Министерство образования Республики Беларусь

Учреждение образования «Белорусский государственный университет

информатики и радиоэлектроники»

Факультет компьютерного проектирования

Кафедра Информатики

Дисциплина «Методы численного анализа»

**ОТЧЕТ**

к лабораторной работе №1

на тему:

**«РЕШЕНИЕ СИСТЕМ ЛИНЕЙНЫХ АЛГЕБРАИЧЕСКИХ УРАВНЕНИЙ (СЛАУ) МЕТОДОМ ГАУССА И С ПОМОЩЬЮ ЕГО МОДИФИКАЦИЙ»**

БГУИР 6-05-0612-02 005

|  |
| --- |
| Выполнила студент группы 353504  АНТОНОВА Лидия Сергеевна |
|  |
| (дата, подпись студента) |
| Проверил доцент каф. Информатики  АНИСИМОВ Владимир Яковлевич |
|  |
| (дата, подпись преподавателя) |

Минск 2024

# 1 ЦЕЛЬ ВЫПОЛНЕНИЯ ЗАДАНИЯ

1 Изучить метод Гаусса и его модификации, составить алгоритм метода и программу его реализации, получить численное решение заданной СЛАУ;

2 Составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ;

3 Составить программу решения СЛАУ по разработанному алгоритму;

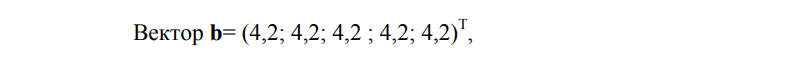
4 Выполнить тестовые примеры и проверить правильность работы программы.  
  
 **2 задание**

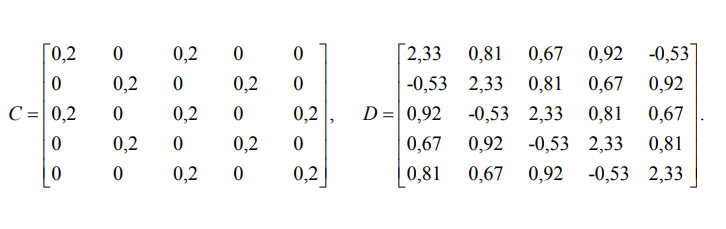
Вариант 1.

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание  
 Также необходимо найти относительную погрешность каждого из методов вычисления СЛАУ.

Исходные данные:

****

****

# 3 КРАТКИЕ ТЕОРЕТИЧЕСКИЕ СВЕДЕНИЯ

СЛАУ обычно записывается в виде:

или

где ; ; .

Здесь и заданы, требуется найти .

Методы решения СЛАУ делятся на прямые и итерационные. Прямые методы дают в принципе точное решение (если не учитывать ошибок округления) за конечное число арифметических операций. Они просты и наиболее универсальны. Для хорошо обусловленных систем небольшого порядка применяются практически только прямые методы.

Метод Гаусса прекрасно подходит для решения систем линейных алгебраических уравнений (СЛАУ). Он обладает рядом преимуществ по сравнению с другими методами:

1 Нет необходимости предварительно исследовать систему уравнений на совместность;

2 Методом Гаусса можно решать не только СЛАУ, в которых число уравнений совпадает с количеством неизвестных переменных и основная матрица системы невырожденная, но и системы уравнений, в которых число уравнений не совпадает с количеством неизвестных переменных или определитель основной матрицы равен нулю;

3 Метод Гаусса приводит к результату при сравнительно небольшом количестве вычислительных операций.

Метод Гаусса включает в себя прямой (приведение расширенной матрицы к ступенчатому виду, то есть получение нулей под главной диагональю) и обратный (получение нулей над главной диагональю расширенной матрицы) ходы. Прямой ход и называется методом Гаусса, обратный - методом Гаусса-Жордана, который отличается от первого только последовательностью исключения переменных.

Метод Гаусса идеально подходит для решения систем, содержащих больше трех линейных уравнений, для решения систем уравнений, которые не являются квадратными (чего не скажешь про метод Крамера и матричный метод). То есть метод Гаусса - наиболее универсальный метод для нахождения решения любой системы линейных уравнений, он работает в случае, когда система имеет бесконечно много решений или.

Вычисления с помощью метода Гаусса заключаются в последовательном исключении неизвестных из системы для преобразования ее к эквивалентной системе с верхней треугольной матрицей. Вычисления значений неизвестных производят на этапе обратного хода.

Рассмотрим сначала простейший вариант метода Гаусса, называемый *схемой единственного деления*.

Прямой ход состоит из шагов исключения.

1-й шаг. Целью этого шага является исключение неизвестного из уравнений с номерами . Предположим, что коэффициент . Будем называть его главным элементом 1-го шага.

Найдем величины:

называемые *множителями 1-го шага*. Вычтем последовательно из второго, третьего, …, -го уравнений системы первое уравнение, умноженное соответственно на . Это позволит обратить в нуль коэффициенты при x1 во всех уравнениях, кроме первого. В результате получим эквивалентную систему

в которой и вычисляются по формулам

2-й шаг. Целью этого шага является исключение неизвестного из уравнений с номерами . Пусть , где – коэффициент, называемый *главным* (или *ведущим*) *элементом 2-го шага*. Вычислим множители 2-го шага:

и вычтем последовательно из третьего, четвертого, …, -го уравнений системы второе уравнение, умноженное соответственно на , , …, . В результате получим систему:

Здесь коэффициенты и вычисляются по формулам:

Аналогично проводятся остальные шаги. Опишем очередной -й шаг.

-й шаг. В предположении, что *главный* (*ведущий*) *элемент k-го шага* отличен от нуля, вычислим *множители k-го шага*:

и вычтем последовательно из -го, …, -го уравнений полученной на предыдущем шаге системы -e уравнение, умноженное соответственно на ,k, …, .

После -го шага исключения получим систему уравнений:

матрица которой является верхней треугольной. На этом вычисления прямого хода заканчиваются.

Обратный ход. Из последнего уравнения системы находим. Подставляя найденное значение в предпоследнее уравнение, получим . Осуществляя обратную подстановку, далее последовательно находим ,…, . Вычисления неизвестных здесь проводятся по формулам:

Необходимость отличия от 0 главных элементов. Заметим, что вычисление множителей, а также обратная подстановка требуют деления на главные элементы . Поэтому если один из главных элементов оказывается равным нулю, то схема не может быть реализована. Здравый смысл подсказывает, что и в ситуации, когда все главные элементы отличны от нуля, но среди них есть близкие к нулю, возможен неконтролируемый рост погрешности.

**Метод Гаусса с выбором главного элемента по столбцу (схема частичного выбора).** На -м шаге прямого хода коэффициенты уравнений системы с номерами преобразуются по формулам:

Интуитивно ясно, что во избежание сильного роста коэффициентов системы и связанных с этим ошибок нельзя допускать появления больших множителей .

В методе Гаусса с выбором главного элемента по столбцу гарантируется, что для всех и . Отличие этого варианта метода Гаусса от схемы единственного деления заключается в том, что на -м шаге исключения в качестве главного элемента выбирают максимальный по модулю коэффициент при неизвестной в уравнениях с номерами . Затем соответствующее выбранному коэффициенту уравнение с номером меняют местами с -м уравнением системы для того, чтобы главный элемент занял место коэффициента . После этой перестановки исключение неизвестного производят, как в схеме единственного деления.

**Метод Гаусса с выбором главного элемента по всей матрице (схема полного выбора).** В этой схеме допускается нарушение естественного порядка исключения неизвестных. На 1-м шаге метода среди элементов определяют максимальный по модулю элемент . Первое уравнение системы и уравнение с номером меняют местами. Далее стандартным образом производят исключение неизвестного из всех уравнений, кроме первого.

На -м шаге метода среди коэффициентов при неизвестных в уравнениях системы с номерами выбирают максимальный по модулю коэффициент . Затем -е уравнение и уравнение, содержащее найденный коэффициент, меняют местами и исключают неизвестное из уравнений с номерами .

На этапе обратного хода неизвестные вычисляют в следующем порядке: .

# 4 Выполнение работы

При написании задания данной лабораторной работы использовалась система компьютерной алгебры Maple 2021. Для решения СЛАУ необходимо было использовать метод Гаусса и 2 его модификации.

Изначально необходимо реализовать функцию для нахождения матрицы A.

      findA := proc(c::Matrix, d::Matrix, k)

local a;

a := c . d;

a := a\*k;

return a;

end proc;

Также необходимо реализовать функцию, проверяющую совпадает ли количество строк матрицы 𝑎 с количеством элементов вектора 𝑏. Если размеры не совпадают, функция генерирует ошибку с сообщением: "Number of rows of the matrix must be equal to number of elements of the vector".

Check := proc(a::Matrix, b::Vector)

local n;

n := RowDimension(a);

if n <> ColumnDimension(a) then

error "Matrix must be square";

elif n <> RowDimension(b) then

error "Number of rows of the matrix must be equal to number of elements of the vector";

end if;

end proc:

Теперь реализуем метод Гаусса, который находит q для каждого главного элемента, приводит матрицу к треугольному виду и с помощью обратного хода находит корни. Листинг представлен ниже.

MyGaussianEliminationWithPivoting := proc(A::Matrix, B::Vector)

local n, i, j, k, q, x, maxIndex, maxValue, temp;

n := RowDimension(A);

for k to n - 1 do

maxValue := abs(A[k, k]);

maxIndex := k;

for i from k + 1 to n do

if maxValue < abs(A[i, k]) then

maxValue := abs(A[i, k]);

maxIndex := i;

end if;

end do;

if maxValue = 0 then

error "Невозможно продолжать: все элементы в столбце равны нулю.";

end if;

if maxIndex <> k then

temp := Row(A, k);

Row(A, k) := Row(A, maxIndex);

Row(A, maxIndex) := temp;

temp := B[k];

B[k] := B[maxIndex];

B[maxIndex] := temp;

end if;

for i from k + 1 to n do

q := A[i, k]/A[k, k];

A[i, k .. n] := A[i, k .. n] - q\*A[k, k .. n];

B[i] := B[i] - q\*B[k];

end do;

end do;

x := Vector(n);

for k from n by -1 to 1 do

if A[k, k] = 0 then

if B[k] = 0 then

error "Ошибка при обратном ходе: система имеет бесконечное множество решений.";

else

error "Ошибка при обратном ходе: система не имеет решений.";

end if;

end if;

x[k] := (B[k] - add(A[k, j]\*x[j], j = k + 1 .. n))/A[k, k];

end do;

return x;

end proc:

Реализуем метод Гаусса с выбором главного элемента по столбцу. Он практически идентичен классического методу Гаусса, отличием является нахождение строки с большим по модулю главным элементом и заменой первой строки на эту.

MyGaussianEliminationWithPivoting := proc(A::Matrix, B::Vector)

local n, i, j, k, q, x, maxIndex, maxValue, temp;

n := RowDimension(A);

for k to n - 1 do

maxValue := abs(A[k, k]);

maxIndex := k;

for i from k + 1 to n do

if maxValue < abs(A[i, k]) then

maxValue := abs(A[i, k]);

maxIndex := i;

end if;

end do;

if maxValue = 0 then

error "Невозможно продолжать: все элементы в столбце равны нулю.";

end if;

if maxIndex <> k then

temp := Row(A, k);

Row(A, k) := Row(A, maxIndex);

Row(A, maxIndex) := temp;

temp := B[k];

B[k] := B[maxIndex];

B[maxIndex] := temp;

end if;

for i from k + 1 to n do

q := A[i, k]/A[k, k];

A[i, k .. n] := A[i, k .. n] - q\*A[k, k .. n];

B[i] := B[i] - q\*B[k];

end do;

end do;

x := Vector(n);

for k from n by -1 to 1 do

if A[k, k] = 0 then

if B[k] = 0 then

error "Ошибка при обратном ходе: система имеет бесконечное множество решений.";

else

error "Ошибка при обратном ходе: система не имеет решений.";

end if;

end if;

x[k] := (B[k] - add(A[k, j]\*x[j], j = k + 1 .. n))/A[k, k];

end do;

return x;

end proc:

Реализуем метод Гаусса с выбором главного элемента по всей матрице. Он также практически идентичен предыдущим: первая строка заменяется на строку с большим по модулю главным элементом, находятся значения q для главным элементов, матрица приводится к треугольному виду, находятся корни. Листинг представлен ниже.

GaussianEliminationFullPivoting := proc(A::Matrix, b::Vector)

local n, i, j, k, maxRow, maxCol, temp, q, x, P, Q, X;

n := LinearAlgebra:-RowDimension(A);

P := Vector(n, i -> i);

Q := Vector(n, i -> i);

for k to n - 1 do

if A[k, k] = 0 then

error "Главный элемент равен нулю, метод не может быть применен";

end if;

maxRow := k;

maxCol := k;

for i from k to n do

for j from k to n do

if abs(A[maxRow, maxCol]) < abs(A[i, j]) then

maxRow := i;

maxCol := j;

end if;

end do;

end do;

if maxRow <> k then

temp := A[k, () .. ()];

A[k, () .. ()] := A[maxRow, () .. ()];

A[maxRow, () .. ()] := temp;

temp := b[k];

b[k] := b[maxRow];

b[maxRow] := temp;

temp := P[k];

P[k] := P[maxRow];

P[maxRow] := temp;

end if;

if maxCol <> k then

temp := A[() .. (), k];

A[() .. (), k] := A[() .. (), maxCol];

A[() .. (), maxCol] := temp;

temp := Q[k];

Q[k] := Q[maxCol];

Q[maxCol] := temp;

end if;

for i from k + 1 to n do

q := A[i, k]/A[k, k];

for j from k to n do

A[i, j] := A[i, j] - q\*A[k, j];

end do;

b[i] := b[i] - q\*b[k];

end do;

end do;

x := Vector(n);

for i from n by -1 to 1 do

x[i] := b[i];

for j from i + 1 to n do

x[i] := -x[j]\*A[i, j] + x[i];

end do;

x[i] := x[i]/A[i, i];

end do;

x := Vector(n, i -> x[Q[i]]);

X := Vector(RowDimension(x));

for i to RowDimension(x) do

X[i] := x[RowDimension(x) - i + 1];

end do;

return X;

end proc:

Также для сверки значений будет использована встроенная функция, с помощью которой будем находить значение корней. Листинг представлен ниже.

X := Vector[column](5, [x1, x2, x3, x4, x5]);

MX := A . X;

Maple\*solution;

Vector([seq(rhs(fsolve({MX(1) = B(1), MX(2) = B(2), MX(3) = B(3), MX(4) = B(4), MX(5) = B(5)}, {x1, x2, x3, x4, x5})[i]), i = 1 .. 5)]);

Root := Vector([seq(rhs(fsolve({MX(1) = B(1), MX(2) = B(2), MX(3) = B(3), MX(4) = B(4), MX(5) = B(5)}, {x1, x2, x3, x4, x5})[i]), i = 1 .. 5)]);

# 5 АНАЛИЗ ПОЛУЧЕННЫХ РЕЗУЛЬТАТОВ

Первый тестовый пример – условие задания. На рисунке 1 представлен результат вычисления матрицы A.

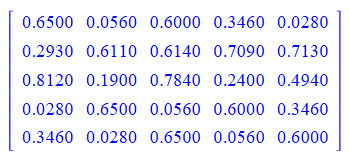


Рисунок 1 – Результат вычисления матрицы A

На рисунке 2 представлены решения, найденные с помощью описанных выше методов.

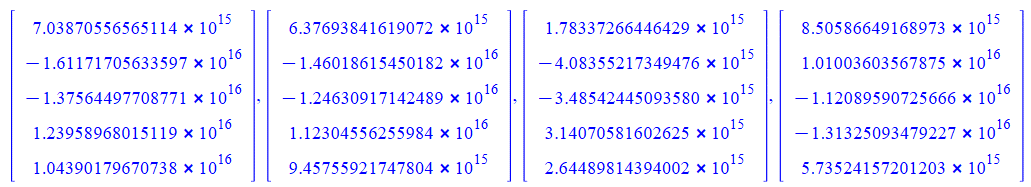


Рисунок 2 – Результат выполнения

Основываясь на рассчитанных решениях, можно сделать вывод, что все решения, использованные в данной работе, дают одинаковые решение с точностью 0,001.

Но это неверно для самого первого тестового примера из-за погрешности при расчётах значений, близких к нулю, и округлениях.   
 Особенно заметны округления во втором и первом способах. Это из-за того, что во втором способе |qik| ≤ 1. В первом же случае невозможно дать точных ограничений. Основываясь на результатах, можно сказать, что третий способ наиболее точный, но точность результатов относительно метода fsolve все равно низкая. На самом деле, причина остаётся той же, что и в случае первого и второго методов. Для хранения чисел с большим количеством неповторяющихся чисел необходимо не учитывать некоторые разряды. В нашем случае это критично.

Погрешности при расчетах. Рассчитаем абсолютную и относительную погрешность для первого метода при нахождении матрицы А.

n := RowDimension(A);

absolute\_errors := Vector(n);

absolute\_errors := abs((A . X1) - B);

relative\_errors := Vector(n);

for i to n do

if B <> 0 then

relative\_errors[i] := absolute\_errors[i]/abs(B[i]);

else

relative\_errors[i] := 0;

end if;

end do;

Полученная относительная погрешность:

Изображение выглядит как текст, Шрифт, Цвет электрик, снимок экрана

Автоматически созданное описание

Погрешность для других двух методов вычисляется аналогично. В итоге получаем следующее:

Изображение выглядит как текст, Шрифт, Цвет электрик, снимок экрана

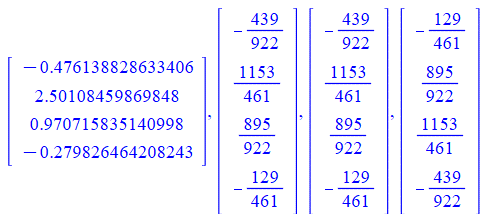
Автоматически созданное описаниеИзображение выглядит как текст, Шрифт, снимок экрана, Цвет электрик

Автоматически созданное описание

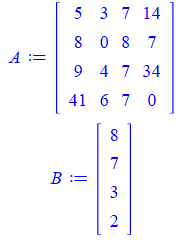
Для следующих тестовых примеров будут выбраны произвольные значения элементов матрицы A. Начальные условия и результат выполнения представлены ниже.

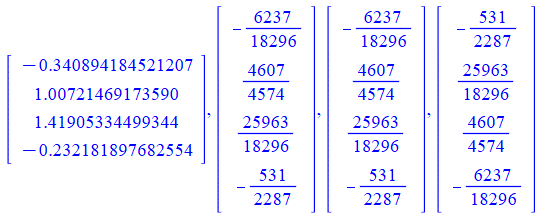
Изображение выглядит как диаграмма, линия, снимок экрана, Шрифт

Автоматически созданное описание

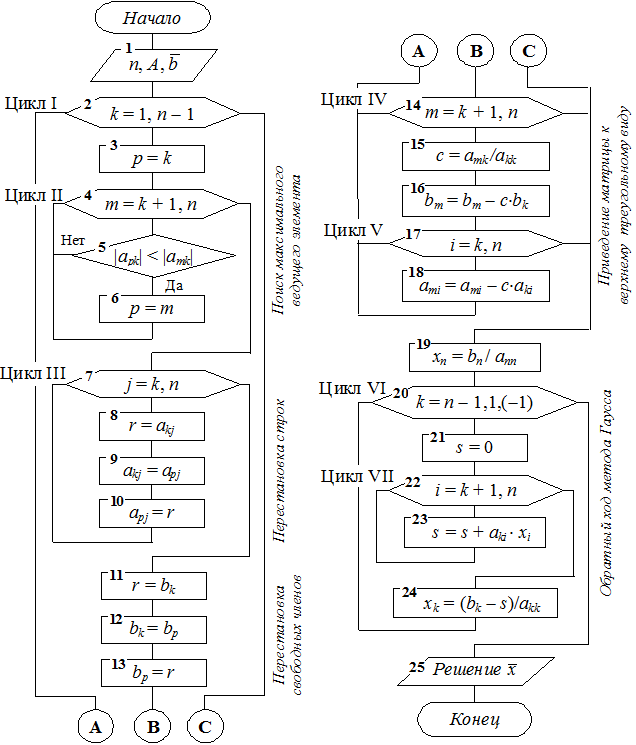
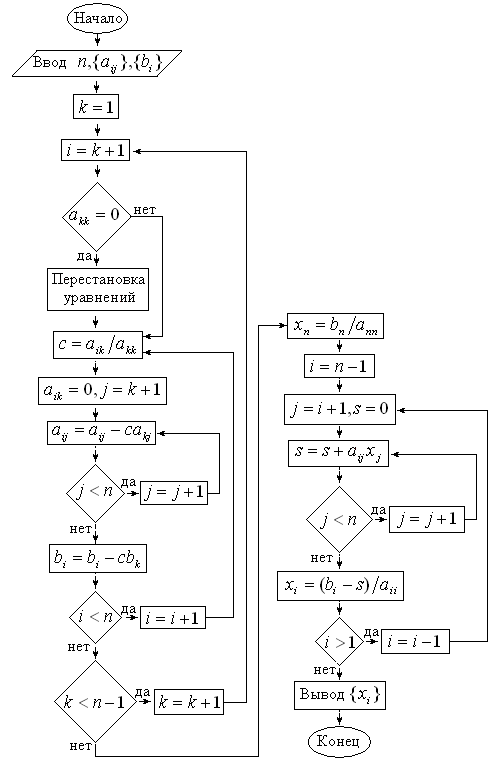


В следующем примере попробуем в матрице А несколько элементов главной диагонали заполнить нулями.



****

# 6 Блок-схемы алгоритмов



# Вывод

Таким образом, в ходе лабораторной работы был применен метод Гаусса, метод Гаусса с выбором главного элемента по столбцу и метод Гаусса с выбором главного элемента по всей матрице для решения системы линейных уравнений, составлены алгоритмы методов и программа их реализации, получено численное решение заданной СЛАУ. А также составлен алгоритм решения СЛАУ указанными методами, применимый для организации вычислений в Maple, составлена программа решения СЛАУ по разработанному алгоритму. Была проверена правильность выполнения на основе нескольких тестовых значениях. А также были проанализированы полученные результаты, на основе которых были сделаны выводы о погрешностях при вычислениях, связанных с проблемой хранения данных большой точности в Maple, из-за чего Maple вынужден обрезать некоторые значения для дальнейшей возможности хранения.