[参考]複素数の関数

①平方根　：$\sqrt{a+bj}=\sqrt{\frac{\sqrt{a^2+b^2}+a}{2}}+j\cdot\sqrt{\frac{\sqrt{a^2+b^2}-a}{2}}$

②対数　：$\log(a+bj)=\log\sqrt{a^2+b^2}+j\cdot\mathrm{Sin}^{-1}\frac{b}{\sqrt{a^2+b^2}}$

③虚数の三角関数　：$\cos(bj)=\frac{e^b+e^{-b}}{2}\quad(=\cosh b)$

$$\sin(bj)=\frac{e^{-b}-e^b}{2j}=j\cdot\frac{e^b-e^{-b}}{2}\quad(=j\cdot\sinh b)$$

$$\tan(bj)=\frac{\sin(bj)}{\cos(bj)}=j\cdot\frac{e^b-e^{-b}}{e^b+e^{-b}}\quad(=j\cdot\tanh b)$$

④複素数の三角関数　：$\sin(a+bj)=\sin a\cosh b+j\cdot\cos a\sinh b$

$$\cos(a+bj)=\cos a\cosh b-j\cdot\sin a\sinh b$$

$$\tan(a+bj)=\frac{\tan a\left(1-\tanh^2 b\right)}{1+\tan^2 a\tanh^2 b}+j\cdot\frac{\tanh b\left(\tan^2 a+1\right)}{1+\tan^2 a\tanh^2 b}$$

$\mathrm{Tan}^{-1}(a+bj)=A+Bj$ として

$$A=\mathrm{Tan}^{-1}\frac{\left(a^2+b^2-1\right)+\sqrt{\left(a^2+b^2-1\right)^2+4a^2}}{2a},\quad B=\frac{1}{2}\log\frac{a\tan A+b+1}{a\tan A-b+1}$$

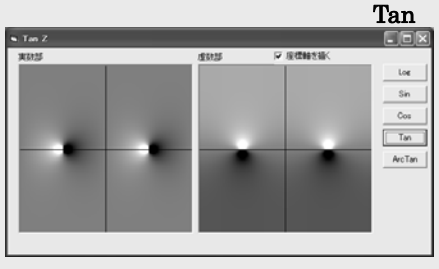
ただし $a\tan A-b+1\to 0$ のとき $B\to\infty$, $a\tan A+b+1\to 0$ のとき $B\to-\infty$

[ちょっと一息]

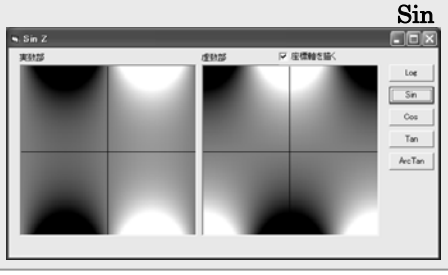
複素数関数の実数部と虚数部の値を色マップで表示すると，興味深い画像になります。以下の図は，入力の複素数の実数部を X 軸に，虚数部を Y 軸にして，結果の実数部と虚数部を色マップで表示したものです。

Log

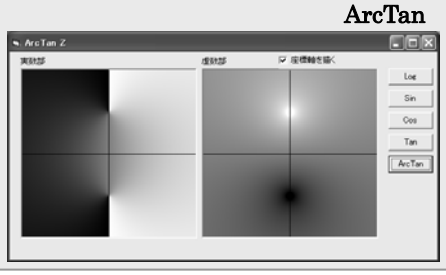
Tan



Sin



ArcTan



■その他

複素数の等値(==)は,差の絶対値が微小な値以下であることで判定します。なお，等値を定義しても

「Equals および GetHashCode はオーバーライドされません」

とのワーニングエラー(軽度のエラー)が表示されます。実行に影響はありませんが，目障りですので，これらの関数のオーバーライドを定義しておきます。

[Program 2-16] 複素数演算子定義の例

基本的演算，比較の定義例を示します。

プログラムリスト１

```
public struct Complex //複素数の構造体
{   public double R, I, E;
    // R:実部, I：虚部, E:等値比較のための誤差
    public Complex(double P1,double P2) // 誤差指定なし
    {   this.R=P1; this.I=P2;this.E=0.0000001;
    }
    public Complex(double P1,double P2,double Err)//誤差指定
    {   this.R=P1; this.I=P2;this.E=Err;
    }
    // 加算
   public static Complex operator +(Complex a,Complex b){
        return new Complex(a.R+b.R,a.I+b.I);
    }
    public static Complex operator +(double a,Complex b){
        return new Complex(a+b.R, b.I);
    }
   public static Complex operator +(Complex a,double b){
        return new Complex(a.R+b, a.I);
   }
   public static Complex operator +(Complex b){
        return new Complex(b.R,b.I);
   }
    //減算
    public static Complex operator -(Complex b){
      return new Complex(- b.R, - b.I);
    }
```