

Министерство образования Республики Беларусь
Учреждение образования
БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет компьютерных систем и сетей
Кафедра информатики
Дисциплина: Методы численного анализа

ОТЧЁТ

к лабораторной работе
на тему

Численное решение систем линейных уравнений методом простых
итераций и методом Зейделя

Выполнил: студент группы 053504
Пекутько Дмитрий Леонидович
Проверил: Анисимов Владимир Яковлевич

Минск 2021

Оглавление

Цели выполнения задания	3
Краткие теоретические сведения	4
Задание	9
Программная реализация	10
Полученные результаты	14
Оценка погрешности для метода простых итераций:	15
Оценка погрешности для метода Зейделя:	15
Выводы	16

Вариант 5

Цели выполнения задания

- изучить итерационные методы решения СЛАУ (метод простых итераций, метод Зейделя);
- составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ;
- составить программу решения СЛАУ по разработанному алгоритму;
- выполнить тестовые примеры и проверить правильность работы программы. Сравнить трудоемкость решения методом простых итераций и методом Зейделя.

Краткие теоретические сведения

Итерационные методы основаны на построении сходящейся к точному решению x рекуррентной последовательности.

Для решения СЛАУ **методом простых итераций** преобразуем систему от первоначальной формы $Ax = b$ или

$$\begin{aligned} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n &= b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n &= b_2, \\ &\dots\dots\dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n &= b_n \end{aligned} \quad (2.1)$$

к виду

$$x = Bx + c. \quad (2.2)$$

Здесь B – квадратная матрица с элементами b_{ij} ($i, j = 1, 2, \dots, n$), c – вектор-столбец с элементами c_i ($i = 1, 2, \dots, n$).

В развернутой форме записи система (2.2) имеет следующий вид:

$$\begin{aligned} x_1 &= b_{11}x_1 + b_{12}x_2 + b_{13}x_3 + \dots + b_{1n}x_n + c_1 \\ x_2 &= b_{21}x_1 + b_{22}x_2 + b_{23}x_3 + \dots + b_{2n}x_n + c_2 \\ &\dots\dots\dots \\ x_n &= b_{n1}x_1 + b_{n2}x_2 + b_{n3}x_3 + \dots + b_{nn}x_n + c_n \end{aligned}$$

Вообще говоря, операция *приведения системы к виду, удобному для итераций*, не является простой и требует специальных знаний, а также существенного использования специфики системы.

Можно, например, преобразовать систему (2.1) следующим образом

$$x_1 = (b_1 - a_{11}x_1 - a_{12}x_2 - \dots - a_{1n}x_n) / a_{11} + x_1,$$

$$x_2 = (b_2 - a_{21}x_1 - a_{22}x_2 - \dots - a_{2n}x_n) / a_{22} + x_2,$$

.....

$$x_n = (b_n - a_{n1}x_1 - a_{n2}x_2 - \dots - a_{nn}x_n) / a_{nn} + x_n$$

если диагональные элементы матрицы A отличны от нуля.

Можно преобразовать систему (2.1) в эквивалентную ей систему

$$x = (E-A)x+b.$$

Задав произвольным образом столбец начальных приближений $x^0 = (x_1^0, x_2^0, \dots, x_n^0)^T$, подставим их в правые части системы (2.2) и вычислим новые приближения $x^1 = (x_1^1, x_2^1, \dots, x_n^1)^T$, которые опять подставим в систему (2.2) и т.д. Таким образом, организуется итерационный процесс

$x^k = Bx^{k-1} + c$, $k = 1, 2, \dots$. Известно, что система (2.1) имеет единственное решение x^* и последовательность $\{x^k\}$ сходится к этому решению со скоростью геометрической прогрессии, если $\|B\| < 1$ в любой матричной норме. Т.е.

Т.е. для того, чтобы последовательность простых итераций сходилась к единственному решению достаточно, чтобы выполнялось одно из следующих условий:

$$1) \max_i \left(\sum_{j=1}^n |b_{ij}| \right) < 1 ;$$

n n

$$2) \sum_{i=1}^n \sum_{j=1}^n b_{ij}^2 < 1;$$

n

$$3) \max_j (\sum_{i=1}^n |b_{ij}|) < 1.$$

Метод Зейделя. Метод Зейделя является модификацией метода простых итераций. Суть его состоит в том, что при вычислении следующего $x_i^k : 2 \leq i \leq n$ в формуле $x^k = Bx^{k-1} + c$, $k = 1, 2, \dots$ используются вместо $x_1^{k-1}, \dots, x_{i-1}^{k-1}$ уже вычисленные ранее x_1^k, \dots, x_{i-1}^k , т.е.

$$x_i^k = \sum_{j=1}^{i-1} g_{ij} x_j^k + \sum_{j=i+1}^n g_{ij} x_j^{k-1} + c_i. \quad (2.3) \quad \text{Такое}$$

усовершенствование позволяет ускорить сходимость итераций почти в два раза. Кроме того, данный метод может быть реализован на ЭВМ без привлечения дополнительного массива, т.к. полученное новое x_i^k сразу засылается на место старого.

Самый простой способ приведения системы к виду, удобному для итераций, состоит в следующем. Из первого уравнения системы выразим неизвестное x_1 :

$$x_1 = a_{11}^{-1} (b_1 - a_{12}x_2 - a_{13}x_3 - \dots - a_{1n}x_n),$$

из второго уравнения – неизвестное x_2 :

$$x_2 = a_{21}^{-1} (b_2 - a_{22}x_2 - a_{23}x_3 - \dots - a_{2n}x_n),$$

и т. д. В результате получим систему

$$x_1 = b_{12}x_2 + b_{13}x_3 + \dots + b_{1,n-1}x_{n-1} + b_{1n}x_n + c_1,$$

$$x_2^{(k+1)} = b_{21}x_1^{(k+1)} + b_{23}x_3^{(k)} + \dots + b_{2n}x_n^{(k)} + c_2,$$

$$x_3^{(k+1)} = b_{31}x_1^{(k+1)} + b_{32}x_2^{(k+1)} + \dots + b_{3n}x_n^{(k)} + c_3,$$

.

$$x_n^{(k+1)} = b_{n1}x_1^{(k+1)} + b_{n2}x_2^{(k+1)} + b_{n3}x_3^{(k+1)} + \dots + c_n.$$

Объединив приведение системы к виду, удобному для итераций и метод Зейделя в одну формулу, получим

$$x_i^{(k+1)} = x_i^{(k)} - a_{ii}^{-1}(\sum_{j=1}^{i-1} a_{ij}x_j^{(k+1)} + \sum_{j=i+1}^n a_{ij}x_j^{(k)} - b_i). \quad (2.9)$$

Тогда достаточным условием сходимости метода Зейделя будет условие доминирования диагональных элементов в строках или столбцах матрицы **A**, т.е.

$$a_{ii} > a_{i1} + \dots + a_{in} \quad \text{для всех } i=1, \dots, n,$$

$$\text{или } a_{jj} > a_{1j} + \dots + a_{nj} \quad \text{для всех } j=1, \dots, n.$$

Задание

ЗАДАНИЕ. Методом простых итераций и методом Зейделя найти с точностью 0,0001 численное решение системы $Ax=b$,

где $A = kC + D$, A – исходная матрица для расчёта, k – номер варианта (0-15), матрицы C, D и вектор свободных членов b задаются ниже.

Вариант 5

$$C = \begin{bmatrix} 0,01 & 0 & -0,02 & 0 & 0 \\ 0,01 & 0,01 & -0,02 & 0 & 0 \\ 0 & 0,01 & 0,01 & 0 & -0,02 \\ 0 & 0 & 0,01 & 0,01 & 0 \\ 0 & 0 & 0 & 0,01 & 0,01 \end{bmatrix}, \quad D = \begin{bmatrix} 1,33 & 0,21 & 0,17 & 0,12 & -0,13 \\ -0,13 & -1,33 & 0,11 & 0,17 & 0,12 \\ 0,12 & -0,13 & -1,33 & 0,11 & 0,17 \\ 0,17 & 0,12 & -0,13 & -1,33 & 0,11 \\ 0,11 & 0,67 & 0,12 & -0,13 & -1,33 \end{bmatrix}.$$

Вектор $b = (1,2; 2,2; 4,0; 0,0; -1,2)^T$.

Программная реализация

```
# Method of simple iterations && Seidel's method

from math import fabs

from statistics import mean

C = [

    [0.01, 0, -0.02, 0, 0],

    [0.01, 0.01, -0.02, 0, 0],

    [0, 0.01, 0.01, 0, -0.02],

    [0, 0, 0.01, 0.01, 0],

    [0, 0, 0, 0.01, 0.01]

]

D = [

    [1.33, 0.21, 0.17, 0.12, -0.13],

    [-0.13, -1.33, 0.11, 0.17, 0.12],

    [0.12, -0.13, -1.33, 0.11, 0.17],

    [0.17, 0.12, -0.13, -1.33, 0.11],

    [0.11, 0.67, 0.12, -0.13, -1.33]

]

b = [1.2, 2.2, 4.0, 0.0, -1.2]

VARIANT = 5

SIZE = len(C[0])

PREC = 5

MAX_ITERATION_COUNT = 100000

def print_matrix(matrix):
```

```

    for i in range(SIZE):

        print(matrix[i])

def simple_iteration_method(matrix):

    x_vector = [0,0,0,0,0]

    for k in range(MAX_ITERATION_COUNT):

        buf_x_vector = [*x_vector]

        is_acc_x = True

        for i in range(SIZE):

            str_sum = 0

            for j in range(SIZE):

                str_sum += matrix[i][j]*x_vector[j]

            buf_x_vector[i] =
(b[i]-str_sum)/matrix[i][i]+x_vector[i]

            if fabs(round(buf_x_vector[i], PREC) -
round(x_vector[i], PREC)):

                is_acc_x = False

        if is_acc_x:

            return [round(x, PREC) for x in x_vector], k

        x_vector = buf_x_vector

    return [0]*SIZE, k

def seidel_method(matrix):

    x_vector = [0,0,0,0,0]

    for k in range(MAX_ITERATION_COUNT):

        buf_x_vector = [*x_vector]

        is_acc_x = True

        for i in range(SIZE):

            str_sum = 0

```

```

        for j in range(SIZE):

            str_sum += matrix[i][j]*buf_x_vector[j]

            buf_x_vector[i] =
(b[i]-str_sum)/matrix[i][i]+x_vector[i]

            if fabs(round(buf_x_vector[i], PREC) -
round(x_vector[i], PREC)):

                is_acc_x = False

        if is_acc_x:

            return [round(x, PREC) for x in x_vector], k

        x_vector = buf_x_vector

    return [0]*SIZE, k

def main():

    standard_matrix = [

        [100,2,100,4,5],

        [10,99,8,7,6],

        [12,23,2,65,77],

        [2,1,11,96,33],

        [96,32,4,5,96]

    ]

    # generate standard matrix

    for i in range(SIZE):

        for j in range(SIZE):

            standard_matrix[i][j] =
round(C[i][j]*VARIANT+D[i][j],PREC)

    for i in range(SIZE):

        if standard_matrix[i][i] == 0.0:

```

```

        exit("There a zero at matrix diagonal.")

    iter_x_vector, iter_count =
simple_iteration_method(standard_matrix)

    seidel_x_vector, seidel_count = seidel_method(standard_matrix)

    print("A matrix:")

    print_matrix(standard_matrix)

    if mean(iter_x_vector):

        print(f"Iteration method. X result vector (steps count =
{iter_count}):")

        print_matrix(iter_x_vector)

    else:

        print(f"Iteration method. Cannot find solution
{iter_x_vector}")

    if mean(seidel_x_vector):

        print(f"Seidel's method. X result vector (steps count =
{seidel_count}):")

        print_matrix(seidel_x_vector)

    else:

        print(f"Seidel's method. Cannot find solution
{iter_x_vector}")

if __name__ == "__main__":

    main()

```

Полученные результаты

A matrix:

[1.38, 0.21, 0.07, 0.12, -0.13]

[-0.08, -1.28, 0.01, 0.17, 0.12]

[0.12, -0.08, -1.28, 0.11, 0.07]

[0.17, 0.12, -0.08, -1.28, 0.11]

[0.11, 0.67, 0.12, -0.08, -1.28]

Iteration method. X result vector (steps count = 12):

1.26092

-1.81605

-2.88953

0.16185

-0.18574

Seidel's method. X result vector (steps count = 6):

1.26092

-1.81605

-2.88953

0.16185

-0.18574

Оценка погрешности для метода простых итераций:

Норма матрицы A:

$$||D|| = \max_{1 \leq i \leq n} \sum_{j=1}^n (d_{ij}) = 0.98$$

Изменим первую компоненту вектора b на 0.001: b1 = 1.20 на b1 = 1.201.

Тогда имеем вектор решений:

$$x = \{1.26164, -1.81608, -2.88945, 0.16194, -0.18569\}$$

Абсолютная погрешность

$$\Delta(x) = \frac{0.98}{1-0.98} (|1.26092 - 1.26164|, 0, 0, 0, 0) = 0.03528$$

$$\delta(x_1) = \frac{\Delta(x)}{\|x_{12}\|} = \frac{0.03529}{2.88953} = 0.0122$$

Оценка погрешности для метода Зейделя:

Норма матрицы A:

$$\|D\| = \max_{1 \leq i \leq n} \sum_{j=1}^n (d_{ij}) = 0.98$$

Изменим первую компоненту вектора b на 0.001: b1 = 1.20 на b1 = 1.201.

Тогда имеем вектор решений:

$$x = \{1.26164, -1.81608, -2.88945, 0.16194, -0.18569\}$$

Абсолютная погрешность

$$\Delta(x) = \frac{0.98}{1-0.98} (|1.26092 - 1.26164|, 0, 0, 0, 0) = 0.03528$$

$$\delta(x_1) = \frac{\Delta(x)}{\|x_6\|} = \frac{0.03529}{2.88953} = 0.0122$$

Выводы

Таким образом, в ходе выполнения лабораторной работы был применен метод простых итераций и метод Зейделя для решения системы линейных уравнений, составлены алгоритмы и созданы реализации соответствующих программ на языке Python для решения поставленной задачи, также проведена оценка погрешности приближенных вычислений методов.