Matt McDermott

Numerical Methods HW #5

p273.) solve for eigenvalues using QR Method (finds all eigenvals simultaneously)

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$$

$$A = [a_1, a_2]$$

$$a_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix} \qquad a_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$q_1 = \frac{a_1}{\|a_1\|} = \frac{\begin{bmatrix} 2 \\ 1 \end{bmatrix}}{\left[2^2 + 1^2\right]^{1/2}} = \begin{bmatrix} 2/\sqrt{5} \\ 1/\sqrt{5} \end{bmatrix}$$

$$q_2 = \frac{a_2}{\|a_2\|} = \sqrt{3^2 + 4^2} = 5$$

$$q_1^T = [0.8944 \quad 0.4472] \qquad q_2^T = [0.6 \quad 0.8]$$

$$Q = [q_1 \quad q_2]$$

$$Q^{(0)} = \begin{bmatrix} 0.8944 & 0.6 \\ 0.4472 & 0.8 \end{bmatrix}$$

$$R^{(0)} = \begin{bmatrix} \|a_1\| & q_1^T a_2 \\ 0 & \|a_2'\| \end{bmatrix}$$

$$R^{(0)} = \begin{bmatrix} 2.2361 & 4.4722 \\ 0 & 5 \end{bmatrix}$$

$$[0.8944, \; 0.4472] \cdot \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$A^{(1)} = Q^{(0)} R^{(0)}$$

$$A^{(1)} = \begin{bmatrix} 0.8944 & 0.6 \\ 0.4472 & 0.8 \end{bmatrix} \begin{bmatrix} 2.2361 & 4.4722 \\ 0 & 5 \end{bmatrix}$$

$$A^{(1)} = \begin{bmatrix} 2 & 5 \\ 1 & 5 \end{bmatrix} \longrightarrow \text{repeat process in script}$$

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$$

$$a_1 = \begin{bmatrix} 2 \\ 1 \end{bmatrix}$$

$$\|a_1\| = \sqrt{2^2 + 1^2} = \sqrt{5}$$

$$q_1^T = a_1 / \|a_1\| = \begin{bmatrix} \frac{2}{\sqrt{5}} & \frac{1}{\sqrt{5}} \end{bmatrix}$$

$$= [0.8944 \quad 0.4472]$$

$$a_2 = \begin{bmatrix} 3 \\ 4 \end{bmatrix}$$

$$a_2' = a_2 - (q_1^T a_2) q_1$$

$$a_2' = \begin{bmatrix} 3 \\ 4 \end{bmatrix} - [0.8944 \quad 0.4472] \begin{bmatrix} 3 \\ 4 \end{bmatrix} \begin{bmatrix} 0.8944 \\ 0.4472 \end{bmatrix}$$

$$a_2' = \begin{bmatrix} 3 \\ 4 \end{bmatrix} - \begin{bmatrix} 4 \\ 2 \end{bmatrix}$$

$$a_2' = \begin{bmatrix} -1 \\ 2 \end{bmatrix}$$

$$a_2' = a_2 - (q_1^T a_2) q_1$$

(python)
» q1.T dot (a2) * q1

after 10 sims,
$\lambda = (1, 5)$

2.8(1.)

solve for eigenvectors of Matrix A corresponding to eigenvalues found in problem 7a via applying shifted inverse power method on time

let first component of $\underline{x}$ be unity component

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix} \qquad \lambda = 1, 5$$

$$A_{shifted} = (A - sI) \overset{\lambda=1}{=} \begin{bmatrix} 1 & 3 \\ 1 & 3 \end{bmatrix}$$

Doolittle LU Method

$$L = \begin{bmatrix} 1 & 0 \\ L_{21} & 1 \end{bmatrix} \cdot U = \begin{bmatrix} U_{11} & U_{12} \\ 0 & U_{22} \end{bmatrix} = \begin{bmatrix} 1 & 3 \\ 1 & 3 \end{bmatrix}$$

$U_{11} = 1$

$U_{11} \cdot L_{21} = 1$

$L_{21} = 1$

$1 \cdot U_{12} + 0 \cdot U_{22} = 3 \rightarrow U_{12} = 3$

$U_{12} \cdot L_{21} + U_{22} = 3$

$3 L_{21} + U_{22} = 3$

$3 + U_{22} = 3 \longrightarrow U_{22} = 0$

$$L = \begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \qquad U = \begin{bmatrix} 1 & 3 \\ 0 & 0 \end{bmatrix}$$

$LX' = X$

$$\begin{bmatrix} 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix} \rightarrow \begin{matrix} x_1' = 1 \\ x_2' = 0 \end{matrix}$$

$UY' = X'$

$$\begin{bmatrix} 1 & 3 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} y_1' \\ y_2' \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$$

$y_1 + 3y_2 = 1 \longrightarrow$

$\lambda = 1$

$\hookrightarrow \begin{bmatrix} 1 \\ 0.\overline{3} \end{bmatrix}$

$$A = \begin{bmatrix} 2 & 3 \\ 1 & 4 \end{bmatrix}$$

$\lambda = 5$

$$A - 5I = \begin{bmatrix} -3 & 3 \\ 1 & -1 \end{bmatrix}$$

$$L = \begin{bmatrix} 1 & 0 \\ l_{21} & 1 \end{bmatrix} \cdot U = \begin{bmatrix} u_{11} & u_{12} \\ 0 & u_{22} \end{bmatrix} = (A - 5I)$$

$u_{11} = -3 \quad u_{12} = 3$

$l_{21} \cdot u_{11} = 1$

$l_{21} = -\frac{1}{3}$

$l_{21} \cdot u_{12} + u_{22} = -1$

$-\frac{1}{3}(3) + u_{22} = -1$

$u_{22} = 0$

$Lx' = x$

$$\begin{bmatrix} 1 & 0 \\ -\frac{1}{3} & 1 \end{bmatrix} \begin{bmatrix} x_1' \\ x_2' \end{bmatrix} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$$

$x_1' = 1$

$x_1' \cdot (-\frac{1}{3}) + x_2' = 1$

$x_2' = \frac{4}{3}$

$Uy = x'$

$$\begin{bmatrix} -3 & 3 \\ 0 & -1 \end{bmatrix} \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 \\ 4/3 \end{bmatrix}$$

$-3y_1 + 3y_2 = 1$

$3y_1 - 4 = 1$

$y_1 = \frac{5}{3}$

$-y_2 = 4/3$

$y_2 = -\frac{4}{3}$

$$\begin{bmatrix} 5/3 \\ 4/3 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 0.8 \end{bmatrix}$$

$\lambda = 5 \rightarrow \begin{bmatrix} 1 \\ 0.8 \end{bmatrix}$

```python
#Matt McDermott
#Numerical Methods HW 5

import numpy as np

def QR(a,runLen):

    A = a
    for _ in range(runLen):

        a1 = A[:,0]
        a2 = A[:,1]

        q1 = - a1/np.linalg.norm(a1)
        # print("q1 = ", q1)

        a2Prime = a2 - q1.T.dot(a2)*q1
        # print("a2' = ",a2Prime)

        q2 = a2Prime/np.linalg.norm(a2Prime)
        Q = np.stack([q1,q2]).T
        # print("Q =  " ,Q)

        # R = np.array([[np.linalg.norm(a1), q1.dot(a1)],[0, np.linalg.norm(a2)]])
        R = np.array([[q1.dot(a1), q1.T.dot(a2)],[0, q2.dot(a2Prime)]])
        # print("R = ", R)

        A = R.dot(Q)
        print("A = ", A)


    ans = np.array([A[0,0], A[1,1]])

    return(ans)

if __name__ == "__main__":

    A = np.array([[2,3],[1,4]])
    runLen = 10

    ans = QR(A,runLen)
    print("ans = ", ans )
```

```
Anaconda Prompt (anaconda3)
A =  [[4.99356946 2.01275901]
 [0.01275901 1.00643054]]
A =  [[ 4.99871877e+00 -2.00255836e+00]
 [-2.55836134e-03  1.00128123e+00]]
ans =  [4.99871877 1.00128123]

(robot) C:\Users\Derm\NumericalMethods>python QR_Method.py
A =  [[ 4. -3.]
 [-1.  2.]]
A =  [[4.82352941 2.29411765]
 [0.29411765 1.17647059]]
A =  [[ 4.96725441 -2.06297229]
 [-0.06297229  1.03274559]]
A =  [[4.99356946 2.01275901]
 [0.01275901 1.00643054]]
A =  [[ 4.99871877e+00 -2.00255836e+00]
 [-2.55836134e-03  1.00128123e+00]]
A =  [[4.99974395e+00 2.00051193e+00]
 [5.11934462e-04 1.00025605e+00]]
A =  [[ 4.99994880e+00 -2.00010240e+00]
 [-1.02397379e-04  1.00005120e+00]]
A =  [[4.99998976e+00 2.00002048e+00]
 [2.04798951e-05 1.00001024e+00]]
A =  [[ 4.99999795e+00 -2.00000410e+00]
 [-4.09599581e-06  1.00000205e+00]]
A =  [[4.99999959e+00 2.00000082e+00]
 [8.19199832e-07 1.00000041e+00]]
ans =  [4.99999959 1.00000041]

(robot) C:\Users\Derm\NumericalMethods>
```