【スミスチャート　Smith chart】

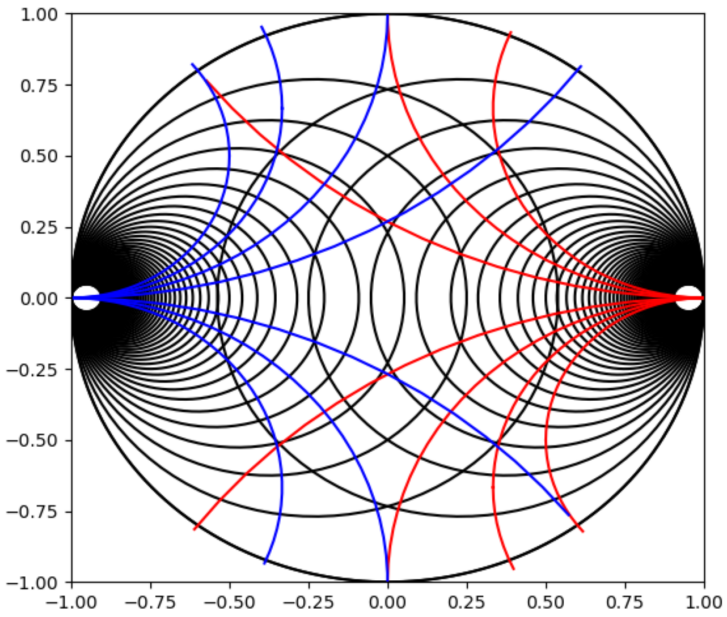

Figure 1: Python 実行結果

Source Code 1: Python

```
#−−−−−−−−−−−−−−−−−−−−−−−−
# Module_Name: Smith Chart
# Author: m_tsutsui
#−−−−−−−−−−−−−−−−−−−−−−−−

#Library_Import##############################
from numpy import*
import math, numpy as np
import matplotlib.pyplot as plt
#Library_Import_end##########################

def tra11(r):
    t=np.linspace(0,2*np.pi,100)
    cx=r/(r+1) #中心 x 座標

    hankei=1/(r+1)

    cf=array([hankei*cos(t)]) #Circle_factor

    cfx=cx+cf

    return cfx

def tra12(r):

    t=np.linspace(0,2*np.pi,100)

    cy=0 #中心y座標
    hankei=1/(r+1)

    cf=array([[hankei*sin(t)]]) #Circle_factor
```

```
    cfy=cy+cf

    return cfy


def tra21(x):
    t=np.linspace(0,2*np.pi,100)

    cx=1
    cy=1/x

    hankei=1/x

    cfx=array([hankei*cos(t)]) #Circle_factor
    cfy=array([hankei*sin(t)]) #Circle_factor

    cfx=cx+cfx
    cfy=cy+cfy

    for i in range(99):
        if (cfx[0,i])**2+(cfy[0,i])**2<=1.08:
            cfx[0,i]=cfx[0,i]
        else:
            cfx[0,i]=NaN
            cfy[0,i]=NaN


    return cfx

def tra22(x):

    t=np.linspace(0,2*np.pi,100)

    cx=1
    cy=1/x

    hankei=1/x

    cfx=array([hankei*cos(t)]) #Circle_factor
    cfy=array([hankei*sin(t)]) #Circle_factor

    cfx=cx+cfx
    cfy=cy+cfy

    for i in range(99):
        if (cfx[0,i])**2+(cfy[0,i])**2<=1.08:
            cfx[0,i]=cfx[0,i]
        else:
            cfx[0,i]=NaN
            cfy[0,i]=NaN

    return cfy

if __name__ == '__main__':

    a=np.zeros([2,100])
    for i in arange(0,20,0.3):
    #for i in range(10):
        a[0,:]=tra11(i)
        a[1,:]=tra12(i)
        plt.plot(a[0,:],a[1,:],"k")

    a2=np.zeros([2,100])
    for i2 in arange(0,20,0.3):
```

```
#for i in range(10):
    a2[0,:]=−tra11(i2)
    a2[1,:]=tra12(i2)
    plt.plot(a2[0,:],a2[1,:],"k")



b=np.zeros([2,100])
for j in arange(−2,2,0.5):
    b[0,:]=tra21(j)
    b[1,:]=tra22(j)
    plt.plot(b[0,:],b[1,:],"r")
    plt.xlim([−1,1])
    plt.ylim([−1,1])

c=np.zeros([2,100])
for k in arange(−2,2,0.5):
    c[0,:]=−tra21(k)
    c[1,:]=−tra22(k)
    plt.plot(c[0,:],c[1,:],"b")
    plt.xlim([−1,1])
    plt.ylim([−1,1])



plt.show()
```