new G4ComptonScattering(), particle);
        ph->RegisterProcess(new G4GammaConversion(), particle);
        ph->RegisterProcess(new G4RayleighScattering(), particle);

    } else if (particleName == "e-") {

        ph->RegisterProcess(new G4eMultipleScattering(), particle);
        ph->RegisterProcess(new G4eIonisation(), particle);
        ph->RegisterProcess(new G4eBremsstrahlung(), particle);

    } else...
}
```

Comment extraire la physique ?

- On peut extraire les quantités physiques via un objet `G4EmCalculator`
- La liste de physique doit avoir été initialisée
- Exemple d'extraction de la **section efficace** du processus de nom `procName`, pour une **particule** donnée et un matériau `matName`

```
#include "G4EmCalculator.hh"
...
G4EmCalculator emCalculator;

G4Material* material = G4NistManager::Instance()->FindOrBuildMaterial(matName);

G4double density = material->GetDensity();

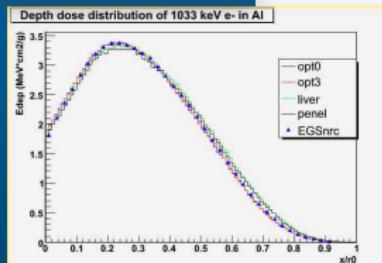
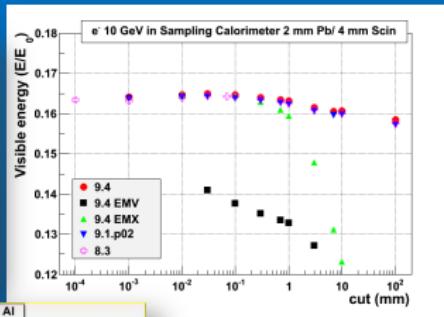
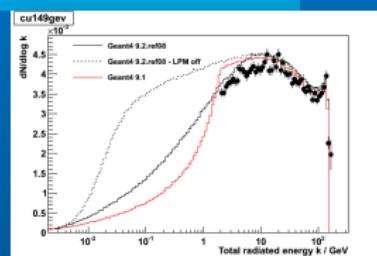
G4double massSigma = emCalculator.ComputeCrossSectionPerVolume
    (energy,particle,procName,material)/density;

G4cout << G4BestUnit(massSigma, "Surface/Mass") << G4endl;
```

- Exemple: `$G4SRC/examples/extended/electromagnetic/TestEm14`
Regarder la classe `RunAction.cc`

Répertoire de validation

- Un outil de vérification basé sur le web a été développé pour comparer les résultats de Physique avec les différentes versions de Geant4 et des mesures expérimentales



<http://geant4.web.cern.ch/geant4/results/EmVali/validation3.php>

25

Des exemples « étendus »

`$G4SRC/examples/extended/electromagnetic`

Section « extended examples » dans le User's Guide : For Application Developer

Quantités de base	
Section efficace totale, libre parcours moyen...	TestEm0, Em13, Em14
Pouvoir d'arrêt, parcours...	Em0, Em1, Em5, Em11, Em12
Etat final: spectre en énergie, distributions angulaires	Em14
Fluctuations de perte d'énergie	Em18
Multiple Coulomb Scattering (MSC)	
Comme un processus isolé	Em15
Résultant du transport	Em5
Vérifications globales	
Couche simple: transmission, absorption, réflexion, fluo.	Em5
Courbes de Bragg, dosimétrie	Em7
Distribution profondeur dose	Em11, Em12
Forme des gerbes, rayon de Molière	Em2
Calorimètre à échantillonage, flot d'énergie	Em3
Calorimètre cristallin	Em9
Autres	
Muons de haute énergie	Em17
Processus rares de haute énergie	Em6
Radiation synchrotron	Em16
Radiation de transition	Em8
Modèle Photo-absorption-ionisation	Em10

En résumé

- Ensemble complet de processus EM
- Pendant de nombreuses années, focalisée sur la **Physique des Hautes Energies**
- Ouverture vers le spatial, le médical et autres domaines
- **Architecture unique** pour l'ensemble de la physique électromagnétique de Geant4
- Nombreux **exemples** disponibles dans Geant4

L'outil Geant4 :

Physique électromagnétique (QED)

BASSE ENERGIE



`PhysicsList.hh (.cc)`



30 minutes

1

Contenu

- Contexte
- Modèles de physique
 - “Livermore” (+ modèles polarisés)
 - “Penelope”
 - “ICRU73”
 - “Geant4-DNA”
 - Désexcitation atomique
- Comment implémenter une liste de physique
- Documentation

Contexte

Objectif

- Etendre la couverture en énergie des interactions électromagnétiques
 - Pour photons, électrons, hadrons et ions
 - Vers les très basses énergies (sub-keV)
- Domaines d'application
 - Spatial
 - Physique médicale
 - Microdosimétrie et radiobiologie
 - ...
- Divers choix de modèles physiques
 - Livermore: électrons et photons [250 eV – 1 GeV]
 - Penelope (code Monte Carlo): électrons, positrons et photons [100 eV – 1 GeV]
 - Microdosimétrie (projet Geant4-DNA): [7 eV – ~100 MeV]

Architecture logicielle

- Une interaction physique ou processus est décrite par une classe de processus
 - Nom: « G4NomDuProcessus »
 - Ex. : « G4ComptonScattering » pour la diffusion Compton
- Un processus physique peut être simulé suivant plusieurs modèles, chacun étant décrit par une classe de modèle
 - Nom: « G4NomDuModèleNomDuProcessusModel »
 - Ex. : « G4LivermoreComptonModel » pour le modèle Livermore Compton
 - Les modèles peuvent être alternatifs et/ou complémentaires sur certains domaines d'énergie
- Suivant le modèle sélectionné, la classe de modèle assure le calcul
 - De la section efficace totale et du pouvoir d'arrêt du processus
 - De l'état final du processus (cinématique, production de secondaires...)

Modèles physiques 1/5

Livermore

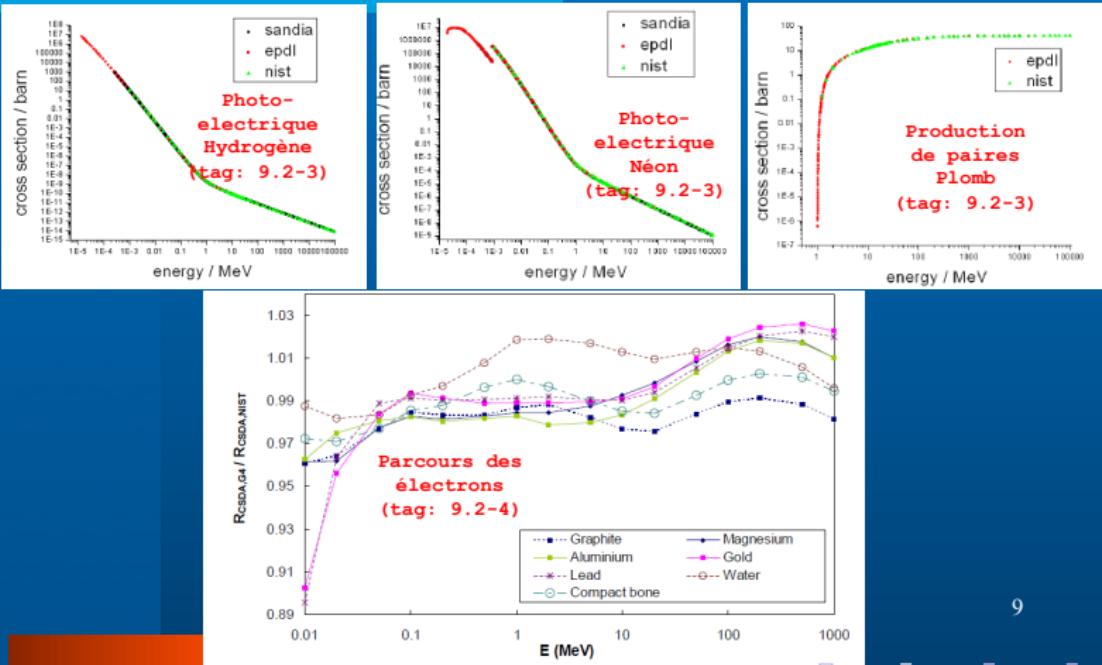
Modèles Livermore

- Pour les électrons et gammas
- Basés sur des **tables de données évaluées publiques** issues de la base de données Livermore
 - EADL : Evaluated Atomic Data Library
 - EEDL : Evaluated Electrons Data Library
 - EPDL97 : Evaluated Photons Data Library
 - Mélanges valeurs expérimentales et théoriques, énergies de liaison d'après Scofield
- Les **tables de données sont interpolées** pour calculer
 - Les sections efficaces totales: photoélectrique, Compton, Rayleigh, production de paires, Bremsstrahlung
 - Prennent en compte les couches atomiques: photo-electrique, ionisation
 - Spectres énergétiques des secondaires: processus créant les électrons secondaires
- Domaine de validité: 250 eV - 100 GeV
 - Les processus peuvent être utilisés jusqu'à 100 eV, mais avec une précision réduite
 - En principe validité jusqu'à ~10 eV
- Éléments inclus de Z=1 à Z=100
 - Incluant les effets atomiques (fluorescence, Auger)
 - Relaxation atomique : Z > 5 (EADL transition data)
- Nom: G4LivermoreXXXModel (ex. G4LivermoreComptonModel)

Modèles Livermore

Processus physique	Classe de processus	Classe de modèle	Limite basse énergie recommandée	Limite haute énergie
Gammas				
Compton	G4ComptonScattering	G4LivermoreComptonModel	250 eV	100 GeV
Polarized Compton	G4ComptonScattering	G4LivermorePolarizedComptonModel	250 eV	100 GeV
Rayleigh	G4RayleighScattering	G4LivermoreRayleighModel	250 eV	100 GeV
Polarized Rayleigh	G4RayleighScattering	G4LivermorePolarizedRayleighModel	250 eV	100 GeV
Conversion	G4GammaConversion	G4LivermoreGammaConversionModel	1.022 MeV	100 GeV
Polarized Conversion	G4GammaConversion	G4LivermorePolarizedGammaConversionModel	1.022 MeV	100 GeV
Photo-electric	G4PhotoElectricEffect	G4LivermorePhotoElectricModel	250 eV	100 GeV
Polarized Photo-electric	G4PhotoElectricEffect	G4LivermorePolarizedPhotoElectricModel	250 eV	100 GeV
Electrons				
Ionization	G4elionisation	G4LivermoreIonisationModel	250 eV	100 GeV
Bremsstrahlung	G4eBremsstrahlung	G4LivermoreBremsstrahlungModel	250 eV	100 GeV

Exemple de vérification Livermore



Modèles Livermore polarisés

- Décrivent les interactions des photons polarisés
- Basés sur la librairie Livermore
- Applications possibles
 - Missions spatiales de détection des gammas polarisés
- Nom: G4LivermorePolarizedXXXModel
 - ex. G4LivermorePolarizedComptonModel

Modèles physiques 2/5

Penelope

Physique de Penelope

- Geant4 contient aussi les modèles physiques du code Monte Carlo **PENELOPE** (PENetration and Energy Loss of PElectrons) version 2008 pour les électrons, positrons et gammas
 - Développé par le groupe de Salvat *et al.*
- Description des effets atomiques, de la fluorescence, dé l'élargissement Doppler
- Approche mixte: analytique, paramétrée et tables de données
- Applicable de 100 eV à 1 GeV
- Peut traiter les positrons
- **G4PenelopeXXXModel** (ex. **G4PenelopeComptonModel**)

Modèles Penelope

Processus physique	Classe de processus	Classe de modèle	Limite basse énergie	Limite haute énergie
Gammes				
Compton	G4ComptonScattering	G4PenelopeComptonModel	100 eV	1 GeV
Rayleigh	G4RayleighScattering	G4PenelopeRayleighModel	100 eV	1 GeV
Conversion	G4GammaConversion	G4PenelopeGammaConversionModel	1.022 MeV	1 GeV
Photo-electric	G4PhotoElectricEffect	G4PenelopePhotoElectricModel	100 eV	1 GeV
Electrons/Positrons				
Ionization	G4ionisation	G4PenelopelonisationModel	100 eV	1 GeV
Bremsstrahlung	G4eBremsstrahlung	G4PenelopeBremsstrahlungModel	100 eV	1 GeV
Positrons				
Annihilation	G4eplusAnnihilation	G4PenelopeAnnihilationModel	100 eV	1 GeV

Modèles physiques 3/5

Les ions

Modèle de perte d'énergie pour les ions

- Décrit la perte d'énergie des **ions plus lourds que l'Hélium** causée par les interactions avec les électrons atomiques
- Applications médicales & spatiales
- Le modèle calcule
 - Les pouvoirs d'arrêt restreints: perte d'énergie continue de l'ion à la traversée d'un absorbeur
 - $T < T_L = 0.025 \text{ MeV/u}$: modèle du gaz d'électrons libres
 - $T_L \leq T \leq T_H$: interpolation des tables ICRU73 révisées
 - $T > T_H = 10 \text{ MeV/u}$: formule de Bethe formula (charge effective) + corrections
 - Les sections efficaces de production d'électrons delta, produits uniquement au-dessus d'un seuil de production (qui gouverne aussi la perte d'énergie continue)
- Inclus dans les constructeurs de physique **Livermore, Penelope, Standard_option3, Standard_option4**
- Pour les composés, utiliser les noms de la base de données NIST (par ex. G4_WATER). Pour les éléments, nom libre.

Modèles physiques 4/5

Geant4-DNA

Geant4 pour la radiobiologie: Geant4-DNA

- Initié en 2001 par l'Agence Spatiale Européenne
- Adapter Geant4 pour la simulation des interactions des radiations ionisantes avec les systèmes biologiques à l'échelle de l'ADN
- Une activité multidisciplinaire du groupe de physique EM de basse énergie de la collaboration Geant4
- Applications
 - Radiobiology, radiotherapy/hadrontherapy
 - ex. Prédire les cassures directes et indirectes à l'ADN
 - Radioprotection pour l'exploration habitée du système solaire

- Eau liquide
- Processus discrets
- Combinables avec les autres processus EM
- Analytiques et/ou interpolés

Aperçu des modèles physiques de Geant4-DNA

	e-	p	H	α , He+, He ⁰	C, N, O, Fe,...
Diffusion élastique	> 9 eV – 1 MeV Screened Rutherford >7.4 eV – 1 MeV Champion	-	-	-	-
Excitation A ₁ B ₁ , B ₁ A ₁ , Ryd A+B, Ryd C+D, diffuse bands	9 eV – 1 MeV Born	10 eV – 500 keV Miller Green 500 keV – 100 MeV Born	10 eV – 500 keV Miller Green	Effective charge scaling from same models as for proton	-
Echange de charge	-	100 eV – 10 MeV Dingfelder	100 eV – 10 MeV Dingfelder	-	-
Ionisation 1b ₁ , 3a ₁ , 1b ₂ , 2a ₁ + 1a ₁	11 eV – 1 MeV Born	100 eV – 500 keV Rudd 500 keV – 100 MeV Born	100 eV – 100 MeV Rudd	1 keV – 400 MeV	Effective charge scaling 0.5 MeV/u – 10 ⁶ MeV/u
Excitation vibrationnelle	2 – 100 eV Sanche	-	-	-	-
Attachement	4 – 13 eV Melton				

Modèles physiques 5/5

Désexcitation atomique

La désexcitation atomique

- Elle est initiée par d'autres processus EM
 - Ex.: effet photoélectrique, ionisation par les e- et les protons (PIXE)
 - Laisse l'atome dans un état excité
- Le données EADL contiennent les transitions de probabilité
 - radiatives: fluorescence
 - non-radiatives
 - Électron Auger: les lacunes initiale et finale appartiennent à des sous-couches atomiques différentes
 - Électron Coster-Kronig : sous-couches identiques
- Compatible avec les catégories EM "standard" et de "basse énergie"

Implémenter une liste de physique

Listes de physique

- Un utilisateur peut
 - Utiliser des **listes de physique de référence** fournies avec Geant4 (QBBC,)
 - Construire **sa propre liste de physique** “à la main” dans son application
 - Ou utiliser des **constructeurs de physique** déjà disponibles
- 1. Si l'on choisit de construire sa propre liste de physique
 - Aller voir **le site web** du groupe de physique EM de basse énergie,
section Processes
 - Et aussi voir les **exemples** de Geant4
 - `$G4SRC/examples/extended/electromagnetic/TestEm14`
- 2. **Plus sûr:** utiliser les **constructeurs** de physique déjà écrits pour vous;
pour la physique électromagnétique de basse énergie, on a:
 - **G4EmLivermorePhysics**
 - **G4EmLivermorePolarizedPhysics**
 - **G4EmPenelopePhysics**
 - **G4EmDNAPhysics**

Comment utiliser les constructeurs de physique déjà existants ?

- Les constructeurs de physique dérivent de la classe de basse abstraite **G4VPhysicsConstructor**
 - Voir par exemple l'utilisation des constructeurs de physique EM
`$G4SRC/examples/extended/electromagnetic/TestEm2`
 - C'est ce que nous ferons en exercice
- Le code source des constructeurs de physique est disponible dans le répertoire suivant: `$G4SRC/source/physics_list/builders`
- On peut les ajouter aux listes de Physique de référence avec la méthode **RegisterPhysics** (**G4VPhysicsConstructor***)
 - Voir le répertoire `$G4SRC/source/physics_lists/lists`
- Si on veut rajouter la physique hadronique en plus de la physique EM, voir:
 - `$G4SRC/examples/extended/electromagnetic/TestEm7`

Documentation

Sites web

- Une page unique pour la physique EM de Geant4
 - <http://geant4.cern.ch/collaboration/EMindex.shtml>
- Depuis là, liens vers:
 - Le groupe de physique **EM standard** de Geant4
 - Le groupe de physique **EM de basse énergie** de Geant4
- Et aussi depuis le site web de Geant4:
 - <http://geant4.org>
 - Who we are
 - Low energy Electromagnetic Physics

Twiki

<https://twiki.cern.ch/twiki/bin/view/Geant4/LowEnergyElectromagneticPhysicsWorkingGroup>

TWiki > Geant4 Web > LowEnergyElectromagneticPhysicsWorkingGroup (11-Jul-2013, Sébastien Incerti) [Edit](#) [Attach](#) [PDF](#)

The Geant4 Low Energy Electromagnetic Physics Working Group

This web site is the [official web site](#) of the Geant4 collaboration Low Energy Electromagnetic Physics working group.

Purpose

The Geant4 Low Energy Electromagnetic Physics Working Group develops and maintains a set of models to describe the electromagnetic interactions of photons, electrons, hadrons and ions with matter down to very low energies (eV scale), including the [Geant4-DNA project](#) ([link1](#) - [link2](#) - [link3](#)). Applications of such models range from high energy physics experiments to space science and astrophysics to the medical and biological fields. After several years of separation, these activities now take place in full collaboration with the [Standard Electromagnetic Physics working group](#) of the Geant4 collaboration.

What's new in Geant4 10 Beta ? (June 2013)

- [Overview](#) of our most recent developments.
- Version [6.33](#) of the set of Low Energy electromagnetic data files is required.

Physics

- [Processes](#) is a link to the catalog of [Geant4 low energy electromagnetic Physics processes](#) and to other useful information related to these processes.
- [Physics Lists](#) describes [recommended Physics lists](#) for applications involving low energy electromagnetic Physics processes.
- [Examples](#) recommended for the usage of Geant4 electromagnetic Physics.

Tools

* [Example recommended for the usage of Geant4 electromagnetic Physics](#)

En résumé

- Utiliser les **modèles EM de basse énergie** (Livermore ou Penelope), comme une alternative aux modèles EM “standards” de Geant4, quand vous
 - Avez besoin de modéliser précisément les interactions EM à **basse énergie** (échelle du **keV**)
 - Êtes intéressés par les **effets atomiques**, comme les X de fluorescence, l’élargissement Doppler, etc...
 - Avez de la **puissance de calcul**
 - Voulez **comparer** vos résultats avec d’autres simulations (par ex. avec un autre modèle)
- **Ne pas utiliser si vous êtes intéressés par la physique EM > MeV**
 - Mêmes résultats que la physique EM Standard de Geant4 mais pénalités en temps de calcul

En savoir plus



28

L'effet photoélectrique

- Section efficace
 - Intégrée sur les couches
 - de EPDL + interpolation
 - Par couche, à partir de celle qui émet l'électron,
 - sélectionnée dans EPDL
- Génération de l'état final
 - Différents générateurs de distributions angulaires (“naïf”, Sauter-Gavrila, Gavrila)
- Désexcitation par le processus de relaxation atomique
 - Lacune initiale et chaîne de lacunes qui en découle

Diffusion Compton (incohérente)

Section efficace Klein-Nishina (E'/E) \times fonction de diffusion (q)

$$q = E \sin^2(\theta/2) \text{ moment transféré}$$

- La distribution en énergie du photon diffusé est obtenue par la formule de Klein-Nishina multipliée par une fonction de diffusion $F(q)$ (facteur atomique de Hubbel) à partir des données de EPDL97
- La fonction de diffusion supprime la diffusion vers l'avant à basse énergie
- La distribution angulaire du photon diffusé et de l'électron de recul viennent de EPDL97

Diffusion Rayleigh (cohérente)

- Dépend de la distribution de charge de l'atome
- Distribution angulaire

$$F(E, \theta) = [1 + \cos^2(\theta)] \sin\theta F^2(q)$$

$$q = 2 E \sin(\theta/2)$$

Formule de Rayleigh multipliée par $F(q)$, le facteur de forme de Hubbel obtenu de EPDL97 (pique vers l'avant à haute énergie)

- N'existe que dans le package basse énergie

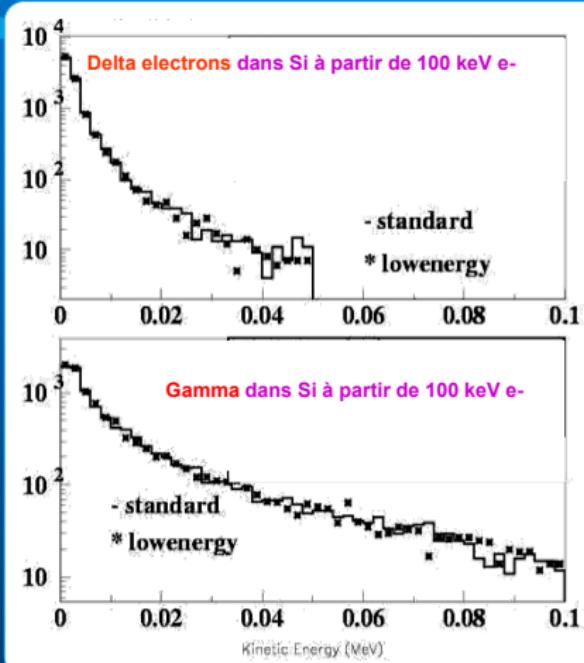
Conversion de paires

- Les énergies des e^- et e^+ secondaires sont calculées à partir de la formule de **Bethe-Heitler** avec correction coulombienne
- On suppose que e^- et e^+ ont une distribution angulaire symétrique
- L'énergie et l'angle polaire sont échantillonnés par rapport au photon incident en utilisant la section efficace différentielle de **Tsai**
- Angle azimuthal isotrope
- Choix au hasard de quelle particule de la paire est e^- ou e^+

Rayonnement de freinage

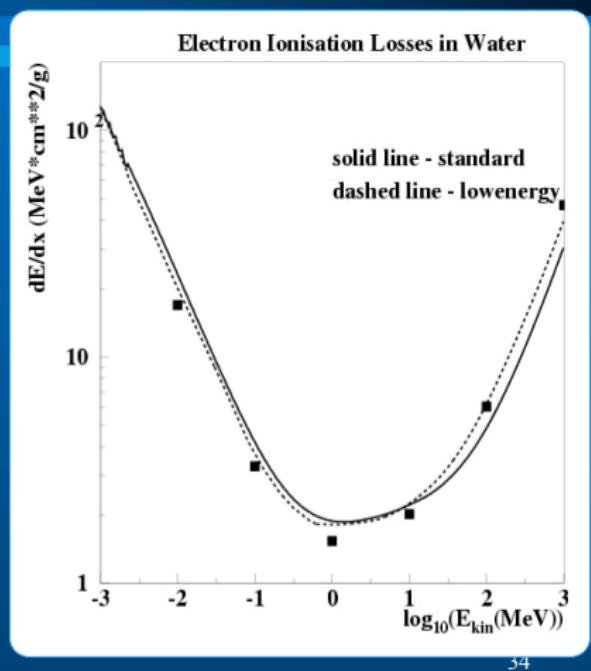
Paramétrisation des données EEDL

- 16 paramètres par atome
- Reproduit la formule de Bethe-Heitler à haute énergie
- Précision ~ 1.5 %



Ionisation

- Paramétrisation basée sur 5 paramètres pour chaque couche
- Precision meilleure que 5% pour 50 % des couches



54

L'outil Geant4 : Physique hadronique (QCD)



`PhysicsList.hh (.cc)`



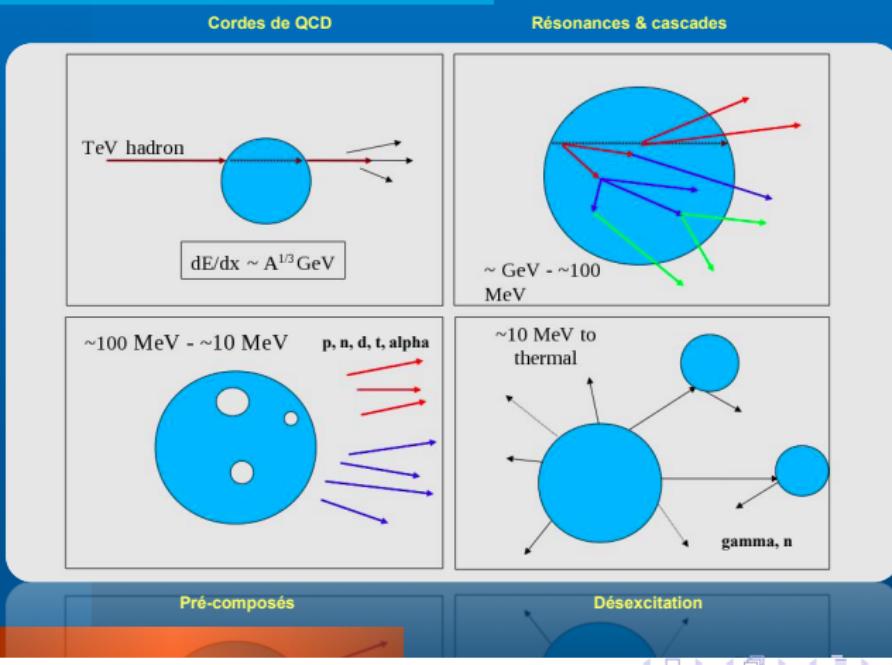
15 minutes

1

Physique hadronique

- Même s'il existe une théorie sous-jacente (QCD), l'appliquer est bien plus complexe que la physique électromagnétique (QED)
- Il existe au moins trois domaines d'énergie différents, jusqu'à 100 TeV
 - Théorie des perturbations chirales (< 100 MeV)
 - Région des résonances et cascades (100 MeV – 20 GeV)
 - Théorie des cordes de QCD (> 20 GeV)
- Dans chaque régime, il existe des sous-modèles:
 - La plupart sont phénoménologiques

Du TeV au meV...



Les processus hadroniques

- **Au repos**
 - Muon, pion, kaon, anti-proton
 - Désintégration radioactive
- **Elastique**
 - Même processus pour tous les hadrons à longue durée de vie
- **Inélastique**
 - Processus différent pour chaque hadron
 - Photo-nucléaire
 - Electro-nucléaire
- **Capture**
 - Pion- et kaon- en vol
- **Fission**

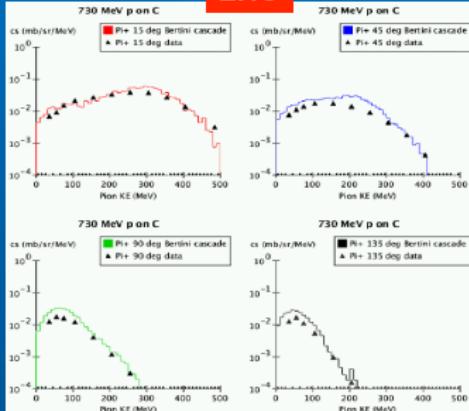
Modèles de gerbes hadroniques

Trois types

- Modèles “Data driven”
- Modèles “Paramétrisation driven” (LEP, HEP)
 - gèrent le plus de types de particules sur le plus grand domaine d'énergie
 - basés sur des ajustements de données et sur la théorie
 - pas très détaillés, rapides
- Modèles “Theory driven” (“Bertini & Binary cascades”)
 - sont valides pour un nombre limité de particules et sur un plus petit domaine d'énergie
 - d'avantage basés sur la théorie
 - plus détaillés, moins rapides

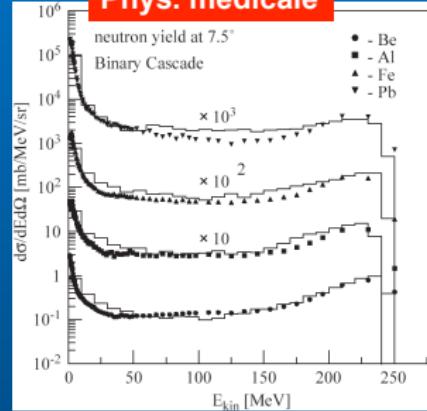
Exemples de modèles theory driven

LHC



Modèle de cascade de Bertini
Production de pions à partir de
protons de 730 MeV sur
Carbone

Phys. médicale



Modèle de cascade binaire
Section efficace différentielle double pour la
production de neutrons par des protons de
256 MeV dans différents matériaux

6

La physique des neutrons de basse énergie

- Energie < 20 MeV
- **High Precision Neutron Models** : modèles data driven basées sur des librairies de données évaluées:
 - Base de données G4NDL (Geant4 Neutron Data Library)
 - Source ENDF/B-VII
 - Elastique
 - Inélastique
 - Capture
 - Fission
- **NeutronHPorLEModel(s)** : modèles alternatifs
- **ThermalScatteringModels**
 - Prend en compte les effets moléculaires

La physique des ions : désintégration radioactive

- Simuler la désintégration radioactive des noyaux
- Modèles empiriques et data driven
- Désintégration α , β^+ , β^- , capture électronique, conversion interne sont implémentées
- Données (RadioactiveDecay) dérivées de Evaluated Nuclear Structure Data File (ENSDF)
 - Demi-vies nucléaires
 - Structure des couches nucléaires
 - Rapport d'embranchement de désintégration
 - Valeur de Q du processus de désintégration
- Si la fille d'une désintégration nucléaire est un isomère excité, sa désexcitation nucléaire prompte est traitée en utilisant un code de photoévaporation

Rappel : la Physics list

- Vous devez implémenter une **physics list**
 - Dériver une classe de **G4VUserPhysicsList**
 - Définir les particules mises en jeu
 - Associer les modèles et les sections efficaces avec les processus
 - Associer les particules avec les processus
 - Fixer les coupures de production des secondaires
 - Dans le **main()**, associer la physics list avec le **run manager**



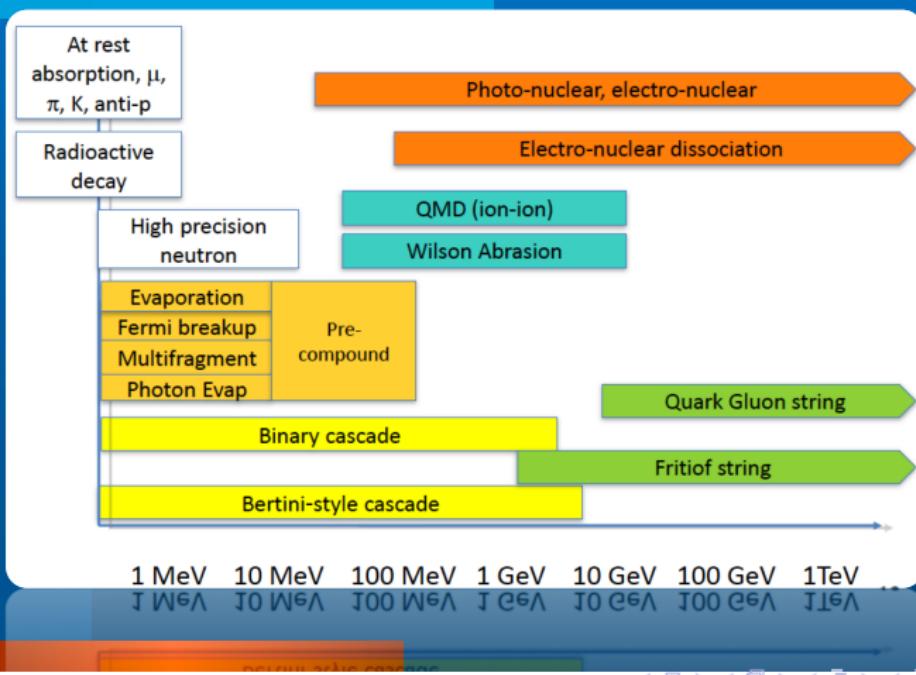
Attention pour la physique hadronique !

- Multiples modèles et sections efficaces autorisés par processus
- Aucun modèle unique pour couvrir toutes les énergies et toutes les particules
- Le choix du modèle dépend fortement de la physique étudiée

Les Physics lists pour la physique hadronique

- D'où la recommandation
 - Utiliser les exemples fournis
 - Et voir depuis la [page web de Geant4](#) :
 - Site web: [User support](#) → physics lists
- De nombreuses physics lists de référence sont disponibles
 - Radioprotection contre les noyaux de basse et haute énergie
 - Applications dosimétriques à basse énergie
 - Applications médicales utilisant des neutrons
 - Expériences à faible bruit de fond (sous terre)
 - Incluent: [physique EM \(standard\)](#), [processus gamma et electron nucléaires](#), [décroissance radioactive](#), [interactions hadroniques élastiques et inélastiques](#), [interactions ions-noyaux...](#) et on peut rajouter d'autres processus (ex. Optiques)

Bref inventaire des modèles hadroniques



En résumé

- Geant4 fournit un **grand nombre de modèles** de physique hadronique pour la simulation
- Les sections efficaces sont calculées ou évaluées à partir de **bases de données**
- Les interactions sont implémentées par modèle, qui **doit être assigné au processus correspondant**
- Il existe de nombreux modèles, ainsi Geant4 fournit des **physics lists** pour des cas d'utilisation spécifiques

L'outil Geant4 : Les Physics Lists



`PhysicsList.hh (.cc)`



15 minutes

1

Qu'est-ce qu'une Physics List ?

- Une classe qui collecte
 - toutes les particules,
 - les processus physiques,
 - les seuils de production de secondaires
- Elle dit au run manager comment et quand utiliser la physique
- C'est un moyen très flexible de construire un environnement de physique
 - On peut choisir les particules que l'on souhaite utiliser
 - On peut choisir la physique associée à chaque particule



Mais l'utilisateur doit avoir une bonne connaissance de la physique nécessaire : l'oubli de particules ou de processus physiques pourrait causer des erreurs

2

A quoi **sert** une Physics List ?

- Geant4 ne devrait-il pas fournir par défaut un ensemble complet de processus physiques que tout le monde pourrait utiliser ?
- **Non !**
 - Il y a trop de modèles et d'approximations différents
 - En particulier pour la physique hadronique
 - Mais aussi pour la physique électromagnétique
 - Problème du temps de calcul
 - Un utilisateur peut vouloir une simulation moins détaillée, plus rapide
 - Aucune application n'a besoin de toute la physique et de toutes les particules que Geant4 peut offrir
 - Ex : la plupart des applications médicales n'ont pas besoin de physique multi-GeV

A quoi sert une Physics List ?

- Pour cette raison, Geant4 prend une **approche atomistique** (et non intégrale) de la Physique
 - Fournit de nombreux composants de physiques (processus) découplés les uns des autres
 - L'utilisateur sélectionne ces composants dans des **listes de physique (physics lists)**, comme on construirait la géométrie d'un détecteur
- Exceptions
 - Quelques processus EM doivent être utilisés ensemble
 - Les futurs processus impliquant des interférences entre interactions EM et interactions fortes pourraient nécessiter un couplage

Principaux processus physiques fournis

- Physique électromagnétique
 - “standard” valide de ~ 1 keV au ~ PeV
 - “basse énergie” valide de ~ eV au ~ GeV
 - Photons optiques
- Physique interaction faible
 - Décroissance des particules subatomiques
 - Désintégration radioactive des noyaux
- Physique hadronique
 - Processus purement hadroniques de 0 à ~100 TeV
 - Processus lepto- et photo-noyau, valides de 10 MeV au ~TeV
- Physique paramétrisée ou simulation rapide

G4VUserPhysicsList

- Toutes les physcis lists doivent dériver de cette classe, puis être déclarées au run manager
- Exemple

```
class myPhysicsList : public G4VUserPhysicsList
{
public:
    myPhysicsList();
    ~myPhysicsList();
    void ConstructParticle();
    void ConstructProcess();
    void SetCuts();
};
```



L'utilisateur doit implémenter les méthodes `ConstructParticle` et `ConstructProcess`. La méthode `SetCuts()` est optionnelle.

6

G4VUserPhysicsList :

3 méthodes dont 2 obligatoires

- **ConstructParticle()** : **obligatoire**
 - choisir les particules dont on a besoin
- **ConstructProcess()** : **obligatoire**
 - Pour chaque particule, assigner tous les processus physiques importants
 - Qu'est-ce qu'un **processus** ?
 - C'est une classe qui définit comment une particule doit interagir avec la matière (c'est là où se trouve la physique !)
- **SetCuts()**
 - Définir les **coupures de parcours (cut)** pour la production de secondaires
 - Qu'est-ce-qu'une **coupure** ?
 - C'est une limite de basse énergie sur la production des secondaires

1) ConstructParticle()

```
void myPhysicsList::ConstructParticle()
{
    G4Electron::ElectronDefinition();
    G4Proton::ProtonDefinition();
    G4Neutron::NeutronDefinition();
    G4Gamma::GammaDefinition();
    ....
    ....
}
```



2) ConstructProcess()

```
void myPhysicsList::ConstructProcess()
{
    AddTransportation();
    // méthode fournie par G4VUserPhysicsList
    // assigne le processus de transport à toutes les particules
    // définies dans ConstructParticle()

    ConstructEM();
    // méthode qui peut être définie à convenance
    // on y place la physique électromagnétique

    ConstructGeneral();
    // autre méthode qui peut être définie à convenance
}
```



Par ex. pour ConstructEM()

```
void myPhysicsList::ConstructEM()
{
    G4PhysicsListHelper* ph = G4PhysicsListHelper::GetPhysicsListHelper();

    theParticleIterator->reset();
    while ( (*theParticleIterator)() )

    {
        G4ParticleDefinition* particle = theParticleIterator->value();
        G4String particleName = particle->GetParticleName();

        if  (particleName == "gamma")
        {
            ph->RegisterProcess(new G4PhotoElectricEffect(), particle);
            ...
        }
        ....
    }
}
```

Par ex. pour ConstructGeneral()

```
void myPhysicsList::ConstructGeneral()
{
    // Ajouter le processus de désintégration radioactive
    G4RadioactiveDecay* radioactiveDecay = new G4RadioactiveDecay();
    radioactiveDecay->SetHTThreshold(-1.*s);
    radioactiveDecay->SetICM(true);                                //Internal Conversion
    radioactiveDecay->SetARM(false);                               //Atomic Rearangement

    G4PhysicsListHelper* ph = G4PhysicsListHelper::GetPhysicsListHelper();
    ph->RegisterProcess(radioactiveDecay, G4GenericIon::GenericIon());

    // Deexcitation (in case of Atomic Rearangement)
    //
    G4UAtomicDeexcitation* de = new G4UAtomicDeexcitation();
    de->SetFluo(true);
    de->SetAuger(true);
    de->SetPIXE(false);
    G4LossTableManager::Instance() ->SetAtomDeexcitation(de);
}
```

3) SetCuts()

```
void myPhysicsList::SetCuts () {  
    defaultCutValue = 1.0*mm;  
    SetCutValue(defaultCutValue, "gamma");  
    SetCutValue(defaultCutValue, "e-");  
    SetCutValue(defaultCutValue, "e+");  
    //  
    // ce sont les cuts de production qu'il faut choisir  
    // - pas nécessaire pour les autres particules  
}
```



En résumé

- Toutes les particules, les processus physiques et les coupures de production doivent être définis dans une physics list
- On peut dériver une physics list simple de la classe `G4VUserPhysicsList`
- On peut ensuite la construire mais il faut faire attention à bien choisir les bons processus physiques...
 - On recommande plutôt d'utiliser les constructeurs de physique et les physics lists de référence
 - La plupart des particules et processus y sont déjà définis pour vous

L'outil Geant4 : Génération des particules primaires



PrimaryGeneratorAction.hh (.cc)



10 minutes

1

Classes d'initialisation et d'action

- Les classes utilisateur d'initialisation
 - Déclarées en utilisant **G4RunManager::SetUserInitialization()**
 - Invoquées à l'initialisation de la simulation
 - G4VUserDetectorConstruction
 - G4VUserPhysicsList
- Les classes utilisateur d'action
 - Déclarées en utilisant **G4RunManager::SetUserAction()**
 - Invoquée à chaque événement (boucle d'événements)
 - G4VUserPrimaryGeneratorAction
 - + G4UserRunAction / G4UserEventAction / G4UserStackingAction / G4UserTrackingAction/ G4UserSteppingAction / ...

Dans votre main() il y aura:

```
G4VUserPrimaryGeneratorAction* gen_action = new PrimaryGeneratorAction;  
runManager->SetUserAction(gen_action);
```

G4VUserPrimaryGeneratorAction

- Cette classe est une **classe obligatoire** pour contrôler la génération des primaires
 - Cette classe ne doit pas générer les primaires mais **invoyer** la méthode `GeneratePrimaryVertex()` de **générateurs primaires**
- Son **constructeur**
 - Doit instantier le(s) générateur(s) primaire(s)
 - Doit lui préciser ses (leurs) valeurs par défaut
- Sa méthode **GeneratePrimaries()**
 - Peut “randomiser” les valeurs particule par particule
 - Passe ces valeurs au(x) générateurs primaires
 - Invoque la méthode `GeneratePrimaryVertex()` du (des) générateur(s) primaire(s)

Générateur primaire

- Tous les générateurs primaires doivent dériver de **G4VPrimaryGenerator** et implémenter la méthode virtuelle pure **GeneratePrimaryVertex()**
 - C'est là que le vertex primaire et les particules primaires sont ajoutés au **G4Event**
- Générateur primaire : Geant4 fournit des implémentations concrètes de **G4VPrimaryGenerator**
 - **G4HEPEvtInterface**
 - **G4HEPMCInterface**
 - **G4ParticleGun**
 - **G4GeneralParticleSource**

1) G4ParticleGun

- G4ParticleGun est une **implémentation de G4VPrimaryGenerator** qui est utilisée pour simuler un faisceau de particules
- Il tire une particule d'une certaine énergie cinétique et avec une direction, d'un point donné, à un instant donné
 - Des méthodes sont disponibles pour construire le “particle gun”
 - Choisir le type de particule
 - Choisir l'énergie, la quantité de mouvement
 - Choisir la polarisation
 - Choisir la charge
 - Choisir le nombre de particules émises simultanément

G4ParticleGun

- Des commandes UI sont aussi disponibles

– <code>/gun/list</code>	List available particles
– <code>/gun/particle</code>	Set particle type to be generated
– <code>/gun/direction</code>	Set momentum direction
– <code>/gun/energy</code>	Set kinetic energy
– <code>/gun/momentum</code>	Set momentum
– <code>/gun/momentumAmp</code>	Set absolute value of momentum
– <code>/gun/position</code>	Set starting position of the particle
– <code>/gun/time</code>	Set initial time of the particle
– <code>/gun/polarization</code>	Set polarization
– <code>/gun/number</code>	Set number of particles to shoot (per event)
– <code>/gun/ion</code>	Set properties of ion to shoot <code>/gun/ion z A Q</code>

G4VUserPrimaryGeneratorAction: exemple

```
void PrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
{
    G4ParticleDefinition* particle;
    particle = G4Proton::ProtonDefinition();

    gun->SetParticleDefinition(particle);

    G4double pp = momentum + (G4UniformRand() - 0.5)*sigmaMomentum;
    G4double mass = particle->GetPDGMass();
    G4double Ekin = sqrt(pp*pp + mass*mass) - mass;
    gun->SetParticleEnergy(Ekin);

    G4double angle = (G4UniformRand() - 0.5)*sigmaAngle;
    gun->SetParticleMomentumDirection(G4ThreeVector(sin(angle), 0.,cos(angle)));

    gun->GeneratePrimaryVertex(anEvent);
}
```

- Répéter autant de fois que nécessaire par événement

2) General Particle Source

- **G4GeneralParticleSource** est une autre implémentation concrète de **G4VPrimaryGenerator**
- Le vertex primaire peut être choisi aléatoirement sur la surface d'un volume donné
 - Utile pour les sources radioactives
 - Des spectres pré-définis en énergie sont disponibles (loi de puissance, etc.)
- Direction et énergie peuvent être choisies aléatoirement
- Utilisation d'un langage de commandes spécifique (type macro)
- Capable de biais par événement (réduction de variance)
 - Biaise la distribution du point vertex
 - Biaise énergie et/ou direction
- Utile pour applications médicales et spatiales
 - Voir exemples avancés : télescope à rayons X, physique souterraine...

General Particle Source exemple

- Très simple à déclarer

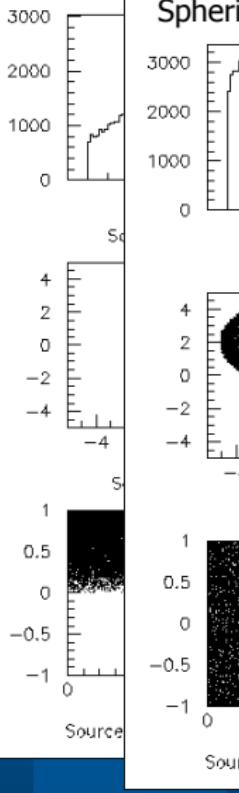
```
PrimaryGeneratorAction::PrimaryGeneratorAction()
{
    fParticleGun = new G4GeneralParticleSource();
}

PrimaryGeneratorAction::~PrimaryGeneratorAction()
{
    delete fParticleGun;
}

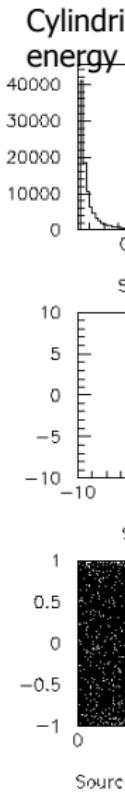
void PrimaryGeneratorAction::GeneratePrimaries(G4Event* anEvent)
{
    fParticleGun->GeneratePrimaryVertex(anEvent) ;
}
```

- Très simple à utiliser
 - Commandes UI
 - Voir liste des commandes dans le **User's Guide: For Application Developers**

Square

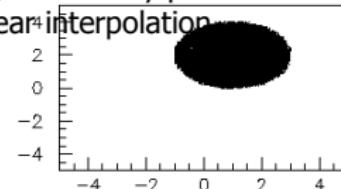
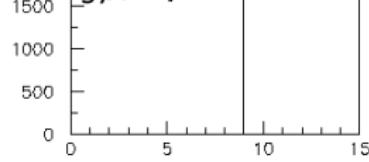


Spherical

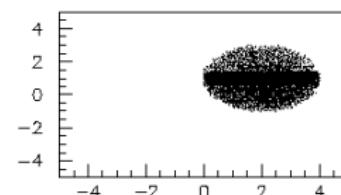
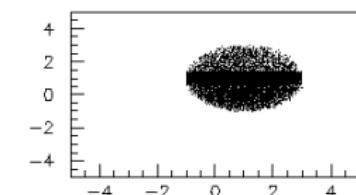


Cylindri
cal

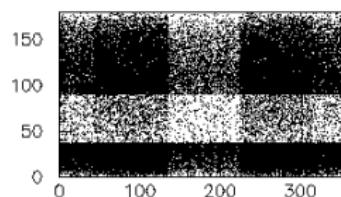
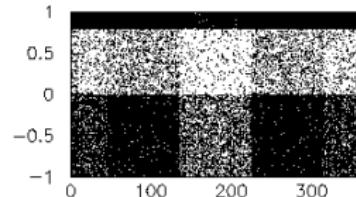
Spherical volume with z biasing, isotropic radiation with theta and phi biasing, integral arbitrary point-wise energy distribution with linear interpolation



Source Energy Spectrum



Source X-Z distribution



Source cos(theta)-phi distribution

Source theta/phi distribution

En résumé

- Nombreuses possibilités pour décrire la génération des particules primaires suivant les besoins de l'utilisateur
- Deux implémentations de **G4VPrimaryGenerator** à privilégier
 - **G4ParticleGun**
 - **G4GeneralParticleSource**

L'outil Geant4 : Particules et processus



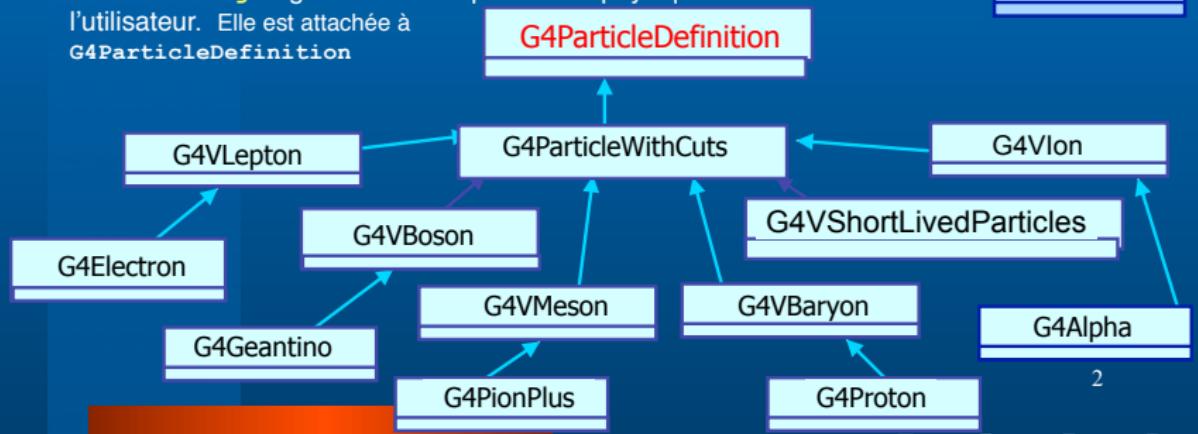
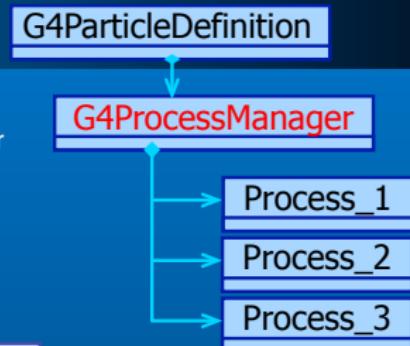
20 minutes

1

G4ParticleDefinition et G4ProcessManager

G4ParticleDefinition est la classe de base concrète pour définir des particules ; elle contient les propriétés intrinsèques (masse, largeur, spin, durée de vie...)
G4ParticleDefinition ne connaît pas sa sensibilité à la physique

G4ProcessManager gère la liste de processus physiques de l'utilisateur. Elle est attachée à **G4ParticleDefinition**



G4DynamicParticle et G4Track

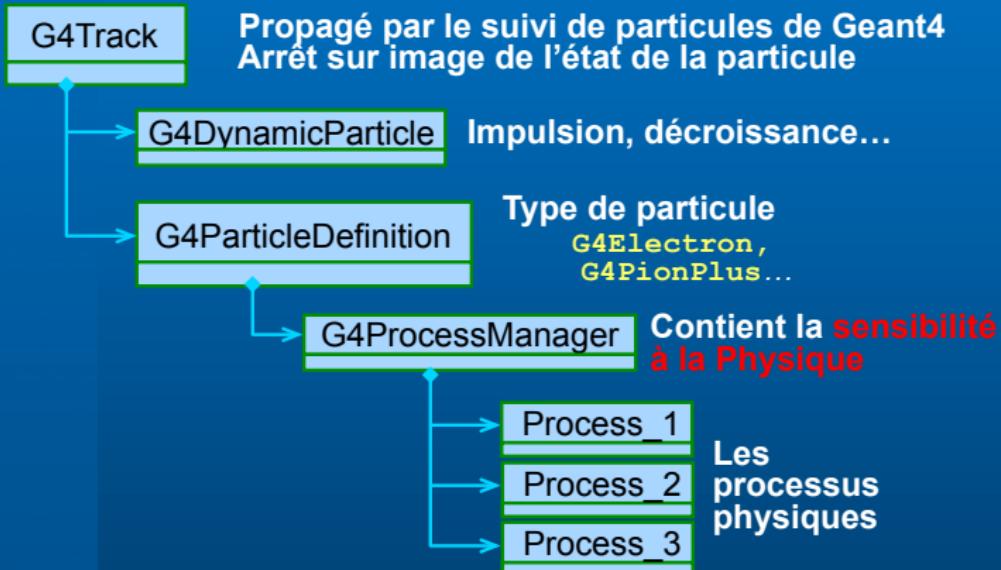
G4DynamicParticle

- Décrit la partie **dynamique** de l'état de la particule
 - **impulsion, énergie, polarisation**
- Contient un pointeur vers **G4ParticleDefinition**
- Retient d'éventuelles informations de décroissance
 - **Produits de désintégration**
 - **Durée de vie**

G4Track

- Définit la classe d'objets propagés par le suivi de particules de Geant4
- Représente l'arrêt sur image de l'état de la particule
- Rassemble
 - une **G4ParticleDefinition**
 - une **G4DynamicParticle**
 - information géométrique
 - position, volume courant...
 - track ID, parent ID
 - Processus qui a créé cette G4Track
 - Poids (event biasing)

Vue d'ensemble



La Physique de Geant4

- **Geant4 fournit une grande variété de composants physiques pour utiliser dans la simulation**
- **Les composants physiques sont codés comme des processus**
 - Un processus est une classe qui indique à une particule comment interagir
 - L'utilisateur peut écrire ses propres processus
- **Les processus sont regroupés en catégories**
 - électromagnétiques, hadroniques et décroissance radioactive

Processus physiques - 1

- Toutes les interactions et désintégrations sont exécutés par des processus
 - Le **transport** est aussi géré par un processus
- Un processus fait deux choses
 - Décide quand et où une interaction doit se produire
 - Méthode : `GetPhysicalInteractionLength()`
 - Elle nécessite une section efficace, une durée de vie, obtenues via un modèle de physique
 - Pour le processus de transport, la distance à l'objet le plus proche le long de la track est nécessaire
 - Génère l'état final de l'interaction (change l'impulsion, génère les secondaires, etc.)
 - Méthode : `DoIt()`
 - Elle nécessite un modèle de physique

Processus physiques - 2

- Il y a trois “**saveurs**” de processus
 - Bien localisé dans l'espace → **PostStep**
 - Distribué dans l'espace → **AlongStep**
 - Bien localisé dans le temps → **AtRest**
- Un processus peut être une combinaison des trois
 - Dans ce cas, 6 méthodes doivent être implémentées (`GetPhysicalInteractionLength()` et `DoIt()` pour chaque action)
- Des processus “**Shortcut**” sont définis et n'invoquent qu'une saveur
 - “**Discrete process**” (possède seulement une physique **PostStep**)
 - “**Continuous process**” (possède seulement une physique **AlongStep**)
 - “**AtRest process**” (possède seulement une physique **AtRest**)

Exemples - 1

- “Discrete” process: diffusion Compton
 - step déterminé par section efficace, interaction à la fin du step
 - `PostStepGPIL()`
 - `PostStepDoIt()`
- “Continuous” process: effet Cerenkov
 - photons créés le long du step, leur nombre en gros proportionnel à la longueur de la trace
 - `AlongStepGPIL()`
 - `AlongStepDoIt()`
- “At rest” process: annihilation de positron au repos
 - Pas de déplacement, le temps est la seule variable
 - `AtRestGPIL()`
 - `AtRestDoIt()`
- On les appelle aussi “pure” processes

Exemples - 2

- “Continuous + discrete” : ionisation
 - La perte d'énergie est continue
 - Diffusion Moller/Bhabha et électrons knock-on (“delta”) sont discrets
- “Continuous + discrete” : rayonnement de freinage
 - La perte d'énergie due aux photons de basse énergie est continue
 - L'émission des photons durs est discrète
- Dans les deux cas, le seuil de production sépare la partie continue de la partie discrète du processus
 - Nous allons y revenir
- La diffusion multiple est continuous + discrete

Gérer des processus multiples

- Plusieurs processus (et donc interactions) peuvent être assignés à la même particule
- Comment Geant4 décide-t-il quelle interaction doit se produire à un instant donné ?
 - La longueur d'interaction ou la longueur de décroissance est calculée pour chaque processus
 - La plus courte est choisie, à moins que
 - La frontière d'un volume soit rencontrée sur une distance plus petite. Alors pas d'interaction physique, seulement du transport.
 - Les processus qui n'ont pas été choisis ont leur longueur d'interaction raccourcie de la distance parcourue au step précédent.
 - On répète la procédure.

Seuil pour la production de secondaires

- Chaque développeur doit se poser la question
jusqu'à quelle limite inférieure en énergie dois-je suivre les particules?
- C'est un compromis
 - Descendre assez bas pour accéder à la physique qui nous intéresse
 - Ne pas aller trop bas car certains processus ont des divergences infrarouges : le CPU va augmenter en conséquence
- La solution traditionnelle en Monte Carlo est d'imposer une coupure ("cut off" ou "cut") absolue en énergie
 - Les particules sont stoppées quand elles atteignent cette énergie
 - L'énergie restante est larguée en ce point

Seuil pour la production de secondaires

- Autre solution adoptée dans Geant4 : imposer une seuil de production
 - Ce seuil est une **distance**, pas une énergie 
 - Valeur par défaut = 1 mm
 - La particule primaire perd de l'énergie en produisant des e- secondaires ou des gammas
 - Si le primaire n'a plus assez d'énergie pour produire des secondaires qui parcoururent au moins 1 mm, alors deux choses se produisent :
 - La perte d'énergie discrète s'arrête (plus de production de secondaires)
 - Le primaire est suivi jusqu'à une énergie nulle en utilisant une perte d'énergie continue
- Une seule valeur est nécessaire pour tous les matériaux car elle correspond à des énergies différentes en fonction du matériau

Seuil pour la production de secondaires

- La valeur par défaut de Geant4 est **1 mm**
 - L'utilisateur doit choisir la meilleure valeur
 - Elle dépendra de la taille des éléments sensibles dans le détecteur et du CPU disponible
- Cette valeur peut être fixée dans la méthode **SetCuts()** de la **physics list**
- Au lieu de “distance seuil de production de secondaires”, on dit simplement “cuts”
 - Mais ne pas oublier que cela ne signifie pas qu'une particule est stoppée avant qu'elle n'ait plus d'énergie

G4UserLimits

- Les user limits sont des limites artificielles appliquées au suivi (tracking)

```
G4UserLimits(G4double ustepMax = DBL_MAX,  
             G4double utrakMax = DBL_MAX,  
             G4double utimeMax = DBL_MAX,  
             G4double uekinMin = 0.,  
             G4double urangMin = 0. );
```

Données membres

- **fMaxStep** taille max de pas (step) dans le volume
- **fMaxTrack** taille max de track
- **fMaxTime** temps global max
- **fMinEkine** énergie cinétique minimale restante (pour particules chargées)
- **fMinRange** parcours minimal restant (pour particules chargées)

Bleu : affectant le step

Jaune : affectant le track

- On peut appliquer ces limites à des volumes logiques et/ou des régions
 - User limits affectées à un volume logique ne se propagent pas aux volumes filles
 - User limits affectées à une région se propagent aux volumes filles, à moins que ces volumes appartiennent à une autre région
 - Si un volume logique ET la région associée ont des user limits, c'est le volume logique qui l'emporte

Processus travaillant avec G4UserLimits

- En plus d'instantier **G4UserLimits** et de l'affecter à un volume logique ou à une région, il faut affecter les processus suivants aux types de particules concernées
- Limites sur le step
 - **G4StepLimiter** : processus doit être défini pour s'appliquer au type de particule concernée
 - Ce processus limite un step (mais ne tue pas une track)
- Limites sur le track
 - **G4UserSpecialCuts** : processus qui doit être défini pour s'appliquer au type de particule concernée
 - Ce processus limite le step et tue le track lorsque le track atteint une de ces limites.
La limitation de step s'applique uniquement au step final.

Résumé

- Geant4 fournit de nombreux processus physiques qui couvrent la physique électromagnétique, hadronique et la décroissance radioactive
- Les processus sont organisés selon le moment où ils doivent être appliqués pendant le suivi d'une particule (discret, continu, au repos, etc.)
- De nombreux processus peuvent être affectés à une particule
 - Celui qui se produit en premier dépend de sections efficaces, durées de vie et distances aux frontières de volumes
- La production des particules secondaires sont déterminés par un seuil de production de secondaires

L'outil Geant4 : Interface



10 minutes

1

Contrôler la simulation

- Une simulation peut être contrôlée par une session **batch** ou par des commandes saisies depuis une session **interactive**
- Mettre en place un **mode batch** est simple : dans le **main()**

```
G4UImanager* UI = G4UImanager::GetUIpointer();
G4String command = "/control/execute";
G4String fileName = argv[1];
UI->ApplyCommand(command+fileName);

UI->ApplyCommand("/control/execute somefile.mac"); // alternatif
```

alors l'exécutable **lira** le fichier macro passé en argument

- Mettre en place un **mode interactif** ("session") est aussi simple, mais il existe plusieurs types d'interfaces utilisateur
 - Classe de base abstraite **G4UIsession**
 - Geant4 fournit plusieurs implémentations

Les types d'interfaces utilisateur

- **G4UIExecutive** – instantiation automatique de la session utilisateur en fonction de variable d'environnement **G4UI_USE_XXX**
- **G4UITerminal** – terminal à caractères de type **C-shell**
 - Fonctionne sur toutes les plates-formes de Geant4
- **G4UItcsh** – terminal à caractères de type **TC-shell**, avec exécution de commandes, historique, etc.
 - Seulement sur Linux et Mac
- **G4UIXm**, **G4UIQt**, **G4UIXWin32** – G4UITerminal implémenté par des librairies **Motif**, **Qt** et **WIN32**
 - A utiliser en combinaison avec les drivers graphiques utilisant les librairies **Xt**, **Qt** et **Win32**
- **G4UIGAG** – GUI basé sur **Java**
 - Fonctionne sur toutes les plateformes de Geant4

Comment utiliser l'interface

- Dans le `main()`, ajouter les lignes :

```
#include "G4UIxxx.hh"    // xxx = Executive, terminal, tcsh, Xm,  
Xaw, XWin32, GAG  
  
G4UIExecutive* session = new G4UIExecutive(argc, argv);  
session->SessionStart();  
delete session;
```

- Pour une **session tcsh (TC-shell)**, la seconde ligne doit être :

```
G4UIsession* session = new G4UITerminal (new G4UITcsh);
```

G4UITerminal

- G4UITerminal est une implémentation concrète dérivée de la classe de base abstraite G4UIsession. Il fournit un terminal à caractères interactif pour saisir les commandes UI de Geant4
- Il comprend les commandes de type Unix pour les répertoires
 - cd, pwd : change et affiche le répertoire courant
 - ls : liste les commandes UI disponibles et les sous-répertoires
- Comprend aussi d'autres commandes
 - history : montre les commandes précédentes
 - !historyID : ré-émet la commande précédente
 - Flèches de contrôle (TC-shell seulement)
 - ?UIcommand : montre les valeurs actuelles des paramètres de la commande UIcommand
 - help UIcommand : help
 - exit : terminer le programme
- Les commandes ci-dessus sont interprétées par le G4UITerminal et ne sont pas passées au noyau de Geant4.
On ne peut pas les utiliser dans un fichier macro.

Variables d'environnement

- Pas de variable nécessaire pour utiliser **G4UITerminal**, **G4UItcsh**, **G4UIGAG**
 - Ces sessions n'ont pas besoin de librairies extérieures, ainsi elles sont automatiquement construites et liées
- Pour construire **G4UIXM**, **G4UIQt**, ou **G4UIXWin32**

```
setenv G4UI_USE_XM 1  
ou  
setenv G4UI_USE_QT 1  
ou  
setenv G4UI_USE_WIN32 1
```

Commandes disponibles

- Geant4 fournit un certain nombre de commandes utilisateur qui peuvent être utilisées
 - Interactivement via une (G)UI
 - Dans un fichier macro, via la commande `/control/execute macroFile`
 - Dans un code C++ utilisant la méthode `ApplyCommand` du `G4UImanager`
- Se reporter au **Geant4 User's Guide For Application Developers**

Exemple de fichier macro

- C'est un fichier **ASCII** qui contient les commandes UI
- Toutes les commandes doivent être indiquées avec leur chemin complet
- Utiliser **#** pour une ligne de commentaires
 - Toute la ligne sera ignorée
 - Les commentaires seront affichés si `/control/verbose` vaut 2
- Nous verrons un exemple de fichier macro dans le code **simulation**
- Le fichier macro peut être exécuté
 - Interactivement ou dans un autre fichier macro

```
/control/execute nomFichier  
– Codé en dur
```

```
G4UImanager* UI = G4UImanager::GetUIpointer();  
UI->ApplyCommand("/control/execute nomFichier");
```

En résumé

- Session **batch** ou **interactive** pour contrôler la simulation
- Utilisation de fichiers **ASCII** («**macro**») pour piloter la simulation
 - Nombreuses commandes disponibles

L'outil Geant4 : Documentation, exemples, forum...



10 minutes

1

Site internet Geant4

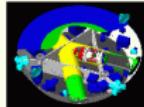
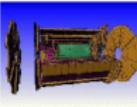
<http://geant4.org>

Download | User Forum | Gallery | Site Index | Contact Us | Search Geant4

Geant4

Geant4 is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The two main reference papers for Geant4 are published in Nuclear Instruments and Methods in Physics Research, [NIM A 506 \(2003\) 250-303](#), and J. Allison et al., IEEE Trans. Nucl. Sci. 53, February 2006 (in press).

Applications **User Support** **Results & Publications** **Collaboration**

A sampling of applications, technology transfer and other uses of Geant4

Getting started, user guides and information for developers

Validation of Geant4, results from experiments and publications

Who we are: collaborating institutions, members, organization and legal information

The original web pages will be available during the transition period.

Applications | User Support | Results & Publications | Collaboration | Site Map | XHTML 1.0 | CSS2 | Contact Webmaster

Last update: Fri 24 Feb 2006 02:30:52 PM PST

News

- 10 February 2006 - Patch 01 of release 8.0 is available from the [download](#) area.
- 16 December 2005 - Geant4 release 8.0 is available from the [download](#) area.
- [less recent news](#)

Events

- 4-day Geant4 tutorial, SLAC, Stanford (USA), 7-10 March 2006.
- 4-day Geant4 tutorial, Jefferson Lab, Newport News, Virginia (USA), 22-25 May 2006.
- [past events](#)

Guide d'installation

- Depuis le site internet, dans user guides
<http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/InstallationGuide/html/index.html>
- Liste les logiciels nécessaires
 - Compilateur C++, Geant4
- Comment installer Geant4 sur Linux
- Comment installer Geant4 sur Windows

Guide de l'utilisateur

- Depuis le site internet, dans user guides
<http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForApplicationDeveloper/html/index.html>
- Introduction à l'utilisation de Geant4
- Description des outils les plus utiles
- Explique comment écrire et exécuter une application de simulation
- Ce n'est qu'un aperçu...

Manuel de référence physique

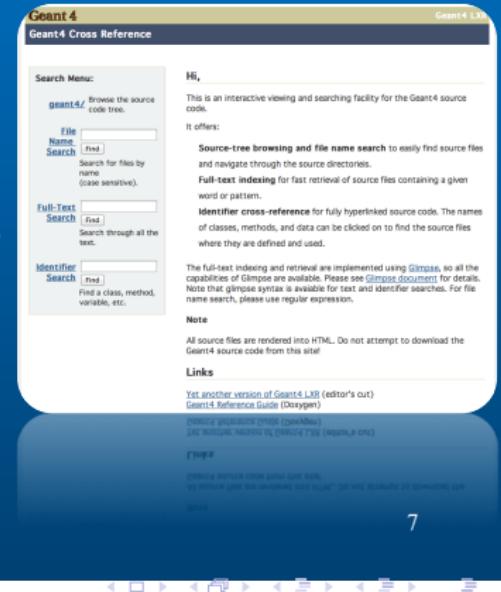
- Depuis le site internet, dans **user guides**
<http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/PhysicsReferenceManual/fo/PhysicsReferenceManual.pdf>
- Référence pour utilisateurs et développeurs qui souhaitent se documenter sur **la physique d'une interaction**
- Présente la **formulation théorique, le modèle ou la paramétrisation** de toutes les interactions physiques fournies dans Geant4

Guide de développement

- Depuis le site internet, dans user guides
<http://geant4.web.cern.ch/geant4/UserDocumentation/UsersGuides/ForToolkitDeveloper/html/index.html>
- Une description de l'architecture OO de Geant4
 - Diagrammes de classes
 - Philosophie expliquant les choix de design
- Un guide pour utilisateurs qui veulent étendre les fonctionnalités de Geant4
 - Ajouter de nouveaux solides, modifier le navigateur, créer de nouveaux champs, etc.

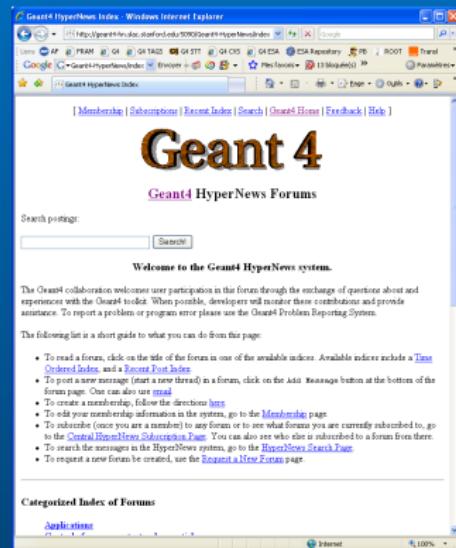
Explorateur de code LXR

- Internet :
<http://www-geant4.kek.jp/LXR>
- Chercher dans le code source de Geant4
 - Un nom de fichier (ex. G4Track.hh)
 - Du texte
 - Un identificateur
- Résultats: un fichier internet contenant des liens vers classes et méthodes
 - Indique où les classes et méthodes sont définies
 - Indique où elles sont référencées
- Version doxygen
<http://www-geant4.kek.jp/Reference>



Forum utilisateurs

- Internet : section **user forum**
<http://hypernews.slac.stanford.edu/HyperNews/geant4/cindex>
- Discuter des problèmes avec d'autres utilisateurs, poser des questions aux experts, proposer des idées, etc...
- **27 forums basés sur les catégories de Geant4**
- **4 forums basés sur des applications spécifiques**
- **Ouvert à la création**
- **Pour joindre : cliquer sur "Membership"**



Exemples

- Des **exemples novices** de facile à complexe
 - Peuvent être utilisés pour démarrer votre application
 - Code dans `$G4SRC/examples/novice`
- De nombreux **exemples étendus**
 - Users' Guide: For Application Developers
 - Code dans `$G4SRC/examples/extended`
- Des **exemples avancés**
 - Users' Guide: For Application Developers
 - Code dans `$G4SRC/examples/advanced`

En résumé

- Documentation riche et variée, en ligne
 - Installation
 - Développement d'application
 - Physique
 - Code
 - Extension de Geant4
- Forum utilisateur
- Exemples novices/étendus/avancés
- Site web **geant4.org**

L'outil Geant4 :

Ecrire votre application utilisateur



20 minutes

1

Application utilisateur

- **Geant4 est un outil**
 - i.e. on ne peut pas l'utiliser comme une **boîte noire**
 - Il faut écrire une application, qui utilise les outils de Geant4
- **Conséquences**
 - Il n'existe **pas de réglages / choix par défaut**
 - Il faut fournir toute l'information pour configurer son application
 - Il faut délibérément choisir quels outils utiliser
- **Guide: de nombreux exemples**
 - **Exemples novices pour un aperçu des outils**
 - **Exemples étendus et avancés pour étudier les outils**

Concepts de base

- Ce qu'il faut faire
 - Décrire votre dispositif expérimental
 - Choisir particules et modèles physiques utiliser et choisir la précision de la simulation
 - cuts pour produire et suivre les particules secondaires
 - Fournir la description des particules primaires
- En option, on peut
 - Interagir avec le noyau de Geant4 pour contrôler la simulation
 - Visualiser la configuration et les résultats
 - Produire des histogrammes et tuples pour analyse ultérieure
 - ...

Rappels sur le noyau de Geant4

- L'architecture de Geant4 fournit des outils pour développer une application
 - Pour indiquer au noyau quelle est la configuration de votre simulation
 - Pour interagir directement avec le noyau
- Les outils pour interagir avec Geant4 sont les classes de base
 - Vous créez vos propres classes concrètes, les classes utilisateur, dérivées des classes de base
 - Le noyau de Geant4 traite vos classes dérivées de façon transparente (polymorphisme)
- Classes de base ABSTRAITES
 - Les classes utilisateurs dérivées sont OBLIGATOIRES
- Classes de base CONCRETES (avec des méthodes virtuelles)
 - Les classes utilisateurs dérivées sont OPTIONNELLES

Classes utilisateur

Classes d'initialisation

Invoquées à l'initialisation

- **G4VUserDetectorConstruction**
- **G4VUserPhysicsList**

En ROUGE,

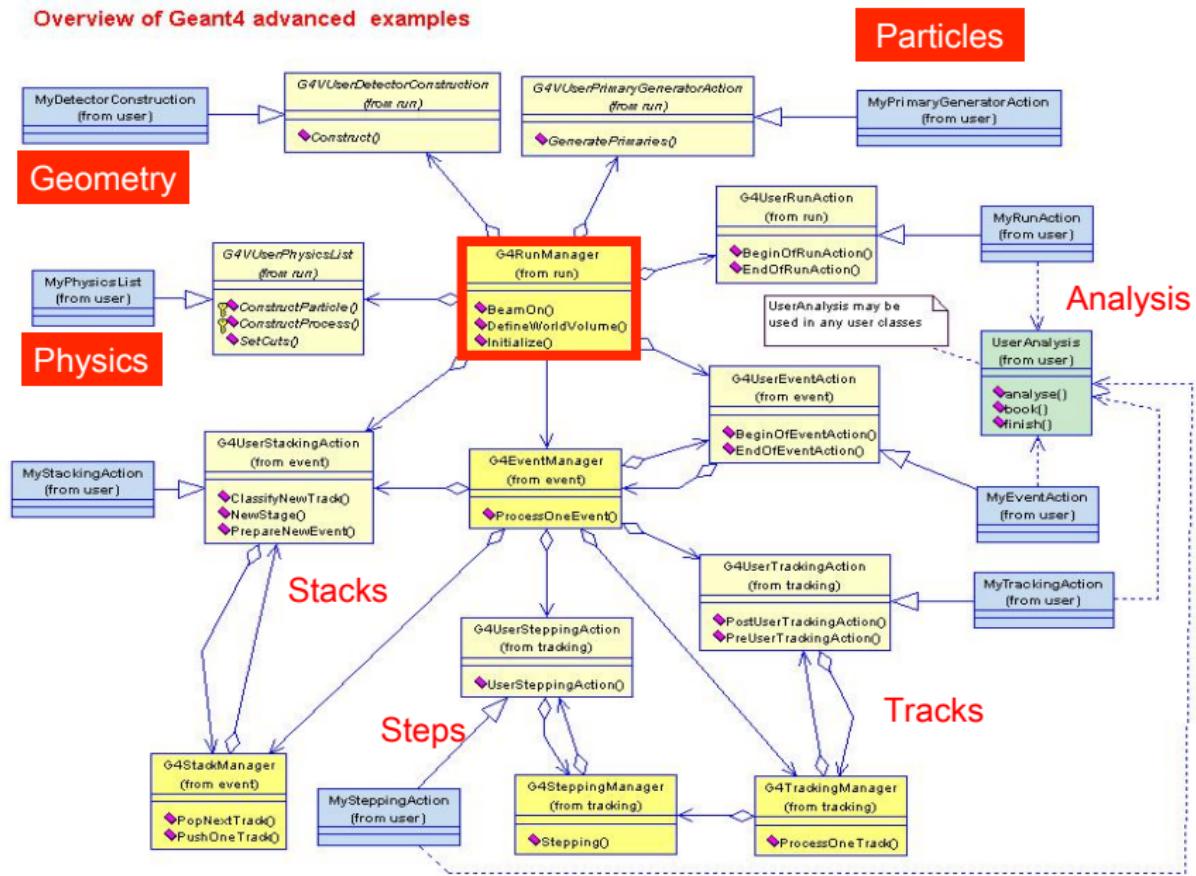
obligatoires !
(classes abstraites)

Classes d'action

Invoquées pdt la boucle d'exécution

- **G4VUserPrimaryGeneratorAction**
- **G4UserRunAction**
- **G4UserEventAction**
- **G4UserTrackingAction**
- **G4UserStackingAction**
- **G4UserSteppingAction**

Overview of Geant4 advanced examples



Construisons une application simple en dosimétrie

Calculer la dose déposée par un alpha de 5 MeV dans une cellule de tissu biologique

1) Le programme `main`

- Geant4 ne fournit pas de `main()`
 - Geant4 est un outil !
 - Le `main()` fait partie de l'application de l'utilisateur
- Dans `main()`, l'utilisateur DOIT
 - Construire `G4RunManager`
 - Notifier au `G4RunManager` les classes utilisateur obligatoires dérivées de
 - `G4VUserDetectorConstruction`
 - `G4VUserPhysicsList`
 - `G4VUserPrimaryGeneratorAction`
- On PEUT aussi y définir
 - Des classes utilisateur d'action
 - VisManager, session (G)UI



`Simulation.cc`

8

2) Décrire le dispositif expérimental

- Dériver votre propre classe concrète de la classe de base abstraite **G4VUserDetectorConstruction**
- Implémenter la méthode **Construct()**
 - Construire tous les matériaux nécessaires
 - Définir les formes/solides nécessaires pour décrire la géométrie
 - Construire et placer les volumes de votre géométrie
 - Définir les détecteurs sensibles et identifier les volumes de détecteur auxquels les associer
 - Définir des attributs de visualisation des éléments du détecteur
 - Associer un champ électromagnétique à des régions de détecteur

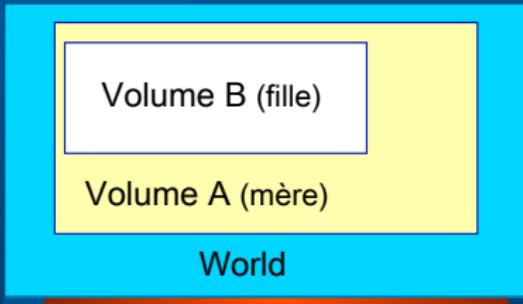


DetectorConstruction.hh et .cc

9

Définir la géométrie (suite)

- Trois niveaux conceptuels
 - **G4VSolid** forme, taille
 - **G4LogicalVolume** matériau, sensibilité, champ magnétique...
 - **G4VPhysicalVolume** position, rotation
- Un volume physique unique, le **WORLD** volume, qui représente la zone expérimentale, doit être obligatoirement défini et doit contenir tous les autres volumes !



ex.: Volume A est mère
du Volume B

Le volume mère doit contenir
entièvement le volume fille



DetectorConstruction.hh et .cc

3) Sélectionner les processus physiques

- Geant4 ne possède pas de particules et processus par défaut
- Dériver votre propre classe concrète de la classe de base abstraite **G4UserPhysicsList**
 - Définir toutes les particules nécessaires
 - Définir tous les processus et les assigner aux bonnes particules
 - Définir les seuils de productions (cut)
- Méthodes virtuelles pures de **G4UserPhysicsList** qu'il faut implémenter
 - **ConstructParticles()**
 - **ConstructProcess()**
 - (et on conseille **SetCuts()**)
- Penser à utiliser les constructeurs de physique de Geant4...



PhysicsList.hh et .cc

4) Tirer les événements primaires

- Dériver votre propre classe concrète de la classe de base abstraite
G4UserPrimaryGeneratorAction
- Définir les particules primaires en fournissant : type, direction initiale, position initiale, énergie initiale
- Implémenter la fonction membre virtuelle
GenerateParticles()

PrimaryGeneratorAction.hh et .cc

5) Classes d'action optionnelles

- Il existe **cinq classes de base concrètes** dont les fonctions membres virtuelles peuvent être réécrites pour contrôler la simulation à divers endroits
 - `G4UserRunAction`
 - `G4UserEventAction`
 - `G4UserSteppingAction`
 - `G4UserTrackingAction`
 - `G4UserStackingAction`
- Chaque fonction membre des classes de base possède une implémentation vide : elle ne font rien
- L'utilisateur peut implémenter ces fonctions membres dans sa classe dérivée
- Les objets des classes d'action utilisateur doivent être déclarées au `G4RunManager`

Classes d'action optionnelles (suite)

G4UserRunAction

- `BeginOfRunAction(const G4Run*)`
 - ex: déclarer des histogrammes
- `EndOfRunAction(const G4Run*)`
 - ex: sauver des histogrammes

G4UserEventAction

- `BeginOfEventAction(const G4Event*)`
 - ex: sélectionner un événement
- `EndOfEventAction(const G4Event*)`
 - ex: analyser l'événement



RunAction.hh et .cc
EventAction.hh et .cc
SteppingAction.hh et .cc

G4UserSteppingAction

- `UserSteppingAction(const G4Step*)`
 - ex: dessiner le step, récupérer des informations physiques
 - ex: tuer, suspendre, remettre à plus tard la track

Classes d'action optionnelles (suite)

G4UserTrackingAction

- `PreUserTrackingAction (const G4Track*)`
 - ex: décider si une trajectoire doit être conservée ou non
- `PostUserTrackingAction (const G4Track*)`

G4UserStackingAction

- `PrepareNewEvent ()`
 - ex: réinitialiser le contrôle de priorité
- `ClassifyNewTrack (const G4Track*)`
 - invoquée à chaque fois qu'une track est poussée
 - ex: classer une nouvelle track (contrôle de priorité)
 - Urgent, en attente, remettre au prochain event, tuer
- `NewStage ()`
 - invoquée quand la pile urgente est vide
 - ex: changer les critères de classification
 - ex: filtrer les événements (annulation d'événement)



TrackingAction.hh et .cc
StackingAction.hh et .cc

6) GUI et visualisation (option)

- Dans le `main()`, selon votre environnement informatique, instantier une classe concrète `G4UIsession` et invoquer la méthode `sessionStart()`
- Geant4 fournit
 - `G4UIExecutive` (Qt)
 - batch job avec un fichier macro
 - ...
- Dans le `main()`, selon votre environnement informatique, instantier `G4VisExecutive` et invoquer la méthode `initialize()`
- Geant4 fournit des interfaces à de nombreux pilotes graphiques:
 - Qt (`G4VisExecutive`)
 - DAWN (*Fukui renderer*)
 - WIRED
 - RayTracer (*ray tracing by Geant4 tracking*)
 - OPACS
 - OpenGL
 - OpenInventor
 - VRML
 - ...

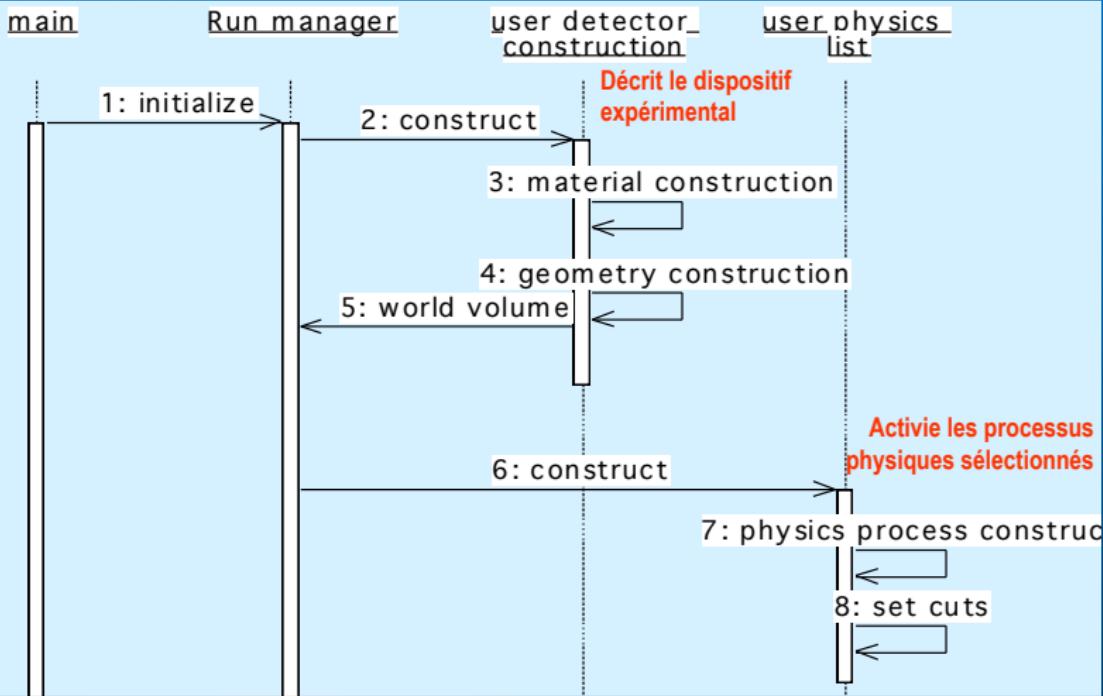


Simulation.cc

7) Histogrammes (option)

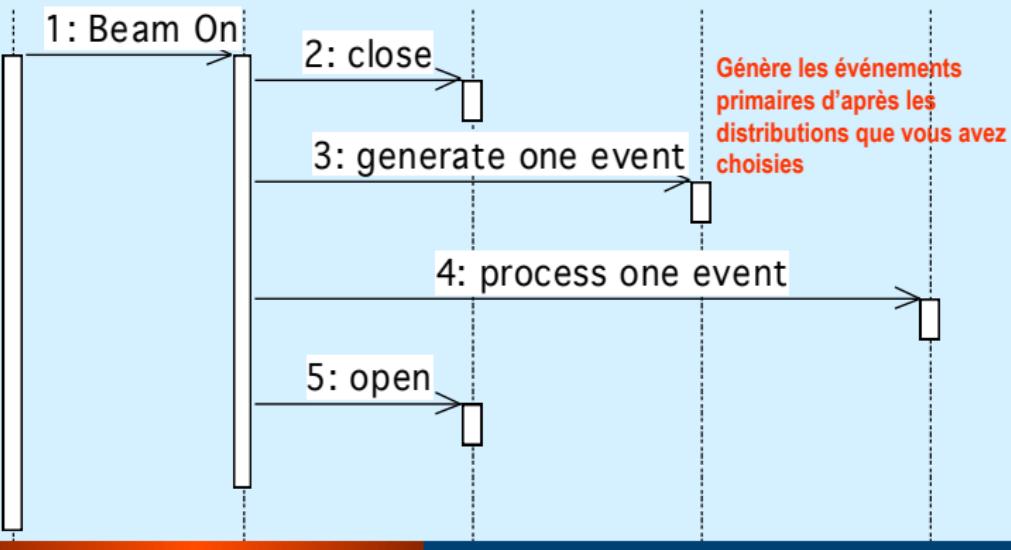
- Vous avez la possibilité de créer votre propre classe (par ex. **HistoManager**) pour récupérer les informations qui vous intéressent à chaque étape de la simulation (step, event, run, ...) et les stocker dans
 - Des **histogrammes**
 - Des **ntuples**
- Utilise les fonctionnalités de l'outil ROOT₁₇

Initialisation

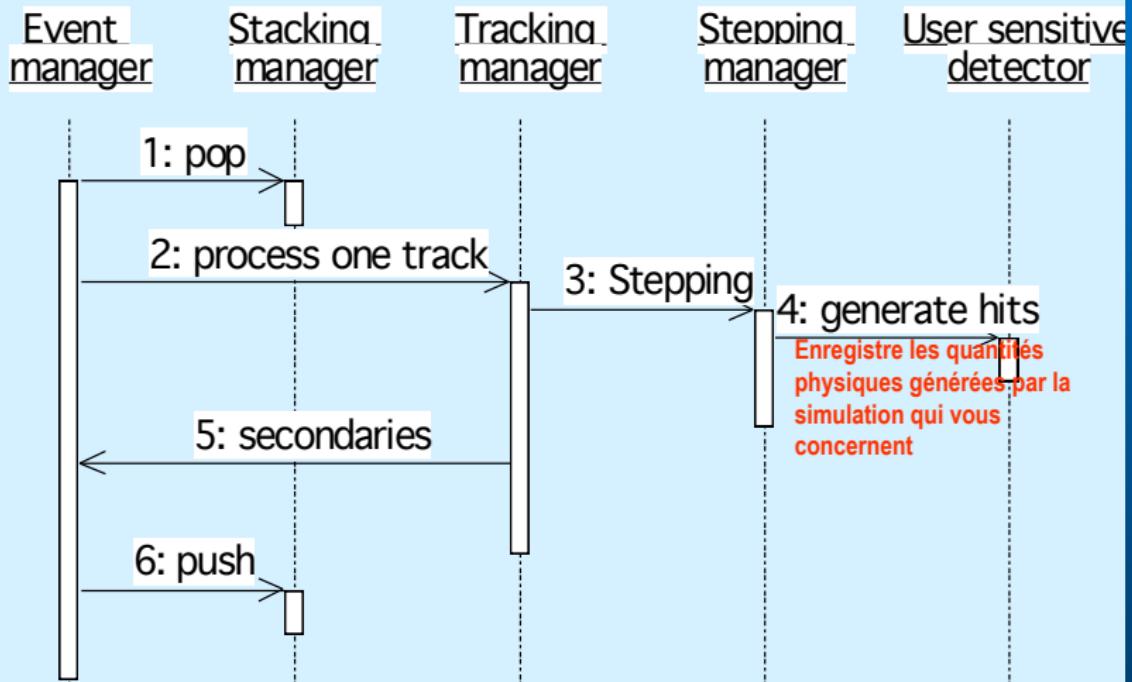


Beam On

main Run Manager Geometry_manager Event generator Event Manager



Calcul de l'event



En résumé

- Construction d'une application utilisateur en quelques étapes
- Nécessite des classes utilisateurs
 - d'initialisation
 - Obligatoires (géométrie, Physique)
 - d'action
 - Obligatoire (primaires) et optionnelles

Les hooks



10 minutes

1

Extraire l'information utile

- Une fois la géométrie, la physique et la génération des primaires définis, Geant4 simule la physique de façon **silencieuse**
 - A vous d'ajouter du code pour **extraire l'information qui vous intéresse**
- Plusieurs façons:
 - Les **user hooks** (`G4UserSteppingAction`, etc.)
 - Accès complet à toute l'information
 - Facile, mais tout faire soi-même
 - Définir des détecteurs **sensibles** : assigner `G4VSensitiveDetector` à un volume et en option générer des “**hits**”
 - Utiliser des **user hooks** (`G4UserEventAction`, `G4UserRunAction`) pour obtenir un résumé de l'événement ou du run
 - Utiliser les **commandes existantes de scoring**
 - La plupart des quantités physiques sont disponibles

User hooks

- Approche “faites-le vous-même”
- Dans Geant4, on a accès à presque toute l'information
 - `G4UserSteppingAction`
 - `G4UserEventAction`
 - `G4UserRunAction`
 - `G4UserTrackingAction`
- Bien adaptés aux **petites applications** & exemples Geant4
- Trop lourd pour les **applications complexes**, où de nombreuses données de nombreux volumes doivent être enregistrées
 - `SteppingAction` trop chargé
 - Subdiviser le problème, avec des « sensitive detectors »

Principe

- Dans le **SteppingAction**, vérifier que la particule est dans le volume A et faites ce dont vous avez besoin
- En général, vos “conteneurs” et histogrammes seront des attributs de **Track**, **Event** ou **Run**
 - ainsi il faudra instancier **TrackingAction** et/ou **EventAction** et/ou **RunAction**
 - passer leur pointeur au **SteppingAction**
- Voir par exemple
 - examples/novice N03, N06,
 - extended/electromagnetic, optical, ...

Information géométrique - 1

- Un objet G4Step est constitué de deux points:

```
G4StepPoint* point1 = step->GetPreStepPoint();  
G4StepPoint* point2 = step->GetPostStepPoint();
```

- Pour obtenir leur position dans le système de coordonnées globales:

```
G4ThreeVector pos1 = point1->GetPosition();  
G4ThreeVector pos2 = point2->GetPosition();
```

- Dans la suite, on appellera volume courant le volume que le step a traversé



L'information géométrique s'obtient du preStepPoint

Information géométrique - 2

- G4VTouchable et ses dérivées conservent l'information géométrique et la hiérarchie

```
G4TouchableHandle touch1 = point1->GetTouchableHandle();
```

- Pour avoir le volume courant

```
G4VPhysicalVolume* volume = touch1->GetVolume();
```

- Son nom

```
G4String name = volume->GetName();
```

- Son numéro de copie

```
G4int copyNumber = touch1->GetCopyNumber();
```

- Son volume logique

```
G4LogicalVolume* lVolume = volume->GetLogicalVolume();
```

Information géométrique - 3

- Pour récupérer le **matériaux**, les deux lignes suivantes sont équivalentes

```
G4Material* material = point1 ->GetMaterial();  
G4Material* material = lVolume ->GetMaterial();
```

- Pour obtenir la **region**

```
G4Region* region = lVolume->GetRegion();
```

- Et le **volume mère**

```
G4VPhysicalVolume* mother = touch1->GetVolume(depth=1);
```

Grand-mère: depth=2 ...etc...

- Et le **numéro de copie de la mère**

```
G4int copyNumber = touch1->GetCopyNumber(depth=1);
```

Grand-mère: depth=2 ...etc...

Information géométrique - 4

- Pour vérifier que la particule a pénétré dans le volume courant, cad se trouve au premier step dans le volume, le `preStepPoint` est à la frontière (`boundary`)

```
if (point1->GetStepStatus() == fGeomBoundary)
```

- Pour vérifier que la particule quitte le volume courant , cad qu'elle se trouve au dernier step du volume, le `postStepPoint` est à la frontière (`boundary`)

```
if (point2->GetStepStatus() == fGeomBoundary)
```

- Alors, on récupère le `touchable` du volume suivant

```
G4TouchableHandle touch2 = point2->GetTouchableHandle();
```

- Depuis `touch2`, toute l'information sur le volume suivant, comme ci-dessus

Information Physique

- Récupérer le processus qui a limité le step courant

```
G4VProcess* aProcess = point2->GetProcessDefinedStep();
```

- Nom de la particule

```
step->GetTrack()->GetDynamicParticle()->GetDefinition()->GetParticleName()
```

- Les quantités physiques sont disponibles depuis le step (**G4Step**) ou le track (**G4Track**)

- Energie déposée, longueur de step, déplacement, temps de vol

```
G4double eDeposit      = step->GetTotalEnergyDeposit();
G4double length       = step->GetStepLength();
G4ThreeVector displace = step->GetDeltaPosition();
G4double tof          = step->GetDeltaTime();
```

- Moment, énergie cinétique, temps global (depuis le commencement de l'event) de la track après le step en cours



```
G4Track* track        = step->GetTrack();
G4ThreeVector momentum = track->GetMomentum();
G4double kinEnergy    = track->GetKineticEnergy();
G4double globalTime   = track->GetGlobalTime();
```

- Pour transformer du système de coordonnées globales au système local du volume en cours, utiliser l'information preStepPoint

```
G4ThreeVector localPosi = touch1->GetHistory()->GetTopTransform().TransformPoint(position); 9
```

Et plus encore...

- De même dans **TrackingAction** on accède aux information de la **track**

```
void MyTrackingAction::PostUserTrackingAction(const G4Track* track)
{
    G4double tracklen = track->GetTrackLength();
    G4double charge   = track->GetDefinition()->GetPDGCharge();
    G4ThreeVector position = track->GetPosition(); // C'est celle du PostStepPoint
    ...
}
```

- Voir plus dans
 - **\$G4SRC/source/track/include/G4Step.hh**
 - **\$G4SRC/source/track/include/G4Track.hh**
 - ...
- On peut récupérer facilement à chaque step des quantités et les cumuler sur les événements ou sur le run en utilisant des **accesseurs/enregistreurs ajoutés à vos classes EventAction et RunAction**

```
G4double dose = (aStep->GetTotalEnergyDeposit()/joule) / (Run->GetMassTarget() /kg);
Run->AddDoseN(dose);
```

10

En résumé

- Nécessaire d'écrire votre propre code pour extraire les informations qui vous intéressent
- Usage de hooks
 - Grande flexibilité (géométrie, physique) mais à réserver à un usage simple
- Alternativement, utilisation de détecteurs sensibles ou de méthodes de scoring

L'environnement LINUX - le minimum pour utiliser Geant4 -



15 minutes



1

Introduction

- Geant4 peut s'installer sous plusieurs environnements
 - Linux sur PC avec g++ (compilateur gcc)
 - MacOS X avec g++ (compilateur gcc)
 - Windows/XP avec Microsoft Visual C++
- Nous allons travailler avec Geant4 sous Linux qui est devenu un OS standard pour le calcul scientifique
 - constamment mis à jour, gratuit, ...

Introduction

- Plus précisément, nous allons travailler avec l'outil de virtualisation VMware (gratuit) qui permet d'émuler simultanément un véritable PC Linux directement sous Windows. Pour en savoir plus

<http://geant4.in2p3.fr>

- Possibilité d'échanger des fichiers entre Linux et Windows
- Nous travaillerons avec Scientific Linux 5.8 (développé par Fermilab et le CERN) et Geant4 9.6P02, déjà installés pour vous

<http://geant4.in2p3.fr>

Tout est gratuit !

The screenshot shows the official website for Geant4, which is a simulation toolkit. The top navigation bar includes links for Overview, Members, Activities, News, Tutorials and teachings, Conferences, workshops and meetings, Geant4 for VMware & VirtualBox, Jobs, Useful links, Publications, The Geant4-DNA project, The ESA BioRad project, BioRad Collaboration, Visualization & Qt, Search On this website, and On the whole CNRS Web.

News front page:

- New Geant4 tutorial at Jefferson Lab, VA, USA
- A five day hands-on course based on the use of the Geant4 toolkit will be held at Jefferson Lab on July 9-13, 2012. See link.
 - Tuesday 15 May 2012 - [Read](#)
- New update of Geant4 9.5+P01 virtual machine + Qt
- Download the software suite from the Geant4 for VMware section.
- Tuesday 17 April 2012 - [Read](#)
- Geant4-DNA collaboration meeting
- The next Geant4-DNA collaboration meeting will be held at CENBG on March 6-7, 2012, see this link.
 - Sunday 4 March 2012 - [Read](#)
- New position opening at Bordeaux 1 U. / CENBG
- A full permanent professor position will be open at Bordeaux 1 University / CENBG, France in 2012. Please see details link.
 - Tuesday 10 January 2012 - [Read](#)
- New update of Geant4 9.5 virtual machine
- Download the software suite from the Geant4 for VMware section.
- Tuesday 3 January 2012 - [Read](#)
- New web site for the Geant4-DNA project

Visualization & Qt:

- Search On this website
- On the whole CNRS Web

<http://twitter.com/geant4vm>

Geant4VM sur Twitter

Geant4VM (Geant4VM) sur Twitter

Geant4 @ IN2P3 ~ Geant4 for VMware & VirtualBox

Rechercher

Vous avez déjà un compte ? Se connecter ~

Geant4VM @Geant4VM

Download freely the Geant4 virtual machine for VMware(TM)/VirtualBox(TM) ! See link below.

CENBG, Bordeaux, France <http://geant4.in2p3.fr/spip.php?rubrique8&lang=en>

Suivre

4 TWEETS

2 FOLLOWING

20 FOLLOWERS

Suivre Geant4VM

Nom complet

Adresse email

Mot de passe

S'inscrire

Tweets

Geant4VM @Geant4VM 17 Avr
Geant 9.5+P01+Qt for VMware and VirtualBox has been released. See geant4.in2p3.fr/spip.php?rubri... Détourer

Geant4VM @Geant4VM 3 Juin
Geant 9.5 for VMware and VirtualBox has been released. See geant4.in2p3.fr/spip.php?rubri... Détourer

Geant4VM @Geant4VM 16 Sept
Geant 9.4+Patch02 for VMware has been released. See geant4.in2p3.fr/spip.php?rubri... Détourer

Geant4VM @Geant4VM 10 Juil 11
Geant4 releases for VMware(TM) and VirtualBox(TM) will be regularly announced on Twitter. Détourer

© 2012 Twitter À propos Aide Conditions Confidentialité Blog Statut du service Applications Ressources Emplois Annonciers Professionnels Médias Développeurs



Recevez les mises à jour de Geant4 pour VMware & VirtualBox

Accéder à Linux

- On se connecte avec :
 - Username: local1
 - Password: local1
- Un clic droit de souris sur le bureau permet d'ouvrir un terminal de travail, (Open Terminal) là où vous allez directement saisir les commandes de contrôle

Les shells

- Ce sont les interpréteurs de commandes disponibles

<code>/bin/sh</code>	Shell POSIX, standardisé, scripts
<code>/bin/ksh</code>	Korn shell, amélioré pour l'interactif
<code>/bin/csh</code>	C-shell, utile une syntaxe proche du C
<code>bash</code>	Shell standard sous Linux, Bourne Again shell
<code>tcsh</code>	C-shell amélioré
<code>zsh</code>	Bash et tcsh amélioré

Naviguer dans les répertoires

<code>pwd</code>	Affiche le répertoire courant
<code>cd <i>répertoire</i></code>	Déplacement dans le répertoire
<code>cd</code>	Déplacement vers le répertoire maison (~)
<code>cd ..</code>	Déplacement vers le répertoire parent
<code>ls</code>	Liste les fichiers
<code>ls -a</code>	Liste aussi les fichiers cachés

Déplacer, copier, créer, supprimer...

<code>mv source cible</code>	Déplace et/ou renomme un fichier
<code>cp source cible</code>	Copie un fichier
<code>cp -R source cible</code>	Copie un répertoire
<code>mkdir repertoire</code>	Crée un répertoire
<code>rmdir repertoire</code>	Supprime un répertoire VIDE
<code>du -ks repertoire</code>	Donne la taille d'un répertoire en ko
<code>rm fichier</code>	Efface un fichier
<code>rm -f fichier</code>	Efface un fichier protégé en écriture
<code>rm -R repertoire</code>	Efface un répertoire

Autres commandes utiles

<code>diff fichier1 fichier2</code>	Affiche les différences entre deux fichiers texte
<code>wc fichier</code>	Compte le nombre de lignes, de mots, d'octets dans fichier
<code>more fichier</code>	Affiche un fichier page après page (espace pour page suivante, entrée pour ligne suivante, u pour remonter)
<code>echo nomDeVariableEnv</code>	Affiche la valeur d'une variable d'environnement

Outils utiles pré-installés

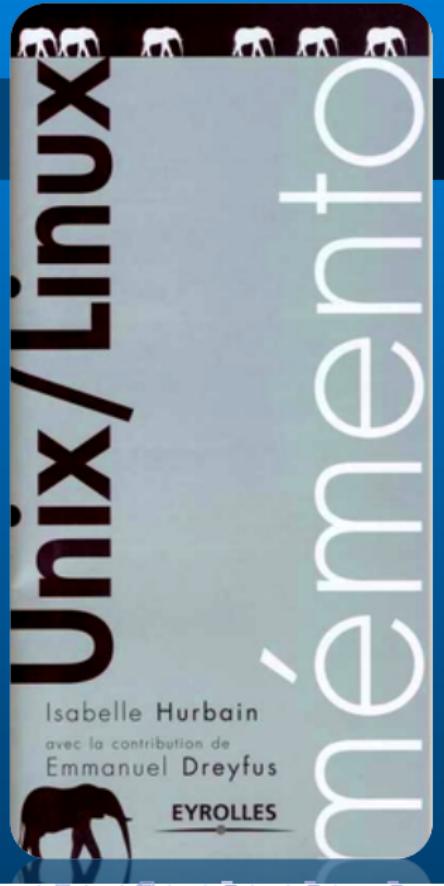
- `nedit` : éditeur de fichiers texte
- `snavigator` : gestionnaire de projet C++ (**I**ntegrated **D**evelopment **E**nvironment), que l'on va utiliser pour développer notre application
- `cmake` : permet de compiler votre application Geant4
- `root` : outil d'analyse de données qui permettra d'analyser les résultats de la simulation
- **Rajouter & après le nom de commande pour garder la main**

Variables d'environnement

- Elles sont déjà définies pour vous dans la machine Vmware, voir:
`/usr/local/Env/SL5-64`
 - Sous **tcsh**, on les définit avec la commande **setenv VARIABLE value**
 - Et on vérifie leur valeur avec **echo \$VARIABLE**
 - Pour Geant4
 - **\$G4INSTALL**
 - Répertoire où est installé Geant4
 - **\$G4SYSTEM**
 - OS
 - Vaut Linux-g++
- et beaucoup d'autres (ex. **\$G4SRC** pour les sources et exemples...)

Pour en savoir plus

**Unix/Linux mémento
chez Eyrolles**



Premiers exemples...



25 minutes

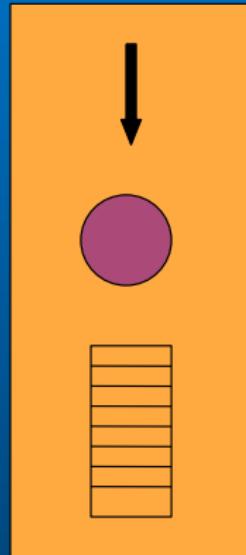
1

Les **exemples** de Geant4

- Des exemples sont déjà disponibles dans Geant4
- Situés dans **\$G4SRC**
- 3 catégories
 - **Novice** : fonctionnalités de base
 - **Extended** : fonctionnalités spécifiques
 - Processus Physiques, « biasing », champs électromagnétiques...
 - **Advanced** : simulations complètes de cas d'utilisation
 - Physique médicale, spatiale, calorimétrie...

Exemple novice N01

- Géométrie
 - Volume de gaz Ar contenant un cylindre de Al et un bloc de Pb avec des tranches de Al
- Particules incidentes: geantino
 - pas d'interactions physiques
- Seul le transport est actif
- Gestion “batch” et “verbosité”



Essayons l'exemple N01

- Copier N01 vers votre « home »

```
cd  
cp -R $G4SRC/examples/novice/N01 .  
ls  
mkdir N01build  
cd N01build
```
- Lire le fichier **README** : exampleN01 est l'exemple **le plus simple** parmi les exemples novices. Il décrit un tube de tracking et un calorimètre sandwich constitué de boîtes. Il tire **3 geantinos** (pseudo-particules qui n'ont pas d'interaction) par événement
 - Permet de montrer le fonctionnement de base de Geant4
 - nedit ..\N01\README &
- Compiler & lier l'exemple

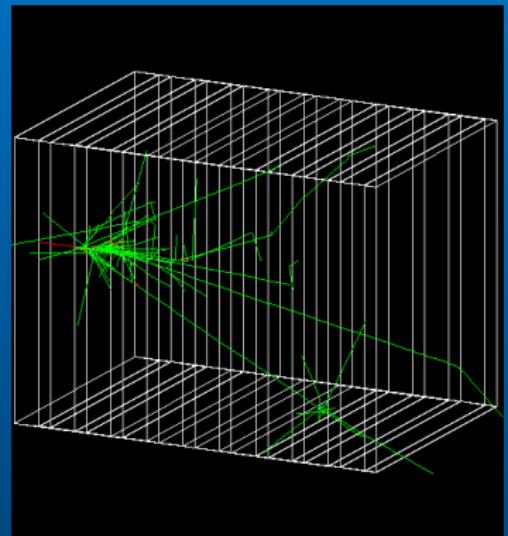
```
cmake -DG4DIR=$G4DIR .../N01  
make -j2
```
- Une fois compilé et lié, lancer l'exécutable avec

```
./exampleN01
```
- **Attention** : il faut toujours que les variables d'environnement soient définies avant de compiler et de lancer votre exécutable

4

Exemple novice N03

- Calorimètre à échantillonnage avec des couches de Pb (absorbeur) et des gaps de lAr (détecteur, **replicas**)
- Liste exhaustive de matériaux
- Interface de commandes
- “**Randomization**” du faisceau incident
- Tous les processus EM + décroissance avec coupures pour γ , e+, e- (à utiliser pour l'étude des gerbes)
- **Réponse du détecteur:** E déposée, longueur des traces dans l'absorbeur et les gaps
- Visualisation
- **Gestion de la “graine aléatoire”**



Essayons l'exemple N03

- Copier N03 vers votre « home »

```
cd  
cp -R $G4SRC/examples/novice/N03 .  
mkdir N03build  
cd N03build
```

- Lire le fichier **README** : simulation d'un simple calorimètre

- Compiler & lier l'example

```
cmake -DG4DIR=$G4DIR .. /N03  
make -j2
```

Exécuter votre code **SANS** macros

- Une fois compilé et lié, vous pouvez exécuter votre code avec

`./exampleN03`

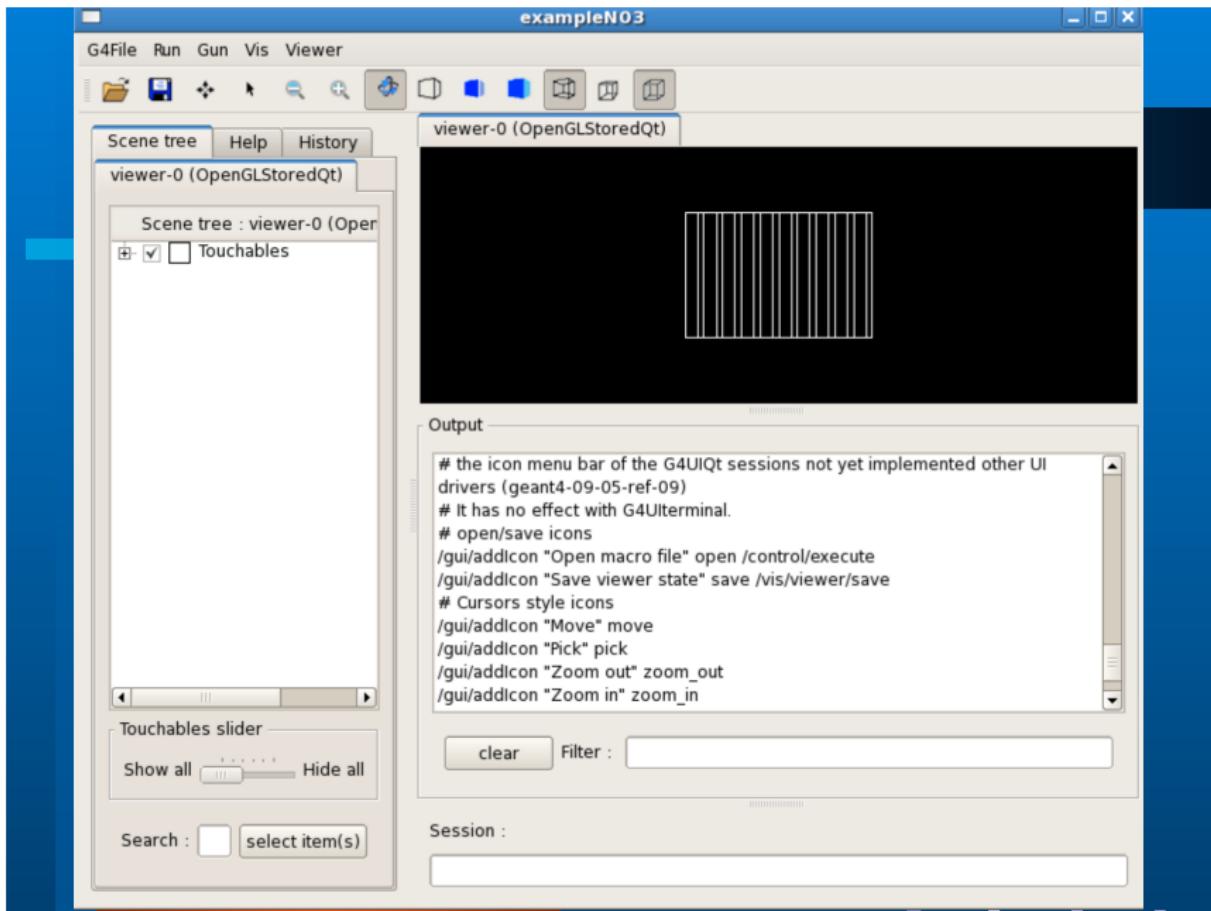


Pas d'arguments après le nom

Utiliser la touche **tabulation** pour une reconnaissance rapide des répertoires existants

- Alors vous obtenez l'affichage suivant

7

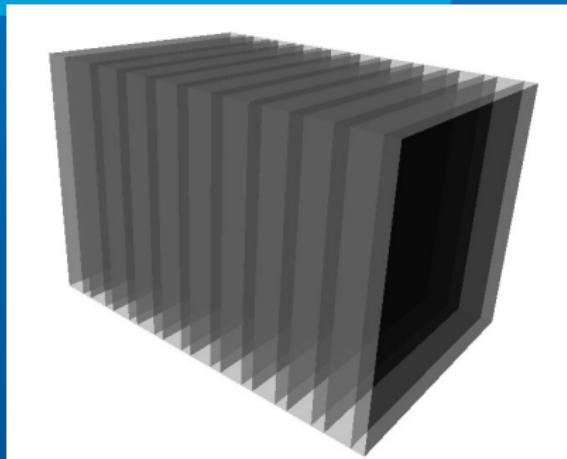


Exécuter votre code **SANS** macros

Que s'est-il passé ?

1. Le **run** est **initialisé**
 1. définition des matériaux,
 2. construction de la géométrie
 3. déclaration des processus physiques
 4. déclaration des coupures
 5. ...
2. Un fichier macro **vis.mac** est automatiquement lu pour activer les pilotes de visualisation et le dispositif simulé apparaît dans une fenêtre graphique (interface Qt)
3. On peut saisir des commandes à la main dans la fenêtre "Session"
 - Par ex., changer la géométrie, choisir quelles particules tirer, à quelle énergie, exécuter un autre fichier macro, ...

Géométrie par défaut



Obtenu avec le
driver VRML

- **10 couches : 10 mm Pb + 5 mm L.Ar**
- **pas de champ magnétique**

10

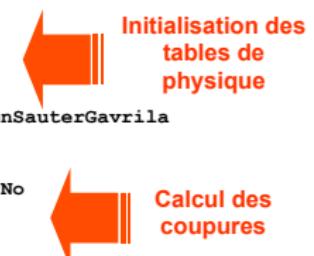
Exécuter votre code sans macros

Essayons de tirer une particule

Dans la fenêtre Session: /run/beamOn 1

Par défaut, un électron de 50 MeV est tiré en incidence normale sur le calorimètre

```
phot:   for gamma SubType= 12
LambdaPrime table from 200 keV to 10 TeV in 54 bins
===== EM models for the G4Region DefaultRegionForTheWorld =====
PhotoElectric : Emin=          0 eV      Emax=        10 TeV    AngularGenSauterGavrila
[...]
Index : 1      used in the geometry : Yes      recalculation needed : No
Material : Lead
Range cuts       : gamma 1 mm     e- 1 mm     e+ 1 mm proton 1 mm
Energy thresholds : gamma 101.843 keV   e- 1.36749 MeV   e+ 1.27862 MeV proton 100 keV
Region(s) which use this couple :
DefaultRegionForTheWorld
```



Initialisation des
tables de
physique

Calcul des
coupures

Exécuter votre code sans macros

```
---> Begin of event: 0
---> End of event: 0
Absorber: total energy: 40.0567 MeV      total track length: 2.9093 cm
Gap: total energy: 5.52904 MeV      total track length: 2.63017 cm
Run terminated.
Run Summary
Number of events processed : 1
User=0.01s Real=0.12s Sys=0.02s
-----End of Run-----
```



Résumé de
l'event

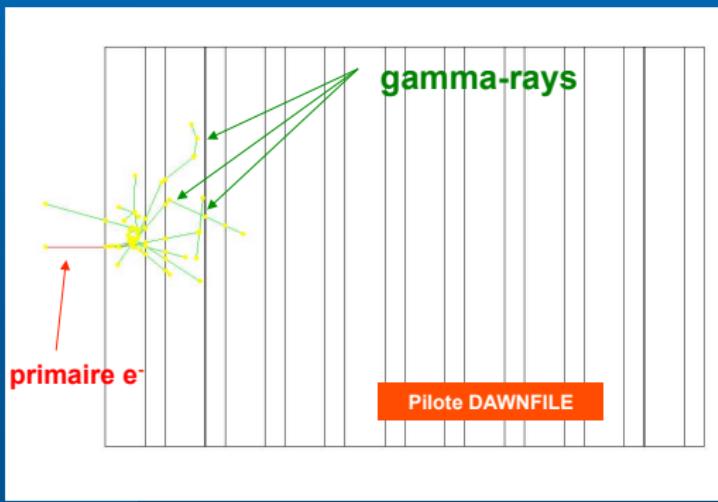
```
mean Energy in Absorber : 40.0567 MeV +- 0 eV
mean Energy in Gap      : 5.52904 MeV +- 0 eV
mean trackLength in Absorber : 2.9093 cm +- 0 fm
mean trackLength in Gap    : 2.63017 cm +- 0 fm
-----
```



Résumé du
run

Exécuter votre code sans macros

On visualise aussi l'événement tiré
(électron de 50 MeV)



Code couleur
par défaut
dans Geant4:
rouge = négative
bleu = positive
vert = neutre

Exécuter votre code AVEC macros

L'argument qui suit le nom de l'exécutable peut être un fichier **macro**, par ex. run1.mac

```
./exampleN03 run1.mac
```

Les macros Geant4 sont des **fichiers ASCII** contentant une séquence de **commandes Geant4**:

```
#  
/run/verbose 2  
/event/verbose 0  
/tracking/verbose 1  
#  
/gun/particle mu+  
/gun/energy 300 MeV  
/run/beamOn 3
```

} **Tire 3 μ^+ d'énergie 300 MeV**

Exécuter votre code AVEC macros

```
---> End of event: 2
    Absorber: total energy: 123.976 MeV
    Gap: total energy: 10.7747 MeV
```

Run terminated.

Run Summary

```
Number of events processed : 3
User=0.01s Real=0.02s Sys=0.01s
```

-----End of Run-----

```
mean Energy in Absorber : 125.191 MeV +- 2.67303 MeV
mean Energy in Gap      : 9.94337 MeV +- 596.048 keV
```

```
mean trackLength in Absorber : 10.5231 cm +- 1.14618 mm
mean trackLength in Gap      : 5.09232 cm +- 524.76 um
```

```
total track length: 10.44 cm
total track length: 5.16631 cm
```



Résumé du
deuxième
événement



Résumé du run
entier

Exécuter votre code AVEC macros

Noter que

`./exampleN03 run1.mac`

équivaut à

`./exampleN03`

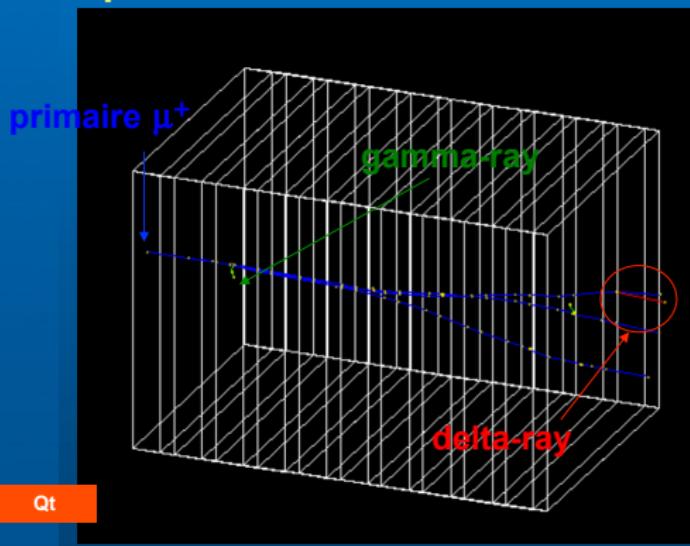
et dans la fenêtre “Session”:

`/control/execute run1.mac`

Commande pour exécuter les commandes contenues
dans le fichier macro run1.mac

Exécuter votre code AVEC macros

Capture d'écran des trois événements



17

Changer la géométrie

```
/control/execute newgeom.mac  
/control/execute run1.mac
```

1) La première macro change la géométrie

- seulement une tranche d'absorbeur (40 cm d'eau), pas d'espace (thickness = 0 cm)
→ un bloc solide d'eau

- change les dimensions transverses,

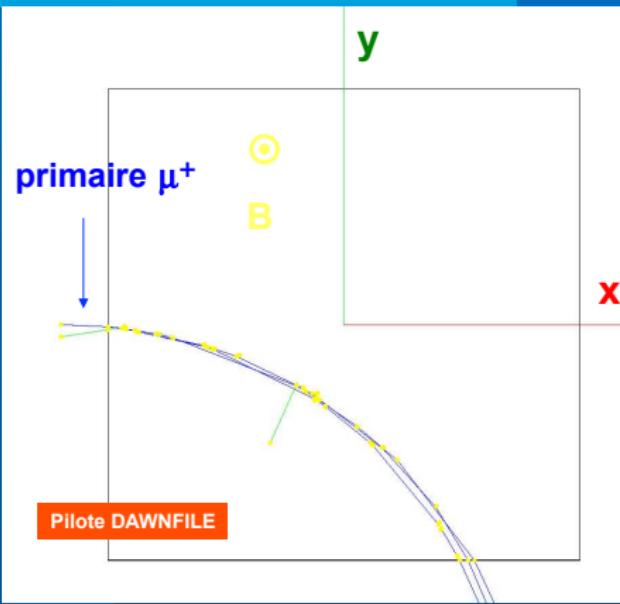
- ajoute un champ magnétique de 3T

2) La seconde macro tire 3 μ^+ de 300 MeV, comme avant

```
/N03/det/setNbOfLayers 1  
/N03/det/setAbsMat Water  
/N03/det/setAbsThick 40 cm  
/N03/det/setGapMat Air  
/N03/det/setGapThick 0 cm  
/N03/det/setSizeYZ 40 cm  
/N03/det/setField 3 tesla  
/N03/det/update
```



Changer la géométrie

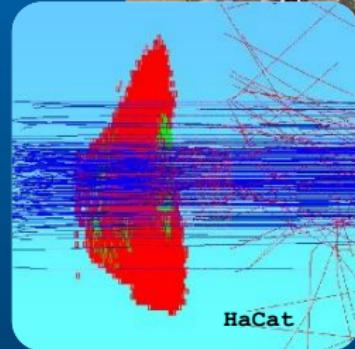


Géometrie,
matériaux, champ
magnétique et
particules primaires
peuvent être
sélectionnés dans
des macros ASCII,
sans recompiler le
code !

Autres exemples...

Microfaisceau pour l'irradiation cellulaire

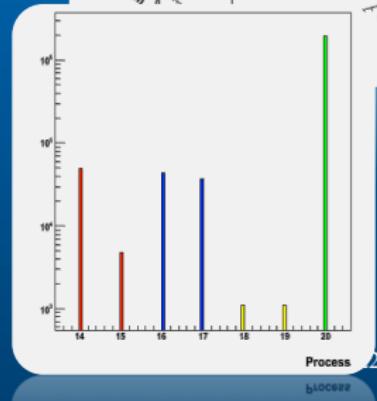
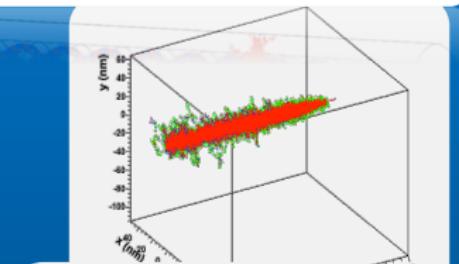
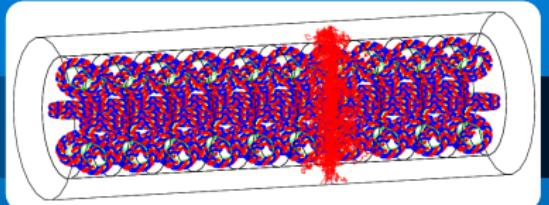
- Simuler la ligne d'irradiation cellulaire par microfaisceau du CENBG, incluant les aimants de focalisation
- Résolution faisceau à l'air ~1 µm
- Fantôme cellulaire obtenu à partir de microscopie confocale et d'analyse par faisceau d'ions (PIXE, STIM) pour la composition chimique
- Dosimétrie à l'échelle sub-cellulaire (noyau, cytoplasme)
- Utilise les constructeurs de physique



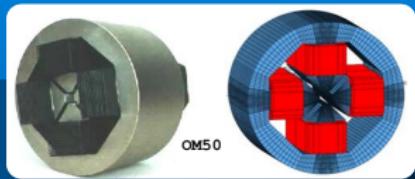
21

Dnaphysics

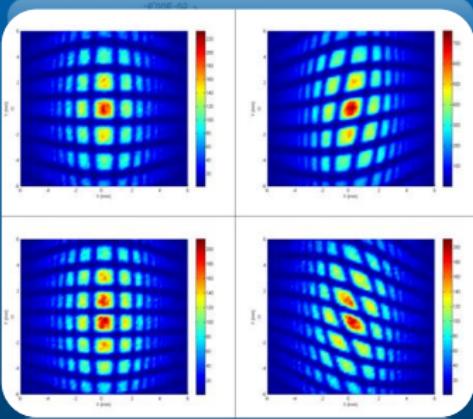
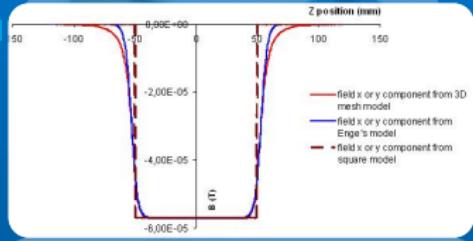
- Explique comment utiliser l'extension Geant4-DNA
 - Ensemble de modèles de Physique jusqu'au **sub-eV** pour la simulation des traces dans l'eau liquide à l'échelle du nanomètre
- Développé dans le cadre du projet Geant4-DNA pour la **simulation des effets biologiques des radiations**



Nanobeam



- Simulation de la **ligne nanofaisceau du CENBG**
 - Résolution du faisceau sous vide de quelques dizaines de nm
 - Nano-technologies
 - Physique EM standard
- Description fine des **quadrupôles**
 - Champs analytiques avec/sans franges
 - Carte de champ interpolée (OPERA3D)
- Geant4 peut calculer
 - Images de spot faisceau
 - Images de grilles
 - Coefficients d'aberration intrinsèques



En résumé

- Nombreux exemples disponibles dans Geant4 pour l'apprentissage et l'illustration d'utilisation dans divers domaines
 - Novices, étendus, avancés
- Localisés dans **\$G4SRC**

Mises à jour de Geant4



15 minutes

Vérifier régulièrement les annonces de mise à jour (MAJ) sur le site web de Geant4

The screenshot shows the Geant4 website homepage. A red arrow points from the top text to the 'News' section on the right.

Geant4 is a toolkit for the simulation of the passage of particles through matter. Its areas of application include high energy, nuclear and accelerator physics, as well as studies in medical and space science. The two main reference papers for Geant4 are published in *Nuclear Instruments and Methods in Physics Research A* 506 (2003) 250-303, and *IEEE Transactions on Nuclear Science* 53 No. 1 (2006) 270-278.

Applications A sampling of applications, technology transfer and other uses of Geant4

User Support Getting started, guides and information for users and developers

Results & Publications Validation of Geant4 results from experiments and publications

Collaboration Who we are: collaborating institutions, members, organization and legal information

News

- 22 April 2010 - Patch-01 to release 9.3 is available from the [download area](#).
- 16 March 2010 - [2010 planned developments](#).
- 19 February 2010 - Patch-03 to release 9.2 is available from the [archive download area](#).

MAJ majeures et mineures

- Les numéros de MAJ sont de la forme X.Y
 - X est le numéro de version **majeure**
 - Y est le numéro de version **mineure**
 - Par ex. Geant4 **9.6** est une mise à jour **mineure** (Décembre 2012)
 - **10.0** sera une mise à jour **majeure**
- Les MAJ mineures n'ont pas le droit de “casser” un code utilisateur
 - Par exemple, en migrant de la MAJ 9.4 vers 9.5,
il n'y aura pas besoin d'adapter vos codes
 - Vous obtiendrez probablement des résultats légèrement différents
 - Et vous pourrez aussi bénéficier de nouveaux développements
 - Mais rien ne doit échouer dans votre code à cause de la migration
- Par contre, les MAJ majeures peuvent casser le code utilisateur
 - En passant de 9.5 à 10, il faudra adapter votre code
 - Les **release notes** vous diront quoi changer.

Les patch

- Un **patch** permet de corriger des bugs dans une MAJ
 - La MAJ se termine alors avec **p01, p02**, etc.
 - Comme dans **Geant4.9.6.p02**
- Ne contient que des corrections
 - Rien de nouveau
- On vous conseille de toujours télécharger le dernier patch
 - Les **release notes** du patch vous diront ce qui a changé, ainsi vous pourrez décider si vous avez besoin d'installer le patch ou non

Quels sont les nouveaux développements à venir ?

- Regarder « planned developments » dans la section News

The screenshot shows the Geant4 website's "Planned features" page. At the top, there is a navigation bar with links for "Download", "User Forum", "Gitter", "Contact Us", and a search bar labeled "Search Geant4". Below the navigation bar, the page title is "Geant 4" and the current location is "Home > User Support > Planned features".

Planned developments for 2010

Items in this list are related to new developments scheduled for the current year. Improvements, fixes, studies and maintenance items are not mentioned here as part of routine activity.

NOTE: items marked with (*) may or may not be achieved in 2010.

Geometry

- Review of navigation verbosity & control at step number - (1)
- Implementation of precise computeSafety() in navigation for EM use - (1)
- Finalisation of interoperability of multiple navigators/geometries - (1)/(2)
- Extension of parameterisation to cylinders for regular navigation - (2)
- New arbitrary trapezoid shape with vertices on parallel planes perpendicular to the z axis - (2)
- Extension to divisions to allow for gaps in replicated daughters - (2)
- Review classes exposed to kernel and thread-safety - (2)

Hadronic Physics

- Complete interface to ENDF high precision neutron database for low energy neutron models - (1)
- Development of interface between INCL and evaporation models - (1)
- Development of capability for incident carbon ions in INCL/ABLA - (1)
- First implementation of code review actions (A2Z types, code cleanup, etc...) - (1)
- Implementation of fast neutron capture model - (2)
- Improved break-up method in de-excitation - (2)
- Complete interface of era compound to Ravelin cascade - (2)

Les release notes

- Chaque MAJ est accompagnée de **release notes**
- Les lire avant d'installer la MAJ !
- Lire en particulier
 - Plates-formes supportées, compilateur
 - Version de CLHEP
 - Comment migrer votre code
 - Effets attendus sur la physique et les performances

The screenshot shows the 'Geant4 Software Download' page. At the top, it says 'Geant4 Software Download' and 'Released 22 April 2010 (sp1-h-01)'. Below that, a note reads: 'The Geant4 source code is freely available. See the license conditions.' It also says 'Please read the Release Notes before downloading or using this release.' A red arrow points from the 'Release Notes' link in the slide to this section of the screenshot.

Geant4 9.3 Release Notes

December 1st, 2009

The code and binary libraries for the supported systems are available through our [Source Code Web page](#).

We are grateful for the efforts of Geant4 users who have provided detailed feedback or comprehensive reports of issues. We thank in particular those who have contributed corrections, improvements or developments included in this release.

Please refer to the [Geant4 User Documentation](#) for further information about using Geant4.

Contents

1. [Supported and Tested Platforms](#)
2. [CLHEP and AIDA](#)
3. [Items for migration of the user code](#)
4. [New Developments and Capabilities](#)
5. [Expected effects on physics and performance](#)
6. [Known Run-Time Problems and Limitations](#)
7. [Compilation Warnings](#)
8. [Known Run-Time Warnings](#)
9. [Geant4 Software License](#)
10. [Detailed list of changes and fixes](#)

1) Plates-formes et compilateurs

- Peuvent changer lors d'une MAJ mineure ou majeure
- Si votre plate-forme et votre compilateur ne sont plus supportés, alors vous pouvez les mettre à jour ou essayer avec ce que vous avez:
 - Si Geant4 compile et démarre, pas de souci
 - Sinon mettre à jour votre plate-forme et compilateur

1. Supported and Tested Platforms

Official platforms:

- Linux, gcc-3.4.6 (SLC4), gcc-4.2.1 (SLC5).
Tested on 32 and 64 bit architectures (Intel or AMD) with Scientific Linux CERN 4.0 (SLC5) (based on RedHat Linux Enterprise 5). Geant4 has also been successfully compiled on Mac OSX 10.5, gcc-4.0.1.
- Windows/XP and CygWin Tools with: Visual C++ 9.0 (Visual Studio 2008)

More verified configurations:

- Linux, gcc-4.3.2
- Linux, Intel-icc 11.0

Platforms configured but neither tested nor supported:

- AIX 4.3.2, xlC 6.0
- DEC V4.0, cxx C++ V6.1-027
- HP 10.20, aCC C++ B3910B A.01.23
- SGI V6.5.5, CC 7.2.1
- SUN Solaris 5.8, C++ CC-5.5.

2) CLHEP

- Computing Library for High Energy Physics
- Peut changer lors d'une MAJ mineure ou majeure
- Ne change pas souvent, souvent pour une bonne raison...
- Si vous n'utilisez pas la bonne version, votre code va lier mais les résultats peuvent être faux !
- Une version allégée est intégrée par défaut dans Geant4

2. CLHEP and AIDA

Geant4 requires the installation of [CLHEP](#), release **2.0.4.2**.

The installation of **CLHEP-2.0.4.2** is mandatory for consistency in the definition of masses and widths of particles. Earlier versions of CLHEP may link, but they will not give correct results.

3) Instructions pour migrer votre code

3. Items for migration of the user code

Listed here is some relevant information on developments included in this release, some of which may require migrations (mainly for users of advanced Geant4 features) in order to upgrade from release 9.1 to release 9.2. Note that for all users a full re-installation of libraries (or a full recompilation) and a recompilation of user applications is required.

- Pas grand chose pour une MAJ mineure
- Lire avec attention pour une MAJ majeure

3. Items for migration of the user code

Listed here is some relevant information on developments included in this release, some of which may require migrations (mainly for advanced uses of features in Geant4) in order to upgrade from release 8.3 to release 9.0. Note that a full re-installation of libraries (or a full recompilation) and a recompilation of user applications is anyhow required.

Geometry

The static constants for the geometrical tolerance `kCartTolerance`, `kRadTolerance` and `kAngTolerance` which used to be defined at global scope, have been now removed.

Advanced applications making use of such constants in the code are required to retrieve the values for Cartesian, Angular and Radial tolerances through the class `G4GeometryTolerance`, a new class providing the methods, respectively: `GetSurfaceTolerance()`, `GetRadialTolerance()` and `GetAngularTolerance()`.

Applications with setups of unusual dimensions (e.g. smaller than 1 mm or larger than 1 km) can now adjust these values, and must do so before creating any part of the geometry description. Please see the User's Guide for Application Developers for the details.

4) Effets attendus sur la physique et les performances

- Cette section vous dit quoi attendre en terme de modification des résultats, du temps d'exécution et d'utilisation mémoire

5. Expected effects on physics and performance

Geometry

- Implementation improvements in G4Cons and G4Tubs bring speed improvements on the order of 20% for those shapes with pure-tracking (i.e. without field propagation or physics).

Low-energy Electromagnetic physics

- An updated version of the G4LogLogInterpolation class is included in this release. This update improves the speed of low energy EM Physics processes that use this class. Only insignificant changes result in the interpolated values obtained and in physics observables.

Standard Electromagnetic physics

- More precise multiple scattering model for electrons and positrons provides wider shower (about 0.5% measured for the CMS calorimeter).
- New relativistic Bremsstrahlung model is applied for electrons and positrons; it provides gamma spectra which agree with thin-target CERN and SLAC data.
- The usage of the Spline approximation of dedx and cross section tables provides a Bragg peak position which is stable within 0.1 mm versus variation of production cut or step limit.
- Visible energy in sampling calorimeters may increase on the order of 1% due to upgraded model of multiple-scattering and usage of Spline approximation.

Hadronic physics

- Changes to the Bertini cascade code have resulted in several changes to energy spectra and shower shapes:
 - the bug fix to multiplicity sampling resulted in cutting the quasi-elastic peak height in half, more in line with the binary cascade;

Comment mettre à jour ?

- Vérifier les release notes pour voir si vos versions de plate-forme et de compilateur sont les bonnes
 - En général pas nécessaire pour les MAJ mineures
 - MAJ au besoin
- Suivre les instructions d'installation de Geant4 pour votre système depuis le site web de Geant4
 - Utilise le système Cmake pour compiler et installer
 - Voir: Installation Guide: For setting up Geant4 in your computing environment
- Ou utiliser notre machine virtuelle Geant4
 - Windows, Mac, Linux
 - Gratuite, pas d'installation et mise à jour régulièrement

<http://geant4.in2p3.fr>



En cas de difficultés

- Avez-vous lu les **release notes** ?
- Allez voir le **forum utilisateur, catégorie installation**
 - <http://geant4-hn.slac.stanford.edu:5090/Geant4-HyperNews/index>
 - Utiliser la fonction **search**
 - Un utilisateur a peut-être eu le même problème et la solution peut y être donnée
 - Si le problème est nouveau, postez votre problème dans le forum

En résumé

- MAJ régulières de Geant4
 - Patch
 - Mineures
 - Majeures
 - Consulter régulièrement le site web de Geant4
- Même procédure d'installation que pour une première installation
- Les **release notes** sont le document de référence pour vous aider à migrer votre code
- Si difficultés, aller voir dans le **user forum**...
 - Catégorie **installation**

L'outil d'analyse de données ROOT



15 minutes

ROOT

An Object-Oriented
Data Analysis Framework

1

Présentation

- Le système ROOT fournit un environnement OO avec toutes les fonctionnalités nécessaires pour analyser un grand nombre de données de manière très efficace.
- Standard d'analyse de données développé au CERN pour les expériences du LHC
- Les données sont définies comme des ensembles d'objets. Des méthodes spécialisées de stockage permettent d'accéder directement à des attributs différents d'objets sélectionnées.
- En particulier :
 - Création d'histogrammes à 1, 2 ou 3 dimensions, de tuples
 - Ajustement de courbes
 - Calcul de fonctions
 - Minimisation
 - Classes de visualisation graphique
 - Traitement batch
 - ...
- Possède un interpréteur C++ (CINT) : tout se fait en C++ (fichiers macro ou programmation). L'interpréteur permet de développer rapidement des macros car il supprime la phase de compilation/lien.
- Et le tout gratuitement... Nous l'avons installé pour vous sous Scientific Linux 5.7



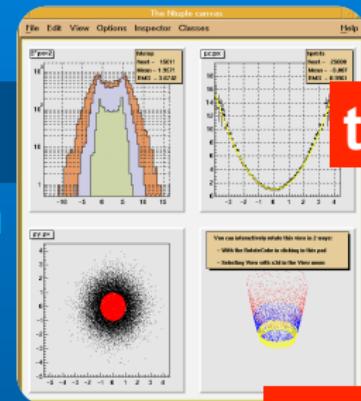
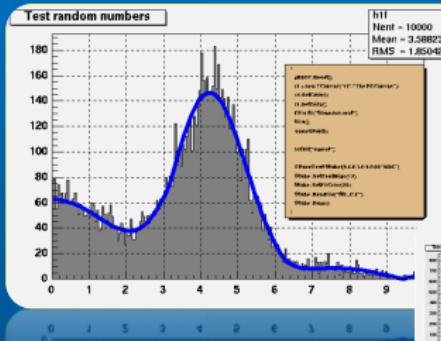
2

Qu'allons-nous faire avec ROOT ?

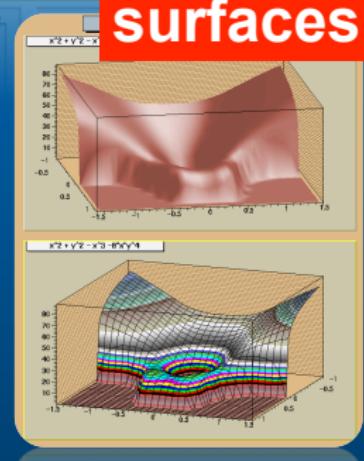
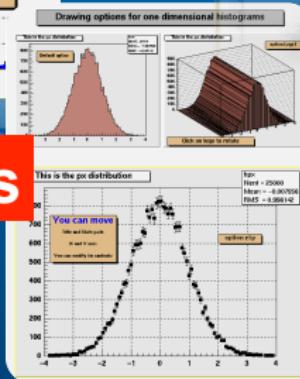
- Votre simulation produit de nombreux résultats : ex : dépôts d'énergie à chaque step, de dose, position, etc...
- Par simplicité, ces résultats sont stockés par votre simulation sous forme de fichiers texte
- ROOT va relire ces fichiers pour créer des histogrammes et tuples pour analyser les résultats de la simulation

Exemples

tuples



surfaces



histogrammes

Palette de couleurs



Utiliser ROOT

- Sous un terminal

`root`

- Exécuter un fichier macro, sous une session ROOT

`.x plot.C`

- Quitter ROOT

`.q ou exit`

Essai

- Editer avec **nedit** un fichier ROOT macro **plot.C**
- A partir de nos résultats de simulation, écrivons un code C++ permettant de
 - Créer un **histogramme 1D** de dose absorbée
 - L'**ajuster** avec une fonction de Gauss
 - Créer un **ntuple**
 - Afficher plusieurs combinaisons de variables

Exemple : histogramme et ntuple

```
{  
    gROOT->Reset();  
    gStyle->SetPalette(1);  
  
    TFile f("result.root");  
  
    TH1F* h1;  
    h1 = (TH1F*)f->Get("dose");  
  
    TNtuple* ntuple;  
    ntuple = (TNtuple*)f->Get("step");  
  
    c1 = new TCanvas ("c1","",20,20,1000,500);  
    c1.Divide(2,1);  
  
    c1.cd(1);  
  
    h1->SetLineColor(3);  
    h1->SetFillColor(3);  
    h1->SetFillStyle(1001);  
    h1->Draw("HIST");  
    h1->GetXaxis()->SetTitle("Absorbed dose (Gy)");  
    h1->GetYaxis()->SetTitle("Events");  
    //h1->Fit("gaus");  
    gPad->SetLogy();  
  
    c1.cd(2);  
  
    ntuple->Draw("x:y","","surf3");  
    gPad->SetLogz();  
  
}
```

Remise à zéro de ROOT
Choix de la palette de couleurs

Ouverture du fichier ROOT
Récupération de l'histogramme
et du ntuple

Division de la page graphique en 2

Dessin de l'histogramme

Dessin du ntuple et fit

Quelques remarques

- De nombreux objets sont directement accessibles par défaut comme :
 - `gROOT` : la session ROOT en cours
 - `gStyle` : le style graphique en cours
 - `gPad` : la fenêtre graphique en cours
- ROOT possède ses propres types : `Int_t`, `Float_t` : ce sont les types du C++ auxquels on enlève la majuscule et on ajoute `_t`

Ouvrir / sauvegarder un fichier ROOT

- Ouvrir un fichier ROOT déjà créé

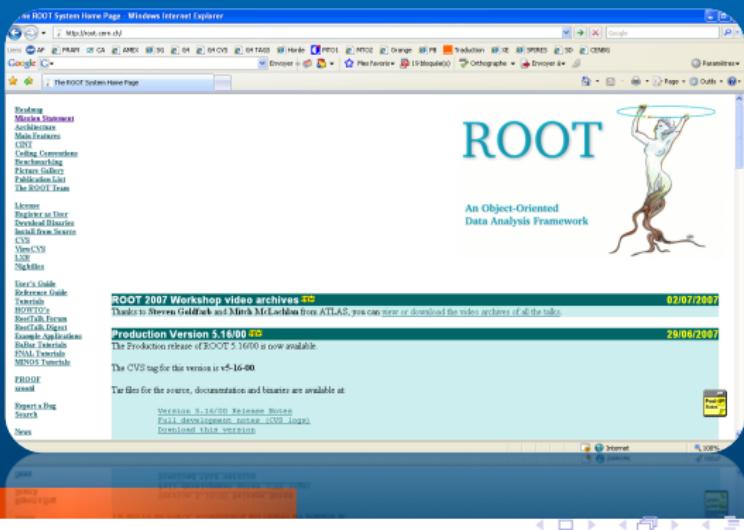
```
Tfile f("hsimple.root");
```

- Sauver dans un fichier ROOT

```
TFile *f = new TFile("basic.root","RECREATE");
...
f->Write();
```

Pour en savoir plus

http://root.cern.ch



11