Méthode de Génération des Événements

Source Europium-152 dans Geant4

Simulation Monte Carlo – Puits Couronne

Documentation Technique

January 13, 2026

1 Introduction

Ce document décrit la méthode de génération des particules primaires pour la simulation Monte Carlo d'une source d'Europium-152 (^{152}Eu) dans le cadre du projet "Puits Couronne" avec Geant4.

1.1 Schéma de désintégration de l'Eu-152

L'^{152}Eu est un radionucléide à double mode de décroissance :

- **Capture électronique (EC)** : 72.1% $\rightarrow$ ^{152}Sm
- **Désintégration** β^- : 27.9% $\rightarrow$ ^{152}Gd

Ces deux voies de décroissance mènent à des états excités des noyaux fils, qui se désexcitent par émission de rayonnements gamma. L'^{152}Eu est ainsi caractérisé par un spectre gamma riche avec de nombreuses raies d'intensités variées.

2 Philosophie de la Simulation

2.1 Définition d'un événement Geant4

Dans notre simulation :

$$\boxed{\textbf{1 événement Geant4} = \textbf{1 désintégration } ^{152}\textbf{Eu}}$$

Chaque désintégration peut émettre **plusieurs photons gamma** de manière indépendante. Le nombre de photons émis par événement suit une distribution statistique déterminée par les intensités des raies.

2.2 Relation avec l'activité

Si la source possède une activité $A_{4\pi}$ (en Bq), alors :

$$\text{Nombre de désintégrations par seconde} = A_{4\pi} \tag{1}$$

La correspondance entre le nombre d'événements simulés N_{evt} et le temps d'irradiation t_{irr} est :

$$t_{\text{irr}} = \frac{N_{\text{evt}}}{A_{4\pi} \times f_\Omega} \tag{2}$$

où f_Ω est la fraction d'angle solide couverte par le cône d'émission.

3 Spectre Gamma Eu-152 Implémenté

3.1 Raies sélectionnées

Le spectre implémenté comprend **13 raies** (2 raies X + 11 raies gamma) avec une intensité totale de 202.78% :

Table 1: Spectre gamma Eu-152 implémenté (Source: NNDC/ENSDF)

Index	Énergie (keV)	Intensité (%)	Type
0	39.52	20.80	X (K_α)
1	40.12	37.70	X (K_β)
2	121.78	28.41	γ
3	244.70	7.53	γ
4	344.28	26.59	γ
5	411.12	2.24	γ
6	443.97	2.83	γ
7	778.90	12.97	γ
8	867.38	4.24	γ
9	964.08	14.63	γ
10	1085.87	10.21	γ
11	1112.07	13.64	γ
12	1408.01	21.01	γ
	Total :	**202.78**	

3.2 Nombre moyen de photons par désintégration

Le nombre moyen de photons émis par désintégration est :

$$\boxed{\langle n_\gamma \rangle = \sum_{i=0}^{12} p_i = \frac{202.78}{100} \approx 2.03} \tag{3}$$

Cette valeur est utilisée dans le code via la méthode `GetMeanGammasPerDecay()`.

4 Algorithme de Génération

4.1 Principe : Tirages de Bernoulli indépendants

Pour chaque événement (désintégration), l'algorithme procède comme suit :

1. Pour chaque raie $i \in \{0, 1, ..., 12\}$:
   - Tirer un nombre aléatoire $r \sim \mathcal{U}(0, 1)$
   - Si $r < p_i$ (où $p_i = I_i/100$), alors la raie i est émise
2. Pour chaque raie émise :
   - Générer une direction dans le cône d'émission
   - Créer un vertex primaire avec l'énergie E_i

4.2 Pseudo-code

Listing 1: Algorithme de génération (simplifié)

```
void GeneratePrimaries(G4Event* event) {
    for (int i = 0; i < 13; i++) {
        double r = G4UniformRand();  // r in [0,1)

        if (r < probability[i]) {
            // Cette raie est emise
            double energy = gammaEnergies[i] * keV;
            G4ThreeVector direction = GenerateDirectionInCone();

            particleGun->SetParticleEnergy(energy);
            particleGun->SetParticleMomentumDirection(direction);
            particleGun->GeneratePrimaryVertex(event);
        }
    }
}
```

4.3 Distribution du nombre de photons par événement

Le nombre de photons n émis par événement suit une loi de **Poisson composée**. En pratique, c'est la somme de 13 variables de Bernoulli indépendantes :

$$n = \sum_{i=0}^{12} X_i, \quad \text{où } X_i \sim \text{Bernoulli}(p_i) \tag{4}$$

Les caractéristiques de cette distribution sont :

$$\mathbb{E}[n] = \sum_{i=0}^{12} p_i = 2.03 \tag{5}$$

$$\text{Var}(n) = \sum_{i=0}^{12} p_i(1-p_i) \approx 1.45 \tag{6}$$

La probabilité d'avoir un événement "vide" (aucun photon émis) est :

$$P(n=0) = \prod_{i=0}^{12}(1-p_i) \approx 0.4\% \tag{7}$$

Cette probabilité est très faible car l'intensité totale est élevée (203%).

5 Génération de la Direction dans le Cône

5.1 Géométrie du cône d'émission

La source émet dans un cône de demi-angle $\theta_{\text{max}} = 45$ orienté selon l'axe $+z$:

$$\begin{aligned}
&\text{Axe du cône : } \vec{u}_z = (0,0,1) \\
&\text{Demi-angle : } \theta_{\text{max}} = 45 \\
&\text{Fraction d'angle solide : } f_\Omega = \frac{1-\cos\theta_{\text{max}}}{2} \approx 14.6\%
\end{aligned}$$

5.2 Distribution uniforme sur la calotte sphérique

Pour obtenir une distribution uniforme sur la surface de la calotte sphérique, on utilise :

$$\cos\theta = 1 - u \cdot (1 - \cos\theta_{\max}), \quad u \sim \mathcal{U}(0, 1) \tag{8}$$

$$\phi = 2\pi v, \quad v \sim \mathcal{U}(0, 1) \tag{9}$$

La direction est alors :

$$\vec{d} = (\sin\theta\cos\phi, \sin\theta\sin\phi, \cos\theta) \tag{10}$$

5.3 Code de génération

Listing 2: Génération de direction dans le cône

```
void GenerateDirectionInCone(G4double coneAngle,
                             G4ThreeVector& direction) {
    // cos(theta) uniforme entre cos(coneAngle) et 1
    G4double cosTheta = 1.0 - G4UniformRand() * (1.0 - cos(coneAngle));
    G4double theta = acos(cosTheta);

    // phi uniforme entre 0 et 2*pi
    G4double phi = G4UniformRand() * 2.0 * CLHEP::pi;

    // Direction cartesienne
    G4double sinTheta = sin(theta);
    direction.set(sinTheta * cos(phi),
                  sinTheta * sin(phi),
                  cosTheta);
}
```

6 Normalisation et Calcul de Dose

6.1 Paramètres de la simulation

Table 2: Paramètres de normalisation

Paramètre	Symbole	Valeur
Activité source (4π)	$A_{4\pi}$	42 kBq
Demi-angle du cône	$\theta_{\max}$	45°
Fraction d'angle solide	f_Ω	14.6%
Activité effective (cône)	A_{eff}	6.15 kBq
Photons moyens/désintégration	$\langle n_\gamma \rangle$	2.03
Position source	z_s	73.5 mm

6.2 Temps d'irradiation équivalent

Pour N_{evt} événements simulés, le temps d'irradiation équivalent est :

$$\boxed{t_{\text{irr}} = \frac{N_{\text{evt}}}{A_{4\pi} \times f_\Omega} = \frac{N_{\text{evt}}}{42000 \times 0.146}\ \text{s} = \frac{N_{\text{evt}}}{6132}\ \text{s}} \tag{11}$$

6.3 Calcul du débit de dose

Si D_{sim} est la dose totale simulée (en Gy) pour N_{evt} événements :

$$\boxed{\dot{D} = \frac{D_{\text{sim}}}{t_{\text{irr}}} = \frac{D_{\text{sim}} \times A_{4\pi} \times f_\Omega}{N_{\text{evt}}}} \tag{12}$$

En nGy/s :

$$\dot{D}\ (\text{nGy/s}) = \frac{D_{\text{sim}}\ (\text{nGy})}{N_{\text{evt}}} \times 6132 \tag{13}$$

7 Résumé des Constantes dans le Code

Table 3: Correspondance code/physique

Variable C++	**Valeur**	**Signification**
`GetMeanGammasPerDecay()`	2.03	$\langle n_\gamma \rangle$
`fMeanGammasPerDecay`	2.03	Idem (RunAction)
`fActivity4pi`	4.2×10^4	$A_{4\pi}$ en Bq
`fConeAngle`	45°	θ_{max}
`GetSolidAngleFraction()`	0.146	f_Ω
`kNbGammaLines`	13	Nombre de raies

8 Validation

8.1 Vérifications recommandées

1. **Nombre moyen de primaires** : Vérifier que $\langle n_\gamma \rangle \approx 2.03$ sur un grand nombre d'événements
2. **Distribution angulaire** : Vérifier que $\cos\theta$ est uniforme dans $[\cos 45, 1]$
3. **Spectre en énergie** : Vérifier que les intensités relatives correspondent aux valeurs du tableau
4. **Événements vides** : Vérifier que $\sim 0.4\%$ des événements n'émettent aucun photon

8.2 Formule de vérification

Après N événements, le nombre total de photons générés devrait être :

$$N_\gamma^{\text{total}} \approx N \times 2.03 \pm \sqrt{N \times 1.45} \tag{14}$$