Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №1

Решение систем линейных алгебраических уравнений (СЛАУ)   
методом Гаусса и с помощью его модификаций

Выполнил:

студент группы 153503

Щиров П.Д.

Руководитель:

доцент

Анисимов В.Я.

Минск 2021

Содержание

[**1.** **Цель работы** 3](#_Toc113794418)

[**2.** **Теоретические сведения** 3](#_Toc113794419)

[**3.** **Программная реализация** 7](#_Toc113794429)

[**4.** **Выводы** 9](#_Toc113794430)

[**5. Код программы** 9](#_Toc113794431)

1. **Цель работы**
2. Изучить метод Гаусса и его модификации, составить алгоритм метода и программу его реализации, получить численное решение заданной СЛАУ;
3. Составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ;
4. Составить программу решения СЛАУ по разработанному алгоритму;
5. Выполнить тестовые примеры и проверить правильность работы программы;
6. **Теоретические сведения**

Задача отыскать решения СЛАУ с n неизвестными является одной из наиболее часто встречающихся вычислительных задач. Хотя задача решения СЛАУ сравнительно редко представляет самостоятельней интерес для приложений, от умения эффективно решать такие системы часто зависит сама возможность математического моделирования с применением ЭВМ разнообразных процессов. Значительная часть численных методов решения различных по своей природе задач (в особенности – нелинейных) включает в себя решение систем линейных уравнений как элементарный шаг соответствующего алгоритма.

СЛАУ обычно записывается в виде

, или коротко Ax=b, где

A = ; a = ; b = .

Здесь А и b заданы и требуется найти x.

Методы решения СЛАУ делятся на прямые и итерационные.

Прямые методы дают в принципе точное решение за конечное число арифметических операций. Они просты и наиболее универсальны. Для хорошо обусловленной системы необходимого порядка n 200 применяются практически только прямые методы.

Наибольшее распространение среди методов получил метод Гаусса и его модификации.

**Метод Гаусса**

Одним из самых распространенных методов решения систем линейных уравнений является метод Гаусса. Этот метод (который также называют *методом последовательного исключения неизвестных*) известен в различных вариантах.

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.

**1. Схема единственного деления.**

Рассмотрим сначала простейший вариант метода Гаусса.

Прямой ход состоит из *n* − 1 шагов исключения.

1-й шаг. Целью этого шага является исключение неизвестного *x*1 из уравнений с номерами *i* = 2, 3, …, *n*. Предположим, что коэффициент *a*11 ≠ 0. Будем называть его *главным элементом* 1-*го шага*.

Найдем величины

*qi*1*= ai*1/*a*11 (*i =*2, 3, …,*n*),

называемые *множителями*1-*го шага*. Вычтем последовательно из второго, третьего, …, *n-*го уравнений системы первое уравнение, умноженное соответственно на *q2*1*, q*31*, …, qn*1. Это позволит обратить в нуль коэффициенты при *x*1 во всех уравнениях, кроме первого. В результате получим эквивалентную систему

*a*11*x*1 + *a*12*x*2 + *a*13*x*3 + … + *a*1*nxn*= *b*1,

*a*22(1)*x*2 + *a*23(1)*x*3 + … + *a*2*n*(1)*xn*= *b*2(1),

Начало формы

Конец формы

*a*32(1)*x*2 + *a*33(1)*x*3 + … + *a*3*n*(1)*xn*= *b*3(1) ,

. . . . . . . . . . . . . . .

*an*2(1)*x*2 + *an*3(1)*x*3 + … + *ann*(1)*xn*= *bn*(1) .

в которой *aij*(1) и *bij*(1) вычисляются по формулам

*aij*(1) = *aij − qi*1*a*1*j*, *bi*(1) = *bi − qi*1*b*1.

2-й шаг. Целью этого шага является исключение неизвестного *x*2 из уравнений с номерами *i =*3, 4, …, *n*. Пусть *a*22(1) ≠ 0, где *a*22(1) ­– коэффициент, называемый *главным* (или *ведущим*) *элементом*2-*го шага*. Вычислим множители 2-го шага

*qi*2 = *ai*2(1) / *a*22(1) (*i =*3, 4, …, *n*)

и вычтем последовательно из третьего, четвертого, …, *n-*го уравнения системы второе уравнение, умноженное соответственно на *q*32,*q*42, …,*qm*2.

В результате получим систему:

*a*11*x*1 + *a*12*x*2 + *a*13*x*3 + … + *a*1*nxn*= *b*1,

*a*22(1)*x*2 + *a*23(1)*x*3 + … + *a*2*n*(1) = *b*2(1),

*a*33(2)*x*3+ … + *a*3*n*(2)*xn* = *b*3(2),

. . . . . . . . . . . . . . . . . . .

*an*3(2)*x*3 + … + *ann*(2)*xn*= *bn*(2).

Здесь коэффициенты *aij*(2) и *bij*(2) вычисляются по формулам

*aij*(2) = *aij*(1) – *qi*2*a*2*j*(1) , *bi*(2) = *bi*(1) – *qi*2*b*2(1).

Аналогично проводятся остальные шаги. Опишем очередной *k-*й шаг.

*k-*й шаг. В предположении, что *главный*(*ведущий*) *элемент k-*го шага *akk*(*k*–1) отличен от нуля, вычислим *множители k-го шага*

*qik = aik*(*k*–1) / *akk*(*k*–1) (*i = k*+ 1, …, *n*)

и вычтем последовательно из (*k* + 1)-го, …, *n*-го уравнений полученной на предыдущем шаге системы *k*-e уравнение, умноженное соответственно на *qk*+1,*k*, *qk*+2,*k*, …, *qnk*.

После (*n -*1)-го шага исключения получим систему уравнений

*a*11*x*1 + *a*12*x*2 + *a*13*x*3 + … + *a*1*nxn* = *b*1,

*a*22(1)*x*2 + *a*23(1)*x*3 + … + *a*2*n*(1)*xn* = *b*2(1),

*a*33(2)*x*3 + … + *a*3*n*(2)*xn* = *b*3(2),

. . . . . . . . . . . . . . . . . . . .

*ann*(*n*–1)*xn* = *bn*(*n*–1).

матрица ***A***(*n*-1) которой является верхней треугольной. На этом вычисления прямого хода заканчиваются.

Обратный ход. Из последнего уравнения системы находим xn. Подставляя найденное значение *xn*в предпоследнее уравнение, получим *xn*–1. Осуществляя обратную подстановку, далее последовательно находим *xn*–1, *xn*–2, …, *x*1. Вычисления неизвестных здесь проводятся по формулам

*xn* = *bn*(*n*–1) / *ann*(*n*–1),

*xk* = (*bn*(*k*–1) – *ak*,*k*+1(*k*–1)*xk*+1 – … – *akn*(*k*–1)*xn*) / *akk*(*k*–1), (*k* = *n* – 1, …, 1).

Необходимость выбора главных элементов. Заметим, что вычисление множителей, а также обратная подстановка требуют деления на главные элементы *akk*(*k*–1). Поэтому если один из главных элементов оказывыется равным нулю, то схема единственного деления 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.

**2. Метод Гаусса с выбором главного элемента по столбцу (схема частичного выбора).**

  Описание метода. На *k*-м шаге прямого хода коэффициенты уравнений системы с номерами *i*= *k*+ 1, …, *n* преобразуются по формулам

*aij*(*k*) = *aij*(*k*–1)*− qikakj*, *bi*(*k*) = *bi*(*k*–1)*− qikbk*(*k*–1), *i* = *k* + 1, …, *n*.

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

В методе Гаусса с выбором главного элементоа по столбцу гарантируется, что |*qik*| ≤ 1 для всех *k*= 1, 2, …, *n* – 1 и *i* = *k* + 1, …, *n*. Отличие этого варианта метода Гаусса от схемы единственного деления заключается в том, что на *k*-м шаге исключения в качестве главного элемента выбирают максимальный по модулю коэффициент *aikk* при неизвестной *xk* в уравнениях с номерами *i = k* + 1, …, *n*. Затем соответствующее выбранному коэффициенту уравнение с номером *ik* меняют местами с *k*-м уравнением системы для того, чтобы главный элемент занял место коэффициента*akk*(*k*-1). После этой перестановки исключение неизвестного *xk* производят, как в схеме единственного деления.

**3. 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 мтода среди элементов *aij* определяют максимальный по модулю элемент *ai*1*j*1. Первое уравнение системы и уравнение с номером *i*1 меняют местами. Далее стандартным образом производят исключение неизвестного *xi*1 из всех уравнений, кроме первого.

На *k*-м шаге метода среди коэффициентов *aij*(*k*–1) при неизвестных в уравнениях системы с номерами *i* = *k*, …, *n* выбирают максимальный по модулю коэффициент *aikjk*(*k*-1). Затем *k*-е уравнение и уравнение, содержащее найденный коэффициент, меняют местами и исключают неизвестное *xjk* из уравнений с номерами *i* = *k* + 1, …, *n*.

На этапе обратного хода неизвестные вычисляют в следующем порядке: *xjn, xjn–*1*, …, xj*1.

# **Программная реализация**

**Вариант 7.**

Тестовый пример 1

Методом Гаусса найти с точностью 0,0001 численное решение СЛАУ:

Ответ:

|  |  |  |
| --- | --- | --- |
| Схема единственного деления | Схема частичного выбора | Схема полного выбора |
|  |  |  |

Тестовый пример 2

Методом Гаусса найти с точностью 0,0001 численное решение СЛАУ:

Ответ:

|  |  |  |
| --- | --- | --- |
| Схема единственного деления | Схема частичного выбора | Схема полного выбора |
|  |  |  |

**ЗАДАНИЕ**

**Вариант 15**

Методом Гаусса найти с точностью 0,0001 численное решение системы **Ax=b**:

Ответ:

|  |  |  |
| --- | --- | --- |
| Схема единственного деления | Схема частичного выбора | Схема полного выбора |
|  |  |  |

# **Выводы**

Таким образом, в ходе выполнения лабораторной работы мы изучили метод Гаусса и его модификации, составили алгоритм метода и программу его реализации, получили численное решение заданной СЛАУ, составили алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ, составили программу решения СЛАУ по разработанному алгоритму, выполнили тестовые примеры и проверили правильность работы программы.

# **5.Код программы**

import numpy

print("Gaussian elimination \n")

def input():

b = numpy.array([[4.2], [4.2], [4.2], [4.2], [4.2]])

C = numpy.array(

[

[0.2, 0.0, 0.2, 0.0, 0.0],

[0.0, 0.2, 0.0, 0.2, 0.0],

[0.2, 0.0, 0.2, 0.0, 0.2],

[0.0, 0.2, 0.0, 0.2, 0.0],

[0.0, 0.0, 0.2, 0.0, 0.2],

]

)

D = numpy.array(

[

[2.33, 0.81, 0.67, 0.92, -0.53],

[-0.53, 2.33, 0.81, 0.67, 0.92],

[0.92, -0.53, 2.33, 0.81, 0.67],

[0.67, 0.92, -0.53, 2.33, 0.81],

[0.81, 0.67, 0.92, -0.53, 2.33],

]

)

A = 15 \* C + D

return (A, b)

def nan(n):

v = numpy.zeros((n, 1))

print("V: ", v, end="\n")

v[:] = numpy.NaN

return v

def basechoice(A, b):

n = A.shape[0]

for i in range(n):

for k in range(i + 1, n):

if A[i, i] == 0.0:

return nan(n)

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

for j in range(n):

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

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

x = numpy.zeros((n, 1))

for i in range(n - 1, -1, -1):

for j in range(i + 1, n):

b[i] -= A[i, j] \* x[j]

if A[i, i] == 0.0:

return nan(n)

x[i] = b[i] / A[i, i]

return x

def partchoice(A, b):

n = A.shape[0]

for i in range(n):

maxi = i

for k in range(i, n):

if abs(A[k, i]) > abs(A[maxi, i]):

maxi = k

A[[i, maxi]] = A[[maxi, i]]

b[[i, maxi]] = b[[maxi, i]]

for k in range(i + 1, n):

if A[i, i] == 0.0:

return nan(n)

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

for j in range(n):

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

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

x = numpy.zeros((n, 1))

for i in range(n - 1, -1, -1):

for j in range(i + 1, n):

b[i] -= A[i, j] \* x[j]

if A[i, i] == 0.0:

return nan(n)

x[i] = b[i] / A[i, i]

return x

def fullchoice(A, b):

n = A.shape[0]

for i in range(n):

maxi = (i, 0)

for k in range(i, n):

for j in range(n):

if abs(A[k, j]) > abs(A[maxi[0], maxi[1]]):

maxi = (k, j)

A[[i, maxi[0]]] = A[[maxi[0], i]]

b[[i, maxi[0]]] = b[[maxi[0], i]]

for k in range(n):

if k != i:

if A[i, maxi[1]] == 0.0:

return nan(n)

q = A[k, maxi[1]] / A[i, maxi[1]]

for j in range(n):

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

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

x = numpy.zeros((n, 1))

for j in range(n):

maxi = (i, 0)

for i in range(n):

if abs(A[i, j]) > abs(A[maxi[0], maxi[1]]):

maxi = (i, j)

if A[maxi[0], maxi[1]] == 0.0:

return nan(n)

x[j] = b[maxi[0]] / A[maxi[0], maxi[1]]

return x

def output(A, x, b):

numpy.set\_printoptions(suppress=True, precision=4, floatmode="fixed")

print("A = ")

print(A)

print("x = ")

print(x.T)

print("Check: b = ")

print(b.T)

print(A.dot(x).T)

print("----------")

def test(method):

(A, b) = input()

x = method(A.copy(), b.copy())

output(A, x, b)

test(basechoice)

test(partchoice)

test(fullchoice)