## Лабораторная работа No6-72 «Сравнение скорости сходимости метода Якоби и Зейделя»

## Цель работы

Реализовать метод Зейделя и Якоби.

## Задание

- 1. Реализовать метод Якоби и Зейделя.
- 2. Ввести критерий остановки итерационного процесс используя равномерную норму.
- 3. Сравнить скорости сходимости методов в зависимости от размера матрицы

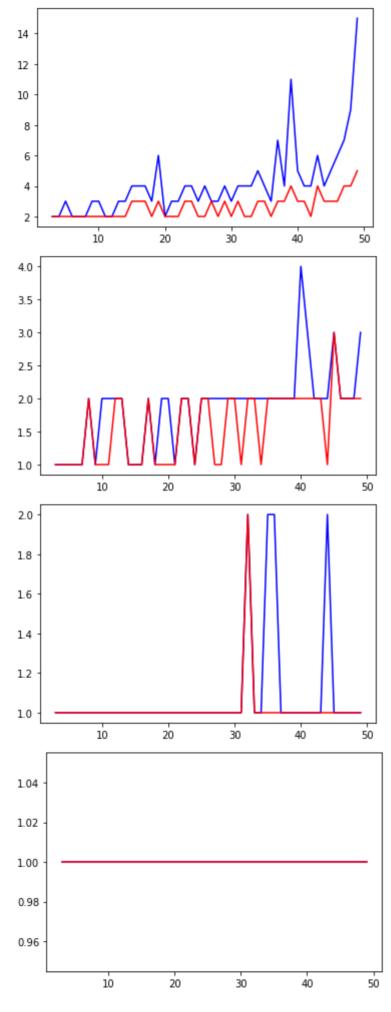
```
import random
from copy import deepcopy
import numpy
import numpy as np
from matplotlib import pyplot as plt
def increase diag(a, pow):
    for i in range(0,len(a)):
        a[i][i] = abs(a[i][i]*(10**pow))
    return a
def norma vec(vec):
   max = -1
    for x in vec:
        if max \le abs(x):
            max = x
    return max
def jacobi(a, b):
    x = np.zeros like(b, dtype=np.double)
    d = np.diagonal(a)
    t = a - np.diag(d)
    iter = 0
    while True:
        x \text{ old} = x.copy()
        x[:] = (b - np.dot(t, x)) / d
        if abs(norma_vec(x) - norma_vec(x_old)) < 10**(-6):
```

```
rter += r
```

```
return x, iter
def seidel(a, b):
   x = np.zeros like(b, dtype=np.double)
   iter = 0
   while True:
       x \text{ old} = x.copy()
        for i in range(a.shape[0]):
           x[i] = (b[i]
                    - np.dot(a[i, :i], x[:i])
                   - np.dot(a[i, (i + 1):], x old[(i + 1):])) \setminus
                   / a[i, i]
        if abs(norma\ vec(x) - norma\ vec(x\ old)) < 10**(-6):
           break
        iter += 1
   return x, iter
def fill matrix(n, a, b):
   m = [[random.uniform(-a, b) for i in range(n)] for j in range(n)]
   return m
def compare jacobi gauss(n, p):
   # matrix = fill_matrix(3, 0, 10)
   \# b = [random.uniform(0, 10) for i in range(3)]
   \# x = [1.0, 1.0, 1.0]
   a = numpy.array(increase diag(fill matrix(n, 0, 10), p))
   b = numpy.array([random.uniform(0, 10) for i in range(n)])
   x jacobi, iter jacobi = jacobi(deepcopy(a), deepcopy(b))
   x seidel, iter seidel = seidel(deepcopy(a), deepcopy(b))
   # print("----")
   # for el in x jacobi:
         print("%.30f " % el)
   # print(x jacobi)
   # print("----")
   # print("----")
   # for el in x seidel:
         print("%.30f " % el)
   # print(x_seidel)
   return iter jacobi, iter seidel
def compare speed():
   for x in range(3, 7):
```

x i = [1]

```
-Δ__J []
        x z = []
        y = []
        for i in range(3, 50):
            j, z = compare jacobi gauss(i, x)
            x j.append(j)
            x_z.append(z)
            y.append(i)
        plt.plot(y, x_j, color="blue")
        plt.plot(y, x_z, color="red")
        plt.show()
def is diag(a):
    sum = []
    flag = True
    for i in range(0, len(a)):
        s = 0.0
        for j in range(0, len(a)):
            if i != j:
                s += abs(a[i][j])
        sum.append(s)
    for i in range(0, len(a)):
        if abs(a[i][i]) < sum[i]:</pre>
            flag = False
            break
    return flag
if __name__ == '__main__':
    compare speed()
 \Box
```



Платные продукты Colab - Отменить подписку

