Ting Fung Lam 2924629375

Feb 21 2020

Programming HW#6 (Due: Feb 28, 11:59 PM):

Implement the algorithm of Gauss-Seidel method. And then run your own code to solve Problem 28 on Page 203.

The starting x is already very close to the true solution. At each iteration, x changes by only a little bit. After 1000 iterations, checking for Ax = b shows that the algorithm gives the correct answer.

[1]:
```
from IPython.display import display, Image
i = Image(filename='image.jpg')
i
```

[1]:

An algorithm for the Gauss-Seidel iteration follows:

input $n, (a_{ij}), (b_i), (x_i), M$
for $k = 1, 2, \ldots, M$ **do**
  for $i = 1, 2, \ldots, n$ **do**

$$x_i \leftarrow \left(b_i - \sum_{\substack{j=1 \\ j \neq i}}^{n} a_{ij} x_j\right) \Big/ a_{ii}$$

  end
output $k, (x_i)$
end

[c]28. Apply the Gauss-Seidel iteration to the system in which

$$A = \begin{bmatrix} 0.96326 & 0.81321 \\ 0.81321 & 0.68654 \end{bmatrix} \qquad b = \begin{bmatrix} 0.88824 \\ 0.74988 \end{bmatrix}$$

Use $(0.33116,\ 0.70000)^T$ as the starting point and explain what happens.

[2]:
```
import numpy as np


def main(A, b, x, M):
    """
    Gauss-Seiden Algorithm
    :param A: Matrix A such that Ax = b
    :param b: Vector b
    :param x: Starting guess of x
    :param M: Number of iterations
    :return:
```

```
    """
    n = np.size(A, 0)
    for k in range(0, M):
        x = (b - (np.dot(A, x) - np.multiply(np.diag(A), x))) / np.diag(A)
    print("k = " + str(k + 1))
    print("x = " + str(x))
    print("Checking Ax = b")
    print(str(np.dot(A, x)))


if __name__ == '__main__':
    A = np.array([[0.96326, 0.81321], [0.81321, 0.68654]])
    b = np.array([0.88824, 0.74988])
    x = np.array([0.33116, 0.7])
    M = 1000
    main(A, b, x, M)
```

```
k = 1000
x = [0.3314485  0.69965827]
Checking Ax = b
[0.88824018 0.74988062]
```

[ ]: