

Ntuple compton_cone_events

Enregistrement des diffusions Compton individuelles
dans le cône graphite — Simulation Geant4 Mini X

February 11, 2026

1 Objectif

Le ntuple `compton_cone_events` enregistre *chaque* diffusion Compton subie par un photon primaire (`parentID=0`) dans le volume logique `logicConeCompton` (cône graphite, $z \in [1.9, 16.95]$ mm). Il permet de tracer :

- la carte (z, r) des interactions Compton dans le cône,
- l'histogramme 2D θ_{in} vs θ_{out} ,
- l'angle de diffusion θ_s en fonction de l'énergie incidente,
- la perte d'énergie ΔE vs E_{in} ,
- la vérification de la formule Compton $\Delta E = \frac{E_{\text{in}}^2 (1 - \cos \theta_s)}{m_e c^2 + E_{\text{in}} (1 - \cos \theta_s)}$.

2 Structure du ntuple

Le ntuple est créé dans `AnalysisManagerSetup.cc` et rempli dans `SteppingAction.cc` au sein du bloc de détection Compton (`procDefined->GetProcessName() == "compt"` dans `logicConeCompton`). Il contient 16 colonnes :

Col.	Nom	Type	Description
0	<code>eventID</code>	int	Identifiant de l'événement
1	<code>trackID</code>	int	Identifiant du track (photon primaire)
2	<code>n_compton_seq</code>	int	n -ième Compton de ce track dans le cône
3	<code>ekin_before_keV</code>	double	E_{in} : énergie cinétique avant (keV)
4	<code>ekin_after_keV</code>	double	E_{out} : énergie cinétique après (keV)
5	<code>delta_ekin_keV</code>	double	$\Delta E = E_{\text{in}} - E_{\text{out}}$ (keV)
6	<code>x_mm</code>	double	Position x de l'interaction (mm)
7	<code>y_mm</code>	double	Position y (mm)
8	<code>z_mm</code>	double	Position z (mm)
9	<code>r_mm</code>	double	$r = \sqrt{x^2 + y^2}$ (mm)
10	<code>theta_in_deg</code>	double	θ_{in} : angle polaire incident ($^\circ$, réf. $+Z$)
11	<code>phi_in_deg</code>	double	φ_{in} : angle azimutal incident ($^\circ$)
12	<code>theta_out_deg</code>	double	θ_{out} : angle polaire sortant ($^\circ$)
13	<code>phi_out_deg</code>	double	φ_{out} : angle azimutal sortant ($^\circ$)
14	<code>scatter_angle_deg</code>	double	$\theta_s = \arccos(\hat{d}_{\text{in}} \cdot \hat{d}_{\text{out}})$ ($^\circ$)
15	<code>cos_scatter</code>	double	$\cos \theta_s$

Table 1: Colonnes du ntuple `compton_cone_events`.

3 Notes d'implémentation

- Les directions incidente et sortante sont lues directement sur les `G4StepPoint` :
`prePoint->GetMomentumDirection()` (\hat{d}_{in}) et `postPoint->GetMomentumDirection()` (\hat{d}_{out}).

- Dans Geant4, lors d'une diffusion Compton le photon diffusé *continue comme le même track* (même `trackID`, `parentID`=0). Seul l'électron de recul est créé comme secondaire. Le champ `n_compton_seq` permet de distinguer les diffusions multiples d'un même photon.
- Le marquage `MyTrackInfo` existant (flag `fComptonInCone`, compteur `fNComptonInCone`, dernière position/énergie) est conservé pour le ntuple `plane_passages` au plan $z = 18$ mm.

4 Fichiers modifiés

Fichier	Modification
<code>include/AnalysisManagerSetup.hh</code>	Déclaration de <code>GetComptonConeNtupleId()</code>
<code>src/AnalysisManagerSetup.cc</code>	Création du ntuple (16 colonnes) + getter
<code>src/SteppingAction.cc</code>	Remplissage à chaque Compton dans <code>logicConeCompton</code>
<code>analyse_compton_cone.C</code>	Macro ROOT d'analyse (9 figures)

5 Exploitation avec ROOT

Après recompilation et exécution du run :

```
root -l -b -q 'analyse_compton_cone.C("output.root")'
```

Exemples de commandes interactives :

```
// Carte (z, r) des Compton
compton_cone_events->Draw("r_mm:z_mm","", "COLZ");

// theta_in vs theta_out
compton_cone_events->Draw("theta_out_deg:theta_in_deg","", "COLZ");

// DeltaE vs E_incident
compton_cone_events->Draw("delta_ekin_keV:ekin_before_keV","", "COLZ");

// Perte relative
compton_cone_events->Draw(
    "100*delta_ekin_keV/ekin_before_keV:ekin_before_keV",
    "ekin_before_keV>0", "COLZ");
```