ИТМО

Отчет
По лабораторной работе №2
Метод решения СЛАУ “Разложение Холецкого”

Выполнил:
Рахматов Нематджон
Р3233
Преподаватель:
Перл Ольга
Вячеславовна

Санкт-Петербург
2024

Оглавление

Задача

Решите систему линейных алгебраических уравнений, реализуя метод разложения Холецкого. Также выведите полученные промежуточные значения y.
Формат входных данных:
n
a11 a12 ... a1n b1
a21 a22 ... a2n b2
...
an1 an2 ... ann bn
Формат вывода:
x1
x2
...
xn
y1
y2
...
yn
, где x1..xn - значения неизвестных, а y1..yn - значения y.
Для систем, которые не имеют решений или имеют неограниченное количество решений, должно быть напечатано только следующее сообщение:
"The system has no roots of equations or has an infinite set of them.". Для этого задайте значение переменной isSolutionExists и сообщение об ошибке.

Описание Метода

Метод разложения Холецкого - это способ решения систем линейных уравнений, который выгодно используется для работы с симметричными и положительно определенными матрицами. Он основан на разложении исходной матрицы на произведение нижнетреугольной матрицы L на её транспонированную сопряжённую форму L^T.Он также позволяет работать с системами уравнений, имеющими множество решений.

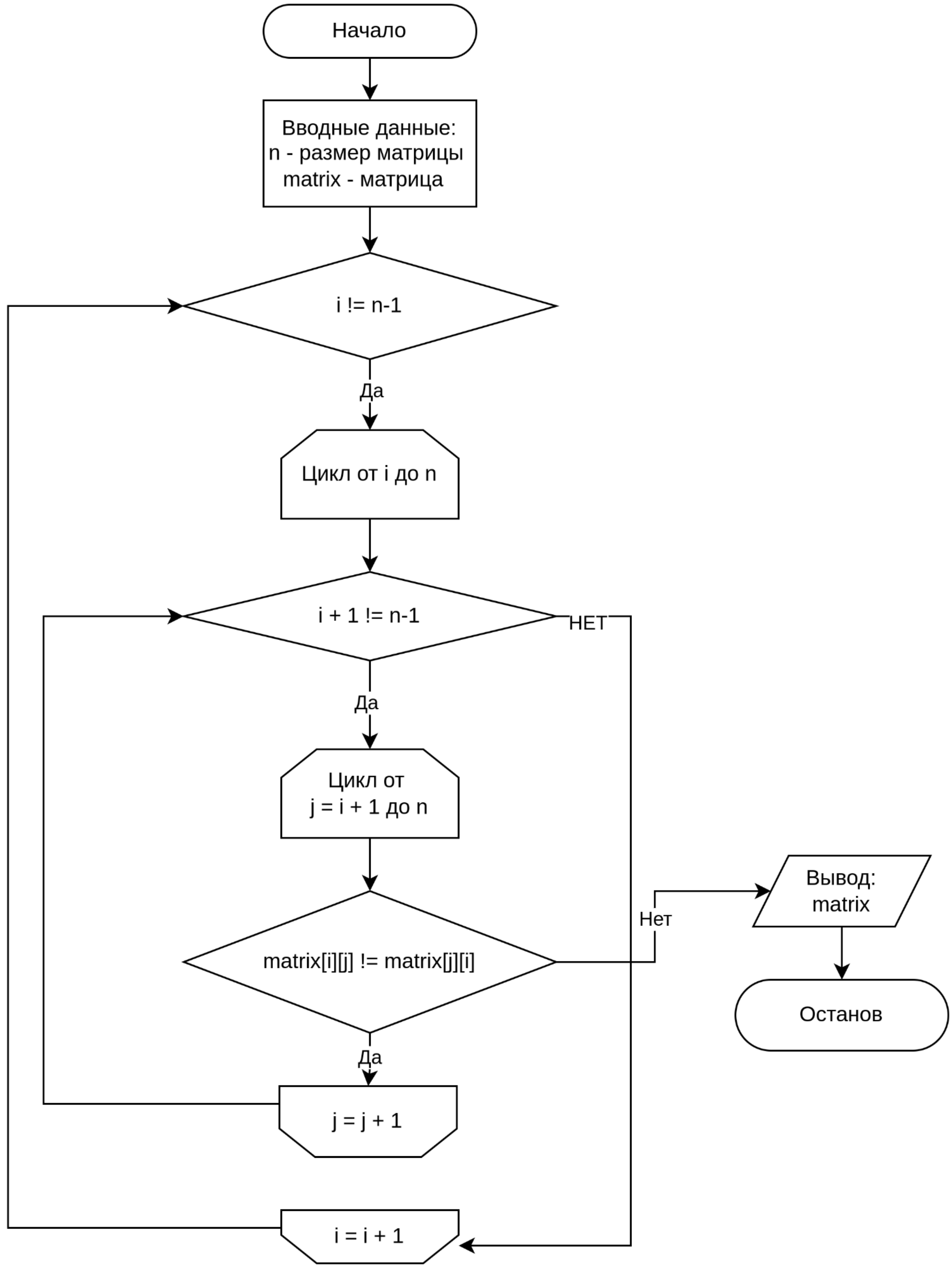
Основные шаги метода разложения Холецкого:

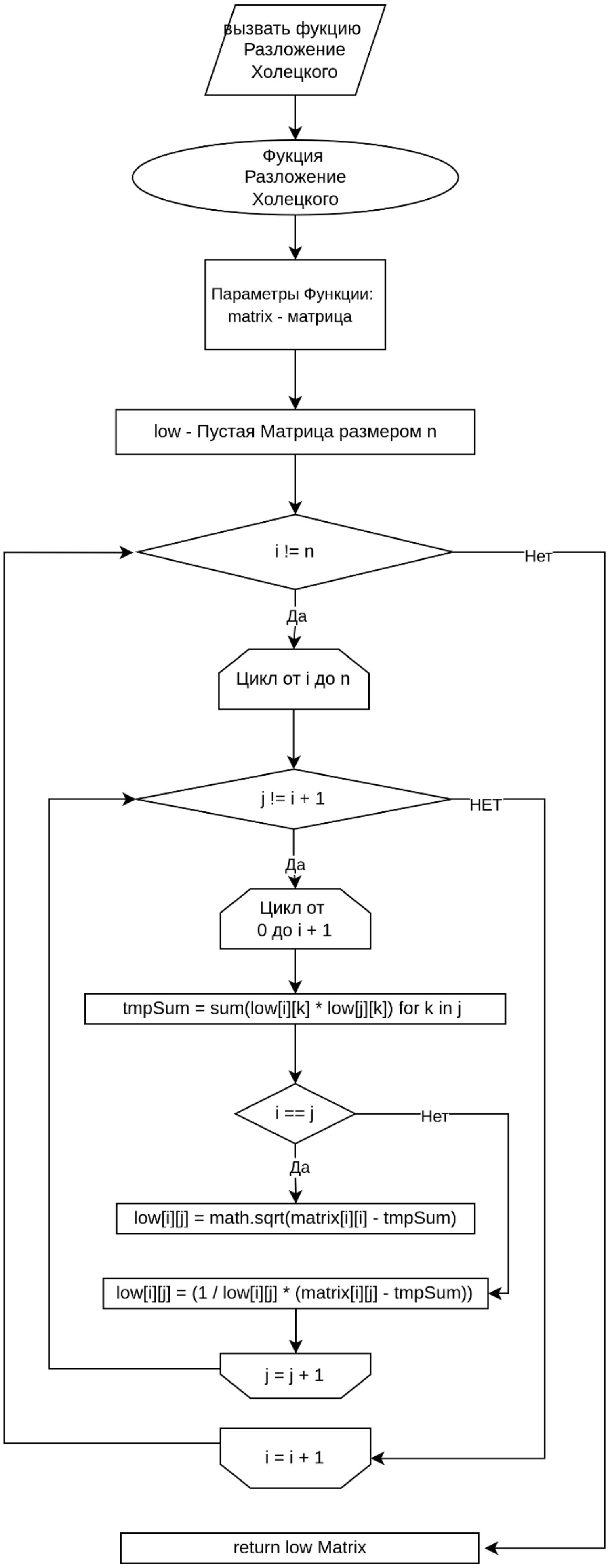
1) Разложение матрицы: Исходная матрица A представляется в виде произведения $A = LL^T$, где L - нижнетреугольная матрица, а LL^t- транспонированная матрица L.
2) Нахождение L: Этот шаг предлагает нахождение нижнетреугольной матрицы L, которая удовлетворяет равенству $A = LL^T$. Это делается по элементно, где каждый элемент L_{ij} вычисляется по следующей формуле:

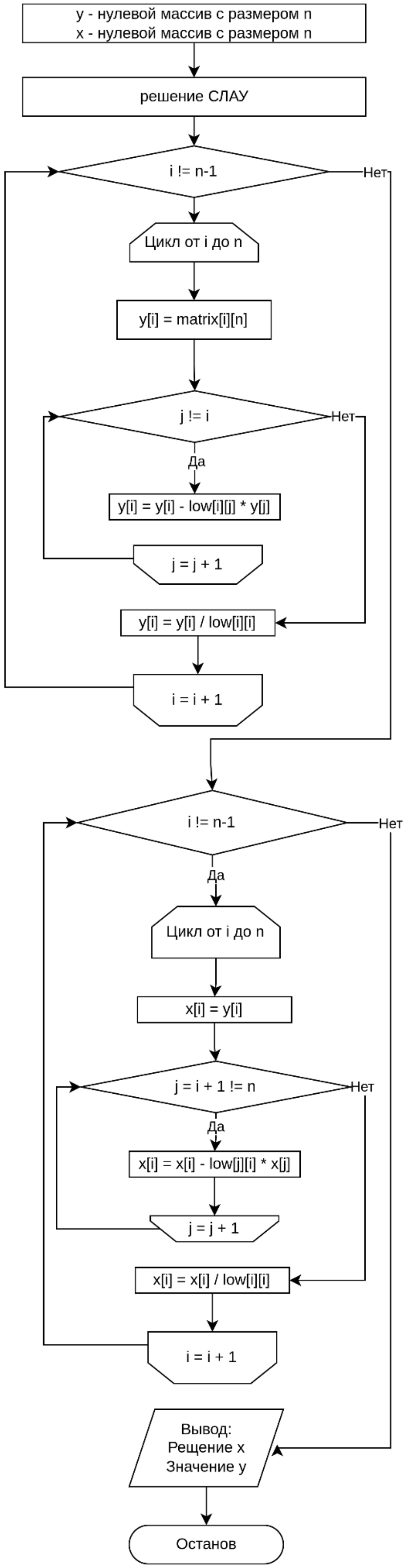
$$L_{ij} = \left\{ \sqrt{A_{ij} - \sum_{k=1}^{j-1} L_{ik}^{\ 2}}, \text{если } i = j \quad ; \qquad \frac{1}{L_{ij}} (A_{ij} - \sum_{k=1}^{j-1} L_{ik} * L_{jk}), \text{ если } i > j \right\}$$

3) Решения системы уравнений: После того как матрица A успешно разложена на LL^T, мы сможем решить сиситему уравнений $Ax = b$. Это делается в два этапа:
3.1. Прямой ход. Решаем систему $Ly = b$ для y, используя метод прямой подстановки.
3.2. Обратный ход. Потом решаем $L^T x = y$ для x, используя метод обратной подстановки.

Блок-схема алгоритма




вызвать фукцию Разложение Холецкого
Фукция Разложение Холецкого
Параметры Функции: matrix - матрица
low - Пустая Матрица размером n
i != n
Нет
Да
Цикл от i до n
j != i + 1
НЕТ
Да
Цикл от 0 до i + 1
tmpSum = sum(low[i][k] * low[j][k]) for k in j
i == j
Нет
Да
low[i][j] = math.sqrt(matrix[i][i] - tmpSum)
low[i][j] = (1 / low[i][j] * (matrix[i][j] - tmpSum))
j = j + 1
i = i + 1
return low Matrix


y - нулевой массив с размером n
x - нулевой массив с размером n
решение СЛАУ
i != n-1
Нет
Цикл от i до n
y[i] = matrix[i][n]
j != i
Нет
Да
y[i] = y[i] - low[i][j] * y[j]
j = j + 1
y[i] = y[i] / low[i][i]
i = i + 1
i != n-1
Нет
Да
Цикл от i до n
x[i] = y[i]
j = i + 1 != n
Нет
Да
x[i] = x[i] - low[j][i] * x[j]
j = j + 1
x[i] = x[i] / low[i][i]
i = i + 1
Вывод:
Рещение x
Значение y
Останов

Код программы

```
def chDec(mt):
   low = [[0.0] * len(mt) for _ in range(len(mt))]

   for i in range(len(mt)):
       for j in range(i+1):
           tmpSum = sum(low[i][k] * low[j][k] for k in range(j))
           if i == j:
               low[i][j] = math.sqrt(mt[i][i] - tmpSum)
           else:
               if low[j][j] == 0:
                   raise ZeroDivisionError("zero devision in
decomposition")
               else:
                   low[i][j] = (1.0 / low[j][j] * (mt[i][j] -
tmpSum))
   return low

def calculation(x, y, mt, n, low):
   for i in range(n):
       y[i] = mt[i][n]
       for j in range(i):
           y[i] -= low[i][j] * y[j]

       if y[i] == 0:
           raise ZeroDivisionError("zero devision in
decomposition")
       else:
           y[i] /= low[i][i]

   for i in reversed(range(n)):
       x[i] = y[i]
       for j in range(i+1, n):
           x[i] -= low[j][i] * x[j]

       if x[i] == 0:
           raise ZeroDivisionError("zero devision in
decomposition")
       else:
           x[i] /= low[i][i]

   return x, y

class Solution:
   isSolutionExists = True
   errorMessage = ""
```

```python
def solveByCholeskyDecomposition(n, matrix):
    # Симметрия?
    for i in range(n):
        for j in range(i + 1, n):
            if matrix[i][j] != matrix[j][i]:
                Solution.isSolutionExists = False
                Solution.errorMessage = "The matrix is not
symmetric."
                return []

    try:
        low = chDec(matrix)
        y = [0.0] * n
        x = [0.0] * n
        newX, newY = calculation(x, y, matrix, n, low)
        return newX, newY

    except:
        Solution.isSolutionExists = False
        Solution.errorMessage = "The system has no roots of
equations or has an infinite set of them."
```

Примеры Работ программи

1) n = 3,

```
matrix = [
    [4, 12, -16, 8],
    [12, 37, -43, 19],
    [-16, -43, 98, -56]
]
```

Ответ:

```
x:  19.111111111111107
x:  -5.555555555555555
x:  0.11111111111111111
y:  4.0
y:  -5.0
y:  0.33333333333333333
```

2) n = 3,

```
matrix = [
    [1, 0, 0, 1],
    [0, 1, 0, 2],
    [0, 0, 1, 3]
]
```

Ответ:
```
The system has no roots of equations or has an infinite set of them.
```

3) n = 3. Симметричная положительно определенная матрица
```
matrix = [
    [4, 1, -1, 8],
    [1, 9, -2, 20],
    [-1, -2, 16, 17]
]
```
Ответ:
```
The matrix is not symmetric.
```

4) n = 0.
```
matrix = [
]
```
Ответ:
```
The system has no roots of equations or has an infinite set of them.
```

5) n = 1
```
matrix = [
   [8, 2]
]
```
Ответ:
```
The matrix is not symmetric.
```

Вывод

После тестирования различных данных я понял, что метод разложения Холецкого отлично справляется с решением систем линейных уравнений, особенно на симметричных и положительно определенных матрицах. Это происходит потому, что он обеспечивает точные и стабильные результаты в таких условиях. Важно отметить, что метод хорошо обрабатывает различные ситуации, включая несимметричные матрицы и неположительно определенные матрицы. Это позволяет избежать ошибок и неправильных результатов при работе с нестандартными данными. По сравнению с другими методами решения систем линейных уравнений, метод разложения Холецкого обладает высокой точностью и стабильностью решений. Однако он требует, чтобы матрица была симметричной и положительно определенной для работы. Алгоритмическая сложность метода кубическая, что делает его эффективным для небольших систем, но менее эффективным для больших. В целом, метод разложения Холецкого является мощным инструментом для решения систем линейных уравнений, но его применимость зависит от характеристик матрицы и требований к точности решений.