pr2_i_math548_midterm_Lazizbek

March 19, 2024

1 Math 548, Midterm. Problem 2. LS

Problem 2

Solve the following systems of linear equations with the Jacobi iteration method using the initial guess as [0, 0, 0].

- In each case, will the Jacobi iteration converge to a solution? Give the justification for your answer?
- If yes, find the solutions.

i.

```
[ 2  1  6][x1]   [ 9 ]

[ 8  3  2][x2] = [ 13]

[ 1  5  1][x3]   [ 7 ]
```

ii.

```
[ 8  3  2][x1]   [ 13]

[ 1  5  1][x2] = [ 7 ]

[ 2  1  6][x3]   [ 9 ]
```

2 Finding eigenstuff of a matrix

Source:

[ ]:

```python
import numpy as np
from numpy.linalg import eig
a = np.array([[0, 2],
              [2, 3]])
w,v=eig(a)
print('E-value:', w)
print('E-vector', v)
```

```
E-value: [-1.  4.]
E-vector [[-0.89442719 -0.4472136 ]
```

```
 [ 0.4472136  -0.89442719]]
```

3 Solving Ax=b matrix equation

Source:

#Problem 2. (i)

To check with the actual(real) solution, here, I'm giving the real solution as well:

[ ]:

```
# To solve Ax=b, We start by constructing the arrays for A and b.

A = np.array([[ 2,  1,  6],
              [ 8,  3,  2],
              [ 1,  5,  1]])
b = np.transpose(np.array([ 9,  13,  7]))
# To solve the system we do

x = np.linalg.solve(A,b)
print("Real solution: ")
print(x)
```

```
Real solution:
[1. 1. 1.]
```

Using Jacobi Iteration:

[ ]:

```
A = np.array([[ 2,  1,  6],
              [ 8,  3,  2],
              [ 1,  5,  1]])

L = np.array([[ 0,  0,  0],
              [ 8,  0,  0],
              [ 1,  5,  0]])

D = np.array([[ 2,  0,  0],
              [ 0,  3,  0],
              [ 0,  0,  1]])

U = np.array([[ 0,  1,  6],
              [ 0,  0,  2],
              [ 0,  0,  0]])

D_inverse =  np. linalg. inv(D)
b = np.transpose(np.array([ 9,  13,  7]))
D_inverse_b = np.dot(D_inverse, b)

BJ = np.dot(-D_inverse, L+U)
w,v=eig(BJ)
```

```

print('BJ evalues:', w)

# x0 = np.transpose(np.array([ 0,   0,   0]))
# x = list();
# x.append(x0)

# for i in range(10):
#   x1 = np.dot(BJ, x0)+ D_inverse_b
#   x0 = x1
#   x.append(x1)
# Aproximations = np.array(x)
# print(Aproximations)
# print((2.083**2 + 2.312**2)**(1/2))
```

```
BJ evalues: [-4.16531114+0.j          2.08265557+2.31207612j
2.08265557-2.31207612j]
```

Because the spectral radius of matrix BJ, $P(BJ) = 4.16531114 > 1$*, Jacobi Iteration does NOT converge, so no need to turn on the part of the code for Jacobi iteration carried out.*

```
[1]: # !sudo apt-get install texlive-xetex texlive-fonts-recommended␣
      ↪texlive-plain-generic
```

```
[ ]: # !jupyter nbconvert --to pdf /content/Math548_hw6_Lazizbek.ipynb
```