­­­Федеральное государственное автономное образовательное учреждение высшего образования

«Национальный исследовательский университет ИТМО»

Факультет Программной Инженерии и Компьютерной Техники

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



Лабораторная работа №1 по дисциплине  
«Вычислительная математика»

"Решение системы линейных алгебраических уравнений"

Выполнил: Сафронов Егор Михайлович

Группа: P3213

Преподаватель: Малышева Татьяна Алексеевна

г. Санкт-Петербург

2022

Цель работы:

Научиться искать решение СЛАУ при помощи численных методов, написать программу, которая будет совершать приближенные вычисления и находить решение, получая на вход матрицу из файла или консоли.

Задание лабораторной работы:

1. № варианта определяется как номер в списке группы согласно ИСУ. (**7**)
2. В программе численный метод должен быть реализован в виде отдельной подпрограммы или класса, в который входные/выходные данные передаются в качестве параметров.
3. Размерность матрицы n <=20 (задается из файла или с клавиатуры - по выбору конечного пользователя).
4. Должна быть реализована возможность ввода коэффициентов матрицы, как с клавиатуры, так и из файла (по выбору конечного пользователя).

**Метод Гаусса-Зейделя (итерационный метод), должно быть реализовано:**

* Точность задается с клавиатуры/файла
* Проверка диагонального преобладания (в случае, если диагональное преобладание в исходной матрице отсутствует, сделать перестановку строк/столбцов до тех пор, пока преобладание не будет достигнуто). В случае невозможности достижения диагонального преобладания - выводить соответствующее сообщение.
* Вывод вектора неизвестных:
* Вывод количества итераций, за которое было найдено решение.
* Вывод вектора погрешностей:

Описание метода, расчетные формулы:

Метод Гаусса-Зейделя – итерационный метод, находящий решение путем последовательного приближения значений переменных. Начальное приближение принимается как . При очередной итерации вычисляются новые значение для всех переменных путем использования в уравнении перменных полученных на предыдущей итерации, то есть для , а для .   
Конечная формула метода Гаусса-Зейделя:   
Итерационный процесс продолжается пока не будет достигнута необходимая точность, погрешность определяется разницей вычисленных переменных на текущем шаге и предыдущем:

Реализация расчета

func entryPoint() {  
 count := 0  
  
 for true {  
 for i := 0; i < size; i++ {  
 matrixX1[i][0] = matrixX2[i][0]  
 }  
 for i := 0; i < size; i++ {  
 sum := 0.0  
 for j := 0; j < size; j++ {  
 if j < i {  
 sum += matrixA[i][j] \* matrixX2[j][0] / matrixA[i][i]  
 } else if j != i {  
 sum += matrixA[i][j] \* matrixX1[j][0] / matrixA[i][i]  
 }  
 }  
 matrixX2[i][0] = matrixB[i][0]/matrixA[i][i] - sum  
 }  
 count++  
 if checkResults() || count >= M {  
 break  
 }  
 }  
  
 fmt.Println("-----result vector-----")  
 for i := 0; i < size; i++ {  
 fmt.Printf("x%d=%e \n", i+1, matrixX2[i][0])  
 }  
  
 fmt.Println("\n-----converges?-----")  
 if count >= M {  
 fmt.Println("no")  
 } else {  
 fmt.Println("yes, converges at", count)  
 }  
  
 fmt.Println("\n-----error vector-----")  
 for i := 0; i < size; i++ {  
 fmt.Printf("x%d=%e \n", i+1, math.Abs(matrixX2[i][0]-matrixX1[i][0]))  
 }  
}

}

Реализация установки начальных значений

func setResultMatrices() {  
 matrixX1 = make([][]float64, size)  
 matrixX2 = make([][]float64, size)  
 for i := range matrixX1 {  
 matrixX1[i] = make([]float64, 1)  
 }  
 for i := range matrixX2 {  
 matrixX2[i] = make([]float64, 1)  
 }  
  
 for i := 0; i < size; i++ {  
 matrixX2[i][0] = matrixB[i][0] / matrixA[i][i]  
 }  
}

Примеры работы программы:

Graphical user interface, text, application, chat or text message

Description automatically generated 🡪 Text

Description automatically generated  
Консольный ввод 🡪 Text

Description automatically generated

Выводы по работе:

Реализован метод решения системы линейных уравнений методов Гаусса-Зейделя. Познакомился с понятием «итерационного метода» (метода Гаусса-Зейделя является именно таким), который постепенно приходит к нужному ответу, что делает данный метод сложным для вычисления без машины, но весьма элегантным по памяти и скорости для машинного вычисления.