МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное автономное образовательное учреждение высшего образования

«Дальневосточный федеральный университет»
(ДВФУ)

Институт математики и компьютерных технологий

Департамент математического и компьютерного моделирования

Курсовой проект по дисциплине
«Вычислительная математика»

на тему «Методы релаксации»

Направление подготовки
01.03.02 «Прикладная математика и информатика»

Выполнил студент
гр. Б9119-01.03.02систпро
Пищиков А.А. ______
(ФИО) *(подпись)*

Проверил доцент, к.ф.-м.н.
Колобов А.Г. ______
(ФИО) *(подпись)*

« 8 » июня 20 22 г.

г. Владивосток
2022

Содержание

Введение

Объектом исследования являются численные методы решения задач линейной алгебры, а также программное обеспечение, реализующее эти методы. Цель работы – ознакомиться с численными методами решения систем линейных алгебраических уравнений, нахождения обратных матриц, решения проблемы собственных значений, решить предложенные типовые задачи, сформулировать выводы по полученным решениям, отметить достоинства и недостатки методов, сравнить удобство использования и эффективность работы каждой использованной программы, приобрести практические навыки и компетенции, а также опыт самостоятельной профессиональной деятельности, а именно:

• создать алгоритм решения поставленной задачи и реализовать его, протестировать программы;

• освоить теорию вычислительного эксперимента; современных компьютерных технологий;

• приобрести навыки представления итогов проделанной работы в виде отчета, оформленного в соответствии с имеющимися требованиями, с привлечением современных средств редактирования и печати.

Работа над курсовым проектом предполагает выполнение следующих задач:

• дальнейшее углубление теоретических знаний обучающихся и их систематизацию;

• получение и развитие прикладных умений и практических навыков по направлению подготовки;

• овладение методикой решения конкретных задач;

• развитие навыков самостоятельной работы;

• развитие навыков обработки полученных результатов, анализа и осмысления их с учетом имеющихся литературных данных;

• приобретение навыков оформления описаний программного продукта;

• повышение общей и профессиональной эрудиции.

Изученный студентом в ходе работы материал должен способствовать повышению его качества знаний, закреплению полученных навыков и уверенности в выборе путей будущего развития своих профессиональных способностей.

Основная часть

Постановка задачи

Теперь рассмотрим следующую задачу: пусть дана система линейных алгебраических уравнений(СЛАУ):

$$Ax = f,$$

где A – квадратная матрица размерности n, f – вектор правой части, x – вектор неизвестных. Необходимо найти решение данной системы, т.е. вектор x, методом релаксации.

Описание метода

Метод релаксации – итерационный метод решения СЛАУ. Суть данного метода заключается в том, чтобы вычислять элементы вектора приближения по следующей формуле:

$$x_i^{(k+1)} = \frac{\omega\left(f_i - \sum\limits_{j=1}^{i-1} a_{ij}x_j^{(k+1)} - \sum\limits_{j=i+1}^{n} a_{ij}x_j^{(k)}\right)}{a_{ii}} + (1-\omega)x_i^{(k)},$$

где ω – релаксационный параметр, до тех пор, пока не будет достигнута требуемая точность. После достижения заданной точности, будет получен вектор, являющий приближённым решением СЛАУ. Известно, что данный метод сходится для релаксационного параметра $\omega \in (0, 2)$, если матрица симметрична и положительно определена.

В зависимости от значения ω выделяют несколько разновидностей этого метода. При параметре $\omega = 1$ данный метод называется методом Зейделя, при $\omega < 1$ – методом нижней релаксации, а при $\omega > 1$ – методом верхней релаксации.

Вычислительный эксперимент

В рамках данного раздела будет рассмотрены СЛАУ, для которых будет найдено численное решение методом релаксации с разными значениями релаксационного параметра ω и с разной точностью ε. Количество итераций решения будет указано в таблице. В качестве первого приближения для всех систем был выбран нулевой вектор. Точность будет определятся по длине вектора невязки.

Матрица размерности 2

Дана система:

$$\begin{pmatrix} 3 & 4 \\ 4 & 7 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 7 \\ 6 \end{pmatrix}.$$

Решение, полученное аналитически с помощью точного метода решения СЛАУ:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 5 \\ -2 \end{pmatrix}.$$

Решим эту систему численно методом релаксации. Для разных ω и ε получим следующее количество итераций:

	$\varepsilon = 0.1$	$\varepsilon = 0.01$	$\varepsilon = 0.001$
$\omega = 0.5$	38	65	92
$\omega = 1$	12	21	29
$\omega = 1.5$	6	10	14

Таблица 1: Количество итераций при разных значениях параметра ω и точности ε для системы размерности 2.

При $\varepsilon = 0.001$ и $\omega = 1.5$ численно было получено следующее решение СЛАУ:

$$\begin{pmatrix} x_1 \\ x_2 \end{pmatrix} = \begin{pmatrix} 4.999995 \\ -1.999944 \end{pmatrix}.$$

Матрица размерности 3

Дана система:

$$\begin{pmatrix} 5 & 4 & 2 \\ 4 & 8 & 4 \\ 2 & 4 & 6 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 11 \\ 4 \\ -14 \end{pmatrix}.$$

Решение, полученное аналитически с помощью точного метода решения СЛАУ:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ -4 \end{pmatrix}.$$

Решим эту систему численно методом релаксации. Для разных ω и ε получим следующее количество итераций:

	$\varepsilon = 0.1$	$\varepsilon = 0.01$	$\varepsilon = 0.001$
$\omega = 0.5$	19	30	41
$\omega = 1$	8	10	13
$\omega = 1.5$	11	16	20

Таблица 2: Количество итераций при разных значениях параметра ω и точности ε для системы размерности 3.

При $\varepsilon = 0.001$ и $\omega = 1$ численно было получено следующее решение СЛАУ:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \end{pmatrix} = \begin{pmatrix} 3.000225 \\ 0.999884 \\ -3.999997 \end{pmatrix}.$$

Матрица размерности 4

Дана система:

$$\begin{pmatrix} 8 & 1 & 3 & 1 \\ 1 & 5 & 0 & 2 \\ 3 & 0 & 6 & 4 \\ 1 & 2 & 4 & 5 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 29 \\ 106 \\ -18 \\ 34 \end{pmatrix};$$

Решение, полученное аналитически с помощью точного метода решения СЛАУ:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4 \\ 18 \\ -9 \\ 6 \end{pmatrix}.$$

Решим эту систему численно методом релаксации. Для разных ω и ε получим следующее количество итераций:

	$\varepsilon = 0.1$	$\varepsilon = 0.01$	$\varepsilon = 0.001$
$\omega = 0.5$	50	82	113
$\omega = 1$	19	29	39
$\omega = 1.5$	11	14	18

Таблица 3: Количество итераций при разных значениях параметра ω и точности ε для системы размерности 4.

При $\varepsilon = 0.001$ и $\omega = 1.5$ численно было получено следующее решение СЛАУ:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} 4.000116 \\ 17.999949 \\ -9.000097 \\ 6.000033 \end{pmatrix}.$$

Матрица размерности 5

Дана система:

$$\begin{pmatrix} 10 & 1 & 5 & 1 & 1 \\ 1 & 7 & 2 & 13 & 2 \\ 5 & 2 & 5 & 5 & 1 \\ 1 & 13 & 5 & 37 & 0 \\ 1 & 2 & 1 & 0 & 2 \end{pmatrix} \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} 0 \\ 55 \\ 8 \\ 17 \\ 57 \end{pmatrix};$$

Решение, полученное аналитически с помощью точного метода решения СЛАУ:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} -2 \\ 10 \\ 0 \\ -3 \\ 13 \end{pmatrix}.$$

Решим эту систему численно методом релаксации. Для разных ω и ε получим следующее количество итераций:

	$\varepsilon = 0.1$	$\varepsilon = 0.01$	$\varepsilon = 0.001$
$\omega = 0.5$	230	477	725
$\omega = 1$	53	134	216
$\omega = 1.5$	41	66	90

Таблица 4: Количество итераций при разных значениях параметра ω и точности ε для системы размерности 5.

При $\varepsilon = 0.001$ и $\omega = 1.5$ численно было получено следующее решение СЛАУ:

$$\begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \end{pmatrix} = \begin{pmatrix} -2.000250 \\ 10.001379 \\ 0.000605 \\ -3.000542 \\ 12.998493 \end{pmatrix}.$$

Заключение

В рамках курсового проекта был реализован и исследован на матрицах разной размерности численный метод решения СЛАУ, а конкретно – метод релаксации. Из полученных результатов можно сделать вывод о том, что релаксационный параметр ω может влиять на скорость сходимости метода как положительно, так и отрицательно, причём значения релаксационного параметра могут оказывать большее влияние на скорость сходимости, чем изменения точности вычисления.

В результате работы над курсовым проектом приобрел практические навыки владения:

• современными численными методами решения задач линейной алгебры;

• основами алгоритмизации для численного решения задач линейной алгебры на одном из языков программирования;

• инструментальными средствами, поддерживающими разработку программного обеспечения для численного решения задач линейной алгебры;

а также навыками представления итогов проделанной работы в виде отчета, оформленного в соответствии с имеющимися требованиями, с привлечением современных средств редактирования и печати.

Приложение

Реализация численного алгоритма.

```
extern crate nalgebra as na;

use na::{DMatrix, DVector};

pub fn sor(a : DMatrix<f32>, f : DVector<f32>, n : usize,
        omega: f32, epsilon : f32) -> DVector<f32> {
    let mut xs = DVector::from_element(n, 0.0);
    let mut xs_prev = xs.clone();

    let mut it_counter = 0;

    loop {
        it_counter += 1;
        for i in 0..n {
            let mut sum1 = 0.0;
            let mut sum2 = 0.0;
            for k in 0..i {
                sum1 += a[(i, k)] * xs[k];
            }
            for k in (i + 1)..n {
                sum2 += a[(i, k)] * xs_prev[k];
            }
            xs[i] = omega * (f[i] - sum1 - sum2) / a[(i, i)]
                + (1.0 - omega) * xs_prev[i];
        }

        xs_prev = xs.clone();

        if (&a * &xs - &f).norm() < epsilon {
            break;
        }
    }

    println!("x␣=␣{}", xs);

    println!("omega␣=␣{},␣epsilon␣=␣{},␣iterations␣=␣{}\n",
        omega, epsilon, it_counter);

    return xs;
}
```

Код программы для решения СЛАУ размерности 2 методом релаксации для разных значений ω и ε.

```
extern crate nalgebra as na;
use na::{DMatrix, DVector};

use crate::sor::sor;

pub fn sor2x2() {
    let n = 2;
    let matrix_row = [3.0, 4.0,
                      4.0, 7.0];

    let vector_row = [7.0, 6.0];


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.1;

    let omega = 0.5;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);

    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.01;

    let omega = 0.5;

    sor(a, f, n, omega, epsilon);
```

```


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);

    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.001;

    let omega = 0.5;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);
}
```

Код программы для решения СЛАУ размерности 3 методом релаксации для разных значений ω и ε.

```
extern crate nalgebra as na;
use na::{DMatrix, DVector};

use crate::sor::sor;

pub fn sor3x3() {
    let n = 3;
    let matrix_row = [5.0, 4.0, 2.0,
                      4.0, 8.0, 4.0,
                      2.0, 4.0, 6.0];

    let vector_row = [11.0, 4.0, -14.0];


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.1;

    let omega = 0.5;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);

    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.01;

    let omega = 0.5;

```

```
    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);

    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.001;

    let omega = 0.5;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);
}
```

Код программы для решения СЛАУ размерности 4 методом релаксации для разных значений ω и ε.

```
extern crate nalgebra as na;
use na::{DMatrix, DVector};

use crate::sor::sor;

pub fn sor4x4() {
    let n = 4;
    let matrix_row = [8.0, 1.0, 3.0, 1.0,
                      1.0, 5.0, 0.0, 2.0,
                      3.0, 0.0, 6.0, 4.0,
                      1.0, 2.0, 4.0, 5.0];

    let vector_row = [29.0, 106.0, -18.0, 34.0];


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.1;

    let omega = 0.5;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);

    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.01;

    let omega = 0.5;
```

```

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);

    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.001;

    let omega = 0.5;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);
}
```

Код программы для решения СЛАУ размерности 5 методом релаксации для разных значений ω и ε.

```
extern crate nalgebra as na;
use na::{DMatrix, DVector};

use crate::sor::sor;

pub fn sor5x5() {
    let n = 5;
    let matrix_row = [10.0, 1.0, 5.0, 1.0, 1.0,
                      1.0, 7.0, 2.0, 13.0, 2.0,
                      5.0, 2.0, 5.0, 5.0, 1.0,
                      1.0, 13.0, 5.0, 37.0, 0.0,
                      1.0, 2.0, 1.0, 0.0, 2.0];

    let vector_row = [0.0, 55.0, 8.0, 17.0, 44.0];


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.1;

    let omega = 0.5;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);

    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.01;

```

```
    let omega = 0.5;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);

    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let epsilon : f32 = 0.001;

    let omega = 0.5;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.0;

    sor(a, f, n, omega, epsilon);


    let a : DMatrix<f32> = DMatrix::from_row_slice(n, n, &matrix_row);
    let f = DVector::from_row_slice(&vector_row);

    let omega = 1.5;

    sor(a, f, n, omega, epsilon);
}
```

Код программы, запускающей вычислительный эксперемент.

```rust
mod sor;

mod sor2x2;
use crate::sor2x2::sor2x2;

mod sor3x3;
use crate::sor3x3::sor3x3;

mod sor4x4;
use crate::sor4x4::sor4x4;

mod sor5x5;
use crate::sor5x5::sor5x5;

fn main() {
    sor2x2();
    sor3x3();
    sor4x4();
    sor5x5();
}
```