Лабораторная работа №4

Научное программирование

Николаев Дмитрий Иванович, НПМмд-02-24

Содержание

Список иллюстраций

Список таблиц

1 Цель работы

Изучение методов решения систем линейных уравнений, включая метод Гаусса, LU-разложение и LUP-разложение, а также их программная реализация.

2 Теоретическое введение

2.1 Метод Гаусса

Запишем исходную систему:

$$\begin{cases} a_{11}x_1 + \dots + a_{1n}x_n = b_1 \\ \vdots \\ a_{m1}x_1 + \dots + a_{mn}x_n = b_m \end{cases}$$

в матричном виде:

$$Ax = b,$$

где

$$A = \begin{pmatrix} a_{11} & \dots & a_{1n} \\ \vdots & \ddots & \vdots \\ a_{m1} & \dots & a_{mn} \end{pmatrix}, \quad x = \begin{pmatrix} x_1 \\ \vdots \\ x_n \end{pmatrix}, \quad b = \begin{pmatrix} b_1 \\ \vdots \\ b_m \end{pmatrix}.$$

Матрица A называется основной матрицей системы, b — столбцом свободных членов.

Алгоритм решения СЛАУ методом Гаусса подразделяется на два этапа:

1. **Прямой ход**: осуществляется последовательное приведение системы к треугольному виду с помощью элементарных преобразований строк. В ходе прямого хода вычитаются строки системы, домноженные на опре-

делённые коэффициенты, чтобы получить в столбце под диагональным элементом нули.

2. **Обратный ход**: после приведения системы к треугольному виду, начиная с последнего уравнения, происходит нахождение значений переменных. Это называется обратной подстановкой, при которой вычисленное значение переменной используется для упрощения последующих уравнений.

Для приведения матрицы к треугольному виду используют расширенную матрицу вида:

$$B = [A|b] = \begin{pmatrix} a_{11} & \dots & a_{1n} & b_1 \\ \vdots & \ddots & \vdots & \vdots \\ a_{m1} & \dots & a_{mn} & b_m \end{pmatrix}.$$

Метод Гаусса позволяет решать совместные системы линейных уравнений или определять их несовместность.

2.2 LU-разложение

LU-разложение — это способ разложения матрицы A на произведение двух матриц L (нижняя треугольная матрица) и U (верхняя треугольная матрица) $A = LU$. Это разложение особенно удобно для решения систем линейных уравнений и нахождения обратной матрицы. Это используется для решения системы $Ax = b$ через два шага:

1. Сначала решаем систему $Ly = b$ методом прямой подстановки..
2. Затем решаем систему $Ux = y$ методом обратной подстановки.

LU-разложение возможно только для невырожденных матриц, для которых существуют обратные матрицы.

2.2.1 Решение систем линейных уравнений

Если известно LU-разложение матрицы A, система $Ax = b$ может быть решена в два шага:

1. $Ly = b$
2. $Ux = y$

2.2.2 Обращение матриц

Обращение матрицы A эквивалентно решению системы $AX = I$, где X — обратная матрица, а I — единичная матрица.

2.2.3 Вычисление определителя матрицы

Определитель матрицы A через LU-разложение:

$$\det(A) = \det(LU) = \det(L)\det(U) = \prod_{i=1}^{n} L_{ii} \prod_{j=1}^{n} U_{jj},$$

где n — размер матрицы A, L_{ii} и U_{jj} — диагональные элементы матриц L и U соответственно.

2.3 LUP-разложение

LUP-разложение представляет собой расширение LU-разложения, которое позволяет работать с системами, требующими перестановки строк для получения нужной формы матрицы. В этом случае матрица A представляется в виде:

$$PA = LU,$$

где P — матрица перестановок. Этот метод является улучшенным вариантом LU-разложения и применяется, когда требуется учитывать перестановку строк для обеспечения вычислительной устойчивости.

3 Выполнение лабораторной работы

Следуем указаниям [1]

3.1 Метод Гаусса

Для системы $Ax = b$:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -2 & -4 \\ 1 & -1 & 0 \end{pmatrix}, \quad b = \begin{pmatrix} 4 \\ 6 \\ 0 \end{pmatrix},$$

построим расширенную матрицу:

$$B = \begin{pmatrix} 1 & 2 & 3 & 4 \\ 0 & -2 & -4 & 6 \\ 1 & -1 & 0 & 0 \end{pmatrix}.$$

Далее приведём матрицу к треугольному виду и решим систему в Octave ([3.1]) и Julia ([3.2,3.3]).

```
>> B = [ 1 2 3 4 ; 0 -2 -4 6 ; 1 -1 0 0 ]
B =

   1   2   3   4
   0  -2  -4   6
   1  -1   0   0

>> B (2, 3)
ans = -4
>> B (1, :)
ans =

   1   2   3   4

>> B(3,:) = (-1) * B(1,:) + B(3,:)
B =

   1   2   3   4
   0  -2  -4   6
   0  -3  -3  -4

>> B(3,:) = -1.5 * B(2,:) + B(3,:)
B =

    1    2    3    4
    0   -2   -4    6
    0    0    3  -13

>> rref(B)
ans =

   1.0000        0        0   5.6667
        0   1.0000        0   5.6667
        0        0   1.0000  -4.3333

>> format long
>> rref(B)
ans =

   1.000000000000000                   0                   0   5.666666666666667
                   0   1.000000000000000                   0   5.666666666666666
                   0                   0   1.000000000000000  -4.333333333333333
```

Рис. 3.1: Метод Гаусса в Octave

```julia
using LinearAlgebra

A = [ 1 2 3 ; 0 -2 -4 ; 1 -1 0 ]
b = [4; 6; 0]

B = hcat(A, b)

println("Расширенная матрица системы уравнений:\n")
for i in 1:size(B)[1]
    for j in 1:size(B)[2]
        print(B[i, j], " ")
    end
    println("\n")
end

# Прямой ход метода Гаусса
B[3,:] = -B[1,:] + B[3,:]
B[3,:] = -1.5*B[2,:] + B[3,:]

println("Расширенная матрица системы, приведённая к верхнедиагональному виду:\n")
for i in 1:size(B)[1]
    for j in 1:size(B)[2]
        print(B[i, j], " ")
    end
    println("\n")
end

x = A \ b
println("Решение системы уравнений: ", x)
```

Рис. 3.2: Код метода Гаусса и встроенного решения систем уравнений на Julia

```
PS C:\Users\User\Documents\work\study\2024-2025\Научное программирование\sciprog\labs\l
jl
Расширенная матрица системы уравнений:

1 2 3 4

0 -2 -4 6

1 -1 0 0

Расширенная матрица системы, приведённая к верхнедиагональному виду:

1 2 3 4

0 -2 -4 6

0 0 3 -13

Решение системы уравнений: [5.666666666666666, 5.666666666666667, -4.333333333333333]
```

Рис. 3.3: Результат кода метода Гаусса и встроенного решения систем уравнений на Julia

3.2 Левое деление

В Octave встроенная операция для решения систем $Ax = b$ называется левым делением и записывается как $A\backslash b$ ([3.4]).

```
>> format short
>> B = [ 1 2 3 4 ; 0 -2 -4 6 ; 1 -1 0 0 ]
B =

   1   2   3   4
   0  -2  -4   6
   1  -1   0   0

>> A = B(:,1:3)
A =

   1   2   3
   0  -2  -4
   1  -1   0

>> b = B (:,4)
b =

   4
   6
   0

>> A\b
ans =

   5.6667
   5.6667
  -4.3333
```

Рис. 3.4: Левое деление в Octave

3.3 LU- и LUP-разложение

Для матрицы A:

$$A = \begin{pmatrix} 1 & 2 & 3 \\ 0 & -2 & -4 \\ 1 & -1 & 0 \end{pmatrix},$$

выполним LU- и LUP-разложение в Octave ([3.5]) и Julia ([3.6,3.7]).

```
>> [L U P] = lu (A)
L =

   1.0000        0        0
   1.0000   1.0000        0
        0   0.6667   1.0000

U =

   1   2   3
   0  -3  -3
   0   0  -2

P =

Permutation Matrix

   1   0   0
   0   0   1
   0   1   0
```

Рис. 3.5: LU- и LUP-разложение в Octave

```julia
# LU-разложение
LU = lu(A)

L = LU.L
U = LU.U
P = LU.P

println("Нижнетреугольная матрица L в LU разложении:\n")
for i in 1:size(L)[1]
    for j in 1:size(L)[2]
        print(L[i, j], " ")
    end
    println("\n")
end
println("Верхнетреугольная матрица U в LU разложении:\n")
for i in 1:size(U)[1]
    for j in 1:size(U)[2]
        print(U[i, j], " ")
    end
    println("\n")
end
println("Матрица перестановок P в LU разложении:\n")
for i in 1:size(P)[1]
    for j in 1:size(P)[2]
        print(P[i, j], " ")
    end
    println("\n")
end
```

Рис. 3.6: Код LU- и LUP-разложения на Julia

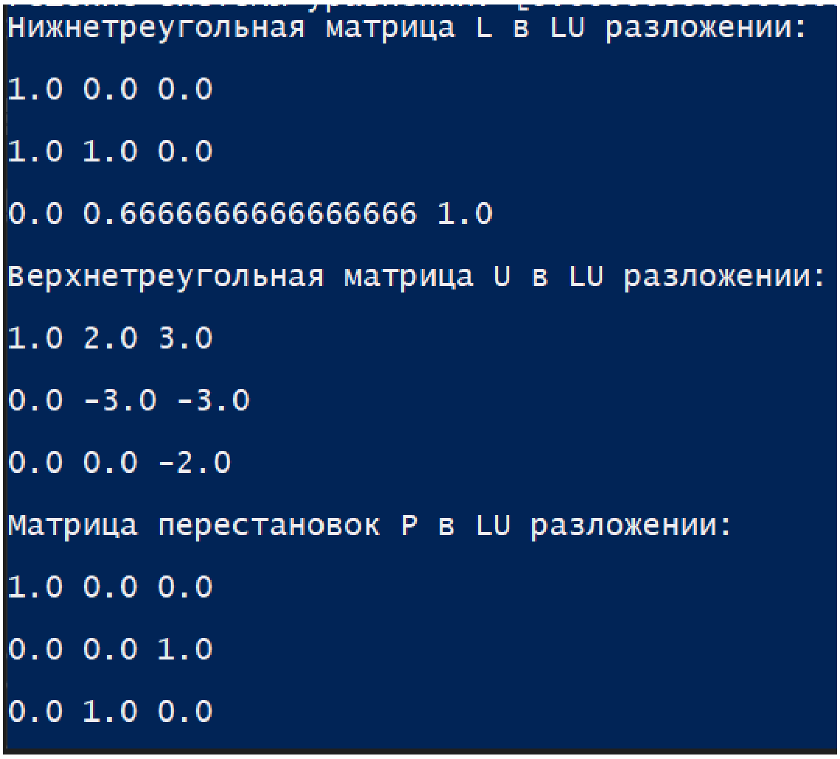

Рис. 3.7: Результат кода LU- и LUP-разложения на Julia

4 Выводы

В ходе выполнения лабораторной работы я изучил метод Гаусса, LU- и LUP-разложения, а также реализовал обозначенные алгоритмы на Octave и Julia.

Список литературы

1. Кулябов Д. С. Лабораторная работа №4. Системы линейных уравнений [Электронный ресурс]. RUDN, 2024. URL: https://esystem.rudn.ru/pluginfile.php/2372904/mod_resource/content/3/004-gauss.pdf.