Лабораторная работа №1

1.1

1.2. Реализовать метод прогонки в виде программы, задавая в качестве входных данных ненулевые элементы матрицы системы и вектор правых частей. Используя разработанное программное обеспечение, решить СЛАУ с трехдиагональной матрицей.



*par\_and\_fun/Matrix.py*

def read\_matrix():  
 n, m = map(int, input("Введите 'n' и 'm': ").split())  
 print("Введите матрицу")  
 matrix = [[float(j) for j in input().split()] for \_ in range(n)]  
 if any(filter(lambda x: len(x) != m, matrix)):  
 raise ValueError("Количество элементов в строке матрицы не соответствует ожидаемому")  
 return matrix  
  
def read\_vector():  
 n = int(input("Введите длину вектора n: "))  
 print("Введите значения вектора, каждый элемент вектора на новой строке")  
 vector = [float(input()) for \_ in range(n)]  
 return vector

*Laba1/Laba1-2.py*

from par\_and\_func import \*  
  
  
def main():  
 stability = 1  
 matrix = read\_matrix()  
 n = len(matrix)  
 if len(matrix) != len(matrix[0]):  
 raise ValueError("Матрица должна быть квадратной")  
 d = read\_vector()  
 pq = []  
  
 # Прямой ход метода  
 if abs(matrix[0][0]) < 10e-12:  
 raise ZeroDivisionError("Элемент b1 == 0")  
 stability \*= abs(matrix[0][0]) >= abs(matrix[0][1])  
  
 pq.append((-matrix[0][1] / matrix[0][0], d[0] / matrix[0][0]))  
 for i in range(1, n - 1):  
 chasn = matrix[i][i] + matrix[i][i - 1] \* pq[i - 1][0]  
 if abs(chasn) < 10e-12:  
 raise ZeroDivisionError("Деление на 0")  
 stability \*= abs(matrix[i][i]) >= (abs(matrix[i][i + 1]) + abs(matrix[i][i - 1]))  
 pq.append((-matrix[i][i + 1] / chasn, (d[i] - matrix[i][i - 1] \* pq[i - 1][1]) / chasn))  
 chasn = matrix[n - 1][n - 1] + matrix[n - 1][n - 2] \* pq[n - 2][0]  
 if abs(chasn) < 10e-12:  
 raise ZeroDivisionError("Деление на 0")  
 pq.append((0, (d[n - 1] - matrix[n - 1][n - 2] \* pq[n - 2][1]) / chasn))  
 # Обратный ход метода  
 result = [0] \* n  
  
 result[-1] = pq[-1][1]  
 for i in range(n - 2, -1, -1):  
 result[i] = pq[i][0] \* result[i + 1] + pq[i][1]  
 if not stability:  
 print("Предупреждение. Результат может быть посчитан с большой погрешностью!")  
  
 print("Ответ: ", end=" ")  
 for i in range(1, n + 1):  
 print(f"x{i} = {result[i - 1]}", end=" ")  
 print()  
  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 main()

Результат решения данной системы:

