Лабораторная работа №2

Факультет прикладной математики и информатики

Кафедра вычислительной математики

**«Метод Гаусса-Зейделя»**

**Преподаватель:** Никифоров Иван Васильевич

**Студент:** Шелег Владислава Михайловна

Минск, 2016

**Постановка задачи**

Реализовать программу решения системы линейных алгебраических уравнений (СЛАУ) вида методом Гаусса-Зейделя.

где ,(для данного варианта ),

а – одномерный вектор правой части.

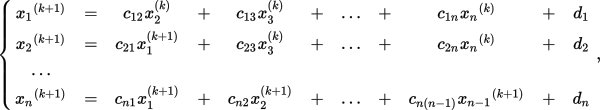
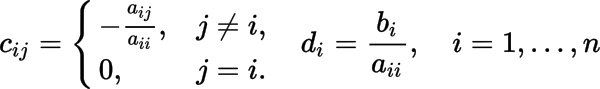
**Теория**

Пусть дана система уравнений

Далее необходимо привести данную систему к виду

Здесь в -м уравнении мы перенесли в правую часть все члены, содержащие для . Эта запись может быть представлена:

где в принятых обозначениях означает матрицу, у которой на главной диагонали стоят соответствующие элементы матрицы , а все остальные нули; тогда как матрицы и содержат верхнюю и нижнюю треугольные части , на главной диагонали которых нули. Итерационный процесс в методе Гаусса-Зейделя строится по формуле

после выбора соответствующего начального приближения .  
Метод Гаусса-Зейделя можно рассматривать как модификацию метода Якоби. Основная идея модификации состоит в том, что новые значения используются здесь сразу же по мере получения, в то время как в методе Якоби они не используются до следующей итерации: , где

Таким образом, -тая компонента -го приближения вычисляется по формуле: 

**Листинг программы**

(Программа реализована на языке С++)

#include <iostream>  
#include <vector>  
#include <cmath>  
  
**int** main() {  
 **int** size = 5;  
  
 std::vector<std::vector<**long double**> > matrix = {{7, 2.5, 2, 1.5, 1, 1},  
 {2.5, 8, 2.5, 2, 1.5, 1},  
 {2, 2.5, 9, 2.5, 2, 1},  
 {1.5, 2, 2.5, 10, 2.5, 1},  
 {1, 1.5, 2, 2.5, 11, 1}  
 };  
  
 **int** lambda = 5;  
 **for** (**int** k = 0; k < size; ++k) {  
 matrix[k][k] += lambda;  
 }  
  
 **for** (**int** i = 0; i < size; ++i) {  
 **for** (**int** j = 0; j < size + 1; ++j) {  
 std::cout.flags(std::ios::left);  
 std::cout.width(5);  
 std::cout << matrix[i][j] << " ";  
 }  
 std::cout << std::endl;  
 }  
  
 std::cout << "Введите точность вычислений:";  
 **long double** eps;  
 std::cin >> eps;  
  
 std::vector<**long double**> previous(size, 0.0);  
  
 **while** (**true**) {  
 std::vector<**long double**> current(size);  
 **for** (**int** i = 0; i < size; i++) {  
 current[i] = matrix[i][size];  
 **for** (**int** j = 0; j < size; j++) {  
 **if** (j < i) {  
 current[i] -= matrix[i][j] \* current[j];  
 }  
  
 **if** (j > i) {  
 current[i] -= matrix[i][j] \* previous[j];  
 }  
 }  
 current[i] /= matrix[i][i];  
 }  
  
 **long double** error = 0.0;  
  
 **for** (**int** i = 0; i < size; i++) {  
 error += abs(current[i] - previous[i]);  
 }  
  
 **if** (error < eps) {  
 **break**;  
 }  
  
 previous = current;  
 }  
  
 **for** (**int** i = 0; i < size; i++) {  
 std::cout << previous[i] << " ";  
 }  
  
 **return** 0;  
}

**Результаты**

