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

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 $\boldsymbol{x}$ рекуррентной последовательности.

Для решения СЛАУ **методом простых итераций** преобразуем систему от первоначальной формы $\boldsymbol{A}\boldsymbol{x} = \boldsymbol{b}$ или

$$a_{11}x_1 + a_{12}x_2 + \ldots + a_{1n}x_n = b_1,$$
$$a_{21}x_1 + a_{22}x_2 + \ldots + a_{2n}x_n = b_2,$$
$$\ldots\ldots$$
$$a_{n1}x_1 + a_{n2}x_2 + \ldots + a_{nn}x_n = b_n \qquad (2.1)$$

к виду

$$\boldsymbol{x} = \boldsymbol{B}\boldsymbol{x} + \boldsymbol{c}. \qquad (2.2)$$

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

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

$$x_1 = b_{11}x_1 + b_{12}x_2 + b_{13}x_3 + \ldots + b_{1n}x_n + c_1$$
$$x_2 = b_{21}x_1 + b_{22}x_2 + b_{23}x_3 + \ldots + b_{2n}x_n + c_2$$
$$. \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad . \quad .$$
$$x_n = b_{n1}x_1 + b_{n2}x_2 + b_{n3}x_3 + \ldots + b_{nn}x_n + c_n$$

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 $\boldsymbol{A}$ отличны от нуля.

Можно преобразовать систему (2.1) в эквивалентную ей систему

$$\boldsymbol{x = (E\text{-}A)x+b}.$$

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

$\boldsymbol{x^k} = \boldsymbol{B}\boldsymbol{x^{k-1}} + \boldsymbol{c}$, k = 1,2, …. ,. Известно, что система (2.1) имеет единственное решение $\boldsymbol{x^*}$ и последовательность $\{\boldsymbol{x^k}\}$ сходится к этому решению со скоростью геометрической прогрессии, если || $\boldsymbol{B}$ || < 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) $\max\limits_{i} \left(\sum\limits_{j=1}^{n} |b_{ij}|\right) < 1$;

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

3) $\max_{j} \left(\sum_{i=1}^{n} | b_{ij} | \right) < 1.$

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

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

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:

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

из второго уравнения – неизвестное x_2:

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

и т. д. В результате получим систему

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

$x_2 = b_{21}x_1 + \quad b_{23}x_3 + \ldots + \; b_{2,n-1}x_{n-1} + \quad b_{2n}x_n + c_2$,

$x_3 = b_{31}x_1 + \quad b_{32}x_2 + \ldots + \; b_{3,n-1}x_{n-1} + \quad b_{3n}x_n + c_3$,

. . . .

$x_n = b_{n1}x_1 + \quad b_{n2}x_2 + b_{n3}x_3 + \quad \ldots + \quad b_{n,n-1}x_{n-1} + \quad c_n$,

в которой на главной диагонали матрицы $\boldsymbol{B}$ находятся нулевые элементы. Остальные элементы выражаются по формулам

$$b_{ij} = -a_{ij} / a_{ii},\ c_i = b_i / a_{ii}\ (i, j = 1, 2, \ldots, n, j \neq i) \qquad (2.4)$$

Конечно, для возможности выполнения указанного преобразования необходимо, чтобы диагональные элементы матрицы $\boldsymbol{A}$ были ненулевыми.

Введем нижнюю $\boldsymbol{B}_1$ (получается из $\boldsymbol{B}$ заменой нулями элементов стоявших на главной диагонали и выше ее) и верхнюю $\boldsymbol{B}2$ (получается из $\boldsymbol{B}$ заменой нулями элементов стоявших на главной диагонали и ниже ее) треугольные матрицы.

Заметим, что $\boldsymbol{B} = \boldsymbol{B}_1 + \boldsymbol{B}_2$ и поэтому решение $\boldsymbol{x}$ исходной системы удовлетворяет равенству

$$\boldsymbol{x} = \boldsymbol{B}_1\boldsymbol{x} + \boldsymbol{B}_2\,\boldsymbol{x} + \boldsymbol{c} \qquad (2.5)$$

Выберем начальное приближение $\boldsymbol{x}^{(0)} = [x_1^{(0)}, x_2^{(0)}, \ldots, x_n^{(0)}]^{\mathrm{T}}$. Подставляя его в правую часть равенства при верхней треугольной матрице $\boldsymbol{B}_2$ и вычисляя полученное выражение, находим первое приближение

$$\boldsymbol{x}^{(1)} = \boldsymbol{B}_1\boldsymbol{x}^{(0)} + \boldsymbol{B}_2\boldsymbol{x}^{(1)} \qquad (2.6)$$

Подставляя приближение $\boldsymbol{x}^{(1)}$, получим

$$\boldsymbol{x}^{(2)} = \boldsymbol{B}_1\boldsymbol{x}^{(1)} + \boldsymbol{B}_2\boldsymbol{x}^{(2)} \qquad (2.7)$$

Продолжая этот процесс далее, получим последовательность $\boldsymbol{x}^{(0)}, \boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(n)}$, … приближений к вычисляемых по формуле

$$\boldsymbol{x}^{(k+1)} = \boldsymbol{B}_1^{(k+1)} + \boldsymbol{B}_2^{(k)} + \boldsymbol{c} \qquad (2.8)$$

или в развернутой форме записи

$\mathrm{x}_1^{(k+1)} = \quad b_{12}x_2^{(k)} \; + \; b_{13}x_2^{(k)} \; + \quad \ldots \quad + \quad b_{1n}x_n^{(k)} + c_1$,

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

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

. . . .

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

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

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

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

$a_{ii} > a_{i1} + \ldots\ldots + a_{in}$ *для всех* $i=1,\ldots n$,

или $a_{jj} > a_{1j} + \ldots\ldots + a_{nj}$ *для всех* $j=1,\ldots,n$.

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:

$$\Delta(x) = ||x_{т} - x_{12}\ ||$$

$$\Delta(x) = max(|0.0000006|, |0.0000008|, |0.0000005|, |0.0000004|, |0.0000005|) = 0.6 * 10^{-6}$$

$$\delta\left(x_1\right) = \frac{\Delta\left(x\ \right)}{||x_{12}||} = \frac{0,6*10^{-6}}{2.88953} = 0.2 * 10^{-7}$$

Оценка погрешности для метода Зейделя:

$$\Delta(x) = ||x_{т} - x_{12}\ ||$$

$$\Delta(x) = max(|0.0000006|, |0.0000008|, |0.0000005|, |0.0000004|, |0.0000005|) = 0.6 * 10^{-6}$$

$$\delta\left(x_1\right) = \frac{\Delta\left(x\ \right)}{||x_{12}||} = \frac{0,6*10^{-6}}{2.88953} = 0.2 * 10^{-7}$$

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.