Министерство образования Республики Беларусь

Учреждение образования

БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ

ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ

Факультет Информационных технологий и управления

Кафедра Высшей математики

Лабораторная работа №2

“РЕШЕНИЯ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ ПРЯМЫМИ
МЕТОДАМИ. ИССЛЕДОВАНИЕ ПОГРЕШНОСТИ РЕШЕНИЯ”

Выполнил:
Заломов Р.А., 121702

Проверил:
Самсонов П.А.

Минск 2022

Цель:
Изучение итерационных методов решения СЛАУ - метода Якоби, метода Зейделя и метода простой итерации; исследование зависимости скорости сходимости методов от требуемой точности, порядка системы и величины параметра итерации; сравнение скорости сходимости итерационных методов.

Вариант: 4

Условия заданий:

Вариант 4

1. Решить систему линейных уравнений итерационными методами Якоби и Зейделя и сравнить число итераций, необходимых для получения точности $\boldsymbol{\varepsilon} = 10^{-2}, 10^{-4}, 10^{-6}$.

$$\begin{cases} 9{,}1x_1 + 5{,}6x_2 + 7{,}8x_3 = 9{,}8; \\ 3{,}8x_1 + 5{,}1x_2 + 2{,}8x_3 = 6{,}7; \\ 4{,}1x_1 + 5{,}7x_2 + 1{,}2x_3 = 5{,}8. \end{cases}$$

2. Решить систему линейных уравнений с диагональным преобладанием порядка n (n = 10, 20, 40). Матрица системы и правая часть имеют вид:

$$a_{ij} = \begin{cases} 1, i \neq j \\ 2n, i = j \end{cases} i = 1,2,...n, \quad j = 1,2,...,n,$$

$$b_i = (2n-1)i+n(n+1)/2+9n - 3,\ i=1,2,...,n$$

Сравнить число итераций, необходимых для получения точности $\boldsymbol{\varepsilon} = 10^{-4}$, этими методами.

Выполнение заданий:

NO: метод Якоби для обоих заданий будем реализовывать, используя пакет Wolfram Mathematica, метод Зейделя реализуем с помощью языка программирования Python

1. Задание 1:
   Для начала преобразуем исходную матрицу в матрицу с диагональным преобладанием, по ходу проверяя с помощью встроенных функций, осталось ли решение системы прежним

"Для преобразования матрицы в матрицу с диагональным преобладанием, решим систему с помощью встроенных функций"

$$\text{Solve}\left[\begin{pmatrix} 9.1\,x1 + 5.6\,x2 + 7.8\,x3 \\ 3.8\,x1 + 5.1\,x2 + 2.8\,x3 \\ 4.1\,x1 + 5.7\,x2 + 1.2\,x3 \end{pmatrix} == \begin{pmatrix} 9.8 \\ 6.7 \\ 5.8 \end{pmatrix}, \{x1, x2, x3\}\right]$$

решить уравнения

"Преобразуем в матрицу с диагональным преобладанием. Для этого отнимем из третьего уравнения второе"

$$\text{Solve}\left[\begin{pmatrix} 9.1\,x1 + 5.6\,x2 + 7.8\,x3 \\ 3.8\,x1 + 5.1\,x2 + 2.8\,x3 \\ 0.3\,x1 + 0.6\,x2 - 1.6\,x3 \end{pmatrix} == \begin{pmatrix} 9.8 \\ 6.7 \\ -0.9 \end{pmatrix}, \{x1, x2, x3\}\right]$$

решить уравнения

"Теперь отнимем второе уравнение из первого"

$$\text{Solve}\left[\begin{pmatrix} 5.3\,x1 + 0.5\,x2 + 5\,x3 \\ 3.8\,x1 + 5.1\,x2 + 2.8\,x3 \\ 0.3\,x1 + 0.6\,x2 - 1.6\,x3 \end{pmatrix} == \begin{pmatrix} 3.1 \\ 6.7 \\ -0.9 \end{pmatrix}, \{x1, x2, x3\}\right]$$

решить уравнения

"И прибавим к первому уравнению третее, умноженное на 3"

$$\text{Solve}\left[\begin{pmatrix} 6.2\,x1 + 2.3\,x2 + 0.2\,x3 \\ 3.8\,x1 + 5.1\,x2 + 2.8\,x3 \\ 0.3\,x1 + 0.6\,x2 - 1.6\,x3 \end{pmatrix} == \begin{pmatrix} 0.4 \\ 6.7 \\ -0.9 \end{pmatrix}, \{x1, x2, x3\}\right]$$

решить уравнения

"Прибавим ко второму уравнению третее"

$$\text{Solve}\left[\begin{pmatrix} 6.2\,x1 + 2.3\,x2 + 0.2\,x3 \\ 4.1\,x1 + 5.7\,x2 + 1.2\,x3 \\ 0.3\,x1 + 0.6\,x2 - 1.6\,x3 \end{pmatrix} == \begin{pmatrix} 0.4 \\ 5.8 \\ -0.9 \end{pmatrix}, \{x1, x2, x3\}\right]$$

решить уравнения

"Проверим выполнения условий для матрицы с диагональным преобладанием"

```
Abs[6.2] > Abs[2.3] + Abs[0.2]
```
абсолютное··· абсолютно··· абсолютное значение

```
Abs[5.7] > Abs[4.1] + Abs[1.2]
```
абсолютное··· абсолютно··· абсолютное значение

```
Abs[-1.6] > Abs[0.3] + Abs[0.6]
```
абсолютное з··· абсолютно··· абсолютное значение

"Условия выполняются, значит матрица приведена к виду с диагональным преобладанием"

Out[1485]=

Для преобразования матрицы в матрицу с диагональным преобладанием, решим систему с помощью встроенных функций

Out[1486]=

{{x1 → -0.370389, x2 → 1.09381, x3 → 0.903231}}

Out[1487]=

Преобразуем в матрицу с диагональным преобладанием. Для этого отнимем из третьего уравнения второе

Out[1488]=

{{x1 → -0.370389, x2 → 1.09381, x3 → 0.903231}}

Out[1489]=

Теперь отнимем второе уравнение из первого

Out[1490]=

{{x1 → -0.370389, x2 → 1.09381, x3 → 0.903231}}

Out[1491]=

И прибавим к первому уравнению третее, умноженное на 3

Out[1492]=

{{x1 → -0.370389, x2 → 1.09381, x3 → 0.903231}}

Out[1493]=

Прибавим ко второму уравнению третее

Out[1494]=

{{x1 → -0.370389, x2 → 1.09381, x3 → 0.903231}}

Out[1495]=

Проверим выполнения условий для матрицы с диагональным преобладанием

Out[1496]=

True

Out[1497]=

True

Out[1498]=

True

Out[1499]=

Условия выполняются, значит матрица приведена к виду с диагональным преобладанием

Метод Зейделя

Реализация

```
{maxiter, e} = {30, 10^-2}; n = Length[b];
                                 длина
y = b; iter = 0; m = 0; SD = {}; NList = {}; Abss = {};
While[And[iter ≤ maxiter, m < n], x = y; iter++; m = 0;
цикл··· логическое И
 For[i = 1, i ≤ n, i++, s = (Σ_{j=1}^{n} a[[i, j]] × x[[j]] - b[[i]]) / a[[i, i]]; y[[i]] = x[[i]] - s; If[Abs[s] < e, m++];];
 цикл ДЛЯ                                                                                            ··· абсолютное значение
 SD = Append[SD, x]; Abss = Append[Abss, s]; NList = Append[NList, Dot[a, x] - b]]
      добавить в конец     добавить в конец             добавить в конец  скалярное произведение
If[iter > maxiter, Print["Решение не найдено"]];
условный оператор   печатать
"Число итераций, за которое было найдено решение с точностью e=10^-2"
Print[iter]
печатать
"Найденное решение"
PaddedForm[x, {9, 8}]
форма числа с заполнением нулями
```

Решение для точности e=10^-2

```
Out[1515]=
        Число итераций, за которое было найдено решение с точностью e=10^-2

        8

Out[1517]=
        Найденное решение

Out[1518]//PaddedForm=
        {-0.38083441,  1.10319200,  0.91111717}

Out[1519]=
        Решение с помощью встроенных функций

Out[1520]//PaddedForm=
        {-0.37038850,  1.09380980,  0.90323083}
```

Решение для точности e=10^-4

```
Out[1528]=
        Число итераций, за которое было найдено решение с точностью e=10^-4

        14

Out[1530]=
        Найденное решение

Out[1531]//PaddedForm=
        {-0.37049937, 1.09394258, 0.90329780}

Out[1532]=
        Решение с помощью встроенных функций

Out[1533]//PaddedForm=
        {-0.37038850, 1.09380980, 0.90323083}
```

Решение для точности e=10^-6

```
Out[1541]=
        Число итераций, за которое было найдено решение с точностью e=10^-6

        20

Out[1543]=
        Найденное решение

Out[1544]//PaddedForm=
        {-0.37038983, 1.09381145, 0.90323161}

Out[1545]=
        Решение с помощью встроенных функций

Out[1546]//PaddedForm=
        {-0.37038850, 1.09380980, 0.90323083}
```

Теперь используем метод Зейделя

Листинг программы, реализующей метод:

```python
import math
import copy


def seidel(a: list, b: list, e: float):
    n = len(b)
    x = copy.deepcopy(b)

    coverge = False
    iters = 0
    while not coverge:
        x_new = copy.deepcopy(x)
        iters += 1
        for i in range(n):
            s1 = sum(a[i][j] * x_new[j] for j in range(i))
            s2 = sum(a[i][j] * x[j] for j in range(i+1, n))
            x_new[i] = (b[i] - s1 - s2)/a[i][i]

        coverge = math.sqrt(sum((x_new[i] - x[i]) ** 2 for i in range(n))) <= e
        x = x_new

    return iters, x
```

Решение для точности e=10^-2

```
Amount of iterations: 6, x = [-0.37257461233389183, 1.0952666282466559, 0.9033672457798914]
```

Решение для точности e=10^-4

```
Amount of iterations: 10, x = [-0.37039694125110106, 1.0938154253350953, 0.9032313580160792]
```

Решение для точности e=10^-6

```
Amount of iterations: 13, x = [-0.37038863148734374, 1.0938098864505694, 0.9032308390150867]
```

Как видно, во всех случаях метод Зейделя справился с решением за меньшее количество итераций, по сравнению с методом Якоби(6<8, 10<14, 13<20)

2. Задание 2:

Для начала посмотрим на получившиеся матрицы A, B и решим систему с помощью встроенных функций(значение n = 10)

Out[1452]=

Формируем матрицу

Out[1456]=

Вывод получившихся матриц

Out[1457]//MatrixForm=

$$\begin{pmatrix} 20 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 20 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 20 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 20 & 1 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 20 & 1 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 20 & 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 20 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 20 & 1 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 20 & 1 \\ 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 20 \end{pmatrix}$$

Out[1458]//MatrixForm=

$$\begin{pmatrix} 161 \\ 180 \\ 199 \\ 218 \\ 237 \\ 256 \\ 275 \\ 294 \\ 313 \\ 332 \end{pmatrix}$$

Out[1459]=

Решение с помощью встроенной функции

Out[1460]//PaddedForm=

{ 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}

Теперь решим данную систему с помощью метода Якоби(точность e=10^-4)

```
Число итераций, за которое было найдено решение с точностью e=10^-4

20

Найденное решение
```

/PaddedForm=

$$\left\{\frac{199997}{50000}, \frac{249997}{50000}, \frac{299997}{50000}, \frac{349997}{50000}, \frac{399997}{50000}, \frac{449997}{50000}, \frac{499997}{50000}, \frac{549997}{50000}, \frac{599997}{50000}, \frac{649997}{50000}\right\}$$

```
Решение с помощью встроенных функций
```

/PaddedForm=

```
{ 4, 5, 6, 7, 8, 9, 10, 11, 12, 13}
```

Как видно, если все элементы в полученном решении округлить до целых то решение будет правильным. Возможно, подобный вид решения(представление решения в виде обыкновенных дробей) обуславливается особенностями пакета Mathematica

Решение системы с помощью метода Зейделя

```
Amount of iterations: 9, x = [4.000000938170884, 5.000000563066775,
6.000000213602832, 6.999999952135093, 7.999999799388175,
8.999999745748243, 9.999999764727525, 10.99999982492563,
11.99999989838108, 12.999999964992687]
```

Как видно, и в этом случае метод Зейделя оказался быстрее метода Якоби.

Вывод:

В ходе лабораторной работы мы познакомились с понятием нормы матрицы, число обусловленности матрицы и методом решения СЛАУ при помощи итерационных методов Якоби и Зейделя. Также были получены знания по решению СЛАУ при помощи пакета MATHEMATICA. Было обнаружено, что метод Зейделя является более выгодным с точки зрения количества итераций, чем метод Якоби.