

# 光線追跡プログラムにおける 角度の処理方法について

システム工学技術系

星 勝広

## 1. 目的

現在、C言語を用いて光線追跡プログラムの開発を行っている。その中で角度の処理は、プログラムの各所で多く用いられ、基本的で重要な部分である。しかし、かなり煩雑な処理を要求され、プログラムミスの発生やプログラムの作成および修正に費やされる時間の拡大が生じる。そこで、より統一的な「角度の処理方法」を考案することによって、プログラムの作成効率を向上させることを目的とする。

## 2. 光線の屈折と反射 (Refraction and Reflection)

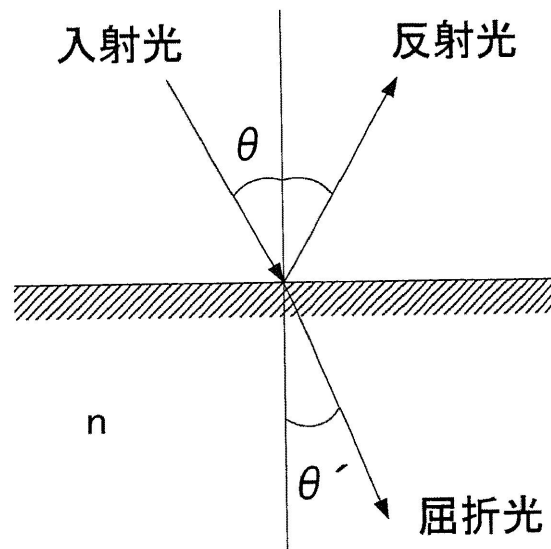


図 光線の屈折と反射

一様な空気中では、光は約  $3 \times 10^8$  [m/s] の速さでまっすぐに進む。ガラスや水のように、透明であっても空気と異なる物質内では速

さがちがう。これらの物質の中では、空気中よりも速さが小さい。このような物質を、空気よりも光学的に密なものに入ると、その境界面で方向が変わる。その変わり方は図のように、入射角 (angle of incidence)  $\theta$ 、屈折角 (angle of refraction)  $\theta'$  の関係が、

$$\sin \theta = n \cdot \sin \theta'$$

で、入射光線、屈折光線、境界面への法線は同一平面上にある。以上の関係が「屈折の法則」－「スネルまたはデカルトの法則」(Snell or Descartes law) である。上式の  $n$  を、媒質が空気に対する屈折率 (refractive index) という。これは、空気中での光の速さが媒質中での光の速さに対する比に等しい。

屈折率は媒質特有の定数であるが、また、光の波長にも関係する。波長が短いほど、屈折率は大きくなる。すなわち、光が余計に曲がる。

### 3. 屈折と反射のパターン

処理系の屈折と反射のパターンとしては、以下のものが上げられる。

#### 3-1. 反射光のみの場合

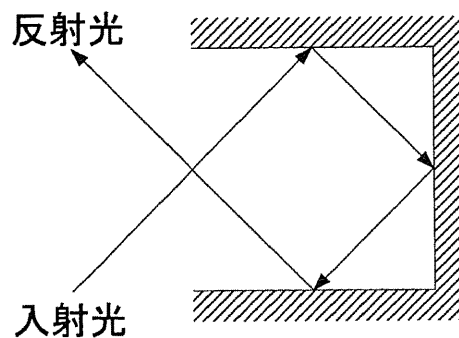


図 反射面での反射パターン

一つの入射光に対して一つの反射光のみが出力なので、処理としては最も簡単である。

#### 3-2. 透明シートの場合

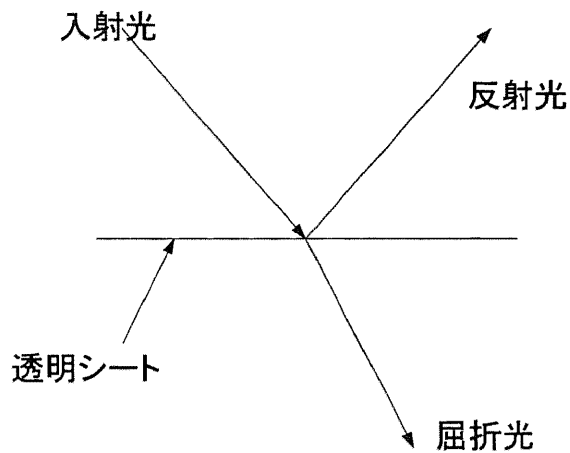


図 透明シートの反射と屈折パターン

一つの入射光に対して反射光と屈折光という複数の出力が発生し、処理が複雑になる。

### 3-3. 透過性の物体の場合

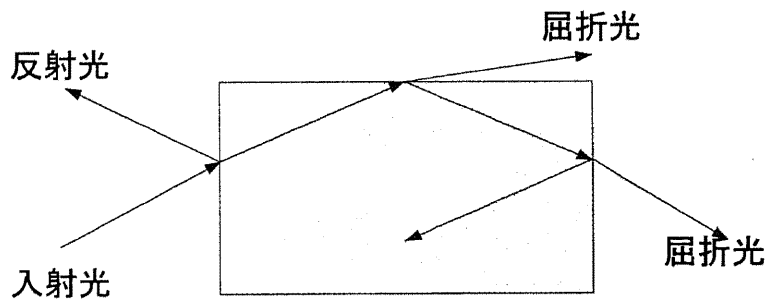


図 物体内の反射と屈折パターン

前述の反射のみの場合と透明シートの場合の性質を併せ持ったもので、一つの入力に対して複数の出力があり、角度の異なった面が複数存在し、更に複雑な処理になる。

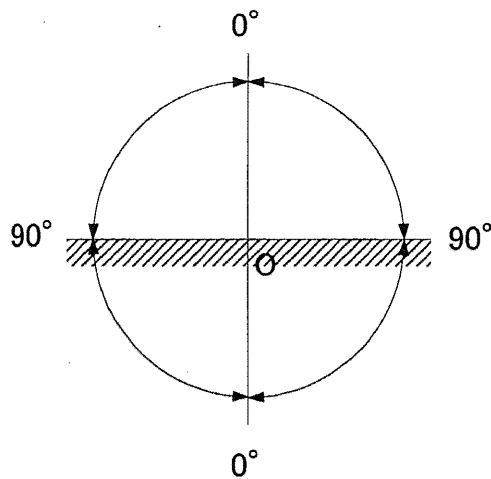
### 4. 角度表現「角」と「方位」

ある面のある点における入射や反射などを考える場合、面の直角方向に基準の $0^\circ$ をとり、 $0 \sim 90^\circ$ 範囲の角度を用いることは、人間側に

としては単純で直感的にもわかりやすい表現方法である。しかし、ある反射点から次の反射点の処理をするといった場合は、ある面の反射角を次の面の入射角に直接結びつけることは、つぎの面の向きが異なるのでできない。そこで、後者の場合には、 $0 \sim 360^\circ$  範囲の表現方法を用いることを試みた。

#### 4-1. 角度表現「角」( $0 \sim 90^\circ$ )

面に対して直角方向を基準の $0^\circ$ とし、 $0 \sim 90^\circ$ を角度範囲とする表現方法。

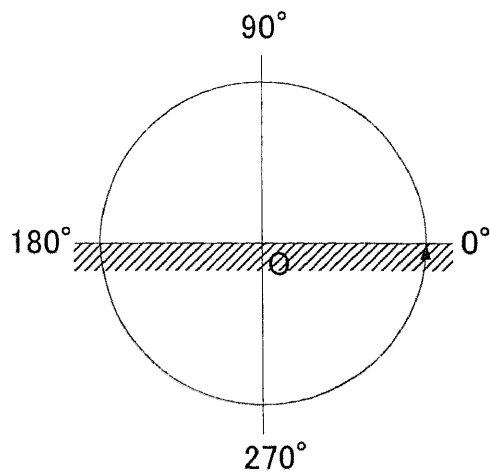


#### 特徴

- 測定データを、 $0 \sim 90^\circ$  範囲のものを用いている。
- 「屈折の法則」における反射や屈折の角度計算に用いられる。
- 人間側にとっては、単純で直感的にもわかりやすい表現で、屈折点や反射点などの局所的な処理に向く。
- 複数の処理面の間で値を引渡すような場合には、不都合。

#### 4-2. 角度表現「方位」( $0 \sim 360^\circ$ )

面の右側を基準の $0^\circ$ とし、 $0 \sim 360^\circ$ を角度範囲とする表現方法。



### 特徴

- 屈折・反射点間の処理に向く。
- 関数の入出力が統一され、関数に与える変数も一つですむと言う意味ではプログラムは、書きやすくなる。
- 面などの局所的な角度を考える意味では、人間側にとっては、わかりにくい。

## 5. 角度処理用の主な関数

### 5 - 1. 前段の出射方位から面に対する入射方位を求める関数

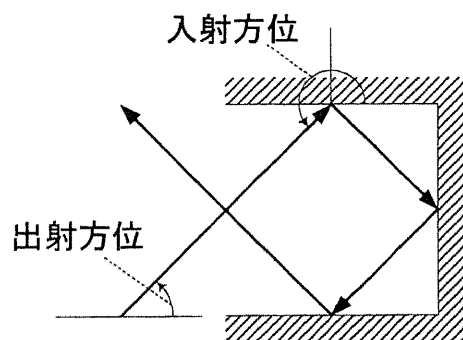


図 関数 `dir_otoi()` の処理

- 関数名 `dir_otoi`
- 使用例 `B = dir_otoi(a);`  
a:出射方位

## B:入射方位

### 5-2. 入射方位から物体の面に対する入射角を求める関数

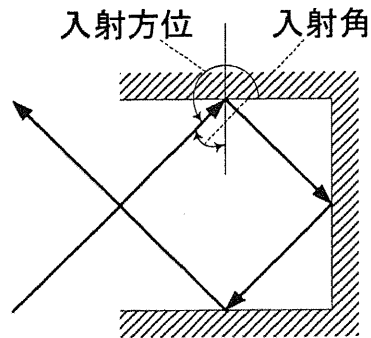
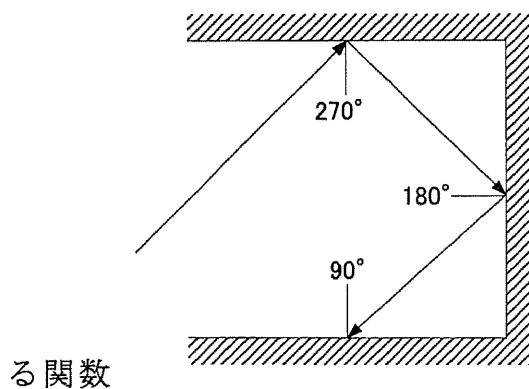


図 関数 `angle_input()` の処理

- 関数名 `angle_input`
- 使用例 `C = angle_input(a, b);`  
a: 屈折面番号 (各面には番号が割当てられている)  
b: 入射方位  
C: 入射角

### 5-3. 入射面基準方位 (各面のどの方向を $0^\circ$ と考えるか) を求め



る関数

図 各面における基準方位

- 関数名 `dir_std`
- 使用例 `B = dir_std(a);`

a: 反射面番号  
 B: 入射面基準方位

## 6. 角度の処理手順の例

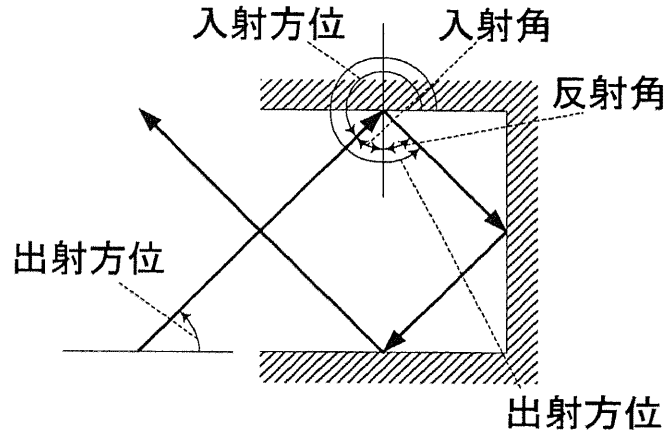
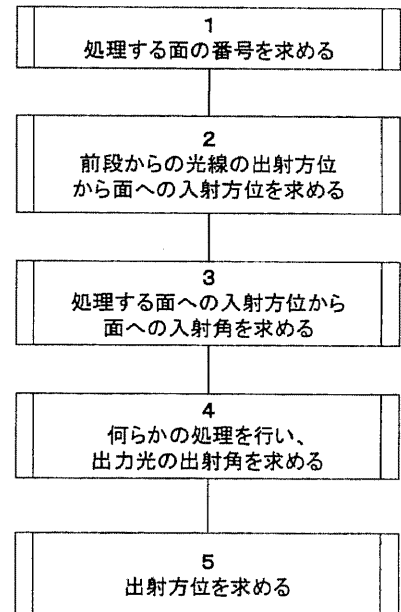


図 角度の処理手順の例

1. `face = get_face();`
  2. `dir = dir_otoi(dir);`
  3. `angle = angle_input(face, dir);`
  4. (5.) `dir = dir_reflect(face, dir);`
- `angle_input()` 関数には、各面における基準方位を求める `dir_std()` 関数が含まれている。



## 7. まとめ

一つの方法として、屈折・反射点における局所的な角度処理の場合は「角」を、屈折・反射面および光源の間における大域的な角度処理の場合は「方位」を用いて区別することによって、面間などの大域的な部分には入射方位をプログラム関数に受渡し、各面などの局所的な部分に

は入射方位を入射角に変換し、何らかの処理をした後で再び出射方位に変換し、次の面などのプログラム関数にその値を引渡すと言うように、角度の処理手順が統一され、プログラム全体の動きがわかりやすくなった。そして、この方法を用いることによって比較の見通し良いプログラムを書くことができるようになり、プログラムの作成効率の向上が見られた。

#### 8. 今後の課題

面の表裏の基準の取り方や面のあらゆる角度の場合について柔軟に対応できるような方法を考案することによって、プログラムを更に発展させる上で、有効になると考えられる。

#### 9. 参考文献

照明工学 電気学会