

その他のスコアリング方法

1) Command-based scoring

Run を通して積算した物理量を取得したい場合に手取り早いのがコマンドラインスコアラを用いる方法です。BOX 形状の領域を設定して、それをメッシュに分割して、各セルにおける物理量を積算してファイルに出力できます。

詳細 [Command-based scoring \(Geant4 documentation\)](#)

Command-based scoring の利用準備の確認

メインプログラムにて、G4ScoringManager を有効化する必要があります。Galet では、すでに有効化した状態となっています。

Galet.cc

1	...
2	#include "G4ScoringManager.hh"
	...
	// Scoring Manager
	G4ScoringManager* scManager = G4ScoringManager::GetScoringManager();
	scManager->SetVerboseLevel(1);
	...

例) run1.mac

1	#
	# Initialize
	/control/execute init.mac
	#
	/gps/particle e-
	/gps/position 0 0 0 m
	/gps/direction 0 0 -1.
	/gps/energy 2 MeV
	#
	#
	/score/create/boxMesh boxMesh_1
	/score/mesh/boxSize 100. 100. 100. cm
	/score/mesh/nBin 30 30 30
	/score/mesh/translate/xyz 0. 0. -10. cm
	/score/mesh/rotate/rotateX 0. degree

	<pre> /score/mesh/rotate/rotateY 0. degree /score/mesh/rotate/rotateZ 0. degree /score/quantity/energyDeposit eDep # /score/close # /score/list # # /run/beamOn 1000 # /score/dumpQuantityToFile boxMesh_1 eDep eDep.txt # </pre>
--	--

例) \$ cat eDep.txt

1	<pre> # mesh name: boxMesh_1 # primitive scorer name: eDep # iX, iY, iZ, total(value) [MeV], total(val^2), entry 0,0,0,0,0,0 0,0,1,0,0,0 0,0,2,0,0,0 0,0,3,0,0,0 0,0,4,0,0,0 </pre>
---	---

2) SteppingAction を利用したスコアリング

シミュレーションの粒子移送の各ステップで情報を収集します。計算効率は非常に悪くなります。ワールドボリューム全体で情報を取得した場合には、この方法もありですし、コーディングを工夫すれば計算効率を上げることはできます（熟練度が必要です）。

（補足）Geant4 でユーザコードを挟むためのクラスとその呼出のタイミング

- RunAction Run をスタートするときと、終了するときに実行されます。
- EventAction Event をスタートするとき、終了する時に実行されます。
- StackingAction シミュレーションされる粒子は2次粒子を含めて、一旦スタックに蓄えられます。スタックから粒子を取り出して輸送を開始する際に、粒子の優先順位などを変更

することができます。

- ・ `TrackingAction` スタックから取り出された粒子を輸送するスタートのとき、終了する時に実行されます。

- ・ `SteppingAction` 粒子の輸送中の各ステップで実行されます。

(補足) 粒子情報へのアクセスは `G4Step` インスタンスから

輸送されている粒子の情報は、`G4Step` インスタンスに格納されています。

ステップの両端の情報として、`PreStepPoint`, `PostStepPoint` があります。ステップ間の情報は、`PreStepPoint` に保存されています。

`Galet` の通常モードでは実行されないようになっていますが、粒子情報に `SteppingAction` でアクセスするコード例を含めています。コードを参考に開発をしてみてください。

\$ code `src/GaletSteppingAction.cc`

`UserSteppingAction(G4Step step)` メソッドが、各ステップで実行されるので、メソッドの引数で与えられた `G4Step` インスタンスから情報を取り出してファイルなどに保存することになります。

*但し、この方法だと、イベント毎に情報をまとめたいとか、あるジオメトリ領域のみを選択したいとか、追加の処理を考えていくと複雑になるのでコーディングが大変になりやすいという欠点があります。