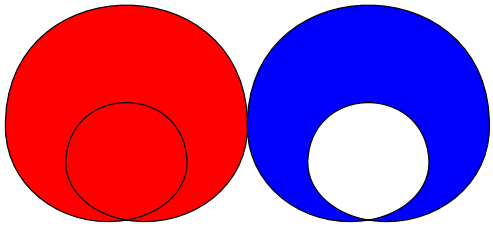
比較対照 METAPOST & Asymptote 〜 第3回

1　塗りつぶす場所

円などでは (通常の意味で)「内部」がどこなのか議論の余地はないと思いますが，次の例のように交差する二重円 (?) のような図形では,「内部」の解釈に違いが出てきます。PostScript では「内部」の判定法として zerowinding と evenodd というものがあり，Asymptote はこの両方に対応しています[1]。それぞれ **pen** 値の引数として指定できます。デフォルトは zerowinding です。以下，具体例で見ていきましょう。

なお，今回 **pair** 値がたくさん出てきますので，宣言と同時に **pair** 値の配列を設定する形にしてます。配列で指定したとき，番号は 0 番から始まることに注意してください。

```
\begin{ASYpic}<8mm>(0,0)(4,-0.5)
  \sendASY{pair[] z = {(0,0),(1,1),(-1,1),(2,1.5),(-2,1.5)};
    path p=z[0]..z[1]..z[2]..z[0]..z[3]..z[4]..cycle;
    filldraw(shift(-2,0)*p,red+zerowinding);
    filldraw(shift(2,0)*p,blue+evenodd);
  }
\end{ASYpic}
```



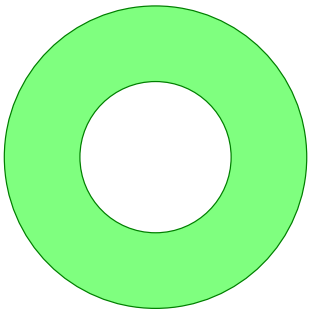
ある点 z が「内部」かどうか判定するときは，z を左端とし右に無限に伸びる半直線を，⟨ 巡回パス ⟩ がどのように横切るかで判定します。以下は上の例を見ながら考えるとわかりやすいと思います。「内部」や「外部」に相当する所に点 z をとり，上述のような半直線を実際に描いてみると，なおわかりやすいと思います。

まず zerowinding について。これは ⟨ 巡回パス ⟩ に沿ってペンを動かしたとき，上述の半直線を下から上に横切る回数と上から下に横切る回数が等しければ「外部」，そうでなければ「内部」という判定方法です[2]。

次に，evenodd について。これは ⟨ 巡回パス ⟩ に沿ってペンを動かしたとき，上述の半直線を横切る回数が偶数回なら「外部」，奇数回なら「内部」という判定方法です[3]。

ちなみに，Asymptote には METAPOST にない ^^ というパス結合演算子があり，これと evenodd を使えば，次のような円環を簡単に塗りつぶせます[4]。

```
\begin{ASYpic}<1cm>(0,0)(3,-2)
  \sendASY*{path p=unitcircle;
            path q=scale(2)*unitcircle;
            filldraw(p^^q,lightgreen+evenodd,deepgreen);
  }
\end{ASYpic}
```



^^ は，記号の左側の ⟨ パス ⟩ の終点と右側の ⟨ パス ⟩ の始点を結び，2 つの ⟨ パス ⟩ から 1 つの ⟨ パス ⟩ を作りますが，-- や .. と異なり，結びつきを表す線を描きませんし，結びつけることによるもとの曲線の変形も起こりません。

2　グラデーション塗り

Asymptote は METAPOST と異なり，初めから「グラデーション塗り」が用意されています[5]。「グラデーション塗り」に使う命令は latticeshade , axialshade , radialshade , gouraudshade , tensorshade で，名前から想像がつくと思いますが,「グラデーション塗り」というよりは立体の影付け (shading) 用の命令のつもりのようです。とくに gouraudshade と tensorshade は完全に立体用 (領域を細かく分けて塗っていく) で敷居が高いので，今回はそれ以外の 3 つを説明します。

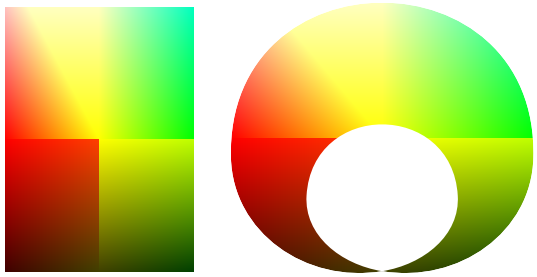
---

[1] METAPOST は zerowinding です。
[2] zerowinding は，下から上に横切る回数から上から下に横切る回数を引いた値が 0 という意味です。直訳すれば 0 回巻き？
[3] evenodd の意味は，読んで字のごとく「偶奇」です。
[4] ^^ は TEX の「特殊文字」ですので，\sendASY ではなく \sendASY* を使って下さい。
[5] MEPOTEX ではマクロで「グラデーション塗り」を実現しています。

2.1 latticeshade

これは塗りつぶしたい領域を含む長方形上に仮想的な「格子点」を設定し，そこに指定した色を置き，格子点間はグラデーションで塗るというものです。言葉で説明するより，実際に見た方がわかりやすいでしょう。

```
\begin{ASYpic}<1cm>(0,0)(7,-0.8)
  \sendASY{pair[] z = {(0,0),(1,1),(-1,1),(2,1.5),(-2,1.5)};
    path p=z[0]..z[1]..z[2]..z[0]..z[3]..z[4]..cycle;
    pen[][] col={{palered,paleyellow,palegreen},
                 {red,yellow,green},
                 {darkred,darkolive,darkgreen}};
    latticeshade(scale(2.5,3.5)*unitsquare,col);
    latticeshade(shift(5,0)*p,evenodd,col);
  }
\end{ASYpic}
```



色は例によって **pen** 値で指定しますが，今回の場合，それを格子状に配置しますので，要素が **pen** 値の 2 次元配列 (行列) で指定することになります。なお，この例は 2 次元配列の設定の仕方の例でもあります。あと，evenodd を指定したいときは，オプションの **pen** 値引数として，上記の **pen** 値配列の前に入れることにも注意して下さい。

2.2 axialshade

これは塗りつぶしたい領域を，ある方向に色を変化させながらグラデーションで塗るというものです。

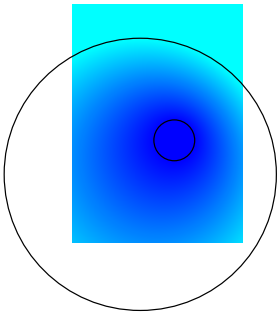
```
\begin{ASYpic}<1cm>(0,0)(7,-0.3)
  \sendASY{pair[] z = {(0,0),(1,1),(-1,1),(2,1.5),(-2,1.5)};
    path p=z[0]..z[1]..z[2]..z[0]..z[3]..z[4]..cycle;
    axialshade(scale(2.5,3.5)*unitsquare,
               red,(0,0),yellow,(2.5,3.5));
    axialshade(shift(5,0)*p,
               red+evenodd,(3,0),yellow,(5.5,3.5));
  }
\end{ASYpic}
```

引数は，塗りつぶしたい ⟨ 巡回パス ⟩ の次に，始点の色 (**pen** 値) , 始点の位置 (**pair** 値) , 終点の色 (**pen** 値) , 終点の位置 (**pair** 値) の順になります。evenodd を指定するときは，始点の色に追加 (+) で指定します。

2.3 radialshade

これは塗りつぶしたい領域を，円形 (?) に色を変えながらグラデーションで塗るというものです。もう少し具体的にいえば，塗りつぶしたい領域の他に，円と対応する色を 2 組指定して，領域内で第 1 の円の外を対応する色で塗り，領域内で第 2 の円の中を対応する色で塗り，最後に領域内で第 1 の円の中で第 2 の円の外にあたる部分をグラデーションで塗ります[6]。以下の例で確認してください。参考までに，指定した円を描いておきます。

```
\begin{ASYpic}<9mm>(0,0)(3,-0.8)
  \sendASY{pair[] z = {(1,1),(1.5,1.5)};
    radialshade(scale(2.5,3.5)*unitsquare,cyan,z[0],2,blue,z[1],0.3);
    draw(shift(z[0])*scale(2)*unitcircle);
    draw(shift(z[1])*scale(0.3)*unitcircle);}
\end{ASYpic}
```



引数は，塗りつぶしたい ⟨ 巡回パス ⟩ の次に，第 1 の円の色[7]，中心，半径，第 2 の円の色，中心，半径です。

[6] 領域内にこの説明に合致する部分がなければ，塗りません。
[7] evenodd を指定するときはここに指定します。指定方法は axialshading と同様です。