МИНИСТЕРСТВО ОБРАЗОВАНИЯ РЕСПУБЛИКИ БЕЛАРУСЬ
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ФАКУЛЬТЕТ ПРИКЛАДНОЙ МАТЕМАТИКИ И ИНФОРМАТИКИ

Лабораторная работа №4
Вариант 5
«Метод Гаусса-Зейделя»
Задание № СЛАУ-04

Выполнил: Благодарный Артём Андреевич,
студент 3 курса, 3 группы
Дисциплина: «Численные методы»
Преподаватель: Будник А.М.

Минск, 2024

Постановка задачи

Найти решение системы линейных алгебраических уравнений $Ax = b$ с основной матрицей A системы вида:

$$A = \begin{pmatrix} 0.8894 & 0.0000 & -0.2323 & 0.1634 & 0.2723 \\ -0.0545 & 0.5808 & 0.0000 & -0.1107 & 0.0363 \\ 0.0182 & -0.1634 & 1.0527 & 0.0200 & 0.0635 \\ 0.0545 & 0.0000 & -0.1325 & 1.0527 & 0.0000 \\ 0.0363 & -0.0545 & 0.2632 & -0.0218 & 0.7623 \end{pmatrix}$$

и столбца свободных членов b вида:

$$b = \begin{pmatrix} 4.2326 \\ -4.1037 \\ -2.6935 \\ 1.6916 \\ 3.1908 \end{pmatrix}$$

методом Гаусса-Зейделя.

Алгоритм решения

Для применения метода Зейделя, как и для применения метода простой итерации, система $Ax=f$ должна быть приведена к виду, пригодному для итераций

$$x=Bx+b.$$

Пусть матрица A обладает свойством диагонального преобладания по строкам. Применим тот же способ получения матрицы B и вектора b, который приводит метод простой итерации к методу Якоби. Для этого каждое уравнение системы разрешим относительно неизвестного, стоящего при диагональном элементе:

$$x_1=0\cdot x_1-\frac{a_{12}}{a_{11}}x_2-\ \dots\ -\frac{a_{1n}}{a_{11}}x_n+\frac{f_1}{a_{11}},$$

$$x_2=-\frac{a_{21}}{a_{22}}x_1-0\cdot x_2-\ \dots\ -\frac{a_{2n}}{a_{22}}x_n+\frac{f_2}{a_{22}},$$

$$\dots\dots\dots\dots\dots\dots$$

$$x_n=-\frac{a_{n1}}{a_{nn}}x_1-\frac{a_{n2}}{a_{nn}}x_2-\ \dots\ -0\cdot x_n+\frac{f_n}{a_{nn}}.$$

Получили

$$B=\begin{bmatrix} 0 & -\frac{a_{12}}{a_{11}} & \cdots & -\frac{a_{1n}}{a_{11}} \\ -\frac{a_{21}}{a_{22}} & 0 & \cdots & -\frac{a_{2n}}{a_{22}} \\ \vdots & \vdots & \vdots & \vdots \\ -\frac{a_{n1}}{a_{nn}} & -\frac{a_{n2}}{a_{nn}} & \cdots & 0 \end{bmatrix},\quad b=\begin{bmatrix} \frac{f_1}{a_{11}} \\ \vdots \\ \frac{f_n}{a_{nn}} \end{bmatrix}.$$

Метод Зейделя (2) с такой матрицей B и таким вектором b называется методом Гаусса-Зейделя:

$$x_i^{k+1}=\frac{1}{a_{ii}}\left(f_i-\sum_{j=1}^{i-1}a_{ij}x_j^{k+1}-\sum_{j=i+1}^{n}a_{ij}x_j^{k}\right),\ \ i=1,2,\dots,n,\ \ k=0,1,2,\dots \qquad \text{(ГЗ)}$$

Метод Гаусса-Зейделя является аналогом метода Якоби (файл «МПИ»). В отличие от метода Якоби, каждая координата $(k+1)$-го приближения сразу после получения используется для вычисления следующих координат.

Алгоритм Гаусса-Зейделя:

Выбрать x^0 (начальное приближение)

$k=0, 1, 2, \dots$ (пока не достигнут критерий остановки):

$i=1, 2, \dots, n$:

$t=f_i$ // Начинается вычисление i–й компоненты x_i^{k+1}

$j=1, \dots, i-1$:

$t=t-a_{ij}x_j^{k+1}$ // Вычисления, порождаемые суммой $\sum_{j=1}^{i-1}a_{ij}x_j^{k+1}$

$j=i+1, \dots, n$:

$t=t-a_{ij}x_j^{k}$ // Вычисления, порождаемые суммой $\sum_{j=i}^{n}a_{ij}x_j^{k}$

$$x_i^{k+1}=\frac{t}{a_{ii}}$$

Следствие (из теоремы 2). *Если диагональное преобладание в каждой строке матрицы A строгое, то метод Гаусса-Зейделя сходится.*

Действительно,

$$\|B\|_\infty = \max_{1\le i\le n}\sum_{j=1}^{n}\left|b_{ij}\right| = \max_{1\le i\le n}\sum_{j=1,\, j\ne i}^{n}\left|\frac{a_{ij}}{a_{ii}}\right| = \max_{1\le i\le n}\frac{1}{\left|a_{ii}\right|}\sum_{j=1,\, j\ne i}^{n}\left|a_{ij}\right| < 1.$$

Приведем (без доказательства) оценку скорости сходимости:

Теорема 4. *Пусть диагональное преобладание в каждой строке матрицы A строгое. Обозначим*

$$q = \max_{1\le i\le n}\frac{1}{\left|a_{ii}\right|}\sum_{j=1,\, j\ne i}^{n}\left|a_{ij}\right|.$$

Тогда метод Гаусса-Зейделя сходится со скоростью геометрической прогрессии со знаменателем q и для ошибки имеет место неравенство

$$\|x^\infty - x^k\| \le q^k \|x^\infty - x^0\|.$$

Листинг

```
import numpy as np
from functools import reduce
A = np.array([[0.8894, 0.0000, -0.2323, 0.1634, 0.2723],
              [-0.0545, 0.5808, 0.0000, -0.1107, 0.0363],
              [0.0182, -0.1634, 1.0527, 0.0200, 0.0635],
              [0.0545, 0.0000, -0.1325, 1.0527, 0.0000],
              [0.0363, -0.0545, 0.2632, -0.0218, 0.7623]])
b = np.array([4.2326, -4.1037, -2.6935, 1.6916, 3.1908])


def gauss_seidel_method(A, b, max_iter=100, epsilon=1e-5, iterations=False):
    n = len(A)
    x = [b[i] / A[i, i] for i in range(n)]
    k = 0
    for iter in range(max_iter):
        k += 1
        x_old = x.copy()
        for i in range(n):
            x[i] = (b[i] - x @ A[i] + x[i] * A[i, i]) / A[i, i]
        if all(abs(x[i]-x_old[i]) <= epsilon for i in range(n)):
            break
    if iterations:
        return (x, k)
    return x

def condition_method(A):
    for i in range(len(A)):
        if 2 * abs(A[i, i]) <= reduce(lambda x, y: x + abs(y), A[i], 0):
            return False
    return True

def residual(A, b, x):
    return b - A @ x

print(f"Are the convergence conditions satisfied: {condition_method(A)}")

eps_values = 10. ** -np.arange(1, 17)

for eps in eps_values:
    sol, k = gauss_seidel_method(A, b, epsilon=eps, iterations=True)
    res = residual(A, b, sol)
    min_res = min(np.where(res == 0, 1, res))
    scientific = f"{min_res:.1e}"
    exponent = int(scientific.split('e')[1])
    print(eps, k, 10. ** exponent)

print(gauss_seidel_method(A, b))
```

Результаты и их анализ

На описанных данных алгоритм выдаёт следующее решение (при eps=1e-5):

$$x = \begin{pmatrix} 1.9996 \\ -6.9999 \\ -4.0003 \\ 0.9999 \\ 4.9999 \end{pmatrix}$$

Так как матрица A имеет строгое диагональное преобладание, то метод Гаусса-Зейделя сходится.

Начальное приближение: `x = [b[i] / A[i, i] for i in range(n)] # элементы матрицы b делённые на диагональные элементы матрицы A`

Вектор невязки Метод Гаусса:

$$r = \begin{bmatrix} 0.0000 \\ 0.0000 \\ 0.0000 \\ 2.22 \times 10^{-16} \\ -4.44 \times 10^{-16} \end{bmatrix}$$

Невязка метода Гаусса-Зейделя при eps=1e-14:

```
r=[-8.88178420e-16
   0.00000000e+00
   0.00000000e+00
  -2.22044605e-16
  -4.44089210e-16]
```

Зависимость вектора невязки от заданного eps и количества шагов алгоритма:

```
eps: 0.1, steps: 3, min exp component r: 0.001
eps: 0.01, steps: 4, min exp component r: 1e-05
eps: 0.001, steps: 5, min exp component r: 1e-06
eps: 0.0001, steps: 5, min exp component r: 1e-06
eps: 1e-05, steps: 6, min exp component r: 1e-07
eps: 1e-06, steps: 7, min exp component r: 1e-08
eps: 1e-07, steps: 8, min exp component r: 1e-10
eps: 1e-08, steps: 9, min exp component r: 1e-11
eps: 1e-09, steps: 9, min exp component r: 1e-11
eps: 1e-10, steps: 10, min exp component r: 1e-12
eps: 1e-11, steps: 11, min exp component r: 1e-14
eps: 1e-12, steps: 12, min exp component r: 1e-15
eps: 1e-13, steps: 12, min exp component r: 1e-15
eps: 1e-14, steps: 13, min exp component r: 1e-16
eps: 1e-15, steps: 14, min exp component r: 1e-16
eps: 1e-16, steps: 100, min exp component r: 1e-16
```

Вывод по полученным результатам

Невязка для **метода Гаусса-Зейделя** при различных значениях `eps` показывает, насколько малы становятся остатки по мере сходимости алгоритма. Это свидетельствует о том, что по мере уменьшения значения `eps` решение становится более точным, и остатки приближаются к машинной точности (порядка `10^-16`), а при `eps=1e-14` невязка метода Гаусса-Зейделя примерно равна невязки метода Гаусса. Даже при очень малых значениях `eps` (например, `1e-16`), получить абсолютно нулевую невязку невозможно из-за **машинной точности**. В вычислениях с плавающей точкой числа представляются с конечной точностью, и операции с ними всегда приводят к некоторым погрешностям округления.

Сравнения методов:

- **Метод Якоби**: Якоби может сходиться медленно, особенно для матриц, у которых слабое диагональное преобладание. Метод Якоби обновляет все переменные одновременно на каждой итерации, что может сделать его сходимость менее эффективной по сравнению с Гауссом-Зейделем.
- **Метод Гаусса-Зейделя**: Этот метод, как правило, работает быстрее, чем метод Гаусса и метод Якоби, потому что он обновляет решение поэтапно и использует обновленные значения сразу на следующих шагах. Это позволяет достичь более быстрой сходимости, как показано в оценке невязки.