МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
Федеральное государственное автономное образовательное учреждение высшего образования
«Дальневосточный федеральный университет»
(ДВФУ)

ИНСТИТУТ МАТЕМАТИКИ И КОМПЬЮТЕРНЫХ ТЕХНОЛОГИЙ (ШКОЛА)

Департамент математического и компьютерного моделирования

Курсовой проект

по дисциплине «Вычислительная математика»

Направление подготовки
01.03.02 «Прикладная математика и информатика»

Выполнил студент гр.
Б9121-01.03.02сп
Держапольский Ю.В. ________
(Ф.И.О.) *(подпись)*

Проверил доцент, к.ф-м.н.
Колобов А.Г. ________
(Ф.И.О.) *(подпись)*

«____» ________ 2024 г.

г. Владивосток

2024

Содержание

1. Введение

Объектом исследования является точный численный метод решения системы линейных алгебраических уравнений, метод отражений, а также программное обеспечение, реализующее этот метод.

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.

2. Основная часть

2.1. Постановка задачи

Пусть дана система линейных алгебраических уравнений. Представим её таким образом: известные квадратная матрица A, вектор f и неизвестный вектор x размерностями n. Нужно найти вектор x в матричном уравнении $Ax = f$, используя метод отражений.

2.2. Описание алгоритма

Метод отражений состоит в выполнении $(n-1)$ шагов, в результате чего матрица A приводится к верхней треугольной форме и в последующей решении системы с такой матрицей.

Для этого на каждом шаге k будем находить вектор нормали p, характеризующий ортогональную матрицу отражения P, которая обнулит все поддиагональные элементы k-того столбца.

Обозначим вектор нормали и матрицу на шаге k: $p^{(k)}$, $A_k = \left(a_{ij}^{(k)}\right)$, тогда

$$p_k^{(k)} = a_{kk}^{(k-1)} + \sigma_k \sqrt{\sum_{l=k}^{n} \left(a_{lk}^{(k-1)}\right)^2}, \quad \sigma_k = \begin{cases} 1, & a_{kk}^{(k-1)} \geq 0, \\ -1, & a_{kk}^{(k-1)} \leq 0, \end{cases}$$

$$p_i^{(k)} = 0, \quad i = 0, \ldots, k-1; \qquad p_i^{(k)} = a_{ik}^{(k-1)}, \quad i = k+1, \ldots, n;$$

Определение матрицы $P_k = I - \dfrac{2p^{(k)}\left(p^{(k)}\right)^*}{\left(p^{(k)}, p^{(k)}\right)}$. Будем применять данную матрицу с обоих сторон уравнения. Связь шагов $A_k = P_k A_{k-1},\ f^{(k)} = P_k f^{(k-1)}$.

Запишем явные формулы элементов матрицы A_k и вектора $f^{(k)}$:

$$a_{kk}^{(k)} = -\sigma_k \sqrt{\sum_{l=k}^{n} \left(a_{lk}^{(k-1)}\right)^2}, \quad a_{ij}^{(k)} = a_{ij}^{(k-1)} - 2p_i^{(k)} \frac{\sum\limits_{l=k}^{n} \left(p_l^{(k)} a_{lj}^{(k-1)}\right)}{\sum\limits_{l=k}^{n} \left(p_l^{(k)}\right)^2},$$

$$f_i^{(k)} = f_i^{(k-1)} - 2p_i^{(k)} \frac{\sum\limits_{l=k}^{n} \left(p_l^{(k)} f_l^{(k-1)}\right)}{\sum\limits_{l=k}^{n} \left(p_l^{(k)}\right)^2}; \quad i = k, \dots, n, \quad j = k+1, \dots, n.$$

В результате выполнения всех $(n-1)$ шагов получится система $A_{n-1}x = f^{(n-1)}$, где матрица A_{n-1} является верхней треугольной, поэтому вектор x можно найти последовательно снизу вверх:

$$x_n = \frac{f_n^{(n-1)}}{a_{nn}^{(n-1)}}, \quad x_i = \frac{f_i^{(n-1)} - \sum\limits_{j=i+1}^{n} a_{ij}^{(n-1)} x_j}{a_{ii}^{(n-1)}}, \quad i = n-1, \dots, 1.$$

Сложность данного алгоритма $O(n^3)$. Это оценка основывается на том, что на для вычисления сумм требуется в среднем $\frac{n}{2}$, при этом такие суммы рассчитываются на каждом для каждого элемента, выше диагонали, что можно оценить как $\frac{n^2}{2}$.

2.2.1. Погрешности вычислений

Данный метод является точным численным методом, поэтому погрешность вычислений зависит от погрешности представления числа в компьютере и количестве проделанных операций деления. Из этого следует, что, используя более точные типы данных, результат будет более точный.

2.3. Вычислительные эксперименты

Для вычислительных экспериментов будут использованы матрицы и вектора, предложенные на практических занятиях по вычислительной математике, из пособия[15], а также несколько специально подобранных. Матрицы можно найти в секции «Приложения».

В таблице 1 приведены модули максимальных погрешностей:

- $\Delta(x)$ – по найденному вектору x и решению, полученному встроенной функции пакета «numpy».
- $\Delta(f)$ – по вычисленному вектору Ax и данному вектору f.

№ данных	$\mu(A)$	$\Delta(x)$	$\Delta(f)$
1	7.3257	2.22e-16	4.44e-16
2	660.085	2.13e-14	1.42e-14
3	954.552	1.37e-14	9.77e-15
4	220.045	5.3e-15	1.69e-15
5	20001	1.57e-12	4.44e-16
6	20000001	1.57e-09	4.44e-16

Таблица 1: Таблица результатов.

2.3.1. Анализ результатов

Как можно увидеть, погрешность увеличивается с увеличением числа обусловленности, потому что число обусловленности показывает насколько сильно небольшое изменение в правой части уравнения ведёт к изменению в решении.

Погрешность в получаемом векторе Ax во всех экспериментах была достаточно близка, и можно сделать вывод, что число обусловленности на него в данных экспериментах не влияет.

3. Заключение

В ходе данной работы был исследован точный численный метод отражения для решения систем линейных алгебраических уравнений. Был получен опыт самостоятельной профессиональной деятельности, подведения итогов работы в виде отчёта. А так же изучены основы алгоритмизации для численного решений задач линейной алгебры на языке программирования «Python».

4. Список использованных источников

1. Бахвалов Н.С. Численные методы / Н.С. Бахвалов, Н.П. Жидков, Г.М. Кобельков. – М.: Наука, 2002 г. – 630 с..
2. Белов С.А., Золотых Н.Ю. Лабораторный практикум по численным методам линейной алгебры. Изд. Нижегородского университета, 2005 г. – 235 с.
3. Богачев К. Ю. Практикум на ЭВМ. Методы решения линейных систем и нахождения собственных значений, Часть 1, Изд.-во МГУ , 1998. – 79 с.
4. Богачев К. Ю. Практикум на ЭВМ. Методы решения линейных систем и нахождения собственных значений, Часть 2, Изд.-во МГУ , 1998. – 137 с.
5. Вержбицкий В. М. Основы численных методов. 2-е изд., перераб. / В. М. Вержбицкий. – М.: Высшая школа, 2005. – 267 с.
6. Волков Е.А. Численные методы, М.: Наука, 1987. - 248 с.
7. Гантмахер Ф.Р. Теория матриц, Изд.-во: Физматлит 2010. — 558 с.
8. Демидович Б. П., Марон И. А. Основы вычислительной математики, Изд-во ”Наука”, 1970. - 664 с.
9. Деммель Дж. Вычислительная линейная алгебра. Теория и приложения - М.: Мир, 2001. -435с.
10. Иванов А.П. Практикум по численным методам. Решение систем линейных алгебраических уравнений. Методические указания. Изд. СПбГУ, Санкт-Петербург, 2013 г. 19 с.
11. Курс лекций: Вычислительные методы линейной алгебры. Изд.-во ДВГУ, Владивосток 2008 г, 27 с.
12. Куксенко С. П., Газизов Т. Р. Итерационные методы решения системы линейных алгебраических уравнений с плотной матрицей . – Томск: Томский государственный университет, 2007 – 208 с.

13. Ланкастер П. Теория матриц, Изд.- во ”Наука”, 1973. - 280 с.

14. Маркус М., Минк Х. Обзор по теории матриц и матричных неравенств, Изд.- во ”Наука”, 1972. - 232 с.

15. Молчанова Л.А. Численные методы линейной алгебры. Методические указания. Изд.-во ДВГУ, Владивосток 2008 г., 37 с.

16. Самарский А.А., Николаев Е.С. Методы решения сеточных уравнений. М.: Наука, 1978 г. – 592 с.

17. Семушин И.В. Численные методы алгебры / И.В. Семушин. – Ульяновск: УлГТУ, 2006.- 180 с.

18. Уилкинсон Дж.Х. Алгебраическая проблема собственных значений - Изд-во ”Наука”, 1970. -565 с.

19. Фаддеев Л.К. Вычислительные методы линейной алгебры / Л.К. Фаддеев, В.Н. Фаддеева. – М.: Физматгиз, 1963.- 656 с.

20. Шарый С.П. Курс вычислительных методов. Институт вычислительных технологий СО РАН, Новосибирск, 2017.-556 с.

21. Прогонки. Пособие в электронном виде.

22. Хорн Р., Джонсон Ч. Матричный анализ, - М.: Мир, 1989. - 666 с.

5. Приложения

5.1. Примеры матриц и векторов

В листингах сначала идёт матрица A, после чего вектор-строка f, в удобном виде для программы.

Листинг 1: Данные 1.

```
6  0  4
0  5 -1
4 -1  5

2 3 2
```

Листинг 2: Данные 2.

```
5  1  0
1 17 10
0 10  6

0 -1 -1
```

Листинг 3: Данные 3.

```
6.03 13 -17
13 29.03 -38
-17 -38 50.03

2.0909 4.1509 -5.1191
```

Листинг 4: Данные 4.

```
0.411 0.421 -0.333 0.313 -0.141 -0.381 0.245
0.241 0.705 0.139 -0.409 0.321 0.0625 0.101
0.123 -0.239 0.502 0.901 0.243 0.819 0.321
0.413 0.309 0.801 0.865 0.423 0.118 0.183
0.241 -0.221 -0.243 0.134 1.274 0.712 0.423
0.281 0.525 0.719 0.118 -0.974 0.808 0.923
0.246 -0.301 0.231 0.813 -0.702 1.223 1.105

0.096 1.252 1.024 1.023 1.155 1.937 1.673
```

Листинг 5: Данные 5.

```
1 1.0001
1.0001 1

2.0001 2.0001
```

Листинг 6: Данные 6.

```
1 1.0000001
1.0000001 1

2.0000001 2.0000001
```

5.2. Вспомогательный модуль

Код вспомогательного модуля, где находятся функции вывода решения и погрешностей, а также чтения данных.

Листинг 7: Модуль utility.py.

```
import numpy as np


def main_solve(solve_func, matrix=None, values=None):
    if matrix is None or values is None:
        print("Matrix or values not present")

    my_sol = solve_func(matrix, values)
    np_sol = np.linalg.solve(matrix, values)
    diff = my_sol - np_sol

    print(f"my_sol:\n{list(my_sol)}\n", f"max diff for my_sol - np_sol = {np.max
        (abs(diff))}",
          sep='\n')

    my_f = matrix.dot(my_sol)
    print(f"max diff for my_f - f = {np.max(abs(values-my_f))}")
    print(np.linalg.cond(matrix), "&", np.max(abs(diff)), "&", np.max(abs(values
        -my_f)))

```

```python

def read_data(file_name: str) -> (np.matrix, np.ndarray):
    with open(file_name, "r") as f:
        start = f.readline().split()
        raw_mat = [start]
        for _ in range(len(start) - 1):
            raw_mat.append(f.readline().split())

        s = f.read().strip()

        return np.matrix(raw_mat).astype(float), np.array(s.split()).astype(
            float)
```

5.3. Метод отражения

Листинг 8: Метод отражения.

```python
import numpy as np
import utility as ut


def sigma(num: float):
    return 1 if num >= 0 else -1


def solve(matrix: np.matrix, values: np.array):
    matrix = matrix.copy().astype(float)
    values = values.copy().astype(float)

    n = len(matrix)

    for k in range(n):
        p = np.zeros(n)
        p[k] = matrix[k, k] + sigma(matrix[k, k]) * (sum(matrix[l, k] ** 2 for l
            in range(k, n))) ** 0.5
        p[k + 1:] = matrix[k + 1:, k].flatten()

        matrix[k, k] = - sigma(matrix[k, k]) * (sum(matrix[l, k] ** 2 for l in
            range(k, n))) ** 0.5
```

```python
        values_new = values.copy()
        matrix_new = matrix.copy()
        for i in range(k, n):
            values_new[i] = values[i] - 2 * p[i] * sum(p[l] * values[l] for l in
                range(k, n)) / sum(
                p[l] ** 2 for l in range(k, n))
            for j in range(k+1, n):
                matrix_new[i, j] = matrix[i, j] - 2 * p[i] * sum(p[l] * matrix[l
                    , j] for l in range(k, n)) / sum(
                    p[l] ** 2 for l in range(k, n))
        matrix_new[k + 1:, k] = np.zeros((n - k - 1, 1))

        matrix = matrix_new
        values = values_new
    sol = np.array([0.] * n).astype(float)

    for i in reversed(range(n)):
        sol[i] = (values[i] - sum(matrix[i, k] * sol[k] for k in range(i + 1, n)
            )) / matrix[i, i]

    return sol


if __name__ == "__main__":
    A1, b1 = ut.read_data("CompMath/s5-kp/source/in6.txt")
    ut.main_solve(solve, matrix=A1, values=b1)
```

6. Решение теоретических задач

6.1. Задание 1

6.1.1. Постановка задачи

Найдите соотношение эквивалентности, связывающее норму $M(A) = n \max\limits_{1 \le i,j \le n} |a_{ij}|$ с $||A||_\infty$. Проверьте экспериментально.

6.1.2. Решение

$$||Ax||_\infty = \max_i \left| \sum_j a_{ij} x_j \right| \le \max_{i,j} |a_{ij}| \sum_j |x_j| = \max_{i,j} |a_{ij}| \cdot ||x||_1$$

Отсюда получаем: $\max\limits_{i,j} |a_{ij}| \ge \dfrac{||Ax||_\infty}{||x||_1}$. Равенство достигается, когда все элементы матрицы одинаковые. Имеем: $\max\limits_{i,j} |a_{ij}| = \sup\limits_{x \ne 0} \dfrac{||Ax||_\infty}{||x||_1}$.

Далее будем использовать неравенство: $||x||_\infty \le ||x||_1 \le n||x||_\infty$. Получим оценку снизу:

$$M(A) = n \max_{i,j} |a_{ij}| = n \sup_{x \ne 0} \frac{||Ax||_\infty}{||x||_1} \ge \sup_{x \ne 0} \frac{n||Ax||_\infty}{n||x||_\infty} = ||A||_\infty.$$

Теперь получим оценку сверху:

$$M(A) = n \max_{i,j} |a_{ij}| = n \sup_{x \ne 0} \frac{||Ax||_\infty}{||x||_1} \le n \sup_{x \ne 0} \frac{||Ax||_\infty}{||x||_\infty} = n||A||_\infty.$$

Таким образом, получили следующее соотношение эквивалентности:

$$||A||_\infty \le M(A) \le n||A||_\infty$$

Проверим его экспериментально и убедимся в выполнении неравенства:

$$A = \begin{pmatrix} 1 & 10 & 8 \\ -5 & 7 & -3 \\ 0 & -6 & 12 \end{pmatrix}, \qquad \begin{array}{c} n = 3, \quad ||A||_\infty = 19, \quad M(A) = 36 \\ 19 \le 36 \le 57 \end{array}$$

6.2. Задание 2

6.2.1. Постановка задачи

Докажите теоретически и проверьте экспериментально, что число обусловленности $\mu(A) = \mu(\alpha A)$, где число $\alpha \neq 0$.

6.2.2. Решение

Для доказательства по формуле распишем число обусловленности.

$$\mu(\alpha A) = ||\alpha A|| \cdot ||(\alpha A)^{-1}|| = |\alpha| \cdot ||A|| \cdot |\alpha^{-1}| \cdot ||A^{-1}|| = 1 \cdot ||A|| \cdot ||A^{-1}|| = \mu(A)$$

Равенство доказано.

Проверим его экспериментально, используя число $\alpha = 2$, и используя формулу $\mu(A) = ||A|| \cdot ||A^{-1}||$:

$$A = \begin{pmatrix} 3 & 4 \\ 1 & 2 \end{pmatrix}, \quad A^{-1} = \begin{pmatrix} 1 & -2 \\ -\frac{1}{2} & \frac{3}{2} \end{pmatrix},$$

$$||A||_1 = 7, \quad ||A^{-1}||_1 = 3, \quad \mu(A) = 21$$

$$\alpha A = \begin{pmatrix} 6 & 8 \\ 2 & 4 \end{pmatrix}, \quad (\alpha A)^{-1} = \begin{pmatrix} \frac{1}{2} & -1 \\ -\frac{1}{4} & \frac{3}{4} \end{pmatrix},$$

$$||\alpha A||_1 = 14, \quad ||(\alpha A^{-1})||_1 = \frac{3}{2}, \quad \mu(\alpha A) = 21$$

Получили: $\mu(A) = \mu(\alpha A) = 21$.