Numerical Analysis Programming Assignment #5 非线性方程求根

陈宇轩 PB16060738

计算结果

Jacobi迭代

迭代步数: 26

数值解: X=[-0.28918999111860827, 0.34542357607815194, -0.7127881887981075, -0.22061338043528445, -0.4303800517315955, 0.15429480803970044, -0.05777671812745311, 0.2010520698807154, 0.2902362576635778]

Gauss-Seidel迭代

迭代步数: 12

数值解：X=[-0.2891803402203373, 0.34547727965684405, -0.7127892125325995, -0.2205921345205877, -0.4303839817707384, 0.15433845237713334, -0.0578011324069756, 0.20105772476105327, 0.2902340082357523]

结果分析

1. 很明显，Gauss-Seidel迭代比Jacobi迭代收敛快得多，所用的迭代步数更少。
2. 由于新数据取代了旧数据，在计算机实现的时候，Gauss-Seidel迭代所需要的存储单元只是Jacobi迭代所需要的一半。

代码实现(Python)

```
'''
@Description: Numerical Caculation lab05
@Author: Yuxuan Chen
@LastEditors: Yuxuan Chen
@Date: 2019-05-05 16:59:56
@LastEditTime: 2019-05-05 18:49:26
'''

import numpy as np
```

```
E = 10e-5


def Jacobi(A, B, X):
    n = 0
    while(True):
        for i in range(len(A)):
            temp = B[i]
            for j in range(len(A[i])):
                if j != i:
                    temp -= A[i][j]*X[j][0]
            X[i][1] = temp/A[i][i]
        n += 1
        if max((np.fabs(X[i][1]-X[i][0]) for i in range(len(A)))) <= E:
            print("Jacobi迭代")
            print("迭代步数:", n)
            print([X[i][1] for i in range(len(A))])
            break
        for i in range(len(A)):
            X[i][0] = X[i][1]

def Gauss_Seidel(A, B, X):
    n = 0
    while(True):
        for i in range(len(A)):
            temp = B[i]
            for j in range(len(A[i])):
                if j != i:
                    temp -= A[i][j]*X[j][1]
            X[i][1] = temp/A[i][i]
        n += 1
        if max((np.fabs(X[i][1]-X[i][0]) for i in range(len(A)))) <= E:
            print("Gauss-Seidel迭代")
            print("迭代步数:", n)
            print([X[i][1] for i in range(len(A))])
            break
        for i in range(len(A)):
            X[i][0] = X[i][1]


def main():
    A = np.array([[31, -13, 0, 0, 0, -10, 0, 0, 0],
                  [-13, 35, -9, 0, -11, 0, 0, 0, 0],
                  [0, -9, 31, -10, 0, 0, 0, 0, 0],
                  [0, 0, -10, 79, -30, 0, 0, 0, -9],
                  [0, 0, 0, -30, 57, -7, 0, -5, 0],
                  [0, 0, 0, 0, -7, 47, -30, 0, 0],
                  [0, 0, 0, 0, 0, -30, 41, 0, 0],
                  [0, 0, 0, 0, -5, 0, 0, 27, -2],
                  [0, 0, 0, -9, 0, 0, 0, -2, 29]])
    B = np.array([-15, 27, -23, 0, -20, 12, -7, 7, 10])
    X = [[0, 0] for p in range(len(B))]
```

```
    Jacobi(A, B, X)
    print("*****************************************")
    X = [[0, 0] for p in range(len(B))]
    Gauss_Seidel(A,B,X)

if __name__ == '__main__':
    main()
```