

---

# 光学光子の記述

## (Optical Photon)

---

Geant4 10.3.P3準拠

Geant4 HEP/Space/Medicine 講習会資料



大学共同利用機関法人  
高エネルギー加速器研究機構

# 本資料に関する注意

---

- 本資料の知的所有権は、高エネルギー加速器研究機構およびGeant4 collaborationが有します
- 以下のすべての条件を満たす場合に限り無料で利用することを許諾します
  - 学校、大学、公的研究機関等における教育および非軍事目的の研究開発のための利用であること
    - ・ Geant4の開発者はいかなる軍事関連目的へのGeant4の利用を拒否します
  - このページを含むすべてのページをオリジナルのまま利用すること
    - ・ 一部を抜き出して配布したり利用してはいけません
  - 誤字や間違いと疑われる点があれば報告する義務を負うこと
- 商業的な目的での利用、出版、電子ファイルの公開は許可なく行えません
- 本資料の最新版は以下からダウンロード可能です
  - <http://geant4.kek.jp/lecture/>
- 本資料に関する問い合わせ先は以下です
  - Email: [lecture-feedback@geant4.kek.jp](mailto:lecture-feedback@geant4.kek.jp)

# 目次

---

1. Geant4での光学光子の扱い
2. 光学光子の生成過程
3. 光学光子と物質の相互作用

**[謝辞]** 本講義は過去にSLAC、IN2P3主催のGeant4チュートリアルで使用され以下のスライドを参考にして作られています。

“Optical Photon Processes in GEANT4”

Peter Gumplinger (TRIUMF) : GEANT4 Users' Workshop at SLAC, February 2002

“Geant4 optics”

Giovanni Santin (ESA/ESTEC) : Ecole Geant4 : Annecy, November 2008

これらのスライドの著者に謝意を表します。

## Geant4での光学光子の扱い

---

# 光学光子(*Optical Photon*)

## ■ 光学光子(optical photon)とは

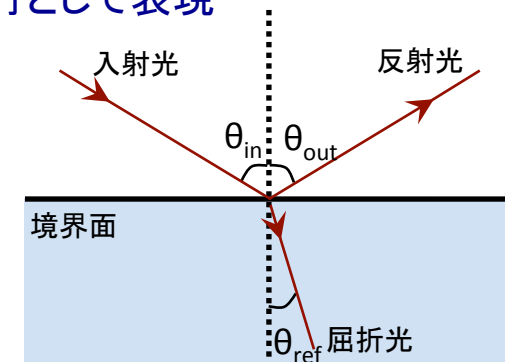
- 光子(photon)のエネルギーが光学(optical)領域のもの ← 赤・紫外を含む可視光領域

[注] 光子は波長が物質の原子間距離に比べ十分に大きい時、光学的に振る舞う

$$\lambda_{\text{optical}} (\sim 500\text{nm} = \sim 2\text{ eV}) \gg \text{原子間距離} (\sim 0.1\text{nm})$$

## ■ Geant4での光学光子の扱い

- 幾何光学的な振る舞い(反射、屈折、分散など)を粒子の進行として表現
- 光の吸収、弾性散乱(Rayleigh散乱など)も扱う
- 光の波動性(回折、干渉など)は考慮しない



## ■ G4OpticalPhoton

- 幾何光学的な光学光子の振る舞いを表現する仮想粒子
- 一方、G4Gammaは光学領域以上のエネルギーを持つ光子(X-ray、γ-ray)を表現

[注] 二つの粒子は全く独立で、G4Gammaの低エネルギー表現がG4OpticalPhotonではない

# G4OpticalPhotonの特徴

---

## ■ 光線の振る舞いをG4OpticalPhoton粒子のtrackingで表現

- 光線の反射屈折は多数のG4OpticalPhoton粒子が確率的に反射または屈折することで表現

[注] 一つのG4OpticalPhotonは境界面で反射されるか屈折されるかのいずれかである

## ■ G4OpticalPhotonは、その偏光(*polarization*)状態を必ず記述する必要がある

- 偏光はG4OpticalPhotonのスピン状態として表現する
- Geant4では光子をtrackingする際に偏光(偏極)方向を考慮する

[注] 偏光情報が設定されていないと、光子は正しくtrackingされない

- Geant4が用意している光学光子発生過程(チェレンコフ、シンチレータ等、後述)は、発生させる光子の偏光情報を自動的に設定する
- ユーザが独自の光学光子を発生させる場合、偏光情報は必ず自分で設定する

# Geant4光学光子の使い方

---

## ■ Geant4で光学光子を使う場合、ユーザは以下の二点に対応する必要:

- どのように光学光子を発生させるか    ← 光学光子発生過程の設定
- 光学光子の物質中での物理過程選択    ← 光学光子と物質の相互作用過程の設定

## ■ 光学光子発生過程の設定

- Geant4が標準で用意している光学光子発生過程を使う（詳細は以下の講義で述べる）
  - Cherenkov発光過程、シンチレーション発光過程、など
  - これは、Physics Listsに記述する
- あるいは、ユーザが独自に用意した光学光子発生過程を使う
  - Primary Particle Generatorを使って光学光子を発生させる、など

## ■ 光学光子と物質の相互作用過程の設定

- Geant4が標準で用意している光学光子相互作用過程を使う（詳細は以下の講義で述べる）
  - 物質境界面での相互作用： 反射、屈折など
  - 物質内での相互作用： 吸収、Rayleigh散乱など
- 物質の光学特性を用意する（詳細は次のスライド参照）
  - Material定義の中に記述する

# 物質の光学特性の設定

## ■ 光学光子の物理相互作用過程は物質の光学特性情報を必要とする

- 光学特性の例: 吸収長、屈折率、透過率、誘電率、表面形状、など

## ■ ユーザは以下の手順で光学特性情報を設定する必要がある

### 1) 光学特性パラメータの設定にはG4MaterialPropertiesTable(特性テーブル)を使う

```
G4MaterialPropertiesTable* MatPropTab = new G4MaterialPropertiesTable();
```

### 2) G4MaterialPropertiesTableの"AddProperty"メソッドでパラメータを設定

```
const G4int NUMENTRIES = 2;  
G4double photmom[NUMENTRIES] = {2.038*eV, 4.144*eV};  
G4double rindex[NUMENTRIES] = {1.35, 1.40};  
MatPropTab -> AddProperty("RINDEX", photmom, rindex, NUMENTRIES);
```

↑ keyword    ↑ energy array    ↑ value array    ↑ array size

[注1]

光は波長でなくエネルギーで表現される

[注2]

エネルギー値の並び順は任意でOK。但し、古いGeant4バージョン(例:9.5)を使う場合、低いものから高いものに順次並べること

[注] 物質の光学特性の各種パラメータ(上例では屈折率)は、ユーザが用意する必要がある

← Geant4が理論的にこれらの値を計算して設定することはない

### 3) 作成したG4MaterialPropertiesTableを物質に付け加える

```
G4NistManager* man = G4NistManager::Instance();  
G4Material* water = man->FindOrBuildMaterial("G4_WATER");  
water -> SetMaterialPropertiesTable(MatPropTab);
```

[注] テーブルはG4Material以外にG4SurfacePropertyにも付け加えられる

← Boundary Processのセクションを参照



# 光学光子の生成過程

---

# Geant4での光学光子の生成過程

## ■ 荷電粒子が物質を通過する際、Geant4では以下の物理過程で光学光子が生成される

- チェレンコフ放射: `G4Cerenkov`
- シンチレーション放射: `G4Scintillation`
- 遷移放射: `G4TransitionRadiation`
- シンクロトロン放射: `G4SynchrotronRadiation`

これらのクラスは以下のディレクトリで管理されている  
`processes/electromagnetic/xrays`

[注] 光学光子の生成においてエネルギー保存則は考慮されていないことに注意

➤ 事象の全エネルギーを足し上げる際は、光学光子エネルギーは無視すること

[注] 上記の過程で生成されたG4OpticalPhotonは偏光の情報設定が自動的にされている

[注] 本講義では上記生成過程の中の「チェレンコフ」と「シンチレーション」のみを扱う

## ■ 上の生成過程を使わず光学光子を扱いたい場合、ユーザは独自に光子を生成することになる

### 1. 独自のプログラム内で生成

```
G4OpticalPhoton* aPhoton = new G4OpticalPhoton();  
aPhoton -> SetPolarization(ux,uy,uz); // ux,uy,uzは単位ベクトル(直線偏光を必ず設定)
```

### 2. primary generatorのマクロを使って生成

```
/gun/particle opticalphoton  
/gun/polarization ux uy uz // ux,uy,uzは単位ベクトル(直線偏光を必ず設定)
```

[注] 直線偏光情報設定がされていないと、光学光子が正しくtrackingされないので、必ず設定すること

[注] 設定する直線偏光の方向は光学光子の進行方向に直角であることを忘れないこと

---

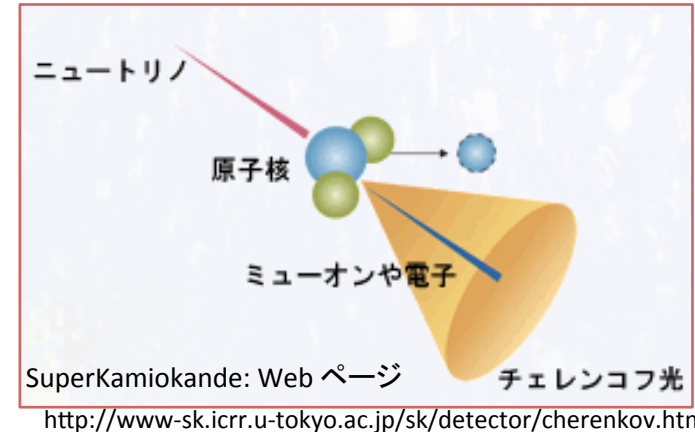
# 光学光子の生成過程

チェレンコフ放射過程

---

# チェレンコフ放射のGeant4による実装

- チェレンコフ光の発生点は親荷電粒子を輸送の一つ一つの直線ステップ(G4Step)上に分布する
  - チェレンコフ光はG4OpticalPhotonとして発生される
  - 磁場中でも発生点は直線ステップ上に分布
  - 光チェレンコフ光の発生を正確にシミュレーションするには、ステップ長を制限すること



- 平均発生G4OpticalPhoton数は屈折率を与えられている光の波長ステップごとに計算される
  - この計算値に対して統計的な揺らぎが考慮される
- 発生G4OpticalPhoton数は親荷電粒子のエネルギーに対して線形に変化する
- G4OpticalPhotonはチェレンコフ円錐面に垂直に直線偏光して発生
- チェレンコフ放射はユーザが屈折率を設定した物質内でのみ生じる

# チェレンコフ放射の使い方

## ■ チェレンコフ放射の使用例 → 以下のコードをPhysics Listに記述

```
#include "G4Cerenkov.hh"
```

```
G4Cerenkov* theCkovProcess = new G4Cerenkov("Cerenkov");  
theCkovProcess -> SetTrackSecondariesFirst(true);  
G4int MaxNumPhotons = 300;  
theCkovProcess->SetMaxNumPhotonsPerStep(MaxNumPhotons);  
theCkovProcess->SetMaxBetaChangePerStep(10.0);
```

← チェレンコフ過程の生成

下記参照

## ■ 発生したチェレンコフ光を輸送するタイミングをコントロール可能:

- SetTrackSecondariesFirst(true)
  - 各ステップで生じた光子を先に輸送する  
← particleスタックが発生光子であふれるのを避けることが可能となる
- SetTrackSecondariesFirst(false)
  - 親の荷電粒子をまず輸送したのちに、発生光子を輸送する  
← 光子輸送をするかしないかをeventの終わりに判断することが可能となる

## ■ 各ステップでのチェレンコフ光の発生最大個数をコントロール可能

- 放射過程は統計的に振る舞うので、実際の最大発生個数には揺らぎがある

## ■ チェレンコフ放射過程による親粒子の輸送ステップ長の制御

- 上記の光発生個数の最大値を設定することで、ステップ長を制御できる
- 親粒子の $\beta(=v/c)$ のステップ内での変化率(%)の最大値を設定により、ステップ長を制御
- 親粒子エネルギーがチェレンコフ放射の閾値より小さくなった時の、ステップ長を設定可能

## G4Cerenkovが使用する物質の光学特性パラメータ

### ■ 物質の光学特性はG4MaterialPropertiesTableに設定

- ユーザはG4MaterialPropertiesTableのAddPropertyメソッドを使いパラメータを設定

➤ AddPropertyメソッドでパラメータを設定する際は、決められているkeywordを使う

[例] MatPropTab -> AddProperty("RINDEX", photmom, rindex, NUMENTRIES);

↑ keyword    ↑ energy array    ↑ value array    ↑ array size

### ■ G4Cerenkovが必要とするユーザ設定パラメータとその'keyword'

Keyword	設定パラメータの内容
RINDEX	屈折率

---

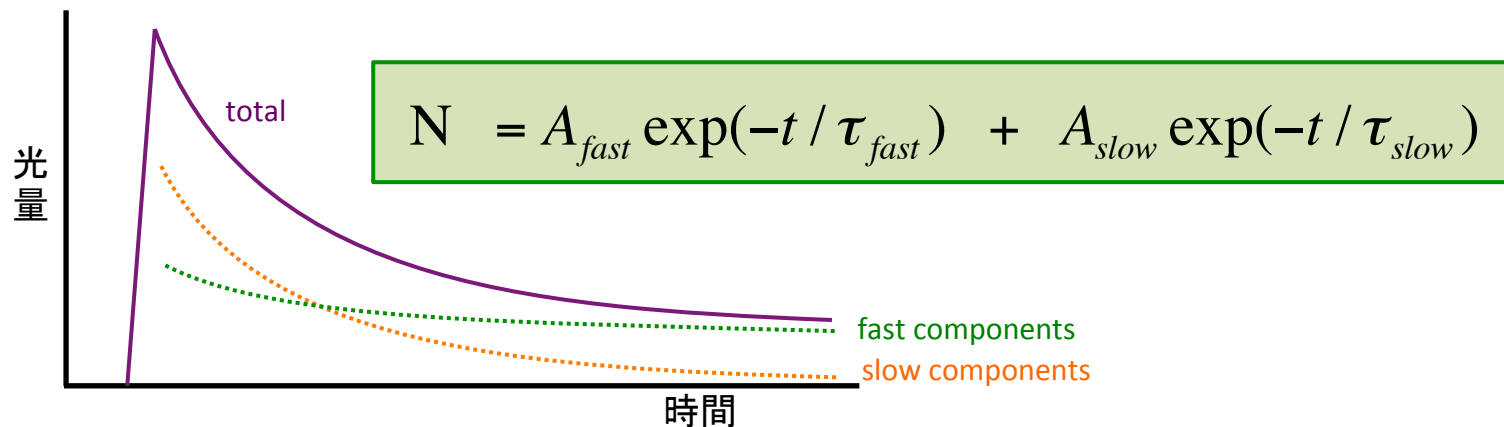
# 光学光子の生成過程

シンチレーション過程

---

# シンチレーション過程のGeant4による実装

- シンチレーション光はG4OpticalPhotonとして発生される
- 発生G4OpticalPhoton数は親荷電粒子の各ステップ間でのエネルギー損失に比例
- 発生G4OpticalPhotonのエネルギー・スペクトラムは観測結果をもとにサンプリング
- 発生G4OpticalPhotonは空間的に一様に放射される
- G4OpticalPhotonの発生点は親荷電粒子のステップ(G4Step)上に分布する
- G4OpticalPhotonは直線偏光している(向きはランダム)
- G4OpticalPhotonの発生の時間構造(time structure)は1要素または2要素(fast/slow)の指数分布





# シンチレーション過程の使い方

- シンチレーション過程の使用例 → 以下のコードをPhysics Listに記述

```
#include "G4Scintillation.hh"
```

シンチレーション過程の生成

```
G4Scintillation* theScintProcess = new G4Scintillation("Scintillation");  
theScintProcess -> SetTrackSecondariesFirst(true);  
theScintProcess -> SetScintillationYieldFactor(0.2);  
theScintProcess -> SetScintillationExcitationRatio(1.0);
```

下記参照

- 必要に応じて親粒子の輸送よりシンチレーション光子の輸送を先に行うことができる
- シンチレーションを生じる物質はそれぞれ固有の発光をする

[注]

- 粒子の種類によるシンチレーション発光量の違いは、“SetScintillationYieldFactor()”を使って設定できる
  - この場合、発光量の異なるシンチレーション過程を定義し、それを対応する粒子に付加
  - fast component / slow componentも同様に扱える
- 発光量の統計的なゆらぎは以下の二つの要素に影響される
    - ドープ結晶内の不純物: ゆらぎ幅を大きくする
    - Fano Factor: ゆらぎ幅を小さくする

# G4Scintillationが使用する物質の光学特性パラメータ

## ■ 物質の光学特性はG4MaterialPropertiesTableに設定

- ユーザはG4MaterialPropertiesTableのAddPropertyメソッドを使いパラメータを設定

➤ AddPropertyメソッドでパラメータを設定する際は、決められているkeywordを使う

[例] MatPropTab -> AddProperty("RINDEX", photmom, rindex, NUMENTRIES);

↑ keyword    ↑ energy array    ↑ value array    ↑ array size

## ■ G4Scintillationが必要とするユーザ設定パラメータとその'keyword'

Keyword	設定パラメータの内容	
FASTCOMPONENT	光発生のfast componentのamplitude	
FASTTIMECONSTANT	光発生のfast componentのtime constant	
FASTSCINTILLATIONRISETIME	光発生のfast componentのrise time	
SLOWCOMPONENT	光発生slow componentのamplitude	
SLOWTIMECONSTANT	光発生slow componentのtime constant	
SLOWSCINTILLATIONRISETIME	光発生slow componentのrise time	
SCINTILLATIONYIELD	平均photon発生数	
RESOLUTIONSCALE	平均photon発生数の揺らぎ幅	
YIELDRATIO	平均photon発生数の比率	
PROTONSCINTILLATIONYIELD	各種親粒子による平均photon発生数	
DEUTERONSCINTILLATIONYIELD		
TRITONSCINTILLATIONYIELD		
ALPHASCINTILLATIONYIELD		
IONSCINTILLATIONYIELD		
ELECTRONSCINTILLATIONYIELD		

[注] 必ずしも全ての  
パラメータを設定  
する必要はない

# 物質のシンチレーション特性の設定例

物質のユーザ定義ファイルに記述する

```
#include "G4Material.hh

// Liquid Xenon material
G4Element* elementXe = new G4Element("Xenon","Xe",54.,131.29*g/mole);
G4Material* LXe = new G4Material ("LXe",3.02*g/cm3,1, kStateLiquid, 173.15*kelvin, 1.5*atmosphere);
LXe -> AddElement(elementXe, 1);

const G4int NUMENTRIES = 9;
G4double LXe_PP[NUMENTRIES] =
    {6.6*eV,6.7*eV,6.8*eV,6.9*eV,7.0*eV, 7.1*eV,7.2*eV,7.3*eV,7.4*eV};
G4double LXe_SCINT[NUMENTRIES] =
    {0.000134, 0.004432, 0.053991, 0.241971, 0.398942, 0.000134, 0.004432, 0.053991,0.241971};
G4double LXe_RIND[NUMENTRIES] =
    { 1.57, 1.57, 1.57, 1.57, 1.57, 1.57, 1.57, 1.57, 1.57};
G4double LXe_ABSL[NUMENTRIES] =
    { 35.*cm, 35.*cm, 35.*cm, 35.*cm, 35.*cm, 35.*cm, 35.*cm, 35.*cm };

G4MaterialPropertiesTable* LXe_MPT = new G4MaterialPropertiesTable();

LXe_MPT -> AddProperty("FASTCOMPONENT",LXe_PP,LXe_SCINT,NUMENTRIES);
LXe_MPT -> AddProperty("RINDEX", LXe_PP,LXe_RIND,NUMENTRIES);
LXe_MPT -> AddProperty("ABSLLENGTH",LXe_PP, LXe_ABSL,NUMENTRIES);
LXe_MPT -> AddConstProperty ("SCINTILLATIONYIELD", 100./MeV);
LXe_MPT -> AddConstProperty("RESOLUTIONSCALE",1.0)
LXe_MPT -> AddConstProperty("FASTTIMECONSTANT",45.*ns);
LXe_MPT -> AddConstProperty("YIELDRATIO",1.0);

LXe -> SetMaterialPropertiesTable(LXe_MPT);
```

[Ref] "Optical Photon Processes in GEANT4"  
Peter Gumplinger (TRIUMF) : GEANT4 Users' Workshop at SLAC, February 2002

# 光学光子と物質の相互作用

---

# Geant4での光学光子と物質との相互作用過程

## ■ Geant4では光学光子が物質を通過する際に、以下の相互作用過程を扱うことが可能

- 物質による吸収: Bulk Absorption

- 光学光子が消滅する

- 物質境界面での反射、屈折: Reflection/Refraction

- 光学光子の波長が散乱体より十分に小さい場合 ( $x \gg 1$ )

- レイリー散乱: Rayleigh Scattering

- 光学光子の波長が散乱体より十分に大きい場合の散乱 ( $x \ll 1$ )

- ミー散乱: Mie Scattering

- 光学光子の波長が散乱体と同程度の大きさの場合の散乱 ( $x \approx 1$ )

- 波長シフト: Wave Length Shift

- 光学光子の波長が物質によって変化する

[注] ユーザが光学光子を扱う場合、上記の相互作用過程の内、最低以下の二つを有効にする

- 物質による吸収
- 物質境界面での反射、屈折

他の相互作用は必要に応じて有効にする

## ■ Geant4では光子をtrackingする際に偏光(偏極)方向の変化を考慮する

[注] ただし、光の位相の変化は考慮しない (すなわち、直線偏光のみで円偏光は考慮しない)

Size parameter ( $x$ )

$$x = 2\pi r / \lambda$$

$r$ : 散乱体の大きさ

$\lambda$ : 光の波長

---

# 光学光子と物質の相互作用

物質による吸収

---

# 光学光子の物質による吸収 – Bulk Absorption

## ■ Geant4での光学光子の物質による吸収過程の記述

- 相互作用過程名: **G4OpAbsorption** このクラスの管理ディレクトリ: `processes/optical`
- 光学光子を物質中で輸送する際に用いる平均自由行程は、物質に設定した吸収長 (Absorption Length) から計算される

```
G4MaterialPropertiesTable *MPT = new G4MaterialPropertiesTable();

G4double PhotonEnergy[nEntries] =
    {6.6*eV,6.7*eV,6.8*eV,6.9*eV,7.0*eV,7.1*eV,7.2*eV,7.3*eV,7.4*eV};
G4double AbsLength[nEntries] =
    {0.1*mm,0.2*mm,0.3*mm,0.4*cm,1.0*cm,10.0*cm,1.0*m,10.0*m,10.0*m};

MPT->AddProperty("ABSLLENGTH", PhotonEnergy, AbsLength, nEntries);
```

[注] 物質の吸収長は、理論的に計算はできない。ユーザが文献等から実測値を入手して設定する必要がある

- G4OpAbsorptionは光子の物質による吸収を、単に光子の輸送を停止する (killする)ことで記述する

## G4OpAbsorptionが使用する物質の光学特性パラメータ

### ■ 物質の光学特性はG4MaterialPropertiesTableに設定

- ユーザはG4MaterialPropertiesTableのAddPropertyメソッドを使いパラメータを設定

➤ AddPropertyメソッドでパラメータを設定する際は、決められているkeywordを使う

[例] MatPropTab -> AddProperty("RINDEX", photmom, rindex, NUMENTRIES);

↑ keyword    ↑ energy array    ↑ value array    ↑ array size

### ■ G4OpAbsorptionが必要とするユーザ設定パラメータとその'keyword'

Keyword	設定パラメータの内容
ABSLNGTH	吸収長



---

# 光学光子と物質の相互作用

物質境界面での反射、屈折

---

# 光学光子の物質境界面での反射、屈折 – Reflection/Refraction

## ■ 光学光子が物質の境界面を通過する際、以下の物理過程が考慮される

- 境界面での反射
- 境界面での屈折

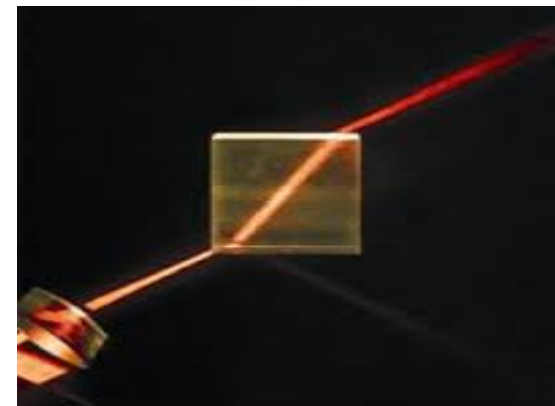
[注] 現バージョンのGeant4は光の回折(diffraction)を扱うことはできない

## ■ Geant4の物質境界面での反射、屈折の扱い

- 相互作用過程名: `G4OpBoundaryProcess` このクラスの管理ディレクトリ: `processes/optical`
- 光学光子が物体の境界面に到達したら、この過程が必ずよばれる
- この過程がtrackingのステップを決めることはない

## ■ 光学光子の物質境界での振る舞いを、隣接物質の性質により三つの場合に分ける

- 誘電体と誘電体:
  - 全反射、あるいはフレネル式に基づく反射／屈折
- 誘電体と金属:
  - 反射あるいは吸収
- 誘電体と黒体(光学特性が定義されていない物質):
  - 吸収



[Ref] <http://www.solitaryroad.com/c1033.html>

# 光学光子と物質表面

- 光学光子の物質境界面での反射、屈折をより現実的にシミュレートしたい場合、その物質の表面状態を記述した情報を用意する必要がある

- 表面が完全に平滑：表面状態に関する情報は不要
- 表面が平滑でない：表面状態を表現する必要がある

[注] 以下では「表面」と「境界面」という言葉は同義とする

- ユーザは物体の平滑でない表面状態の記述に以下の二つクラスが使える

- G4LogicalBorderSurface

- 二つの物体(Physical Volumes)が接している時に、その境界面を記述

```
G4LogicalBorderSurface( const G4String& name,      コンストラクタ  
                        G4VPhysicalVolume* vol1, G4VPhysicalVolume* vol2, ← 境界を形成する二つの物体  
                        G4SurfaceProperty* surfaceProperty ); ← 表面状態を記述するオブジェクト
```

- G4LogicalSkinSurface

- 一つの物体(Logical Volume)の全表面がスキン(反射ペイント等)で覆われている境界面を記述

```
G4LogicalSkinSurface( const G4String& name,      コンストラクタ  
                      G4LogicalVolume* vol,      ← スキンで覆われた物体  
                      G4SurfaceProperty* surfaceProperty ); ← 表面状態を記述するオブジェクト
```

[注] 上の二つのクラスはベース・クラスG4LogicalSurfaceの導出クラスとして実装されている

- ユーザは必要に応じて、G4LogicalSurfaceをベースクラスとして独自の表面記述を作ることが

[注] 表面状態は引数のG4SurfacePropertyクラスのオブジェクトで記述されている： 次のスライド参照

# 物質表面状態の記述クラス (G4OpticalSurface)

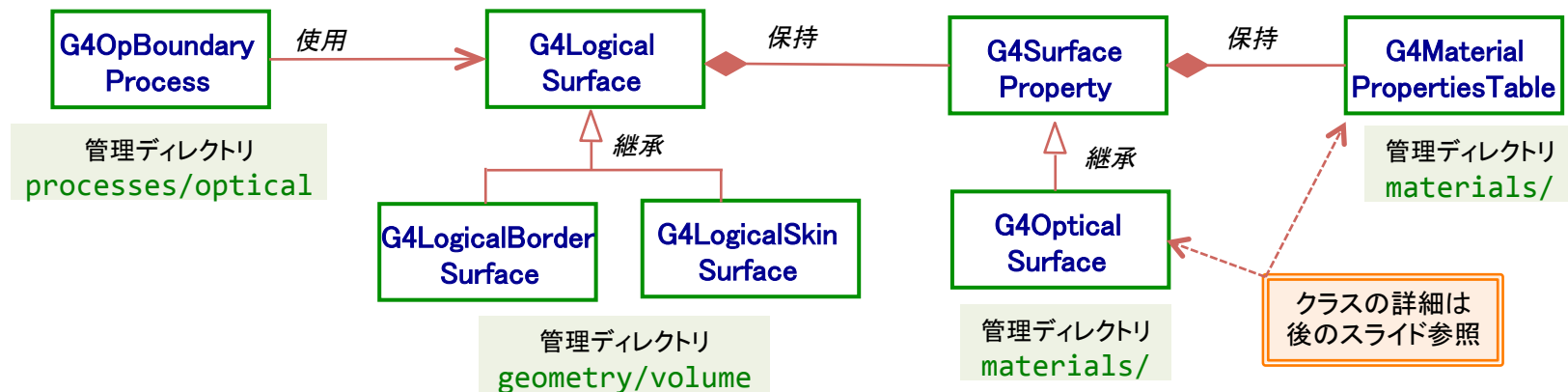
## ■ 光学光子に対する物質の表面状態はG4OpticalSurfaceクラスのオブジェクトで表現

- このクラスはG4SurfaceType(物質表面状態記述の一般的なクラス)からの導出クラス

```
G4OpticalSurface : public G4SurfaceProperty
```

- G4OpticalSurfaceオブジェクトはG4OpBoundaryProcessが光学光子をtrackingする時に必要な情報をすべて保持している:
  - 使用する表面モデル: glisur, unified, etc (G4OpticalSurfaceで定義されているenum)
  - 境界面の仕上げ状態: polished, ground, etc (G4OpticalSurfaceで定義されているenum)
  - 境界面を作る物体のタイプ: dielectric\_metal, dielectric\_dielectric, etc (G4SurfacePropertyのenum)
  - G4MaterialPropertiesTable: 境界面の付随的な光学情報が設定されている

## ■ 光学光子の物質境界面での物理過程(G4OpBoundaryProcess)と境界表面記述に関連するクラスと関係を以下にまとめる



# G4OpticalSurfaceクラスの標準的な使い方

- 1) G4OpticalSurfaceオブジェクトを作り”Set”メソッドで基本情報を設定

```
G4OpticalSurface* OpSurface = new G4OpticalSurface("surface_name");
```

光学光子のエネルギーに  
依存しないパラメータ

- 2) G4OpticalSurfaceの”Set”メソッドで表面表現の基本パラメータを設定

```
OpSurface -> SetModel(glisur);  
OpSurface -> SetFinish(polished);  
.....
```

setメソッドの詳細は後述

- 3) 基本パラメータ以外の情報を設定するためにG4MaterialPropertiesTableオブジェクトを作る

```
G4MaterialPropertiesTable* MatPropTab = new G4MaterialPropertiesTable();
```

光学光子のエネルギーに  
依存するパラメータ

- 4) G4MaterialPropertiesTableの”AddProperty”メソッドで詳細パラメータを設定

```
MatPropTab -> AddProperty("RINDEX",pp,rindex,NUM);  
MatPropTab -> AddProperty("SPECULARLOBECONSTANT",pp,specularlobe,NUM);  
.....
```

詳細パラメータの種類は後述

- 5) 作成したG4MaterialPropertiesTableオブジェクトをG4OpticalSurfaceオブジェクトに設定

```
OpSurface -> SetMaterialPropertiesTable(MatPropTab);
```

- 6) 作成が完了したG4OpticalSurfaceオブジェクトと表面を構成する物体を結合する

```
G4VPhysicalVolume* v1; G4VPhysicalVolume* v2;  
G4LogicalBorderSurface* Surface = new G4LogicalBorderSurface("volume_name",v1,v2,OpSurface);
```

[注] G4OpticalSurfaceはユーザが設定したパラメータをもとに表面の特性を表現

➤ パラメータの組み合わせで表面の振る舞いが異なってくることに注意

# G4OpticalSurfaceの”set”メソッド

## ■ SetModel(val-enum)

- 表面記述モデルを選択設定する
- メソッドの引数(val-enum)は以下の値のいずれかを設定

選択可能な値は以下の'enum'に定義されている  
`enum G4OpticalSurfaceModel{...}`

### [引数の値]

- glisur          単純な表面モデル : モデルを設定しないとデフォルトでこれが設定
- unified        詳細な表面モデル : glisurでは十分でない時に使用
- 上記以外にLUT、dichroicがあるが通常の使用では上の二つから選択

モデルの  
詳細は後述

[設定例]      `OpSurface -> SetModel(glisur);`

## ■ SetType(val-enum)

- 表面／境界面を形成する物体の種類を設定する
- メソッドの引数(val-enum)は以下の値のいずれかを設定

選択可能な値は以下の'enum'に定義されている  
`enum G4OpticalSurfaceType {...}`

### [引数の値]

- dielectric\_metal      誘電体と金属が作る境界面  
                            ← 光子は透過できない、ユーザが与える吸収係数で吸収、あるいは反射
- dielectric\_dielectric      誘電体と誘電体が作る境界面  
                                    ← 光子はユーザが与える反射率で境界両面で全反射、フレネル反射・回折
- 上記以外にdielectric\_LUT, dielectric\_dichroic等があるが通常の使用では上の二つから選択

[設定例]      `OpSurface -> SetType(dielectric_metal);`

# G4OpticalSurfaceの”set”メソッド (つづき)

## ■ SetFinish(val-enum)

- 表面の仕上げ状態を選択設定する
- メソッドの引数(val-enum)は以下の値のいずれかを設定

選択可能な値は以下の'enum'に定義されている  
`enum G4OpticalSurfaceFinish {...}`

[引数の値]

- polished 完全な光沢面
- polishedfrontpainted 光沢面で前面がペイント
- polishedbackpainted 光沢面で背面がペイントあるいはフイル
- ground 下地面
- groundfrontpainted 下地面で前面がペイント
- groundbackpainted 下地面で背面がペイントあるいはフイル
- その他多数ある
  - SetType(dielectric\_LUT)とすることで、G4 DataFileの'RealSurface1.0'から多数の表面状態を使える (Geant4 User's Guide for Application Developers Sec. 5.2を参照)
  - G4OpticalSurface.hhに定義されているenum G4OpticalSurfaceFinish {...}も参照のこと

[設定例]

```
OpSurface -> SetFinish(polished);
```

## ■ SetMaterialPropertiesTable(G4MaterialPropertiesTable)

- 境界面の付随的な光学情報を設定

[引数の値]

- G4MaterialPropertiesTableのオブジェクト

[設定例]

```
G4MaterialPropertiesTable* MatPropTab = new G4MaterialPropertiesTable();  
OpSurface -> SetMaterialPropertiesTable(MatPropTab);
```

# G4OpticalSurfaceの”set”メソッド (つづき)

---

## ■ SetPolish(*G4double*)

- **glisur**モデルで使われる表面の仕上げ状態を表す数値パラメータ

[引数の値]

- $0 \leq \text{value} \leq 1.0$ : 0は面の荒さが最大

[設定例] `OpSurface -> SetPolish(0.8);`

## ■ SetSigmaAlpha (*G4double*)

- **unified**モデルで使用する表面パラメータの設定

[引数の値]

- [radian]単位の角度値

`OpSurface -> SetSigmaAlpha (0.1);`

[注] 上記のモデル及び使用しているパラメータの詳細は後述



## G4OpBoundaryProcessが使用するユーザ設定のパラメータ

### ■ 物質の光学特性はG4MaterialPropertiesTableに設定

- 光学相互作用過程は光子をtrackingする際に、G4MaterialPropertiesTableから境界面の光学特性情報を取得
- ユーザはG4MaterialPropertiesTableのAddPropertyメソッドを使いパラメータを設定
  - AddPropertyメソッドでパラメータを設定する際は、決められているkeywordを使う

[例] MatPropTab -> AddProperty("RINDEX", photmom, rindex, NUMENTRIES);

keyword      energy array      value array      array size

### ■ G4OpBoundaryProcessが必要とするユーザ設定パラメータとその'keyword'

Keyword	設定パラメータの内容	
RINDEX	屈折率	[注] 必ずしも全てのパラメータを設定する必要はない
REFLECTIVITY	反射率	
REALRINDEX	屈折率の実数部	
IMAGINARYRINDEX	屈折率の虚数部	
EFFICIENCY	量子効率	
TRANSMITTANCE	透過率	
SURFACEROUGHNESS	反射率の角度依存性を考慮	
SPECULARLOBECONSTANT	unifiedモデルのパラメータ (後述)	
SPECULARSPIKECONSTANT	unifiedモデルのパラメータ (後述)	
BACKSCATTERCONSTANT	unifiedモデルのパラメータ (後述)	

# 物質表面状態の設定例

物質のユーザ定義ファイルに記述する

```
G4VPhysicalVolume* volume1;
G4VPhysicalVolume* volume2;

G4OpticalSurface* OpSurface = new G4OpticalSurface("name");

G4LogicalBorderSurface* Surface = new G4LogicalBorderSurface("name",volume1,volume2,OpSurface);

G4double sigma_alpha = 0.1;

OpSurface -> SetType(dielectric_dielectric);
OpSurface -> SetModel(unified);
OpSurface -> SetFinish(groundbackpainted);
OpSurface -> SetSigmaAlpha(sigma_alpha);

const G4int NUM = 2;
G4double pp[NUM] = {2.038*eV, 4.144*eV};
G4double specularlobe[NUM] = {0.3, 0.3};
G4double specularspike[NUM] = {0.2, 0.2};
G4double backscatter[NUM] = {0.1, 0.1};
G4double rindex[NUM] = {1.35, 1.40};
G4double reflectivity[NUM] = {0.3, 0.5};
G4double efficiency[NUM] = {0.8, 0.1};

G4MaterialPropertiesTable* SMPT = new G4MaterialPropertiesTable();

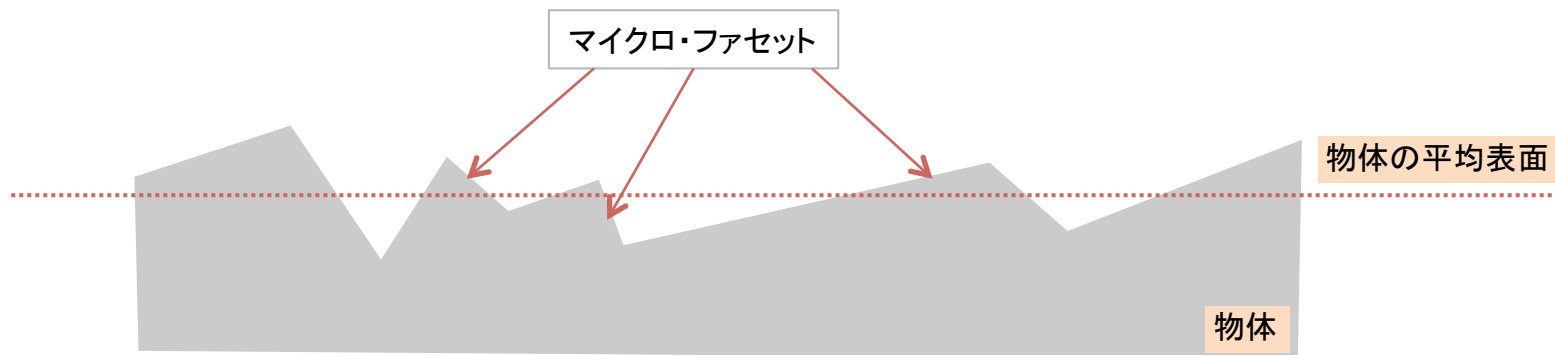
SMPT -> AddProperty("RINDEX",pp,rindex,NUM);
SMPT -> AddProperty("SPECULARLOBECONSTANT",pp,specularlobe,NUM);
SMPT -> AddProperty("SPECULARSPIKECONSTANT",pp,specularspike,NUM);
SMPT -> AddProperty("BACKSCATTERCONSTANT",pp,backscatter,NUM);
SMPT -> AddProperty("REFLECTIVITY",pp,reflectivity,NUM);
SMPT -> AddProperty("EFFICIENCY",pp,efficiency,NUM);

OpSurface -> SetMaterialPropertiesTable(SMPT);
```

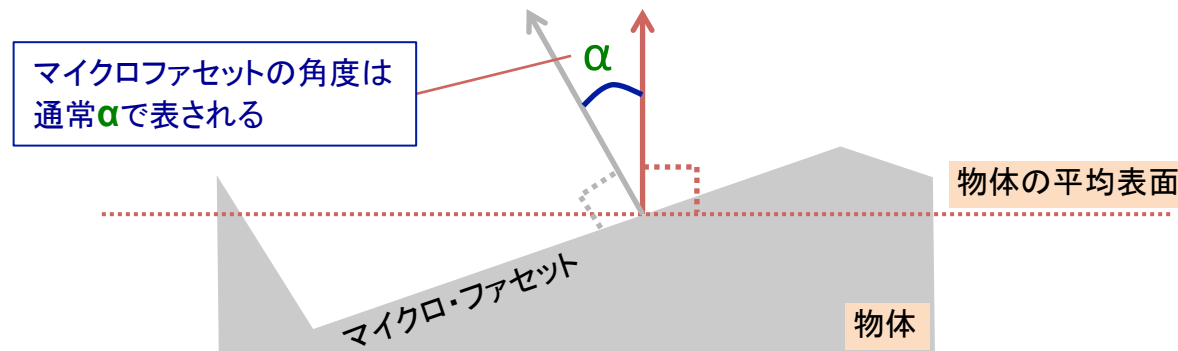
[Ref] Geant4 User's Guide for Application Developers  
Sec. 5.2: Example 5.8

# Geant4での表面モデル – glisurとunified

- Geant4では物体表面をマイクロ・ファセット(microfacet)の集合として表現
  - マイクロ・ファセットとは微少な平面
  - 物体表面はランダムに分布した微少表面で覆われているとする



- マイクロ・ファセットの面方向は物体の平均表面に対して定義される



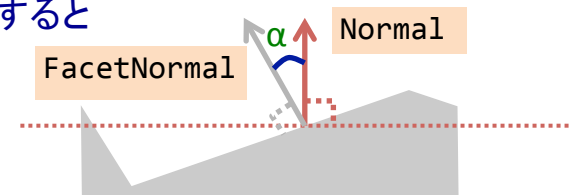
# Geant4での表面モデル – glisur

## ■ glisurモデルとは

- Geant4が採用している最も単純な表面モデル
- GEANT3(Geant4の前身のFORTRANプログラム)で開発されたモデルを基礎とする
  - ‘GEANT3.21’ CENR Program Library Long Writeup: W5013, p.230

## ■ glisurモデルの表面表現

- モデルが必要とするパラメータは‘polish’一つだけ
  - ‘G4OpticalSurface’クラスの‘SetPolish(G4double polish)’メソッドで設定（前述）
- マイクロ・ファセットの角度 $\alpha$ は以下の手続きで決定される
  - 半径が $(1.0 - \text{polish})$ の球を考える
  - 球表面上にランダムな一点を選び、それと球原点とが作るベクトル(smear)を作る
  - 物体の平均表面の垂直ベクトルを‘Normal’とする（Geant4のワールド座標系で）
  - マイクロ・ファセットの面の垂直ベクトルを‘FacetNormal’すると
  - $\text{MatFacetNormal} = \text{Normal} + \text{smear}$
- マイクロ・ファセットの表面反射は完全な光沢面として扱う



[注] GEANT3のオリジナル・モデルではマイクロ・ファセットはあらわに出てこないが、考え方は同じ

# Geant4での表面モデル – unified

## ■ unifiedモデルとは

- glisurモデルに比べて複雑な表面反射を扱うことが可能となる
- 表面を構成する二つの物質がともに誘電体である場合(dielectric-dielectric)に使われる
- 以下の文献で述べられているモデルに基づいている

A. Levin and C. Moisan, 'A More Physical Approach to Model the Surface Treatment of Scintillation Counters and its Implementation into DETECT': TRIUMF Preprint TRI-PP-96-64, Oct. 1996  
(<http://geant4.slac.stanford.edu/UsersWorkshop/G4Lectures/Peter/moisan.ps>)

## ■ unifiedモデルの表面表現

- モデルでは物体表面からの反射過程は以下の4種類に分類され、それらを統一的に記述
  - specular spike 物体の平均表面からの反射
  - specular lobe reflection 向きがgauss分布(幅: $\sigma_\alpha$  - 以下参照)のマイクロ・ファセットからの反射
  - backscatter reflection 入射光学光子の方向の逆側への反射
  - lambertian reflection 物体の平均表面に対してcos分布での反射 (ランバート反射)

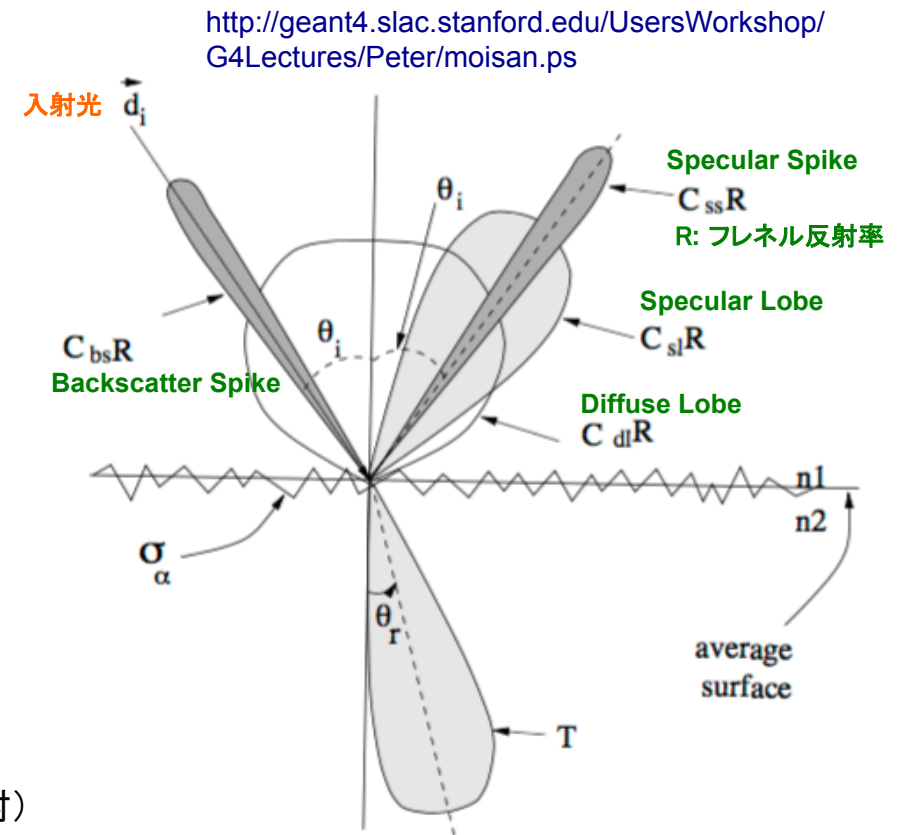
[注] 4種類の過程のそれぞれが起こる確率の総和は 1.0

- $\sigma_\alpha$ (sigma\_alpha) について
  - 上述のspecular lobe reflectionでのマイクロ・ファセットの方向を記述するパラメータ
  - 'G4OpticalSurface'クラスの' SetSigmaAlpha(G4double)'メソッドで設定

# Geant4での表面モデル – unified (つづき)

## ■ Material TableにAddProperty()を使って設定するunifiedモデルのkeywords

- $C_{sl}$  : Specular Lobe Constant
  - マイクロ・ファセット面での反射率
  - keyword: **SPECULARLOBECONSTANT**
- $C_{ss}$  : Specular Spike Constant
  - 物体の平均表面での反射率
  - keyword: **SPECULARSPIKECONSTANT**
- $C_{bs}$  : Backscatter Spike Constant
  - 後方反射(散乱)の確率
  - keyword: **BACKSCATTERCONSTANT**
- $C_{dl}$  : Diffuse Lobe Constant
  - ランバート反射率 (表面より内部からの反射)
  - keyword: **無し** ← 次式から計算 ( $C_{dl} = 1.0 - C_{sl} - C_{ss} - C_{bs}$ )



---

# 光学光子と物質の相互作用

波長シフト

---

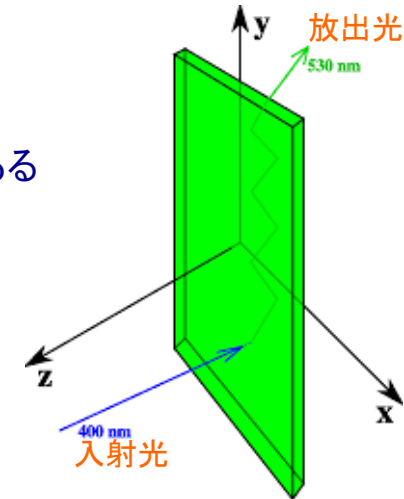
# 光学光子の波長シフト – Wavelength Shifting (WLS)

## ■ Wavelength Shifting(WLS)とは

- シンチレータの集光に使われる素材
- 通常、プラスチック板に芳香族分子を混入したもので様々な形状がある

## ■ WLSは以下の性質で特徴付けられる

- その光吸収と光放出スペクトラム
- 吸収光と放出光の吸収係数
- 光吸収と光放出の時間差
- 放射光は一様に分布し、ランダムな直線偏光



“Wavelength Shifters as (new) light sensors”  
M. Voge et al : PINGU workshop,  
Amsterdam, March 2011

## ■ Geant4でのWLSの扱い

- 相互作用過程名: **G4OpWLS**      このクラスの管理ディレクトリ: **processes/optical**
- 収波長シフトは入射光学光子の消失と、エネルギーの異なる新たな光学光子の発生で記述される
- ユーザは以下の情報を'G4MaterialPropertiesTable'に記述する
  - WLSの反射係数(光のエネルギーの関数)
  - WLS内での光の吸収係数(光のエネルギーの関数)
  - WLSの放出光スペクトラム(光のエネルギーの関数)



## G4OpWLSが使用する物質の光学特性パラメータ

### ■ 物質の光学特性はG4MaterialPropertiesTableに設定

- ユーザはG4MaterialPropertiesTableのAddPropertyメソッドを使いパラメータを設定

➤ AddPropertyメソッドでパラメータを設定する際は、決められているkeywordを使う

[例] MatPropTab -> AddProperty("RINDEX", photmom, rindex, NUMENTRIES);

↑ keyword    ↑ energy array    ↑ value array    ↑ array size

### ■ G4OpWLSが必要とするユーザ設定パラメータとその'keyword'

Keyword	設定パラメータの内容
WLSCOMPONENT	WLS光の発生強度
WLSABSLENGT	WLS光の減衰長

# Wavelength Shifting (WLS)特性の設定例

```
const G4int nEntries = 9;

G4double PhotonEnergy[nEntries] = { 6.6*eV, 6.7*eV, 6.8*eV, 6.9*eV,
                                     7.0*eV, 7.1*eV, 7.2*eV, 7.3*eV, 7.4*eV };

G4double RIndexFiber[nEntries] =
    { 1.60, 1.60, 1.60, 1.60, 1.60, 1.60, 1.60, 1.60, 1.60 };
G4double AbsFiber[nEntries] =
    { 0.1*mm, 0.2*mm, 0.3*mm, 0.4*cm, 1.0*cm, 10*cm, 1.0*m, 10.0*m, 10.0*m };
G4double EmissionFiber[nEntries] =
    { 0.0, 0.0, 0.0, 0.1, 0.5, 1.0, 5.0, 10.0, 10.0 };

G4Material* WLSFiber;
G4MaterialPropertiesTable* MPTFiber = new G4MaterialPropertiesTable();

MPTFiber->AddProperty("RINDEX", PhotonEnergy, RIndexFiber, nEntries);
MPTFiber->AddProperty("WLSABSLLENGTH", PhotonEnergy, AbsFiber, nEntries);
MPTFiber->AddProperty("WLSCOMPONENT", PhotonEnergy, EmissionFiber, nEntries);
MPTFiber->AddConstProperty("WLSTIMECONSTANT", 0.5*ns);

WLSFiber->SetMaterialPropertiesTable(MPTFiber);
```

物質のユーザ定義ファイルに記述する

[Ref] Geant4 User's Guide for Application Developers  
Sec. 5.2: Example 5.7

---

# 光学光子と物質の相互作用

レイリー散乱

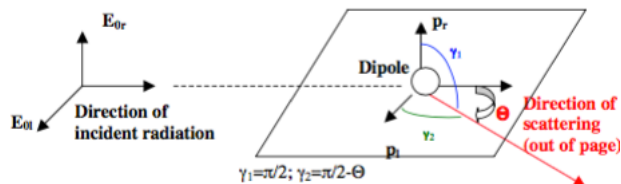
---

# 光学光子のレイリー散乱 – Rayleigh Scattering

## ■ レイリー散乱とは

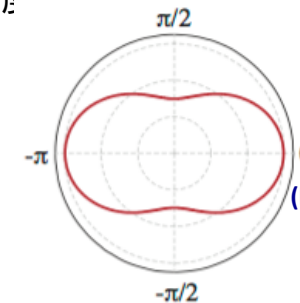
- 光学光子の波長が散乱体より十分に大きい場合の散乱 ( $x \ll 1$ )
- 散乱体を光の波長より十分に小さい電気双極子として扱い、Maxwell方程式を解くことで以下のように記述できる

$$I(\theta) = I_0 \frac{\alpha^2}{r^2} \left( \frac{2\pi}{\lambda} \right)^4 \frac{1 + \cos^2(\theta)}{2}$$



(Ref: [http://irina.eas.gatech.edu/ATOC5235\\_2003/Lec9.pdf](http://irina.eas.gatech.edu/ATOC5235_2003/Lec9.pdf))

$I_0$  : 入射光の強度  
 $I$  : 散乱光の強度  
 $\alpha$  : 分曲率 ( $p = \alpha E$ )  
 $r$  : 電気双極子からの距離  
 $\lambda$  : 入射光の波長 (=散乱光の波長)  
 $\theta$  : 散乱光の角度



(<https://www.cs.dartmouth.edu/~wjarosz/publications/dissertation/chapter4.pdf>)

## ■ レイリー散乱の特徴

- 散乱強度が $\lambda^{-4}$ に比例: 短波長ほど強く散乱される
- 散乱角は前後方向に対称で直角方向への散乱は少ない(繭形: 上図参照)
- レイリー散乱過程をシミュレーションで考慮すべきかどうかは、対象となる散乱体物質での平均自由行程( $\lambda_{\text{free}}$ )で判断 ← 多くのアプリケーションでは重要でない
  - 波長200nmの光: 空気中の $\lambda_{\text{free}} \sim 2 \text{ km}$
  - 波長200nmの光: Quartz中の $\lambda_{\text{free}} \sim 1 \text{ m}$

# 光学光子のレイリー散乱 – Rayleigh Scattering (つづき)

## ■ Geant4でのレイリー散乱の記述

- 相互作用過程名: `G4OpRayleigh` このクラスの管理ディレクトリ: `processes/optical`

## ■ Geant4でのレイリー散乱の偏光の扱い

- 散乱光学光子の直線偏光ベクトルは以下のように扱われる
  - 入射光学光子に運動量ベクトルに対して、以下の式に従って光学光子の散乱角をサンプルする

$$I(\theta) = I_0 \frac{\alpha^2}{r^2} \left( \frac{2\pi}{\lambda} \right)^4 \frac{1 + \cos^2(\theta)}{2} \quad (\text{式の内容は前スライドを参照})$$

- 入射光学光子の直線偏光ベクトルと散乱光学光子の運動量ベクトルが作る平面内に散乱光学光子の直線偏光ベクトルが向く

## ■ Geant4でのレイリー散乱平均自由行程( $\lambda_{\text{free}}$ )の扱い

- 様々な物質のレイリー散乱平均自由行程を計算することは非常に複雑である
  - ➔ ユーザが実測値などを用いて'G4MaterialPropertiesTable'に記述するのが基本
- ユーザが実測値を入手できず、レイリー散乱平均自由行程を設定しなかった場合、物質が水に設定されれば、Geant4はG4OpRayleighクラスが持っている'RayleighAttenuationLengthGenerator'メソッドを使い10°Cの水に対する $\lambda_{\text{free}}$ を自動的に設定
  - ➔ 詳細はユーザガイド、'ForApplicationDeveloper'の5.2章のOptical Photon参照

## G4OpRayleighが使用する物質の光学特性パラメータ

### ■ 物質の光学特性はG4MaterialPropertiesTableに設定

- ユーザはG4MaterialPropertiesTableのAddPropertyメソッドを使いパラメータを設定

➤ AddPropertyメソッドでパラメータを設定する際は、決められているkeywordを使う

[例] MatPropTab -> AddProperty("RINDEX", photmom, rindex, NUMENTRIES);

↑ keyword    ↑ energy array    ↑ value array    ↑ array size

### ■ G4OpRayleighが必要とするユーザ設定パラメータとその'keyword'

Keyword	設定パラメータの内容
RINDEX	屈折率
RAYLEIGH	Rayleigh散乱の平均自由行程

---

# 光学光子と物質の相互作用

ミ-散乱

---

# 光学光子のミー散乱 – Mie Scattering

## ■ ミー散乱とは

- 光学光子の波長が散乱体と同じ程度の場合の散乱 ( $x \approx 1$ )
- レイリー散乱導出の仮定( $x \ll 1$ )を緩め、 $x \approx 1$ 領域でも記述できるMaxwell方程式の解析解
- 様々な大きさの散乱体を扱えるが、実際の物質への応用は複雑となる  
(解析解としてはレイリー散乱も包含している)
- Geant4では解析解を“Henyey-Greensterin”とよばれる近次解(HG近似)を用いて散乱を表現

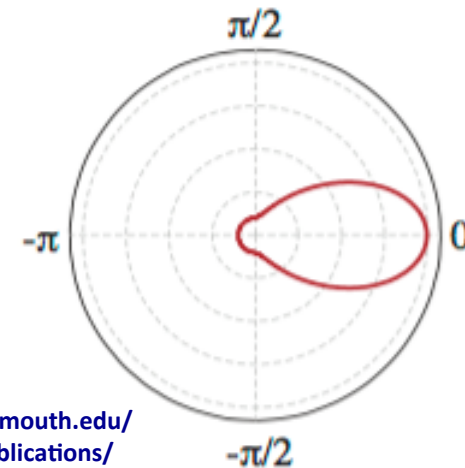
$$I(\theta) \propto I_0 \frac{1 - g^2}{(1 + g^2 - 2g \cos(\theta))^{3/2}}$$

$g = \langle \cos(\theta) \rangle$   
:  $\cos(\theta)$ の平均値(定数)

(詳細: Geant4 Physics Reference Manual 19.1 参照)

## ■ ミー散乱の特徴

- ミー散乱の考慮が必要となるのは $x \approx 1$ 領域  
→ 水滴程度の大きさの散乱体を持つ物質(大気中の雲など)
- 散乱角は前方にピークを持つ(葉巻形: 右図参照)



(<https://www.cs.dartmouth.edu/~wjarosz/publications/dissertation/chapter4.pdf>)



# 光学光子のミー散乱 – Mie Scattering (つづき)

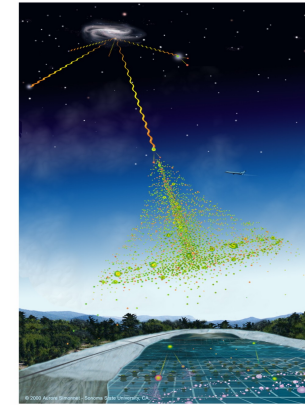
## ■ Geant4でのミー散乱の扱い

- 相互作用過程名: **G4OpMieHG** このクラスの管理ディレクトリ: [processes/optical](#)

- Geant4でのミー散乱はMilagro実験への応用を念頭に実装

Milagro: TeV All-Sky Gamma Ray Observatory

(<http://umdgrb.umd.edu/cosmic/milagro.html>)



## ■ Geant4でのミー散乱平均自由行程( $\lambda_{\text{free}}$ )の扱い

- レイリー散乱と同様、平均自由行程を計算することは非常に複雑である
  - ➔ ユーザが実測値などを用いて'G4MaterialPropertiesTable'に記述する
- ユーザは $\lambda_{\text{free}}$ を与えると同時に、HG近似パラメータである $g = \langle \cos(\theta) \rangle$ の値を設定する必要がある
  - ➔ 詳細はユーザガイド、'ForApplicationDeveloper'の5章のOptical Photon参照

## G4OpMieHGが使用する物質の光学特性パラメータ

### ■ 物質の光学特性はG4MaterialPropertiesTableに設定

- ユーザはG4MaterialPropertiesTableのAddPropertyメソッドを使いパラメータを設定

➤ AddPropertyメソッドでパラメータを設定する際は、決められているkeywordを使う

[例] MatPropTab -> AddProperty("RINDEX", photmom, rindex, NUMENTRIES);

↑ keyword    ↑ energy array    ↑ value array    ↑ array size

### ■ G4OpMieHGが必要とするユーザ設定パラメータとその'keyword'

Keyword	設定パラメータの内容
MIEHG	Mie散乱の減衰長

最後に

---

## Optical Photonを使用するための標準Physics List

- Optical Photonユーザの利便性のために以下のPhysics Listが用意されている
  - **G4OpticalPhysics**
- このPhysics Listを使うことで、Optical Photon関連の機能を使えるようになる
- G4OpticalPhysicsの使用例 (G4 Extended Example: LXeより)

```
LXePhysicsList::LXePhysicsList() : G4VModularPhysicsList() {  
    // default cut value (1.0mm)  
    defaultCutValue = 1.0*mm;  
  
    // General Physics  
    RegisterPhysics( new LXeGeneralPhysics("general") );  
  
    // EM Physics  
    RegisterPhysics( new LXeEMPhysics("standard EM"));  
  
    // Muon Physics  
    RegisterPhysics( new LXeMuonPhysics("muon"));  
  
    // Optical Physics  
    G4OpticalPhysics* opticalPhysics = new G4OpticalPhysics();  
    RegisterPhysics( opticalPhysics );  
  
    opticalPhysics->SetWLSTimeProfile("delta");  
    opticalPhysics->SetScintillationYieldFactor(1.0);  
    opticalPhysics->SetScintillationExcitationRatio(0.0);  
    opticalPhysics->SetMaxNumPhotonsPerStep(100);  
    opticalPhysics->SetMaxBetaChangePerStep(10.0);  
    opticalPhysics->SetTrackSecondariesFirst(kCerenkov,true);  
    opticalPhysics->SetTrackSecondariesFirst(kScintillation,true);  
}
```

G4OpticalPhysicsの生成・登録

Optical Photonの取り扱い  
に関する機能設定をする

[注]

特定の機能はdefaultで  
offになっているものが  
あるので注意が必要

