Учреждение образования

«БЕЛОРУССКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ИНФОРМАТИКИ И РАДИОЭЛЕКТРОНИКИ»

Кафедра информатики

Отчет по лабораторной работе №2

Численное решение СЛАУ методом простых итераций и методом

Зейделя

Выполнил: cтудент гр. 953506

Кондрашов И.Д.

Руководитель: доцент Анисимов В. Я.

Минск 2021

Содержание

Цель выполнения задания 2

Краткие теоретические сведения 3

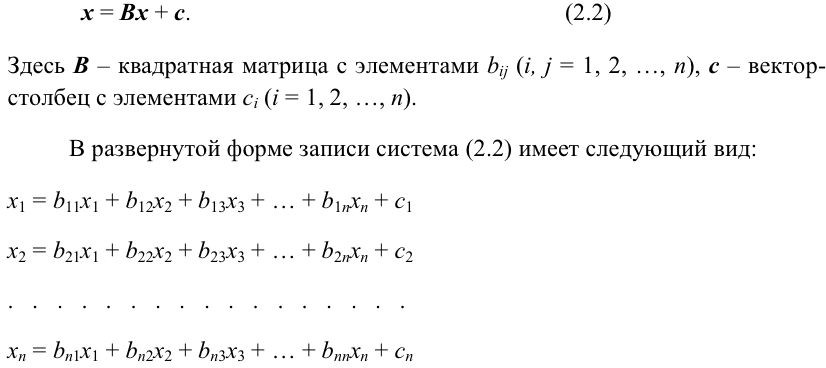
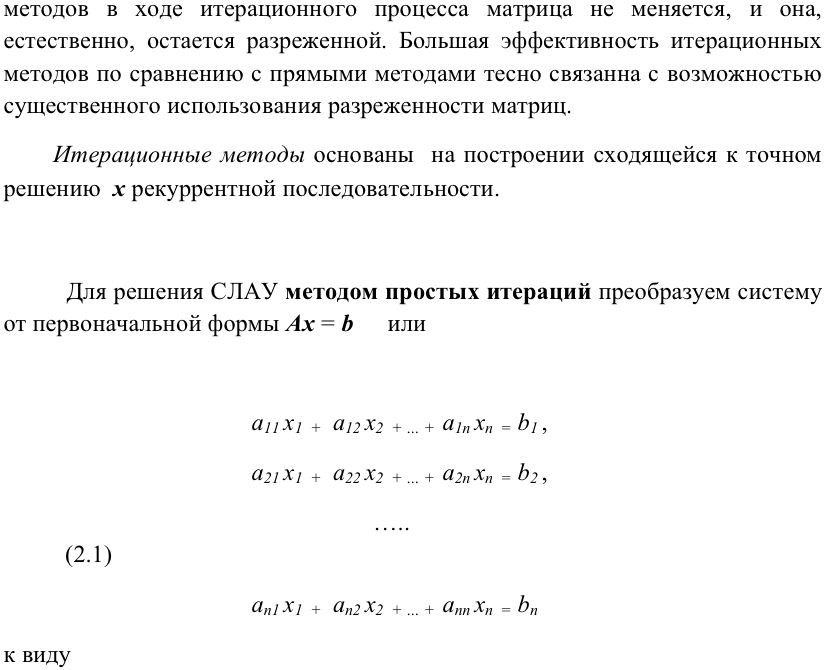
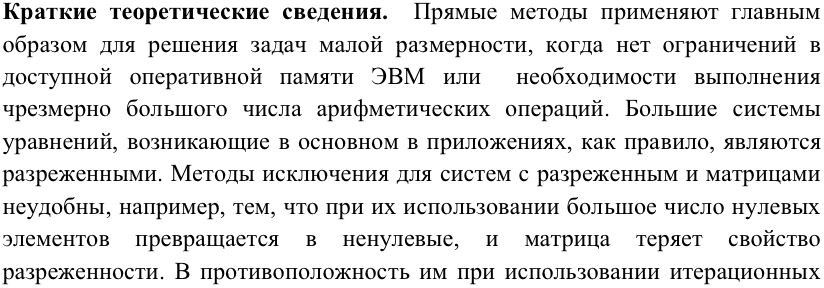
Программная реализация 7

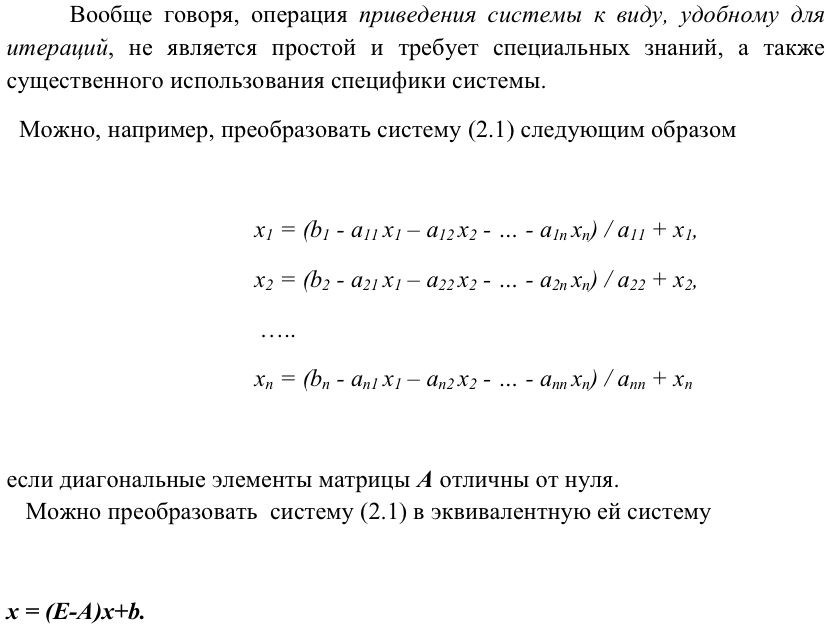
Выводы 9

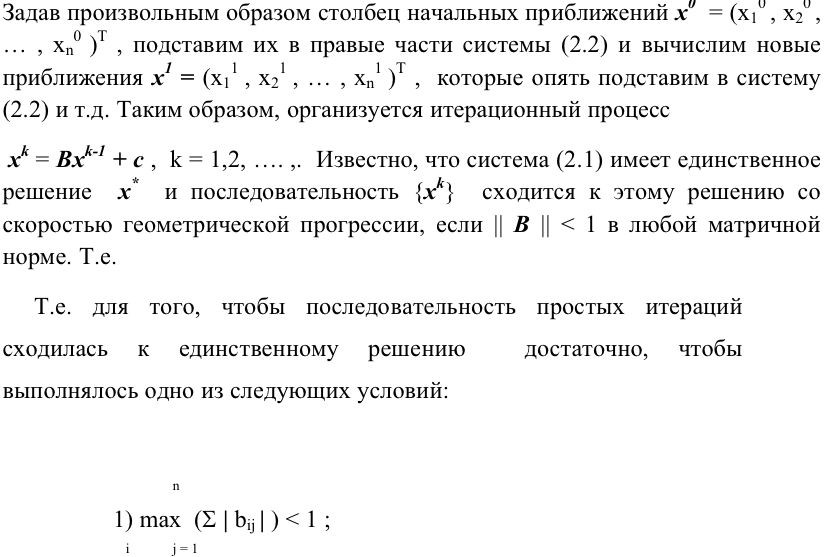
# Цель выполнения задания

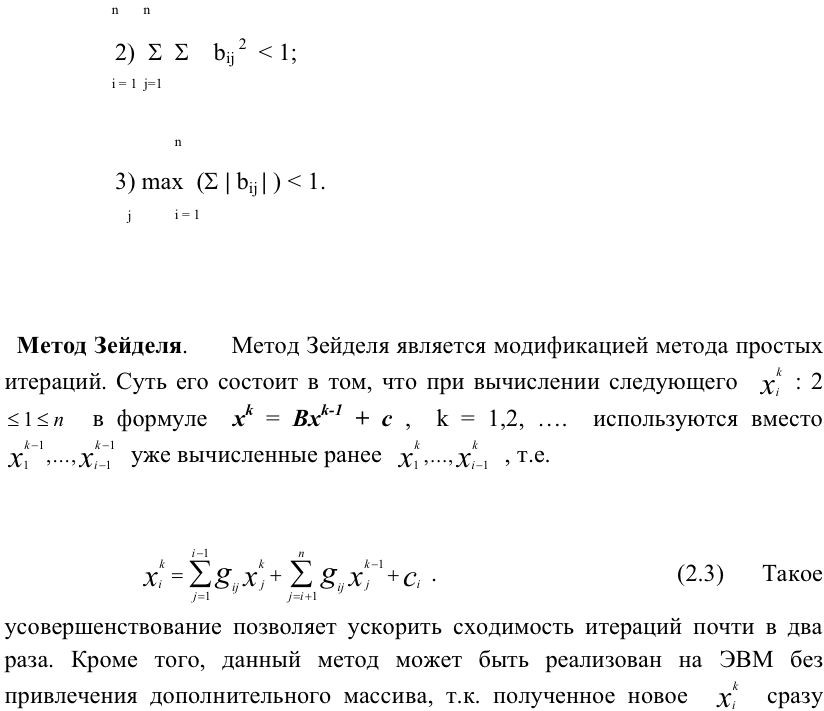
* изучить итерационные методы решения СЛАУ(метод простых итераций и метод Зейделя);
* составить алгоритм решения СЛАУ указанными методами, применимый для организации вычислений на ЭВМ;
* составить программу решения СЛАУ по разработанному алгоритму;
* выполнить тестовые примеры и проверить правильность работы программы.

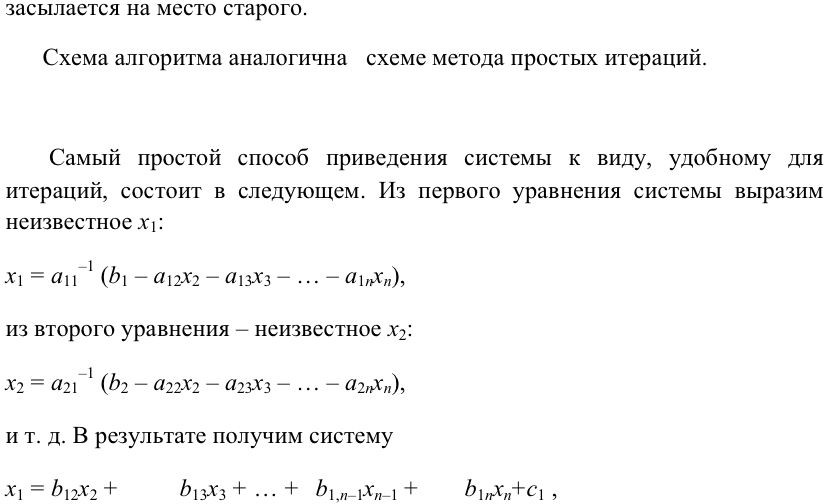
# Краткие теоретические сведения

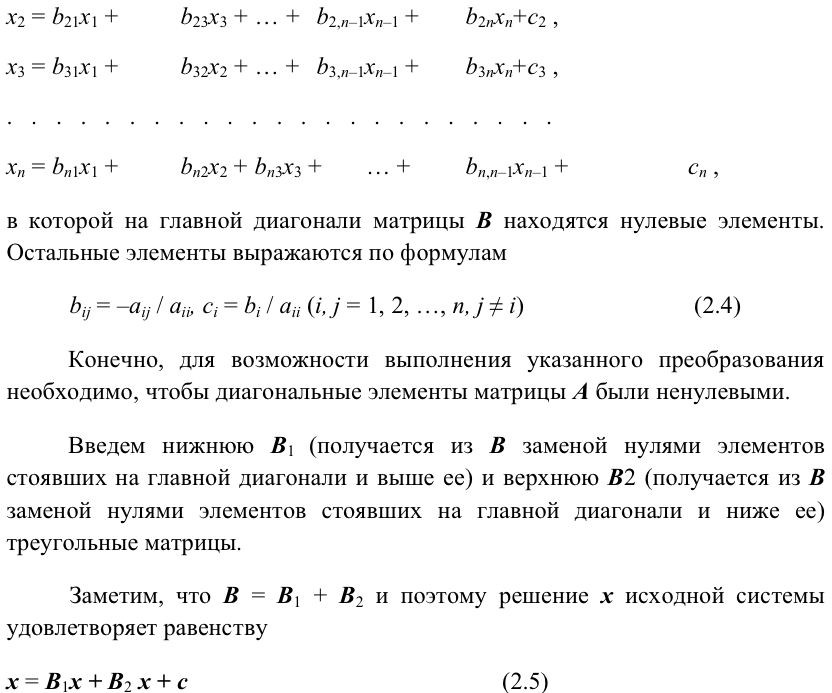


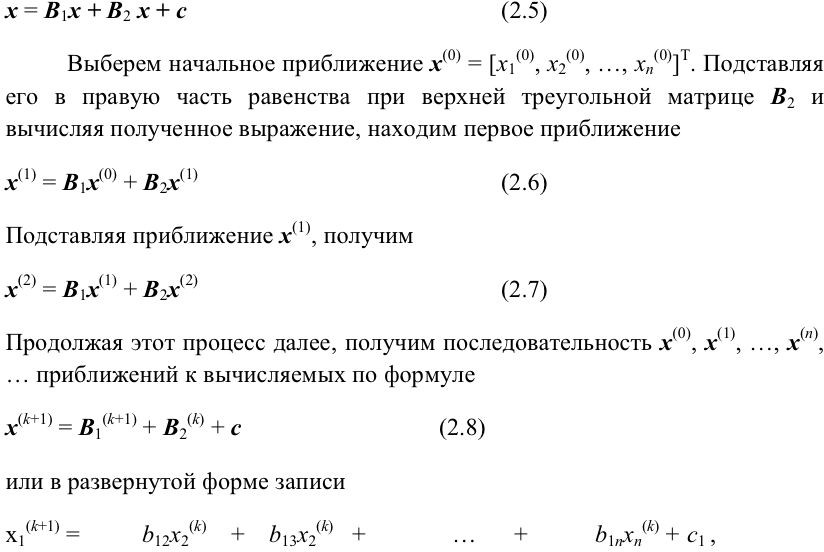


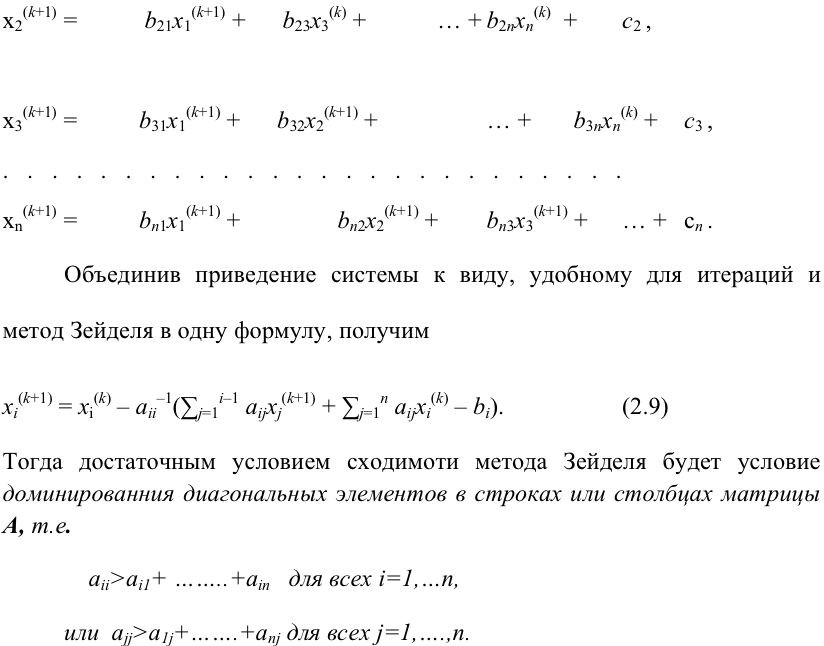






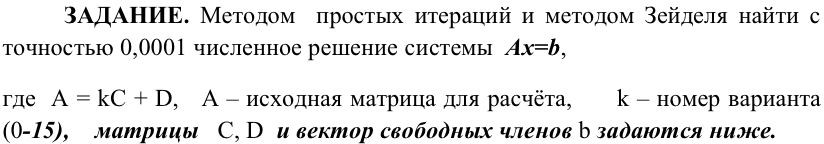


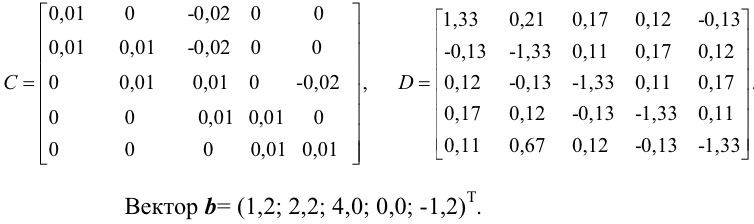




# Программная реализация

Вариант 11





## Метод простых итераций

import numpy as np  
  
np.set\_printoptions(precision=4, floatmode='fixed')  
option = 11.0  
  
C = np.array([[0.01, 0, -0.02, 0, 0],  
 [0.01, 0.01, -0.02, 0, 0],  
 [0, 0.01, 0.01, 0, -0.02],  
 [0, 0, 0.01, 0.01, 0],  
 [0, 0, 0, 0.01, 0.01]], dtype=float)  
  
D = np.array([[1.33, 0.21, 0.17, 0.12, -0.13],  
 [-0.13, -1.33, 0.11, 0.17, 0.12],  
 [0.12, -0.13, -1.33, 0.11, 0.17],  
 [0.17, 0.12, -0.13, -1.33, 0.11],  
 [0.11, 0.67, 0.12, -0.13, -1.33]], dtype=float)  
  
b = np.array([[1.2], [2.2], [4.0], [0.0], [-1.2]], dtype=float)  
  
A = np.add(np.multiply(C, option), D)  
  
  
def jacobi(A, b, init\_vector=None):  
 B = np.eye(A.shape[0]) - A / np.diag(A).reshape(-1, 1)  
 if min(np.linalg.norm(B, ord=norm)  
 for norm in (np.inf, 1, 'fro')) < 1:  
 c = b / np.diag(A).reshape(-1, 1)  
 if init\_vector is None:  
 c.dtype = float  
 xk = c.copy()  
 else:  
 init\_vector.dtype = float  
 xk = init\_vector.copy()  
 k = 1  
 while True:  
 xk\_1 = xk.copy()  
 xk = B @ xk\_1 + c  
 error = np.linalg.norm(B) \* np.linalg.norm(xk - xk\_1) / (1 - np.linalg.norm(B))  
 if error < 1e-3:  
 break  
 xk\_1 = xk.copy()  
 k += 1  
 return xk  
 raise ValueError("Jacobi method cannot be performed")

## Метод выбора главного элемента по столбцу

def gauss\_zeidel(A, b, init\_vector=None):  
 B = np.eye(A.shape[0]) - A / np.diag(A).reshape(-1, 1)  
 if min(np.linalg.norm(B, ord=norm)  
 for norm in (np.inf, 1)) < 1:  
 c = (b / np.diag(A).reshape(-1, 1)).reshape(-1)  
 if init\_vector is None:  
 c.dtype = float  
 xk = c.copy().reshape(-1)  
 else:  
 init\_vector.dtype = float  
 xk = init\_vector.copy().reshape(-1)  
 k = 1  
 while True:  
 xk\_1 = xk.copy()  
 for i in range(B.shape[0]):  
 t = np.sum(B[i, :i] \* xk[:i])  
 m = np.sum(B[i, i:] \* xk\_1[i:])  
 xk[i] = t + m + c[i]  
 error = np.linalg.norm(B) \* np.linalg.norm(xk - xk\_1) / (1 - if error < 1e-3:  
 break  
 k += 1  
 return xk.reshape(-1, 1)  
 raise ValueError("Gauss-Seidel method cannot be performed")

## Подтверждение работы программы:

Graphical user interface, text

Description automatically generated with medium confidence

**Выводы**

Таким образом, в ходе выполнения лабораторной работы был применён метод простых итераций (Якоби) и метод Зейделя для решения системы линейных уравнений, составлены алгоритмы и созданы реализации соответствующих программ на языке Python для решения поставленной задачи.