МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

Факультет прикладной математики и информатики
Кафедра вычислительной математики

Отчет по лабораторной работе №3
по теме: “Метод Гаусса-Зейделя”

Вариант 9

Выполнила
Бодина Виктория Александровна
студентка 3 курса 7а группы

Преподаватель
Будник Анатолий Михайлович

Минск 2024

Введение

Дана СЛАУ вида Ax=b, где A – матрица коэффициентов, x – вектор неизвестных, b – вектор свободных членов.

Итерационный процесс:

Метод Гаусса-Зейделя вычисляет приближения решения $x = (x_0, x_1, \ldots, x_{n-1})$ итерационно, обновляя каждую компоненту $x_i^{(k+1)}$ с использованием значений из текущей или предыдущей итерации:

$$x_i^{(k+1)} = \frac{1}{a_{ii}}(b_i - \sum_{j=0}^{i-1} a_{ij} x_j^{(k+1)} - \sum_{j=i+1}^{n-1} a_{ij} x_j^{(k)}),$$

где k – номер текущей итерации, a_{ii} – диагональный элемент матрицы A, a_{ij} – оставшиеся элементы матрицы A

Критерий остановки:

Итерационный процесс останавливается, если выполняется критерий

$\|x^{(k+1)} - x^{(k)}\| \leq \varepsilon$, где $\varepsilon = 10^{-5}$, то есть $\sqrt{\sum_{i=0}^{n-1} (x_i^{(k+1)} - x_i^{(k)})^2} \leq 10^{-5}$

(евклидова норма)

Условия сходимости:

Метод сходится, если в матрице A выполняется диагональное преобладание:

$|a_{ii}| > \sum_{j \neq i} |a_{ij}|$

```
([[0.7277, -0.0958, 0.0192, 0.0383, 0.1341],
[0.0996, 1.1394, 0.0000, -0.0766, 0.0766],
[0.0575, 0.0000, 0.9154, -0.2681, 0.1532],
[-0.1149, 0.2413, 0.0000, 0.9001, -0.0383],
[0.4788, 0.0000, 0.1724, 0.0192, 1.0724]])
```

i=0:   0.7277 > 0.2874

i=1:   1.1394 > 0.2498
i=2:   0.9154 > 0.4788
i=3:   0.9001 > 0.3945
i=4:   1.0724 > 0.6704

Матрица строго диагонально доминирующая, и метод Гаусса-Зейделя для нее сходится.

Листинг кода

```
import numpy as np

EPS = 1e-5
def gauss_seidel(A, b):
   n = len(A)
   x = np.array(b, dtype=float)
   x_old = np.zeros(n, dtype=float)
   iterations = 0

   while True:
       iterations += 1
       x_old[:] = x

       for i in range(n):
           sigma = sum(A[i][j] * x[j] for j in range(n) if j != i)
           x[i] = (b[i] - sigma) / A[i][i]

       norm = np.linalg.norm(x - x_old)
       print(f"Итерация номер {iterations}: ||x(k+1) - x(k)|| = {norm}")
       if norm < EPS:
           break

   r = A@x - b

   return x, iterations, r

A = np.array([[0.7277, -0.0958, 0.0192, 0.0383, 0.1341],
   [0.0996, 1.1394, 0.0000, -0.0766, 0.0766],
   [0.0575, 0.0000, 0.9154, -0.2681, 0.1532],
   [-0.1149, 0.2413, 0.0000, 0.9001, -0.0383],
   [0.4788, 0.0000, 0.1724, 0.0192, 1.0724]])

b = np.array([1.4363, -1.6431, 6.0514, -3.4508, 5.5727])

x, iterations, r = gauss_seidel(A, b)

print("Решение системы:", np.round(x, 4))
print("Понадобилось итераций:", iterations)
print("Невязка:", r)
```

```
print("Величина невязки:", np.linalg.norm(r))
```

Результаты и их анализ:

Итерация номер 1: ||x(k+1) - x(k)|| = 2.229906153885526
Итерация номер 2: ||x(k+1) - x(k)|| = 0.47031909249603177
Итерация номер 3: ||x(k+1) - x(k)|| = 0.04010424138850462
Итерация номер 4: ||x(k+1) - x(k)|| = 0.004397070248280587
Итерация номер 5: ||x(k+1) - x(k)|| = 0.0004772255604417147
Итерация номер 6: ||x(k+1) - x(k)|| = 5.254238674638902e-05
Итерация номер 7: ||x(k+1) - x(k)|| = 5.7641331293507206e-06
Решение системы: [0.9993 -2.   4.9998 -2.9999 4.0003]
Понадобилось итераций: 7
Невязка: [-3.79568533e-07 -1.45339698e-07 -2.29464790e-07
9.30747182e-08 0.00000000e+00]
Величина невязки: 4.759337060420125e-07

Величина невязки получилась больше, чем в методе Гаусса, так как итерационный процесс останавливается при достижении определенной точности. При необходимости можно задать $\varepsilon < 10^{-5}$ и проделать дополнительные итерации, чтобы достичь той же точности, как в точных методах, при этом потеряв в скорости действия метода.