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 №2
на тему:

«Численное решение систем линейных уравнений методом простых итераций и методом Зейделя»

Выполнил: студент группы 353502
Згирская Дарья Денисовна
Проверил: Анисимов Владимир Яковлевич

Минск 2024

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.
- Численно решить тестовые примеры и проверить правильность работы программы. Сравнить трудоемкость решения методом простых итераций и методом Зейделя.

ЗАДАНИЕ

Вариант 5. 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 (рис. 1).

k=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 $\boldsymbol{b}$= (1,2; 2,2; 4,0; 0,0; -1,2)^{T}.

Рисунок 1 – Исходные данные

ХОД РАБОТЫ

Перед началом выполнения работы были изучены теоретические материалы на тему «Численное решение систем линейных уравнений методом простых итераций и методом Зейделя».

Далее для создания программы решения СЛАУ был разработан алгоритм для обоих методов: метода простых итераций и метода Зейделя.

1 Метод простых итераций (рис. 2).

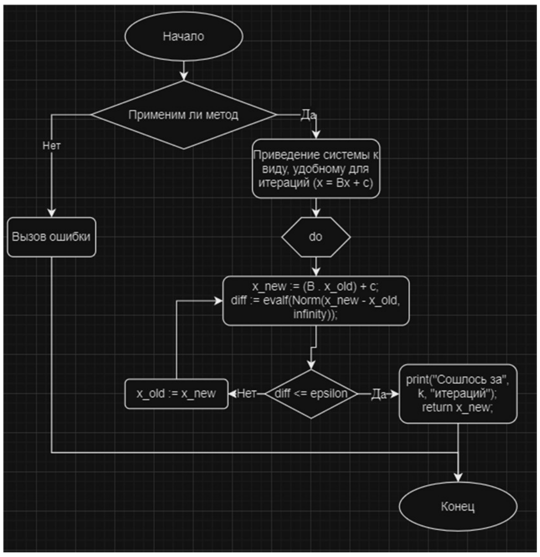

Рисунок 2 – Алгоритм метода простых итераций

2 Метод Зейделя (рис. 3).

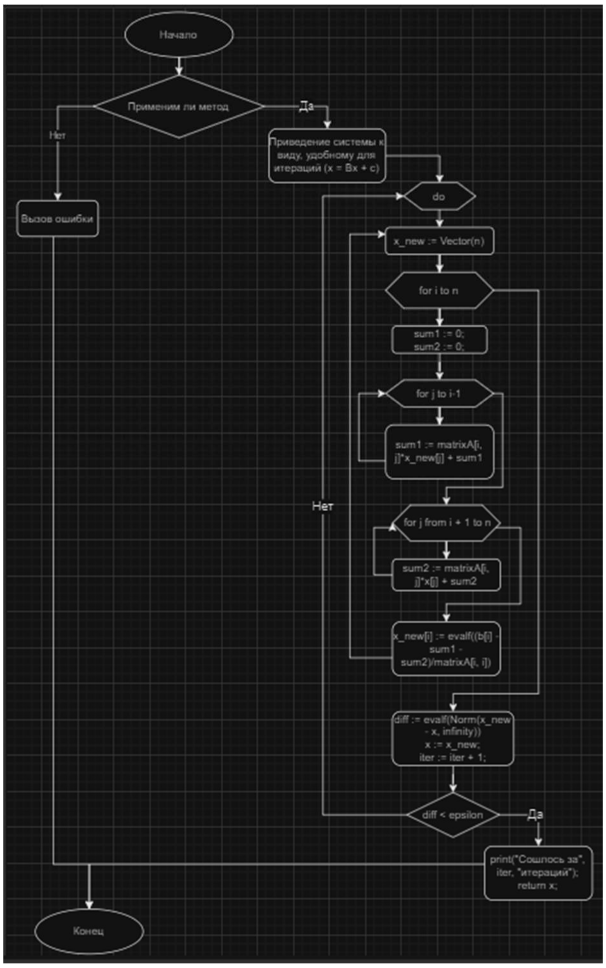

Рисунок 3 – Алгоритм метода Зейделя

После этого была составлена программа решения СЛАУ по составленным алгоритмам, причем каждый алгоритм реализован в виде процедуры.

1 Процедура, реализующая метод простых итераций (рис. 4).

```
> simple_iteration_method := proc(matrixA, b, x0, epsilon)
    local n, B, c, x_new, x_old, i, j, k, diff, check_solution_result, check_diag_zero_result;

   n := RowDimension(matrixA);
   B := Matrix(n, n);
   c := Vector(n);
   check_solution_result := check_system_on_solution(matrixA, b);

   if check_solution_result = 1 then
     check_diag_zero_result := check_matrix_on_zero_diag_el(matrixA, b);

     if check_diag_zero_result = 0 then
        for i from 1 to n do
          for j from 1 to n do
           if i ≠ j then
             B[i,j] := - matrixA[i,j] / matrixA[i, i];
           else
             B[i,j] := 0;
           end if;
          end do;
          c[i] := b[i] / matrixA[i, i];
        end do;

       x_old := x0;
       k := 0;
       do
         x_new := B · x_old + c;
         diff := evalf(Norm(x_new - x_old, infinity));
         k := k + 1;

         if diff ≤ epsilon then
           print("Сошлось за", k, "итераций");
           return x_new;
         end if;

         x_old := x_new;
       end do;

     else
       error "Основная матрица системы имеет хотя бы 1-ин нулевой элемент на диагонали. Следовательно, метод невозмлжно применить"
     end if;
```

Рисунок 4

2 Процедура, реализующая метод Зейделя (рис. 5).

```
zeidel_method := proc(matrixA, b, x0, epsilon)
   local x, x_new, iter, i, j, sum1, sum2, diff, n, check_solution_result, check_diag_zero_result;

  n := RowDimension(matrixA);
  x := x0;
  iter := 0;
  check_solution_result := check_system_on_solution(matrixA, b);

 if check_solution_result = 1 then
     check_diag_zero_result := check_matrix_on_zero_diag_el(matrixA, b);

     if check_diag_zero_result = 0 then
       do
         x_new := Vector(n);

         for i to n do
           sum1 := 0;
           sum2 := 0;

         for j to i - 1 do
            sum1 := sum1 + matrixA[i,j]·x_new[j];
         end do;
         for j from i + 1 to n do
            sum2 := sum2 + matrixA[i,j]·x[j];
         end do;

         x_new[i] := evalf((b[i] - sum1 - sum2) / matrixA[i, i]);
       end do;

       diff := evalf(Norm(x_new - x, infinity));
       x := x_new;
       iter := iter + 1;

       if diff < epsilon then
         print("Сошлось за", iter, "итераций");
         return x;
       end if;
     end do;

     else
       error "Основная матрица системы имеет  хотя бы 1—ин нулевой элемент на диагонали. Следовательно, метод невозможно применить"
     end if;
```

Рисунок 5

Для корректной работы реализованных алгоритмов дополнительно была создана процедура, проверяющая расширенную матрицу на совместность (рис. 6).

```
> check_system_on_solution := proc(matrixA, b)
    local n, matrix_rank, system, system_rank;

    n := RowDimension(matrixA);
    system := ⟨matrixA|b⟩;

    matrix_rank := Rank(matrixA);
    system_rank := Rank(system);

    if matrix_rank = system_rank then
        print("Система совместна");

        if system_rank < n then
          print("Система имеет бесконечно много решений");
          return 2;
        else
          print("Система имеет единственное решение");
          return 1;
        end if;
    else
        print("Система несовместна");
        return 0;
    end if;
  end proc:
```

Рисунок 6

Эта процедура также выводит информацию о том, имеет ли система единственное решение или нет. Она также учитывает то, что основная матрица системы может содержать нулевой столбец.

Еще была создана процедура, которая проверят основную матрицу системы на наличие нулевых элементов на главной диагонали (рис. 7).

```
> check_matrix_on_zero_diag_el := proc (matrixA, b)
    local i, n;

    n := RowDimension(matrixA);
    for i from 1 to n do
      if matrixA[i, i] = 0 then
        print("Основная матрица системы имеет нулевой элемент на главной диагонали");
        return 1;
      end if;
    end do;

    return 0;
  end proc:
```

Рисунок 7

ТЕСТОВЫЕ ПРИМЕРЫ

Тестовый пример 1. Несовместная система (рис. 8).

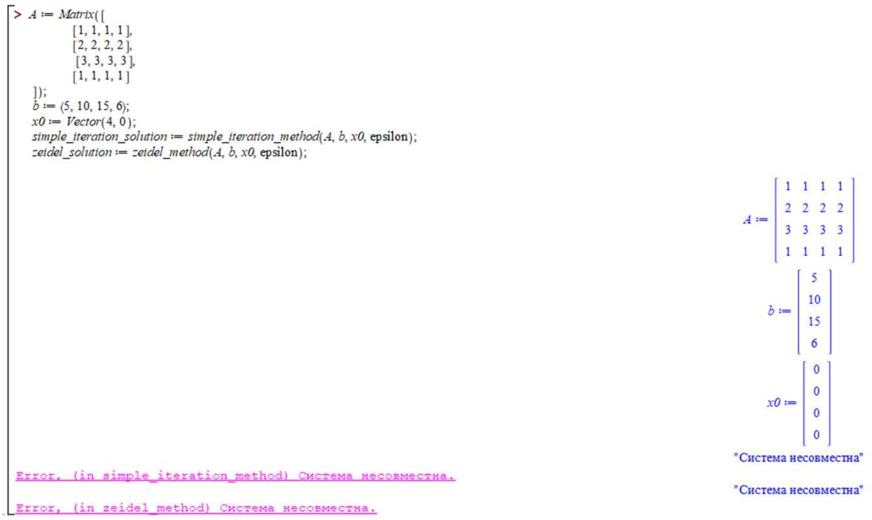

Рисунок 8

Тестовый пример 2. Совместная система с бесконечным множеством решений (рис. 9).

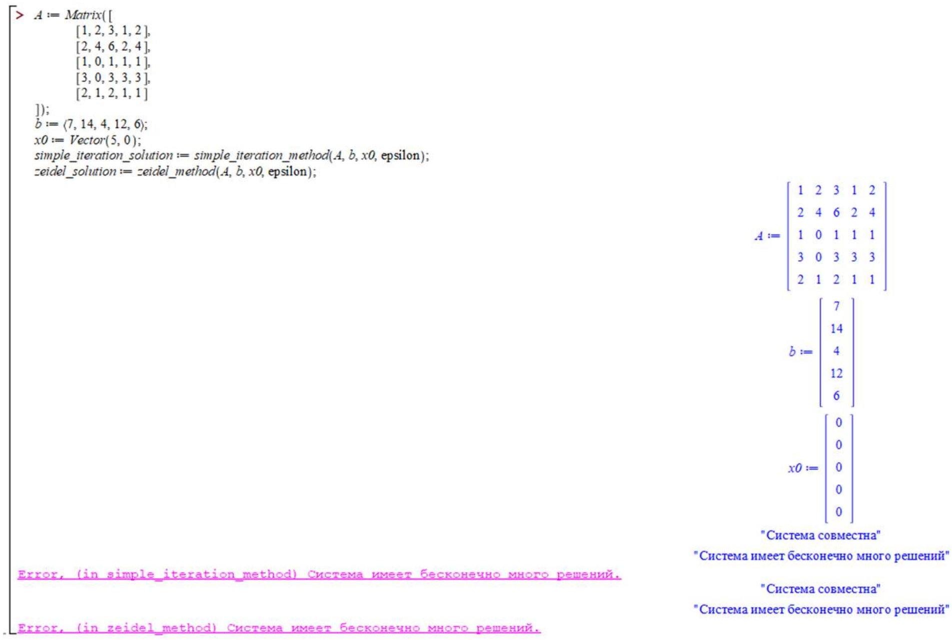

Рисунок 9

Тестовый пример 3. Совместная система, основная матрица которой имеет нулевые элементы на главной диагонали (рис. 10).

```
> A := Matrix([
        [1, 2, 3, 1],
        [2, 0, 5, 1],
        [1, 1, 0, 2],
        [3, 4, 1, 1]
  ]);
  b := ⟨4, 5, 6, 7⟩;
  x0 := Vector(4, 0);
  simple_iteration_solution := simple_iteration_method(A, b, x0, epsilon);
  zeidel_solution := zeidel_method(A, b, x0, epsilon);
```

$$A := \begin{bmatrix} 1 & 2 & 3 & 1 \\ 2 & 0 & 5 & 1 \\ 1 & 1 & 0 & 2 \\ 3 & 4 & 1 & 1 \end{bmatrix}$$

$$b := \begin{bmatrix} 4 \\ 5 \\ 6 \\ 7 \end{bmatrix}$$

$$x0 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

"Система совместна"
"Система имеет единственное решение"
"Основная матрица системы имеет нулевой элемент на главной диагонали"

```
Error, (in simple_iteration_method) Основная матрица системы имеет хотя бы 1-ин нулевой элемент на диагонали. Следовательно, метод невозмлжно применить
```

"Система совместна"
"Система имеет единственное решение"
"Основная матрица системы имеет нулевой элемент на главной диагонали"

```
Error, (in zeidel_method) Основная матрица системы имеет  хотя бы 1-ин нулевой элемент на диагонали. Следовательно, метод невозможно применить
```

Рисунок 10

Тестовый пример 4. Совместная система с единственным решением (рис. 11).

```
b := ⟨10, 18, 16, 25⟩;
x0 := Vector(4, 0);
simple_iteration_solution := simple_iteration_method(A, b, x0, epsilon) :
simple_iteration_solution := evalf(simple_iteration_solution);
zeidel_solution := zeidel_method(A, b, x0, epsilon);
```

$$A := \begin{bmatrix} 6 & -2 & 1 & 2 \\ 3 & 9 & -1 & 3 \\ 2 & 4 & 8 & -2 \\ 1 & 3 & 2 & 7 \end{bmatrix}$$

$$b := \begin{bmatrix} 10 \\ 18 \\ 16 \\ 25 \end{bmatrix}$$

$$x0 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

"Система совместна"
"Система имеет единственное решение"
"Сошлось за", 17, "итераций"

$$simple\_iteration\_solution := \begin{bmatrix} 0.9049951362 \\ 1.086483985 \\ 1.843027943 \\ 2.449957627 \end{bmatrix}$$

"Система совместна"
"Система имеет единственное решение"
"Сошлось за", 8, "итераций"

$$zeidel\_solution := \begin{bmatrix} 0.9050097967 \\ 1.086461260 \\ 1.843002138 \\ 2.449943164 \end{bmatrix}$$

Рисунок 11

Тестовый пример 5. Совместная система с единственным решением (рис. 12).

```
> A := Matrix([
        [4, 2, 3, 1],
        [2, 5, 1, 2],
        [3, 1, 4, 2],
        [1, 3, 2, 4]
  ]);
  b := <20, 25, 18, 16>;
  x0 := Vector(4, 0);
  simple_iteration_solution := simple_iteration_method(A, b, x0, epsilon);
  zeidel_solution := zeidel_method(A, b, x0, epsilon);
```

$$A := \begin{bmatrix} 4 & 2 & 3 & 1 \\ 2 & 5 & 1 & 2 \\ 3 & 1 & 4 & 2 \\ 1 & 3 & 2 & 4 \end{bmatrix}$$

$$b := \begin{bmatrix} 20 \\ 25 \\ 18 \\ 16 \end{bmatrix}$$

$$x0 := \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

"Система совместна"

"Система имеет единственное решение"

"Система совместна"

"Система имеет единственное решение"

"Сошлось за", 48, "итераций"

$$zeidel\_solution := \begin{bmatrix} -0.4496229525 \\ 5.049834650 \\ 4.549626318 \\ -1.949783410 \end{bmatrix}$$

Рисунок 12

В примерах 4 и 5 обе системы являются совместными и имеют единственное решение. Однако для примера 4 сошлись оба реализованных метода, в то время как для примера 5 сошелся только метод Зейделя (за 48 итераций).

Это связано с тем, что СЛАУ в примере 4 обладает свойством диагонального доминирования (рис. 13), а СЛАУ в примере 5 – нет.

Матрица A системы линейных уравнений $Ax = b$ называется **диагонально доминирующей**, если для каждой строки i выполняется следующее условие:

$$|a_{ii}| > \sum_{\substack{j=1 \\ j \neq i}}^{n} |a_{ij}|$$

где:

- a_{ii} — элемент на главной диагонали матрицы (в i-й строке и i-ом столбце),
- $|a_{ij}|$ — абсолютное значение элементов в i-й строке (кроме диагонального элемента),
- n — количество переменных (размерность матрицы).

Рисунок 13

Тестовый пример 6. Совместная система с единственным решением (рис. 14).

```
> A := Matrix([
        [4, 1, 1],
        [2, 5, 2],
        [1, 2, 4]
  ]);
  b := <12, 22, 13>;
  x0 := Vector(3, 0);
  simple_iteration_solution := simple_iteration_method(A, b, x0, epsilon) :
  simple_iteration_solution := evalf(simple_iteration_solution);
  zeidel_solution := zeidel_method(A, b, x0, epsilon);
```

$$A := \begin{bmatrix} 4 & 1 & 1 \\ 2 & 5 & 2 \\ 1 & 2 & 4 \end{bmatrix}$$

$$b := \begin{bmatrix} 12 \\ 22 \\ 13 \end{bmatrix}$$

$$x0 := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

"Система совместна"
"Система имеет единственное решение"
"Сошлось за", 30, "итераций"

$$simple\_iteration\_solution := \begin{bmatrix} 1.912259158 \\ 3.157865024 \\ 1.192952983 \end{bmatrix}$$

"Система совместна"
"Система имеет единственное решение"
"Сошлось за", 8, "итераций"

$$zeidel\_solution := \begin{bmatrix} 1.912272346 \\ 3.157902468 \\ 1.192980680 \end{bmatrix}$$

Рисунок 14

Тестовый пример 7. Совместная система с единственным решением (рис. 15).

```
> A := Matrix([
        [1, 0, 1],
        [-1, 1, 0],
        [1, 2, -3]
  ]);
  b := <1, 2, 3>;
  x0 := Vector(3, 0);
  simple_iteration_solution := simple_iteration_method(A, b, x0, epsilon) :
  simple_iteration_solution := evalf(simple_iteration_solution);
  zeidel_solution := zeidel_method(A, b, x0, epsilon);
```

$$A := \begin{bmatrix} 1 & 0 & 1 \\ -1 & 1 & 0 \\ 1 & 2 & -3 \end{bmatrix}$$

$$b := \begin{bmatrix} 1 \\ 2 \\ 3 \end{bmatrix}$$

$$x0 := \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$$

"Система совместна"
"Система имеет единственное решение"
"Сошлось за", 173, "итераций"

$$simple\_iteration\_solution := \begin{bmatrix} 0.3333187130 \\ 2.333246598 \\ 0.6666002291 \end{bmatrix}$$

"Система совместна"
"Система имеет единственное решение"

Рисунок 15

В тестовом примере 7 метод простых итераций сходится, а метод Зейделя – нет.

ВЫВОДЫ

В ходе выполнения данной работы были изучены итерационные методы решения СЛАУ – метод простых итераций и метод Зейделя. Также были составлены алгоритмы решения СЛАУ изученными методами и реализована методы в соответствии с разработанными алгоритмами.

Было выполнено задание и решены 5 тестовых примеров, показывающих различные случаи СЛАУ (совместаная система с ед./бесконеными решениями, несовместная система, диагонально доминирующая система и система без этого свойства).

На основании полученных данных задания (метод простых итераций отработал за 9 итераций, а метод Зейделя – за 5 итераций) и тестовых примеров 4 (метод простых итераций отработал за 17 итераций, а метод Зейделя – за 8 итераций) и 6 (метод простых итераций отработал за 30 итераций, а метод Зейделя – за 8 итераций) можно подтвердить то, что метод Зейделя сходимость итераций ускоряется почти в 2 раза – или даже больше: $9/5 = 1.8 \approx 2$, $17/8 \approx 2.13 \approx 2$, $30/8 = 3.75 > 2$.