|  |  |
| --- | --- |
|  | **Министерство науки и высшего образования Российской Федерации**  **Федеральное государственное бюджетное образовательное учреждение**  **высшего образования**  **«Московский государственный технический университет**  **имени Н.Э. Баумана**  **(национальный исследовательский университет)»**  **(МГТУ им. Н.Э. Баумана)** |

ФАКУЛЬТЕТ Информатики и систем управления

КАФЕДРА Теоретической информатики и компьютерных технологий

**Лабораторная работа № 1**

**«Решение СЛАУ** **с трёхдиагональной матрицей»**

***по курсу «Численные методы»***

Студент *Ионов Т.Р. 61Б*

Преподаватель *Домрачева А.Б.*

*Москва, 2022 г.*

СОДЕРЖАНИЕ

[1 Постановка задачи 3](#_Toc96202771)

[2 Теоретическая часть 3](#_Toc96202772)

[2.1 Решение трехдиагональной СЛАУ методом прогонки 3](#_Toc96202773)

[2.2 Решение трехдиагональной СЛАУ методом Гаусса 4](#_Toc96202774)

[3 Практическая часть 4](#_Toc96202775)

[4 Тестирование 6](#_Toc96202776)

[Вывод 7](#_Toc96202777)

# 1 Постановка задачи

Необходимо реализовать программу для решения СЛАУ с трёхдиагональной матрицей методом прогонки и методом Гаусса, а также сравнить эффективность этих методов.

Дано:

=

Найти:

# 2 Теоретическая часть

Имеем систему линейных уравнений:

## 2.1 Решение трехдиагональной СЛАУ методом прогонки

Выражая через и так далее, получаем:

Решение имеется только при выполнении трёх условий диагонального преобладания:

## 2.2 Решение трехдиагональной СЛАУ методом Гаусса

Составим расширенную матрицу системы , приведем ее к верхнему треугольному виду:

=

Затем выразим все через линейную комбинацию :

# 3 Практическая часть

Листинг 1 − Решение трехдиагональной СЛАУ методом прогонки

def solve\_matrix(a, b, c, d):

def check\_matrix(a, b, c):

n = len(a)

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

if abs(b[i]) < abs(a[i] + c[i-1]):

return 0

if abs(c[i]/b[i]) > 1:

return 0

if abs(a[i+1] / c[i]) > 1:

return 0

return 1

n = len(a)

f = d.copy()

a = a.copy()

b = b.copy()

c = c.copy()

x = [0 for \_ in range(n)]

if not check\_matrix(a, b, c):

print('incorrect system')

return

for i in range(1, n):

m = a[i] / b[i-1]

b[i] = b[i] - m\*c[i-1]

f[i] = f[i] - m\*f[i-1]

x[n-1] = f[n-1]/b[n-1]

for i in range(n-2, -1, -1):

x[i] = (f[i] - c[i] \* x[i+1]) / b[i]

return x

Листинг 2 − Решение трехдиагональной СЛАУ методом Гаусса

def solve\_matrix\_gauss(a, d):

a = a.copy()

d = d.copy()

n = len(a[0])

x = [0 for \_ in range(n)]

for i in range(n):

a[i].append(d[i])

for i in range(n):

if a[i][i] == 0:

print('error')

return

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

r = a[j][i]/a[i][i]

for k in range(n+1):

a[j][k] = a[j][k] - r \* a[i][k]

x[n-1] = a[n-1][n]/a[n-1][n-1]

for i in range(n-2, -1, -1):

x[i] = a[i][n]

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

x[i] = x[i] - a[i][j]\*x[j]

x[i] = x[i]/a[i][i]

return x

# 4 Тестирование

Для тестирования были выбраны матрица А и вектор d:

Метод Гаусса и метод прогонки выдали одинаковые ответы – единичные вектора без погрешностей. Поэтому воспользуемся для вычисления погрешности вектор невязки.

Вектор невязки:

Результаты метода прогонки:

Результаты метода Гаусса:

# Вывод

В результате выполнения данной лабораторной работы были реализованы два метода решения трехдиагональной СЛАУ: метод прогонки и метод Гаусса, а также найдены значения вектора невязки для обоих решений.