Федеральное государственное автономное образовательное учреждение высшего образования «Национальный исследовательский университет ИТМО»
Факультет программной инженерии и компьютерной техники

Лабораторная работа №1

По вычислительной математике
Вариант №9

Выполнил:
Пчелкин Илья Игоревич
Группа № P3206
Проверил:
Зинченко Антон Андреевич

Санкт-Петербург 2026

Оглавление

Цель работы

Изучить численные методы решения СЛАУ

Описание метода

Итерационные методы позволяют решить систему посредством последовательного приближения решения к истинному значению, решение будет приближенно настолько на сколько нам это нужно.

При каждом приближении (на k-той итерации) получается вектор $x^{(k)}$, в пределе при k стремящемся к бесконечности вектор равен точному решению системы,

т. е. $\lim\limits_{k\to\infty} x^{(k)} = x^{(*)}$. Когда достигается желаемая точность ε, т. е. вектор погрешности

$\varepsilon^* = | x^{(k+1)} - x^{(k)}| <= \varepsilon$ приближение завершается.

Метод простой итерации:

Пусть есть исходная матрица:

$$\begin{cases} a_{11}x_1 + \cdots + a_{1n}x_n = b_1 \\ \vdots \\ a_{n1}x_1 + \cdots + a_{nn}x_n = b_n \end{cases}$$

Сначала нужно проверить преобладают ли в матрице диагональные элементы (модуль главного элемента больше, чем сумма оставшихся коэффициентов), если данное условие не выполняется, то нужно привести матрицу эквивалентными преобразованиями к такому виду чтобы выполнялось данное условие, если это невозможно, это значит что нет гарантии что решение сойдется, поэтому в таком случае лучше применить другой метод. Если же все хорошо, то далее нужно разделить каждую строку на коэффициент при главном элементе чтобы он стал равен 1.

После этих преобразований наша СЛАУ будет иметь вид

$$\begin{cases} x_1 = -\frac{a_{12}}{a_{11}}x_2 - \cdots - \frac{a_{1n}}{a_{11}}x_n + \frac{b_1}{a_{11}} \\ \vdots \\ x_n = -\frac{a_{n2}}{a_{nn}}x_2 - \cdots - \frac{a_{1,n-1}}{a_{nn}}x_{n-1} + \frac{b_n}{a_{nn}} \end{cases}$$

Нашим нулевым приближением будет являться вектор

$X_0 = (x_1^{(0)}, \ldots, x_n^{(0)}) = (\frac{b_1}{a_{11}}, \ldots, \frac{b_n}{a_{nn}})$

И далее нужно подставить эти иксы в СЛАУ, где каждый x уже будет 1-м приближением

$$\begin{cases} x_1 = x_1^{(1)} = -\frac{a_{12}}{a_{11}}x_2^{(0)} - \cdots - \frac{a_{1n}}{a_{11}}x_n^{(0)} + \frac{b_1}{a_{11}} \\ \vdots \\ x_n = x_n^{(1)} = -\frac{a_{n2}}{a_{nn}}x_2^{(0)} - \cdots - \frac{a_{1,n-1}}{a_{nn}}x_{n-1}^{(0)} + \frac{b_n}{a_{nn}} \end{cases}$$

И так до k-той итерации

$$\begin{cases} x_n^{(k)} = -\dfrac{a_{12}}{a_{11}} x_2^{(k-1)} - \dots - \dfrac{a_{1n}}{a_{11}} x_n^{(k-1)} + \dfrac{b_1}{a_{11}} \\ \vdots \\ x_n^{(k)} = -\dfrac{a_{n2}}{a_{nn}} x_2^{(k-1)} - \dots - \dfrac{a_{1,n-1}}{a_{nn}} x_{n-1}^{(k-1)} + \dfrac{b_n}{a_{nn}} \end{cases}$$

Листинг программы

main.py

```
from matrixUtil import MatrixUtil
from printUtil import PrintUtil

printUtil = PrintUtil()
matrixUtil = MatrixUtil(printUtil)
if __name__ == "__main__":
    while True:
        try:
            print("=" * 387)
            print("Выберите откуда считать матрицу:\n1 - ввод в консоли\n2 - файл")
            print("Введите \"exit\" для выхода\n")
            userInput = input().strip()
            if userInput == "1":
                matrixUtil.getMatrixFromInput()
            elif userInput == "2":
                print("Введите название файла: ")
                fileName = input()
                matrixUtil.getMatrixFromFile(fileName)
            elif userInput == "exit":
                break
            else:
                print("Введите корректный вариант")
        except FileNotFoundError:
            print("Введите корректное имя файла!")
```

matrixUtil.py

```
from printUtil import PrintUtil


class MatrixUtil:
    def __init__(self, printUtil: PrintUtil):
        self.printUtil = printUtil

    def getMatrixFromFile(self, filename):
        matrix = list()
        n = 0
        with open(filename) as f:
            while True:
                for line in f:
                    if line.strip() == "":
                        break
                    else:
                        lineNums = list(map(int, line.strip().split()))
                        if len(lineNums) == 1:
                            n = lineNums[0]
                        else:
                            matrix.append(lineNums)
                break
        print("Изначальная матрица:")
```

```
        self.printUtil.printMatrix(matrix)
        print()
        self.zeroApproximation(matrix, n)

    def getMatrixFromInput(self):
        n = 0
        while True:
            try:
                n = int(input("Введите размерность матрицы(от 1 до 20): "))
                if n < 1 or n > 20:
                    print("Размерность матрицы должна быть от 1 до 20!")
                    continue
                else:
                    break
            except ValueError:
                print("Размерность должна быть числом!")
                continue

        while True:
            try:
                print("Для подтверждения матрицы нажмите enter 2 раза")
                print("Введите матрицу\n")
                matrix = self.inputMatrix()
                if len(matrix) != n:
                    print("Введите корректную матрицу")
                    continue

                else:
                    print("Изначальная матрица:")
                    self.printUtil.printMatrix(matrix)
                    print()
                    self.zeroApproximation(matrix, n)
                    break
            except ValueError:
                print("Введите корректную матрицу")

    def inputMatrix(self):
        matrix = list()
        while True:
            line_input = input()
            if line_input == "":
                break
            else:
                line = list(map(int, line_input.strip().split()))
                matrix.append(line)
        return matrix

    def diagElemDominate(self, matrix, lineIndex):
        s = 0  # сумма всех коэф. кроме главного
        matrixLine = matrix[lineIndex]
        for i in range(len(matrixLine) - 1):
            if i != lineIndex:
                s += abs(matrixLine[i])
        if abs(matrixLine[lineIndex]) < s:
            return False
        return True

    def zeroApproximation(self, matrix, n):
        for i in range(n):
            for j in range(i + 1, n):
                if not self.diagElemDominate(matrix, i):  # условие преобладания
диагональных элементов
```

```
                    matrix[i], matrix[j] = matrix[j], matrix[i]
                    if self.diagElemDominate(matrix, i):
                        break
        if all(self.diagElemDominate(matrix, i) for i in range(n)):
            print("Преобладание диагональных элементов достигнуто")
        else:
            print(
                "Решение может не сходиться т.к. в матрице не выполняется условие
преобладания диагональных элементов")

        self.printUtil.printMatrix(matrix)
        c = []  # список для коэф. при главных элементах
        divideZeroFlag = False
        for i in range(n):
            c.append(matrix[i][i])
        for i in range(n):
            for j in range(n):
                if i != j:
                    if c[i] != 0:
                        matrix[i][j] = -1 * matrix[i][j] / c[i]
                    else:
                        divideZeroFlag = True
                        matrix[i][-1] = matrix[i][-1]
                else:
                    matrix[i][j] = 0
        d = [0] * n  # список свободных членов равных решению x_0 (нулевому
приближению)
        for i in range(n):
            if c[i] != 0:
                matrix[i][-1] = matrix[i][-1] / c[i]
            else:
                divideZeroFlag = True
                matrix[i][-1] = matrix[i][-1]
            d[i] = matrix[i][-1]

        X = []  # решения системы x_0, ..., x_k; где k = номер итерации
        X.append(d)

        self.printUtil.printX(0, X[0])

        k = 1
        i = 0
        if not divideZeroFlag:
            while not self.absoluteDeviation(X):
                i += 1
                x = self.kApproximationSimpleIteration(matrix, n, X)
                X.append(x)
                self.printUtil.printX(k, X[-1])
                k += 1
                if i > 10: break
        else:
            self.printUtil.printX(0, d)
        print(f"Итого: {i + 1} итераций")

    def absoluteDeviation(self, X):
        epsilon = 0.01  # абсолютное отклонение
        maxDeviation = -1
        l = len(X)
        solutionLen = len(X[0])
        if l == 1:
            print(f"Вектор погрешности: {max(X[0]):10.4f}")
            return False
```

```
        for i in range(solutionLen):
            e = abs(X[l - 1][i] - X[l - 2][i])
            if e >= maxDeviation:
                maxDeviation = e
        print(f"Вектор погрешности: {maxDeviation:10.4f}")
        return maxDeviation <= epsilon

    def kApproximationSimpleIteration(self, matrix, n, X):
        x = []  # k-тое приближение решения
        for i in range(n):
            x_k = 0
            for j in range(n):
                x_k += matrix[i][j] * X[-1][j]
            x_k += + X[0][i]
            x.append(x_k)
        return x
```

printUtil.py

```
class PrintUtil:
    def printSoulutionX(self, k, d1, d2):
        print(f"x_{k} = ({round(d1, 3)}, {round(d2, 3)})")

    def printX(self, k, x):
        s = f"Вектор неизвестных: x_{k} = ("
        for i in range(len(x)):
            if i == len(x) - 1:
                s += f"{round(x[i], 4)})"
                break
            s += f"{round(x[i], 4)}, "

        print(f"{s:60}", end=" | ")

    def printMatrix(self, matrix: list):
        print("=" * 387)
        for i in range(len(matrix)):
            for j in range(len(matrix) + 1):
                if not isinstance(matrix[i][j], float):
                    print(f"{matrix[i][j]:6}", end=" ")
                else:
                    print(f"{matrix[i][j]:6.4f}", end=" ")
            print()
        print("=" * 387)
```

Примеры работы программы

```
==========================================================================================================================================================================================
==========================================================================================================================================================================================
====================================
Выберите откуда считать матрицу:
1 - ввод в консоли
2 - файл
Введите "exit" для выхода

1
Введите размерность матрицы(от 1 до 20): 3
Для подтверждения матрицы нажмите enter 2 раза
Введите матрицу

 2   2 10 14
10   1  1 12
 2  10  1 13

Изначальная матрица:
==========================================================================================================================================================================================
==========================================================================================================================================================================================
====================================
     2      2     10     14
    10      1      1     12
     2     10      1     13
==========================================================================================================================================================================================
==========================================================================================================================================================================================
====================================
```

```
Преобладание диагональных элементов достигнуто
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================
    10      1      1     12
     2     10      1     13
     2      2     10     14
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================
Вектор неизвестных: x_0 = (1.2, 1.3, 1.4)                    | Вектор погрешности:     1.4000
Вектор неизвестных: x_1 = (0.93, 0.92, 0.9)                  | Вектор погрешности:     0.5000
Вектор неизвестных: x_2 = (1.018, 1.024, 1.03)               | Вектор погрешности:     0.1300
Вектор неизвестных: x_3 = (0.9946, 0.9934, 0.9916)           | Вектор погрешности:     0.0384
Вектор неизвестных: x_4 = (1.0015, 1.0019, 1.0024)           | Вектор погрешности:     0.0108
Вектор неизвестных: x_5 = (0.9996, 0.9995, 0.9993)           | Вектор погрешности:     0.0031
Итого: 6 итераций
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================
Выберите откуда считать матрицу:
1 - ввод в консоли
2 - файл
Введите "exit" для выхода

1
Введите размерность матрицы(от 1 до 20): 10
Для подтверждения матрицы нажмите enter 2 раза
Введите матрицу

  0  1 59  0 -3  2  0  0  0  0  35
  1  0  2 64  0  3  1  0  0  0  46
  3  0  0  0  0  0  0  0  2 57  44
 -2  0  0  0  3 78  1  0 -1  0  62
  0  0  0  0  0  0  2 66  4 -1  51
  2 67 -1  0  4  0  0  0  0  0  48
  0  0  0  1 71 -2  0  4  0  0  55
  0  0  0  0  0 -1 53  3  0  2  39
 52  3  0  1  0  0  0  0  0  2  41
  0  0  0  0  0  0  0  1 84  0  68

Изначальная матрица:
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================
     0      1     59      0     -3      2      0      0      0      0     35
     1      0      2     64      0      3      1      0      0      0     46
     3      0      0      0      0      0      0      0      2     57     44
    -2      0      0      0      3     78      1      0     -1      0     62
     0      0      0      0      0      0      2     66      4     -1     51
     2     67     -1      0      4      0      0      0      0      0     48
     0      0      0      1     71     -2      0      4      0      0     55
     0      0      0      0      0     -1     53      3      0      2     39
    52      3      0      1      0      0      0      0      0      2     41
     0      0      0      0      0      0      0      1     84      0     68
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================

Преобладание диагональных элементов достигнуто
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================
    52      3      0      1      0      0      0      0      0      2     41
     2     67     -1      0      4      0      0      0      0      0     48
     0      1     59      0     -3      2      0      0      0      0     35
     1      0      2     64      0      3      1      0      0      0     46
     0      0      0      1     71     -2      0      4      0      0     55
    -2      0      0      0      3     78      1      0     -1      0     62
     0      0      0      0      0     -1     53      3      0      2     39
     0      0      0      0      0      0      2     66      4     -1     51
     0      0      0      0      0      0      1     84      0     68     41
     3      0      0      0      0      0      0      0      2     57     44
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================
Вектор неизвестных: x_0 = (0.7885, 0.7164, 0.5932, 0.7188, 0.7746, 0.7949, 0.7358, 0.7727, 0.8095, 0.7719) | Вектор погрешности:     0.8095
Вектор неизвестных: x_1 = (0.7036, 0.6555, 0.5935, 0.6391, 0.7434, 0.7862, 0.678, 0.7131, 0.8003, 0.702) | Вектор погрешности:     0.0848
Вектор неизвестных: x_2 = (0.7114, 0.6599, 0.5933, 0.6418, 0.7476, 0.7859, 0.6838, 0.7143, 0.801, 0.7068) | Вектор погрешности:     0.0077
Итого: 3 итераций
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================
Выберите откуда считать матрицу:
1 - ввод в консоли
2 - файл
Введите "exit" для выхода

2
Введите название файла:
matrix20x20.txt
Изначальная матрица:
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================
     0      0      0      0      0      0      0      0      0      0      3     68      2      0      0      0      0      0      0      0    875
     0      0      0      0      0      0      0    100     70      1      0      0      0      0      0      0      0      0      0      0    576
     0      0      0      0      0      0      0      0      1     62      3      0      0      0      0      0      0      0      0      0    662
     0      0      0      0      0      0      0      0      0      0      0      0      2     59      3      0      0      0      0      0    897
     0      0      0      3     48      1      0      0      0      0      0      0      0      0      0      0      0      0      0      0    258
     0      0      0      0      2     65      3      0      0      0      0      0      0      0      0      0      0      0      0      0    421
     0      2     55      1      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0    173
     0      0      0      0      0      0      0      3     58      2      0      0      0      0      0      0      0      0      0      0    566
     1     45      3      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0     97
     0      0      0      0      0      0      0      0      0      2     44      1      0      0      0      0      0      0      0      0    515
     0      0      0      0      0      0      0      0      0      0      0      1     53      1      0      0      0      0      0      0    715
     0      0      0      0      0      0      0      0      0      0      0      0      0      1     66      2      0      0      0      0   1036
    50      2      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      1     74
     0      0      0      0      0      1     52      2      0      0      0      0      0      0      0      0      0      0      0      0    386
     0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      3     63      1   1236
     0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      2     72      3      0      0   1310
     0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      3     51      1      0      0    877
     1      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      2     57   1179
     0      0      1     60      2      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0    253
     0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      1     49      2      0    937
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================

Преобладание диагональных элементов достигнуто
================================================================================================================================================================================================================
================================================================================================================================================================================================================
====================================
    50      2      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      0      1     74
```

```
    1    45     3     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0    97
    0     2    55     1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0   173
    0     0     1    60     2     0     0     0     0     0     0     0     0     0     0     0     0     0     0   253
    0     0     0     3    48     1     0     0     0     0     0     0     0     0     0     0     0     0     0   258
    0     0     0     0     2    65     3     0     0     0     0     0     0     0     0     0     0     0     0   421
    0     0     0     0     0     1    52     2     0     0     0     0     0     0     0     0     0     0     0   386
    0     0     0     0     0     0     0   100    70     1     0     0     0     0     0     0     0     0     0   576
    0     0     0     0     0     0     0     3    58     2     0     0     0     0     0     0     0     0     0   566
    0     0     0     0     0     0     0     0     1    62     3     0     0     0     0     0     0     0     0   662
    0     0     0     0     0     0     0     0     0     2    44     1     0     0     0     0     0     0     0   515
    0     0     0     0     0     0     0     0     0     0     3    68     2     0     0     0     0     0     0   875
    0     0     0     0     0     0     0     0     0     0     0     1    53     1     0     0     0     0     0   715
    0     0     0     0     0     0     0     0     0     0     0     0     2    59     3     0     0     0     0   897
    0     0     0     0     0     0     0     0     0     0     0     0     0     1    66     2     0     0     0  1036
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     3    51     1     0     0   877
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     2    72     3     0  1310
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     1    49     2   937
    0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     3    63     1  1236
    1     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     0     2    57  1179
==========================================================================================================================================================================================================
==========================================================================================================================================================================================================
====================================
Вектор неизвестных: x_0 = (1.48, 2.1556, 3.1455, 4.2167, 5.375, 6.4769, 7.4231, 5.76, 9.7586, 10.6774, 11.7045, 12.8676, 13.4906, 15.2034, 15.697, 17.1961, 18.1944, 19.1224,
19.619, 20.6842) | Вектор погрешности:    20.6842
Вектор неизвестных: x_1 = (0.9801, 1.913, 2.9904, 3.9851, 4.9765, 5.9689, 7.077, -1.1778, 9.0925, 9.9537, 10.9268, 11.9545, 12.9609, 13.9479, 14.9455, 15.916, 16.92, 17.9504,
18.3801, 19.9699) | Вектор погрешности:     6.9378
Вектор неизвестных: x_2 = (1.0041, 1.9344, 3.0034, 4.0009, 5.0016, 5.9972, 7.3536, -0.7043, 9.4763, 10.0021, 10.9804, 12.0044, 13.0018, 14.0041, 15.0033, 15.9852, 17.0044,
18.0269, 18.4473, 20.0221) | Вектор погрешности:     0.4735
Вектор неизвестных: x_3 = (1.0022, 1.933, 3.0024, 3.9999, 5.0, 5.9836, 7.3348, -0.9734, 9.4502, 9.9933, 10.9771, 12.0008, 12.9998, 13.9998, 15.0004, 15.9801, 16.9993, 18.0225,
18.4428, 20.0193) | Вектор погрешности:     0.2692
Вектор неизвестных: x_4 = (1.0023, 1.9331, 3.0024, 4.0, 5.0003, 5.9845, 7.3454, -0.955, 9.4644, 9.9938, 10.9776, 12.001, 13.0, 14.0, 15.0006, 15.9804, 16.9996, 18.0228,
18.4431, 20.0195) | Вектор погрешности:     0.0184
Вектор неизвестных: x_5 = (1.0023, 1.9331, 3.0024, 3.9999, 5.0003, 5.984, 7.3447, -0.965, 9.4634, 9.9936, 10.9775, 12.001, 13.0, 14.0, 15.0006, 15.9804, 16.9996, 18.0227,
18.4431, 20.0195) | Вектор погрешности:     0.0100
Итого: 6 итераций
==========================================================================================================================================================================================================
==========================================================================================================================================================================================================
====================================
Выберите откуда считать матрицу:
1 - ввод в консоли
2 - файл
Введите "exit" для выхода

exit

Process finished with exit code 0
```

matrix20x20.txt

```
20
  0   0   0   0   0   0   0   0   0   3  68   2   0   0   0   0   0   0   0   875
  0   0   0   0   0   0   0   100  70   1   0   0   0   0   0   0   0   0   0   576
  0   0   0   0   0   0   0   0   1  62   3   0   0   0   0   0   0   0   0   662
  0   0   0   0   0   0   0   0   0   0   2  59   3   0   0   0   0   0   0   897
  0   0   0   3  48   1   0   0   0   0   0   0   0   0   0   0   0   0   0   258
  0   0   0   0   2  65   3   0   0   0   0   0   0   0   0   0   0   0   0   421
  0   2  55   1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   173
  0   0   0   0   0   0   0   3  58   2   0   0   0   0   0   0   0   0   0   566
  1  45   3   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0    97
  0   0   0   0   0   0   0   0   0   0   2  44   1   0   0   0   0   0   0   515
  0   0   0   0   0   0   0   0   0   0   0   1  53   1   0   0   0   0   0   715
  0   0   0   0   0   0   0   0   0   0   0   0   1  66   2   0   0   0   0  1036
  50  2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1    74
  0   0   0   0   0   1  52   2   0   0   0   0   0   0   0   0   0   0   0   386
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3  63   1  1236
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2  72   3   0  1310
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   3  51   1   0   0   0   877
  1   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   2  57  1179
  0   0   1  60   2   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   253
  0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   0   1  49   2   0   937
```

Вывод

Данная лабораторная помогла понять, как решаются СЛАУ численными итерационными методами, в частности методом простой итерации.