Matt McDermott

Numerical Methods HW #4

2.12) Solve for largest (in magnitude) eigenvalue of matrix E and corresponding eigenvector x via power method with $x^{(0)T} = [1.0, 0.0, 0.0], [0.0, 1.0, 0.0], \omega [0.0, 0.0, 1.0]$

a) let first component of x be unity component

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

Power Method

↳ works poorly if largest eig close to another

$$Ex^0 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 1 \\ 1 & 1 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \rightarrow \lambda^{(1)} = 2, \quad x^{(1)} = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}$$

$$Ex^1 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 1 \\ 1 & 1 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} = \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix} \rightarrow \lambda^2 = 5, \quad x_2 = \begin{bmatrix} 5 \\ 5 \\ 5 \end{bmatrix}$$

scale to 1

$$Ex^2 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 1 \\ 1 & 1 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 4 \\ 4 \\ 5 \end{bmatrix} \rightarrow \lambda^3 = 5, \quad x_3 = \begin{bmatrix} 4 \\ 4 \\ 5 \end{bmatrix}$$

$$Ex^3 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 1 \\ 1 & 1 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 1 \\ 1.25 \end{bmatrix} = \begin{bmatrix} 4.5 \\ 4.25 \\ 5.75 \end{bmatrix} \rightarrow \lambda_3 = 5.75, \quad x_3 = \begin{bmatrix} 4.5 \\ 4.25 \\ 5.75 \end{bmatrix}$$

$$Ex^4 = \begin{bmatrix} 1 & 1 & 2 \\ 2 & 1 & 1 \\ 1 & 1 & 3 \end{bmatrix}\begin{bmatrix} 1 \\ 0.9\overline{4} \\ 1.2\overline{7} \end{bmatrix} = \begin{bmatrix} 4.5 \\ 5.1\overline{6} \\ 5.8\overline{3} \end{bmatrix} \rightarrow \lambda_4 = 5.8\overline{3}, \quad x_4 = \begin{bmatrix} 4.5 \\ 5.1\overline{6} \\ 5.83 \end{bmatrix}$$

⋮

I got bored so I wrote a python script to do this for me

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

$$\left[\begin{array}{cc|cc} 2 & 3 & 1 & 0 \\ 1 & 4 & 0 & 1 \end{array}\right] \xrightarrow{4R_1 - 3R_2} \left[\begin{array}{cc|cc} 5 & 0 & 4 & -3 \\ 1 & 4 & 0 & 1 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{cc|cc} 1 & 0 & 0.8 & -0.6 \\ 1 & 4 & 0 & 1 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{cc|cc} 1 & 0 & 0.8 & -0.6 \\ 0 & 4 & -0.8 & 1.6 \end{array}\right]$$

$$\downarrow$$

$$\left[\begin{array}{cc|cc} 1 & 0 & 0.8 & -0.6 \\ 1 & 0 & -0.2 & 0.4 \end{array}\right]$$

$$\underbrace{\begin{array}{cc} 0.8 & -0.6 \\ -0.2 & 0.4 \end{array}}_{A^{-1}}$$

2.23) Solve for smallest (in magnitude) eigenvalue matrix A & corresponding eigenvector x by inverse power method using matrix inverse (found using gauss-jordan elimination)

a.) let first element of x be unity component

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

$$A^{-1} = \frac{1}{\underset{5}{|A|}} \begin{bmatrix} 4 & -3 \\ -1 & 2 \end{bmatrix} = \begin{bmatrix} 0.8 & -0.6 \\ -0.2 & 0.4 \end{bmatrix}$$

do power method on inv(A) to get smallest eig

$x^{(0)} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$

$$A^{-1}x^{(0)} = \begin{bmatrix} 0.8 & -0.6 \\ -0.2 & 0.4 \end{bmatrix}\begin{bmatrix} 1 \\ 0 \end{bmatrix} = \begin{bmatrix} 0.8 \\ -0.2 \end{bmatrix}$$

$\lambda = 0.8 \quad x = \begin{bmatrix} 0.7 \\ 0.2 \end{bmatrix}$

$$\begin{bmatrix} 0.8 & -0.6 \\ -0.2 & 0.4 \end{bmatrix}\begin{bmatrix} 1 \\ 4 \end{bmatrix} = \begin{bmatrix} -1.6 \\ 1.4 \end{bmatrix}$$

$\lambda = -1.6 \quad x = \begin{bmatrix} -1.6 \\ 1.4 \end{bmatrix}$

$$\begin{bmatrix} 0.8 & -0.6 \\ -0.2 & 0.4 \end{bmatrix}\begin{bmatrix} 1 \\ -0.875 \end{bmatrix} =$$

⋮

using 500 iterations of my script I got

$$\boxed{\lambda = 1, \ x = \begin{bmatrix} 1 \\ -0.\overline{3} \end{bmatrix}}$$

÷ both elements by $-0.\overline{3}$ → $\begin{bmatrix} -3 \\ 1 \end{bmatrix}$

$x^{(0)} = \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$$\begin{bmatrix} 0.8 & -0.6 \\ -0.2 & 0.4 \end{bmatrix}\begin{bmatrix} 0 \\ 1 \end{bmatrix} = \begin{bmatrix} -0.6 \\ 0.4 \end{bmatrix}$$

$x = \begin{bmatrix} -0.6 \\ 0.4 \end{bmatrix} \quad \lambda = -0.6$

$$\begin{bmatrix} 0.8 & -0.6 \\ -0.2 & 0.4 \end{bmatrix}\begin{bmatrix} -1.5 \\ 1 \end{bmatrix} = \begin{bmatrix} 0.6 \\ -1 \end{bmatrix}$$

$x = \begin{bmatrix} 0.6 \\ -0.1 \end{bmatrix} \quad \lambda = 0.6$

⋮

500 iterations in script

$$\boxed{\lambda = 1, \ x = \begin{bmatrix} -3 \\ 1 \end{bmatrix}}$$

eigenvectors are the same

```python
#Matt McDermott
#Numerical Methods HW #4

import numpy as np

def powerMethod(A,x,runLen,unityComponent):
    # print(A,x,runLen)

    for _ in range(runLen):
        xNext = np.dot(A,x)
        lam = np.amax(abs(xNext))
        if x[unityComponent] != 0:
            x = xNext/(x[unityComponent])

    return x, lam


if __name__ == "__main__":

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

    x0 = np.array([[1],[0],[0]])

    runLen = 500
    unityComponent = 0

    x, lam = powerMethod(A,x0,runLen,unityComponent)
    print("problem 1: x = ", x, '  Lambda = ', lam)

    A = np.array([[0.8,-0.6],
                  [-0.2,0.4]])

    x0 = np.array([[0],[1]])

    runLen = 500
    unityComponent = 1

    x, lam = powerMethod(A,x0,runLen,unityComponent)
    print("problem 2: x = ", x, '  Lambda = ', lam)
```

Anaconda Prompt (anaconda3)

```
(robot) C:\Users\Derm\NumericalMethods>python powerMethod.py
problem 1: x =  [[4.50701864]
 [4.22187616]
 [5.79216113]]   Lambda =  26.105378184124454
problem 2: x =  [[-3.]
 [ 1.]]   Lambda =  3.0000000000000004

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