

メッシュに関するまとめ

- ・ 1 つの ParallelWorld に対して 1 つのスコアジオメトリが対応する。
- ・ ユーザに ParallelWorld の存在を意識させない。
- ・ メッシュ名と ParallelWorld 名は、1 対 1 対応
- ・ メッシュ名を指定すると、どの ParallelWorld かが限定される。
- ・ 基本的に次にあげる分割方法をとるが、さしあたり、**太字**の項目を優先する。

形状	分割軸	Replica (depth=2)	Replica (depth=1)	Parameterised (depth=0)
Box	XYZ	X	Y	Z
Tubs	R Z	R		Z
Tubs	ZR		Z	R
Sphere	R			

I) コマンド構造

```

/score/mesh/open  <mesh 名>
/score/filter/open  <filter 名>
/score/filter/close
/score/quantity/open  <スコアラ名>
/score/filter/open  <filter 名>
/score/filter/close
/score/quantity/close
/score/mesh/close

```

open-close で固有名称の設定と、その指定する範囲設定を行う。

mesh 名は、MultiFunctionalDetector 名と同一であり、かつ ParalellWorld 名と同一である。

Quantity および Filter 共に、異なるメッシュで共有されることはないものとする。

mesh の open 直下で定義されたフィルタは、MultiFunctionalDetector へのフィルタとして設定される。quantity 内で定義されたフィルタは、その quantity へのフィルタとして設定される。

但し、Filter は、同一メッシュ内であれば、共有できるようにするため、一度 open-close で定義されたフィルタは、そのフィルタ名を引用して、他のスコアラに添付できるようにする。

filter, quantity は mesh の open-close 内のみで利用可能

filter は、mesh の open-close 内および quantity の open-close 内で利用可能

II) メッシュ定義

1) 概要

メッシュは、メッシュ名と共に開いた際に、未定義であれば新規に定義されるものとする。既に定義されているメッシュがある場合は、既存メッシュを再度開くことを表す。メッシュ名は固有名称でなければならない。指定したメッシュに対する設定は、次のコマンドの範囲で有効である。

```
/score/mesh/open <mesh 名/S>
```

```
/score/mesh/close
```

この範囲内では、/score/mesh/のほかに、/score/quantity/および/score/filter/が利用可能である。(S は、String 型で与える。その他/I は int 型、/D は double 型とする) 既にメッシュが開かれている状態で、他のメッシュを開いた場合はエラーとなる。

2) メッシュ形状

2 - 1) Box 型

```
/score/mesh/shape/box <dX/D> <dY/D> <dZ/D> <unit/S>
```

dX, dY, dZ : BOX 形状の X,Y,Z 方向のハーフサイズ値

unit : dX, dY, dZ の単位を表す (default: mm)

2 - 2) Tubs 型

```
/score/mesh/shape/tubs <Rin/D> <Rout/D> <dZ/D> <unit/S>
```

dRin, dRout, dZ : tubs 形状の内径, 外径値と, Z 方向のハーフサイズ値

unit : dRin, dRout, dZ の単位を表す (default: mm)

2 - 3) Sphere 型

```
/score/mesh/shape/sphere <Rin/D> <Rout/D> <unit/S>
```

dRin, dRout : sphere の内径, 外径値

unit : dRin, dRout の単位を表す (default: mm)

3) メッシュ分割

形状によらず共通する次のコマンドにより分割する。

```
/score/mesh/bin/numberOfBin <N1/I> <N2/I> <N3/I>  
                                <P1/D> ... <Pn-1/D> <unit/S>
```

N1,N2,N3 : 分割数 (原則として等間隔で分割される)

Box (Nx, Ny, Nz), Tubs (Nr, Nphi, Nz)

P1~Pn-1 : 分割ベクタ。N3 で指定する軸を、与えた仕切り値で分割する
仕切り値は、最小値からの分割の厚さの累積値とする。

unit : 分割ベクタで与える数字の単位

4) メッシュの配置

形状によらず共通する次のコマンドにより，移動と回転を行う。

実験室系座標で，絶対位置として指定する。

```
/score/mesh/translate/set <x/D> <y/D> <z/D> <unit/S>
```

x, y, z : 実験室座標系での位置

unit : 座標で与えた数値の単位

実験室系座標に対する回転として指定する。

```
/score/mesh/rotate/set <Rx/D> <Ry/D> <Rz/D> <unit/S>
```

Rx, Ry, Rz : 実験室座標系での各軸方向での回転角度

unit : 回転で与えた数値の単位

現在の配置位置に対して，与えた距離だけ移動させる

```
/score/mesh/translate/add <dx/D> <dy/D> <dz/D> <unit/S>
```

dx, dy, dz : 現在の配置位置に対する移動量

unit : 座標で与えた数値の単位

現在の回転にたいして，さらに回転を指定する。

```
/score/mesh/rotate/add <dRx/D> <dRy/D> <dRz/D> <unit/S>
```

dRx, dRy, dRz : 追加する各軸方向での回転角度

unit : 回転で与えた数値の単位

5) メッシュの削除*

メッシュに含まれる全てのオブジェクトを消去する。

```
/score/mesh/delete <メッシュ名>
```

このコマンドは，mesh が open されていない状態で実行されなければならない。

(open と同レベルのコマンド)

III) スコア量の定義

1) 概要

スコアする物理量を quantity と呼ぶ。quantity は、定義された mesh の open-close 内で固有なスコアラ名と共に指定されなければならない。

/score/quantity/open <スコアラ名/S>

/score/quantity/close

この範囲内では、/score/quantity/のほかに、/score/filter/が利用可能である。

2) スコア量のタイプ

/score/quantity/type <スコアタイプ/S> <Option1> <OptionN>

スコアタイプ： スコア量のタイプを表す名称（プリミティブスコアラ名）

（名前規則）PrimitiveScorer の G4PS を省略し、初めの文字を大文字に直したもの。例) G4PSEnergyDeposit は、EnergyDeposit

Option1 ... N： スコアタイプに依存するオプションを設定する。

対応は、表 1 を参照すること。

3) スコア量の削除*

指定する物理量をスコアするオブジェクトを削除する。

/score/quantity/delete <スコアラ名/S>

このコマンドは、指定されたスコアラが利用されている mesh を open した状態で実行されなければならない。

（/score/quantity/open と同レベルのコマンド）

表1 スコアタイプと対応オプション

スコアタイプ	Option=default 値	備考
CellCharge	オプションなし	
TrackLength	<Weighted=FALSE/B> <MultiplyKineticEnergy=FALSE/B> <DivideByVelocity=FALSE/B>	
CellFlux	オプションなし	
PassageCellFlux	オプションなし	
PassageCellCurrent	<Weighted=TRUE/B>	
PassageTrackLength	<Weighted=TRUE/B>	
EnergyDeposit	オプションなし	
DoseDeposit	オプションなし	
NofStep	オプションなし	
FlatSurfaceCurrent	<Direction=0, 1 or 2/I> <Weighted=TRUE/B> <DivideByArea=TRUE/B>	0=InOut 1=In 2=Out
FlatSurfaceFlux	<Direction=0, 1 or 2/I>	
SphereSurfaceCurrent	<Direction=0, 1 or 2/I> <Weighted=TRUE/B> <DivideByArea=TRUE/B>	
SphereSurfaceFlux	<Direction=0, 1 or 2/I>	
CylinderSurfaceCurrent	<Direction=0, 1 or 2/I> <Weighted=TRUE/B> <DivideByArea=TRUE/B>	
CylinderSurfaceFlux	<Direction=0, 1 or 2/I>	
MinKinAtGeneration	オプションなし	
NofCollision	<Weighted=FALSE/B>	
Population	<Weighted=FALSE/B>	
TrackCounter	<Weighted=FALSE/B>	
NofSecondary	オプションなし	
Termination	<Weighted=FALSE/B>	

III) フィルタの定義

1) 概要

filter は、スコアする物理量の条件を指定する。filter は、定義された mesh または quantity の open-close 内で固有なフィルタ名と共に指定されなければならない。

/score/filter/open <フィルタ名>

/score/filter/close

この範囲内では、/score/filter/のみが利用可能である。

2) フィルタのタイプ

/score/filter/type <フィルタタイプ> <Option1> ... <OptionN>

フィルタタイプ: フィルタタイプの名称 (センシティブフィルター名)

(名前規則) SensitiveFilter の G4SD を省略し、初めの文字を大文字に

直したもの。例) G4SDKineticEnergyFilter は、KineticEnergyFilter

Option1 ... N: フィルタタイプに依存するオプションを設定する。

対応は、表 2 を参照すること。

3) フィルタの共有

既に定義済みのフィルタを再定義することなしに利用することが可能である。

この場合には、以下のコマンドを用いる。

/score/filter/attach <フィルタ名/S>

フィルタ名: 定義済みのフィルタ固有名称

フィルタを適用する mesh または quantity が open された状態で実行される。

(/score/filter/open と同レベルのコマンド)

4) フィルタの削除

定義済みのフィルタを削除する。同一のフィルタが複数のスコアラまたはメッシュに適用されている可能性がある。そのため、フィルタを参照する mesh または quantity がある場合は削除を行わない。つまり、削除コマンドが mesh で実行された場合は、mesh へのフィルタを抜き取る。quantity で実行された場合には、その quantity へのフィルタを抜き取る。mesh 内で、フィルタを参照する mesh または quantity がなくなった場合に、フィルタオブジェクトを削除する。

/score/filter/delete <フィルタ名/S>

フィルタ名: 定義済みのフィルタ固有名称

mesh または quantity が open された状態で実行される。

(score/fileter/open と同レベルのコマンド)

表2 フィルタタイプと対応オプション

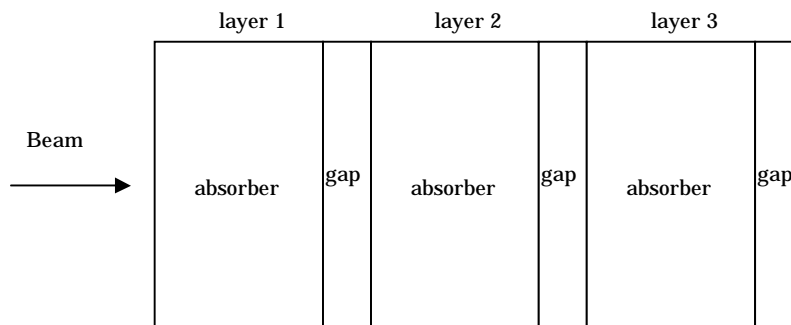
フィルタタイプ	Option=default 値	備考
Charged	なし	
Neutral	なし	
KineticEnergy	<Emin=0.0/D> <Emax=DBL_MAX/D> <unit=MeV/S>	
Particle	<ParticleName0/S> <ParticleNameN/S>	
ParticleWithKineticEnergy	<Emin/D> <Emax /D> <unit/S> <ParticleName0/S> ... <ParticleNameN/S>	
Process	<ProcessName/S>	
ProcessType	<ProcessType/S>	
CreatorProcess	<ProcessName/S>	

Ⅴ) コマンド適用シミュレーション

1) exampleN03

1 - 1) 概要

サンドイッチ・カロリメタでの各領域でのエネルギー付与とトラック長を計測する。エネルギー付与は、全ての粒子にたいして行い、トラック長は荷電粒子に対してのみ行う。形状の構造は、以下の通り。デフォルトでは10個のレイヤーまでである。



1 - 2) コマンド検証

各部でのエネルギー損失分布をビーム軸方向(X 軸)にとる場合のコマンドを検証する。

```
/score/mesh/open  SocreMesh
/score/mesh/shape/box  5.  5.  75.  cm  (ハーフサイズ。全長は 10x10x150cm)
#/score/mesh/bin/numberOfBin  1  1  10  (Layer  ごとのスコアを行う場合)
/score/mesh/bin/numberOfBin  1  1  10
                                10. 15.  25. 30.   40. 45.   55. 60.  70. 75
                                85. 90.  100. 105.  115. 120.  130. 135.
(Y 軸の周りに回転し、メッシュのZ 軸を実験室のX 軸に一致させる)
/score/mesh/rotate/set  0.  90.  0.  deg
/score/quantity/open  ChargedTrackLength
/score/quantity/type  TrackLength
/score/filter/open    ChargedFilter
/score/filter/type    ChargedFilter
/score/filter/close
/score/quantity/close
/score/quantity/open  EnegyDeposited
/score/quantity/type  EnergyDeposit
/score/quantity/close
/score/mesh/close
```


2) exampleN07

2 - 1) 概要

サンドイッチ・カロリメタでの物理量を計測する。ビームはZ軸方向である。カロリメタのデフォルトサイズは、全長2 mで20レイヤーからなる。それぞれのレイヤーは同じ厚さ。3つの同じカロリメタが配置される。

2 - 2) コマンド検証

```

/score/mesh/open CalorimeterA
  /score/mesh/shape/box 0.5 0.5 1.0 m (全長は、1x1x2m)
  /score/mesh/bin/numberOfBin 1 1 20 (Layer ごとのスコアを行う)
  /score/mesh/translate/set 0.0 -1.0 0.0 m
    /score/quantity/open eDep
      /score/quantity/type EnergyDeposit
    /score/quantity/close
    /score/quantity/open nGamma
      /score/quantity/type NofSecondary
      /score/filter/open gammaFilter
        /score/filter/type ParticleFilter gamma
      /score/filter/close
    /score/quantity/close
    /score/quantity/open nElectron
      /score/quantity/type NofSecondary
      /score/filter/open electronFilter
        /score/filter/type ParticleFilter e-
      /score/filter/close
    /score/quantity/close
    /score/quantity/open nPositron
      /score/quantity/type NofSecondary
      /score/filter/open positronFilter
        /score/filter/type ParticleFilter e+
      /score/filter/close
    /score/quantity/close
    /score/quantity/open minEkinGamma
      /score/quantity/type MinKinEAtGeneration
      /score/filter/attach gammaFilter
    /score/quantity/close
    /score/quantity/open minEkinElectron

```

```

        /score/quantity/type    MinKinEAtGeneration
        /score/filter/attach    electronFilter
    /score/quantity/close
    /score/quantity/open    minEkinPositron
        /score/quantity/type    MinKinEAtGeneration
        /score/filter/attach    positronFilter
    /score/quantity/close
    /score/quantity/open    trackLength
        /score/quantity/type    MinKinEAtGeneration
        /score/filter/open    epFilter
            /score/filter/type    ParticleFilter    e-    e+
        /score/filter/close
    /score/quantity/close
    /score/quantity/open    nStep
        /score/quantity/type    NofStep
    /score/quantity/close
/score/mesh/close
/score/mesh/open    CalorimeterB
    /score/mesh/translate/set    0.0    0.0    0.0    m
    他は CalorimeterA と同じなので省略
/score/mesh/close
/score/mesh/open    CalorimeterC
    /score/mesh/translate/set    0.0    1.0    0.0    m
    他は CalorimeterA と同じなので省略
/score/mesh/close

```

3) exampleN04

3 - 1) 概要

ROGeometry を用いた例題となっている。Tubs 型のカロリメタで , Phi 方向に 4 8 セグメントと Z 方向に 2 0 スライスのセグメントを用いている。Tracker 部と Muon カウンター部もあるが , ここでは , カロリメタ部のみを取り上げる。(Muon カウンター部は , LogicalVolume のコピーが利用できると良い)

3 - 2) コマンド検証

```
/score/mesh/open CaloRO
  /score/mesh/shape/tubs 50. 300. 200.0 cm
  /score/mesh/bin/numberOfBin 1 48 20
/score/quantity/open Edep
  /score/quantity/type EnergyDeposit
/score/quantity/close
/score/mesh/close
```