**Задание № .**

**Умова :**

**АЛГОРИТМ ЧИСЛЕННОГО РЕШЕНИЯ СЛАУ МЕТОДОМ ГАУССА-ЗЕЙДЕЛЯ**

1. Перевірити виконання умови діагонального переважання

http://ok-t.ru/studopediaru/baza5/1780124706440.files/image167.gifhttp://ok-t.ru/studopediaru/baza5/1780124706440.files/image165.gif

Якщо умова виконується для будь-якого i, то обчислюємо елементи матриці B і вектора d:

bij= - аij/aii  di= bi/aii при условии,что bii=0 i,j=1,2,…,n

2. Якщо умова діагонального переважання не виконується, то привести систему до виду зручному для ітерацій

3. Обчислити норму матриці В

||B||1 = max ∑ |bij|

Якщо ||B||1 < 1, то ітераційний процес сходиться (триває)

Якщо ||B||1 > 1, то ітераційний процес розходиться (обчислення закінчуються)

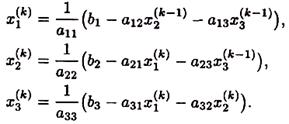
4 Обчислимо норму вектора d

||d||1 = max |di|

5. Вибрати в якості початкового наближення вектор d, де х0 = d

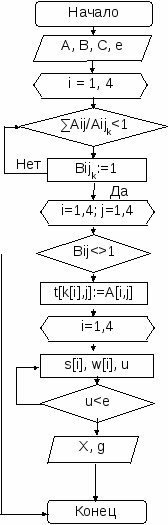
6. Обчислити необхідну кількість ітерацій для досягнення необхідної точності рішення системи

7. Реалізувати ітераційний процес у вигляді



і реалізувати його k раз.

БЛОК-СХЕМА МЕТОДА ГАУССА – ЗЕЙДЕЛЯ



КОД ПРОГРАМИ МЕТОДА ГАУССА-ЗЕЙДЕЛЯ

package gaussZeidel;  
  
import javafx.print.Printer;  
  
import java.io.PrintWriter;  
import java.util.Locale;  
import java.util.Scanner;  
  
*/\*\*  
 \* Created by anton on 03.12.2016.  
 \*/*public class gaussZeidel {  
 public static void main(String[] args) {  
 Scanner scanner = new Scanner(System.*in*);  
 scanner.useLocale(new Locale("Russian"));  
 PrintWriter printWriter = new PrintWriter(System.*out*);  
 System.*out*.println("Введите количество переменных:");  
 int size;  
 size = scanner.nextInt();  
 double[][] matrix = new double[size][size + 1];  
 System.*out*.println("Введите коеффициенты: ");  
 for (int i = 0; i < size; i++) {  
 for (int j = 0; j < size + 1; j++) {  
 matrix[i][j] = scanner.nextDouble();}}  
  
 System.*out*.println("Введите точность:");  
 double eps;  
 eps = scanner.nextDouble();  
  
 double[] previousVariableValues = new double[size];  
 for (int i = 0; i < size; i++) {  
 previousVariableValues[i] = 0.0;  
 }  
  
 while (true) {  
 double[] currentVariableValues = new double[size];  
 4б for (int i = 0; i < size; i++) {  
 currentVariableValues[i] = matrix[i][size];  
 for (int j = 0; j < size; j++) {  
 if (j < i) {  
 currentVariableValues[i] -= matrix[i][j] \* currentVariableValues[j];}}  
 if (j > i) {  
 currentVariableValues[i] -= matrix[i][j] \* previousVariableValues[j];}}  
 currentVariableValues[i] /= matrix[i][i];  
 }  
 double err = 0.0;  
 for (int i = 0; i < size; i++) {  
 err += Math.*abs*(currentVariableValues[i] - previousVariableValues[i]);  
 }  
 if (err < eps) { break;  
 }  
 previousVariableValues = currentVariableValues;  
 }  
 printWriter.println("Ответ: ");  
 for (int i = 0; i < size; i++) {  
 printWriter.print("X[" + (i + 1) + "] = " + previousVariableValues[i] + "\n");}  
 scanner.close();  
 printWriter.close();}}

РІЧНИЙ ПРОРАХУНОК Метод Гауса – Зейделя

Початкові дані: ε = 0,001.

Перевіряємо чи є діагональне переважання :

1,7 < |2,8| + | 1,9|

34 < |21| + |18|

1,3< |4,2| + |-17|

Ні в одному з рівнянь немає діагонального переважання. Значить необхідно виконати перетворення системи.

Поміняємо перше і третє рівняння і запишемо в перший; розділимо друге рівняння на десять і віднімемо перше і запишемо в друге; від першого віднімемо різницю другого (поділена на 10) і першого, і помножимо на третє і запишемо в третє.

Отримаємо систему має наступний вигляд:

Приблизний xi

x1 = = 0.6666666666666666

x2 = = -0.6666666666666667,

x3 = = 1.409090909090909

Приведемо до вигляду :

x1=0.6666666666666666 – 1,7x2 – 1,3x3  
x2=-0.6666666666666667-0.4x1+0,1x3  
x3= 1.409090909090909 – 0,5x­1 – 1x­2

Ітерація #1

= 0.667 - (0.405 \* -0.667) - (0.31 \* 1.409) = 0.5

= -0.667 - (0.667 \* 0.5) - (-0.167 \* 1.409) = -0.765

= 1.409 - (0.227 \* 0.5) - (0.455 \* -0.765) = 1.643

Ітерація #2

= 0.667 - (0.405 \* -0.765) - (0.31 \* 1.643) = 0.468

= -0.667 - (0.667 \* 0.468) - (-0.167 \* 1.643) = -0.705

= 1.409 - (0.227 \* 0.468) - (0.455 \* -0.705) = 1.623

Ітерація #3

= 0.667 - (0.405 \* -0.705) - (0.31 \* 1.623) = 0.45

= -0.667 - (0.667 \* 0.45) - (-0.167 \* 1.623) = -0.696

= 1.409 - (0.227 \* 0.45) - (0.455 \* -0.696) = 1.623

Ітерація #4

= 0.667 - (0.405 \* -0.696) - (0.31 \* 1.623) = 0.446

= -0.667 - (0.667 \* 0.446) - (-0.167 \* 1.623) = -0.693

= 1.409 - (0.227 \* 0.446) - (0.455 \* -0.693) = 1.623

Ітерацдія #5

= 0.667 - (0.405 \* -0.693) - (0.31 \* 1.623) = 0.445

= -0.667 - (0.667 \* 0.445) - (-0.167 \* 1.623) = -0.693

= 1.409 - (0.227 \* 0.445) - (0.455 \* -0.693) = 1.623