Markdown

Golub–Kahan SVD step μ constant calculation

$$C = \begin{bmatrix} a_{1,1} & a_{1,2} \\ 0 & a_{2,2} \end{bmatrix} \begin{bmatrix} a^*_{1,1} & 0 \\ a^*_{1,2} & a^*_{2,2} \end{bmatrix} = \begin{bmatrix} |a_{1,1}|^2 + |a_{1,2}|^2 & a_{1,2}a^*_{2,2} \\ a^*_{1,2}a_{2,2} & |a_{2,2}|^2 \end{bmatrix}$$

$c_{1,1} = |a_{1,1}|^2 + |a_{1,2}|^2 \equiv c_u$
$c_{1,2} = a_{1,2}a^*_{2,2} = c^*_{2,1} \equiv c_d$
$c_{2,2} = |a_{2,2}|^2 \equiv c_l$

So
$$C = \begin{pmatrix} c_u & c_d{}^* \\ c_d & c_l \end{pmatrix}$$

$\det C = c_u c_l - |c_d|^2 = \lambda_1 \lambda_2 \equiv D$
$c_u + c_l = \lambda_1 + \lambda_2 \equiv T$

$\lambda_1 = D/\lambda_2$
$T = D/\lambda_2 + \lambda_2$
$\lambda_2 T - D - {\lambda_2}^2 = {\lambda_2}^2 - T\lambda_2 + D = 0$

$\lambda_1, \lambda_2 = \frac{T \pm \sqrt{T^2 - 4D}}{2}$

Want $\mu = \lambda$ such that $\lambda = \lambda_1, \lambda_2$ closest to c_l.

Function gks_mu

In [1]:
```python
import pyJvsip as pv
def gks_mu(a11,a12,a22):
    cu=a11 * a11.conjugate() + a12 * a12.conjugate()
    cl=a22 * a22.conjugate()
    cd = a12 * a22.conjugate()
    D = (cu * cl - cd * cd.conjugate()).real
    T = (cu + cl).real
    root = pv.vsip_sqrt_d(T*T - 4 * D)
    lambda1 = (T + root)/(2)
    lambda2 = (T - root)/(2)
    if abs(lambda1 - cl.real) < abs(lambda2 - cl.real):
        mu = lambda1
    else:
        mu = lambda2
    #un-comment below for testing
    #print('det: %.6f, trace: %.6f, closest to: %.6f'%(D, T, cl.real))
    #print('lambda1: %.6f, lambda2: %.6f'%(lambda1,lambda2))
    return mu
```

Simple Test of gks_mu

In [2]:
```python
a11 = complex(2,3); a12 = complex(.5,.6); a22 = complex(-1,-3)
```

In [3]:
```python
mu = gks_mu(a11,a12,a22)
print('%.6f'%mu)
```

```
8.745911
```