

МОСКОВСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИМЕНИ М. В. ЛОМОНОСОВА
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ МАТЕМАТИКИ И КИБЕРНЕТИКИ

**Компьютерный практикум по курсе
«ВВЕДЕНИЕ В ЧИСЛЕННЫЕ МЕТОДЫ»**

**ЗАДАНИЕ №1
Подвариант №2**

ИТЕРАЦИОННЫЕ МЕТОДЫ РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ
АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ (НА ПРИМЕРЕ МЕТОДОВ
ЗЕЙДЕЛЯ И ВЕРХНЕЙ РЕЛАКСАЦИИ)

ОТЧЕТ
о выполненном задании
студента 202 учебной группы факультета ВМК МГУ
Кузнецова Михаила Константиновича

Москва
2020

Содержание

Цель работы	2
Постановка задачи	2
Задачи практической работы	2
Описание метода решения	3
Описание программы	3
Код программы	4
Тестирование программы	5
Выводы	8

Цель работы

Изучить классические итерационные методы (Зейделя и верхней релаксации), используемые для численного решения систем линейных алгебраических уравнений; изучить скорость сходимости этих методов в зависимости от выбора итерационного параметра.

Постановка задачи

Дана система уравнений $Ax=f$ порядка $n \times n$ с невырожденной матрицей A . Написать программу численного решения данной системы линейных алгебраических уравнений (n – параметр программы), использующую численный алгоритм итерационного метода Зейделя:

$$(D + A^{(-)}) (x^{k+1} - x^k) + Ax^k = f$$

где D , $A^{(-)}$ – соответственно диагональная и нижняя треугольные матрицы, k – номер текущей итерации; в случае использования итерационного метода верхней релаксации итерационный процесс имеет следующий вид:

$$(D + \omega A^{(-)}) \frac{x^{k+1} - x^k}{\omega} + Ax^k = f$$

где ω – итерационный параметр (при $\omega = 1$ метод верхней релаксации переходит в метод Зейделя). Предусмотреть возможность задания элементов матрицы системы и ее правой части как во входном файле данных, так и путем задания специальных формул.

Задачи практической работы

1. Решить заданную СЛАУ итерационным методом Зейделя (или более общим методом верхней релаксации);
2. Разработать критерий остановки итерационного процесса, гарантирующий получение приближенного решения исходной системы СЛАУ с заданной точностью;
3. Изучить скорость сходимости итераций к точному решению задачи (при использовании итерационного метода верхней релаксации провести эксперименты с различными значениями итерационного параметра (в случае симметрической положительно определенной матрицы системы известно, что для сходимости итераций следует выбирать $0 < \omega < 2$; при $\omega = 1$ метод верхней релаксации совпадает с методом Зейделя));
4. Правильность решения СЛАУ подтвердить системой тестов.

Описание метода решения

Рассмотрим произвольную квадратную матрицу:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1m} \\ a_{21} & a_{22} & \dots & a_{2m} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mm} \end{pmatrix}$$

Разложим её на сумму трех матриц $A = D + T_H + T_B$, где D - диагональная часть матрицы A , которая содержит элементы a_{ii} , стоящие на главной диагонали, T_H - нижняя треугольная матрица, T_B - верхняя треугольная матрица.

Рассмотрим рекуррентное соотношение:

$$(D + \omega T_H) \frac{(\mathbf{x}_{k+1} - \mathbf{x}_k)}{\omega} + A\mathbf{x}_k = \mathbf{f} \quad B = D + \omega T_H, \tau = \omega > 0$$

Это можно переписать в виде:

$$\left(\frac{1}{\omega}D + T_H\right) (\mathbf{x}_{k+1} - \mathbf{x}_k) + A\mathbf{x}_k = \mathbf{f}$$

$$\begin{aligned} \left(\frac{1}{\omega}D + T_H\right) \mathbf{x}_{k+1} + \left[\left(1 - \frac{1}{\omega}\right)D + T_B\right] \mathbf{x}_k &= \mathbf{f} \\ x_i^{k+1} = x_i^k + \frac{\omega}{a_{ii}} \left(f_i - \sum_{j=1}^{i-1} a_{ij}x_j^{k+1} - \sum_{j=i}^n a_{ij}x_j^k\right), i = 1, \dots, n \end{aligned}$$

В качестве критерия остановки было выбрано следующее условие: $\|Ax^k - f\| < \epsilon$, где ϵ - заданная погрешность

Описание программы

- Вводится откуда брать данные: 1- с файла, 2- по уравнению
- Вводится ω и eps

Программа домнажает слева на транспонированную матрицу и преобразует последующую так, чтобы на главной диагонали не было 0.

Код программы

```
import numpy as np

def produce_A_f_eq(n=50, m=15):
    A = np.zeros((n, n), dtype=np.float32)
    f = np.zeros((n, 1), dtype=np.float32)
    for i in np.arange(n):
        for j in np.arange(n):
            if i != j:
                A[i][j] = (i + j + 2) / (m + n)
            else:
                A[i][j] = n + m ** 2 + (j + 1) / m + (i + 1) / n
    for i in np.arange(n):
        f[i, 0] = m * n - (i + 1) ** 3
    Af = np.concatenate((A, f), axis=1)
    return A, f, Af

def produce_A_f_path(path):
    Af = np.loadtxt(path)
    return Af[:, :-1], Af[:, -1:], Af

def upper_rel_m(A, f, w=0.01, eps=0.01):
    f = A.T @ f
    A = A.T @ A

    for i in range(A.shape[0]):
        if (A[i, i] == 0):
            j = 0
            while (A[j, i] == 0):
                j += 1
            A[i] += A[j]
            f[i] += f[j]

    it = 0
    xk = np.zeros((A.shape[0], 1))
    xk_1 = np.zeros((A.shape[0], 1))
    err = np.max(np.absolute((A @ xk) - f))
    while (err > eps):
        it += 1
        for i in range(A.shape[0]):
            sum1 = 0.0
            sum2 = 0.0
            for j in range(i):
                sum1 += A[i, j] * xk_1[j]
            for j in range(i, A.shape[0]):
                sum2 += A[i, j] * xk[j]
            xk_1[i] = xk[i] + w/A[i, i] * (f[i] - sum1 - sum2)

        xk = xk_1
        err = np.max(np.absolute((A @ xk) - f))
    return it, xk

#start of the program
inc = int(input())
if inc == 1:
    A, f, Af = produce_A_f_path(input())
elif inc == 2:
    A, f, Af = produce_A_f_eq(n=int(input()), m=int(input()))
w = float(input())
eps = float(input())
count_iter, x = upper_rel_m(A, f, w=w, eps=eps)
print(x, count_iter)
```

Тестирование программы

Приложение 1. Вариант 13

1. Матрица $A = \left[\begin{array}{cccc|c} 3 & -2 & 2 & -2 & 8 \\ 2 & -1 & 2 & 0 & 4 \\ 2 & 1 & 4 & 8 & -1 \\ 1 & 3 & -6 & 2 & 3 \end{array} \right]$

Решение системы методом верхней релаксации: (1.999340, -2.998590, -1.499171, 0.499712)

Решение системы numpy : (2, -3, -1.5, 0.5)

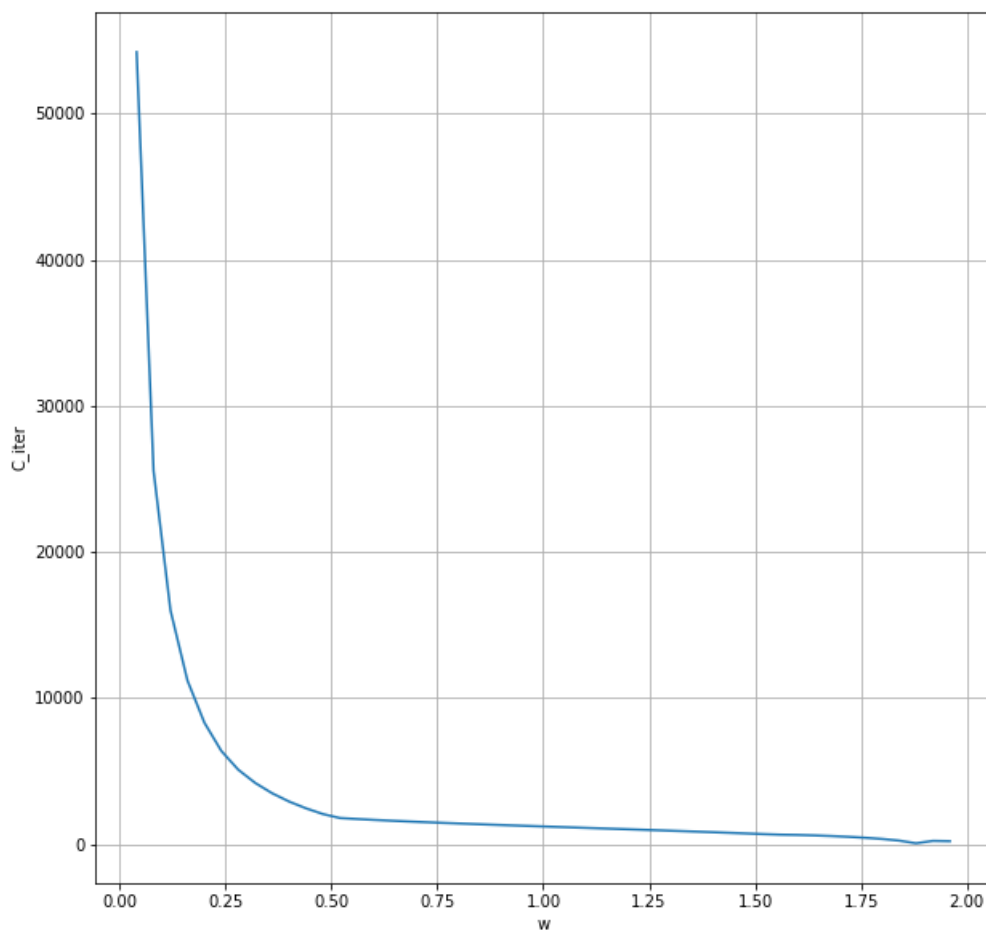


Рис. 1:

Наилучшая сходимость наблюдается при $w = 0.92$

2. Матрица $A = \left[\begin{array}{cccc|c} 2 & 3 & 1 & 2 & 4 \\ 4 & 3 & 1 & 1 & 5 \\ 1 & -7 & -1 & -2 & 7 \\ 2 & 5 & 1 & 1 & 1 \end{array} \right]$

Решение системы методом верхней релаксации: $(-2.993370, -4.995409, 38.945714, -6.985508)$

Решение системы пипру : $(-3, -5, 39, -7)$

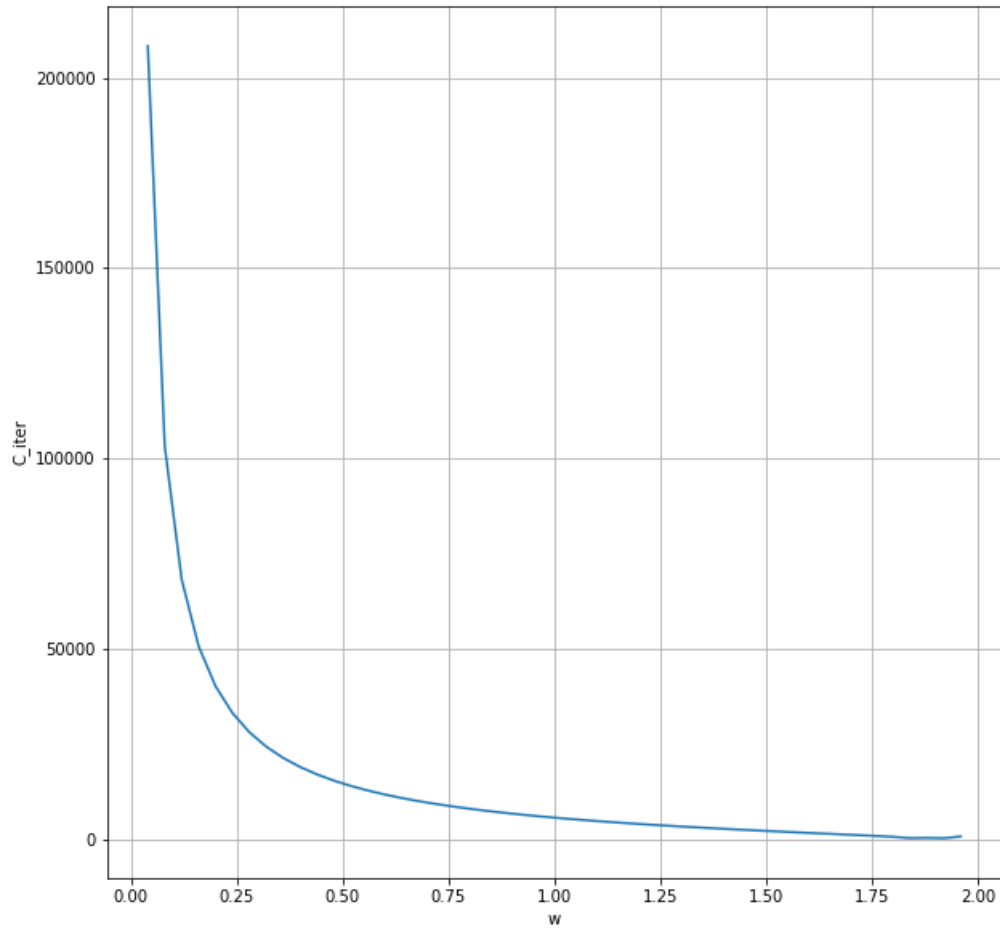


Рис. 2:

Наилучшая сходимость наблюдается при $w = 1.88$

3. Матрица $A = \left[\begin{array}{cccc|c} 2 & 3 & 1 & 2 & 4 \\ 4 & 3 & 1 & 1 & 5 \\ 1 & -7 & -1 & -2 & 7 \\ 2 & 5 & 1 & 1 & 1 \end{array} \right]$

$$|A| = 0$$

4. Матрица $A = A_{ij} = \begin{cases} \frac{i+j}{m+n}, & i \neq j \\ n + m^2 + \frac{j}{m} + \frac{i}{n}, & i = j \end{cases}$, где $i, j = 1, \dots, n$

Решение системы методом верхней релаксации: (14.57970047, 14.82029091, 15.01717258, 15.14857629, 15.19273856, 15.12792845, 14.93242055, 14.58451253, 14.06252053, 13.3447701, 12.4096211, 11.23543332, 9.80059393, 8.0835019, 6.06258159, 3.71626915, 1.0230146, -2.03870211, -5.49040015, -9.35356123, -13.64967248, -18.40017947, -23.62653873, -29.35016356, -35.59247564, -42.37482526, -49.71861896, -57.64521467, -66.17592088, -75.33207225, -85.13499292, -95.60589056, -106.76614769, -118.6369238, -131.23947449, -144.59507712, -158.7248453, -173.65000072, -189.39172144, -205.97114867, -223.40936385, -241.72756469, -260.94684065, -281.08817396, -302.17272935, -324.22150041, -347.25550291, -371.29581885, -396.36338209, -422.47917118)

Решение системы нутру : отличается в 5 и далее знаках после запятой
Наилучшая сходимость наблюдается при $\omega = 0.92$

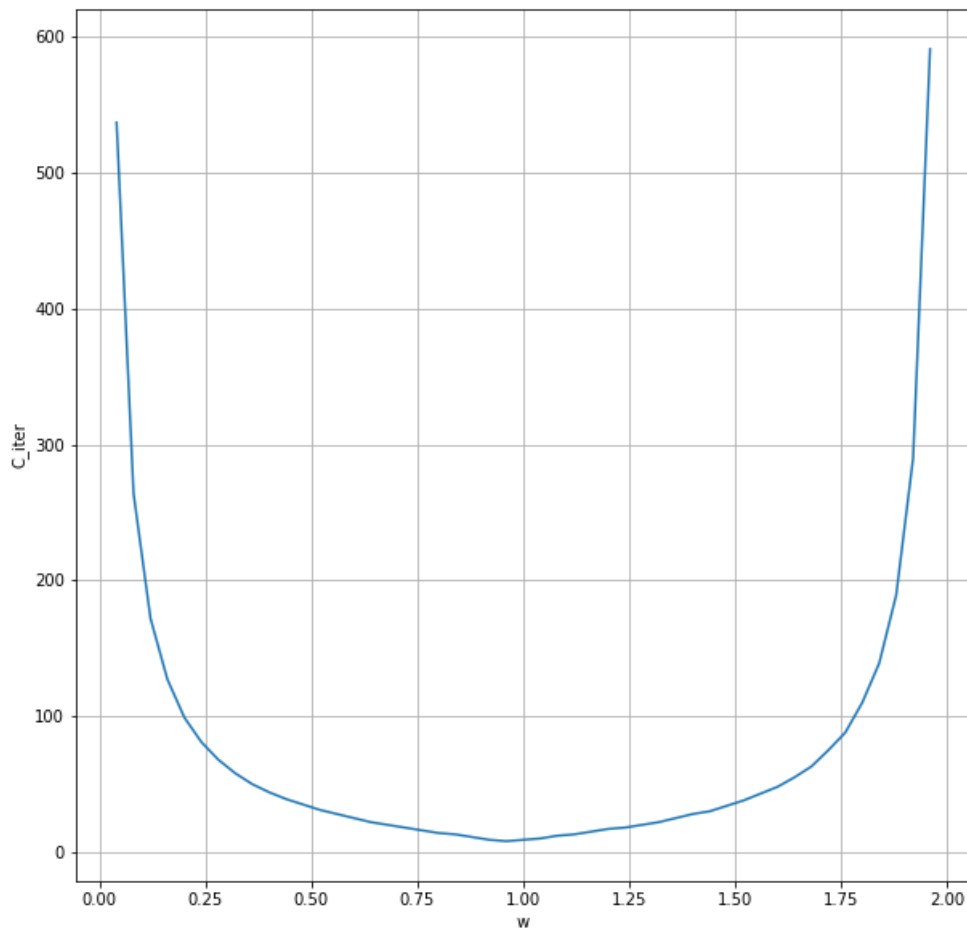


Рис. 3:

Выводы

Изучен метод верхней релаксации и его частный случай - метод Зейделя. У метода есть свои достоинства, такие как: легкая реализация на ЭВМ и сходжение метода за малое число итераций при применении к "простым" матрицам, и недостатки: скорость сходимости метода для каждой системы сильно зависит от выбора итерационного параметра, большие ограничения, накладываемых на матрицу.