Санкт-Петербургский Национальный Исследовательский

Университет Информационных Технологий, Механики и Оптики

Факультет Программной Инженерии и Компьютерной Техники

Лабораторная работа №1

по дисциплине

«Вычислительная математика»

**Вариант - 13**

Студент: Провоторов Александр Владимирович

Группа: P3210

Преподаватель: Малышева Татьяна Алексеевна

Санкт-Петербург

2021 г.

**Текст задания**

Варианты:

* Метод Гаусса
* Метод Гаусса с выбором главного элемента
* **Метод простых итераций**
* Метод Гаусса-Зейделя

Размерность n<=20 (задается из файла или с клавиатуры - по выбору конечного пользователя)

* Должно быть предусмотрено чтение исходных данных как из файла, так и ввод с клавиатуры.
* Должна быть реализована возможность ввода коэффициентов матрицы как с клавиатуры, так и из файла. Также предусмотреть случайные коэффициенты.
* Обязательно: Тестовые данные на матрице большого размера (5\*5 / 6\*6...) + в отчёт с решением.
* Для точных методов(Гаусс и главные элементы) должно быть реализовано:
* Вычисление определителя
* Вывод треугольной  матрицы (включая преобразованный столбец В)
* Столбец неизвестных
* Столбец невязок
* Для итерационных методов:
* Точность задается с клавиатуры/файла
* Проверка диагонального преобладания  
  //В случае, если диагональное преобладание в изначальной матрице отсутствует - предлагается сделать перестановку строк/столбцов до тех пор, пока преобладание не будет достигнуто. В случае невозможности достижения диагонального преобладания - выводить сообщение.
* Столбец неизвестных
* Количество итераций, за которое было найдено решение
* Столбец погрешностей

**Цель работы**

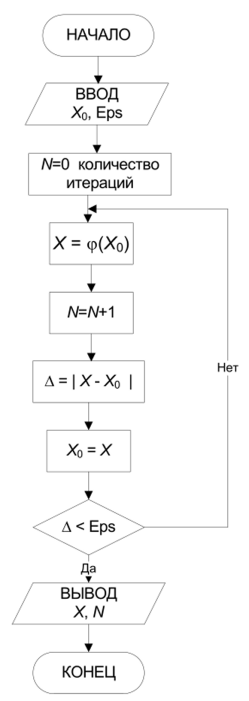
Реализовать и протестировать программу для решения системы линейных уравнений методом Гаусса, размерность до 20 неизвестных.

**Описание метода**

Суть метода заключается в последовательном приближении за счет задания какого-то изначального приближения и погрешности приближения. Один ход данного цикла называется итерацией.

**Рабочие формулы**

**Блок-схема для метода простых итераций**



**Программная реализация**

**public** SimpleIteration (**double** eps, **double**[][] matrix, **int** size) {**previousVariableValues** = **new double**[size];  
 **for** (**int** i = 0; i < size; i++) {  
 **previousVariableValues**[i] = 0.0;  
 }**while** (**true**) {**double**[] currentVariableValues = **new double**[size];**for** (**int** i = 0; i < size; i++) {currentVariableValues[i] = matrix[i][size];**for** (**int** j = 0; j < size; j++) {  
 **if** (i != j) {  
 currentVariableValues[i] -= matrix[i][j] \* **previousVariableValues**[j];  
 }  
 }currentVariableValues[i] /= matrix[i][i];  
 }**double** error = 0.0;  
 **double**[] a = **new double**[size];  
 **for** (**int** i = 0; i < size; i++) {  
 error += Math.*abs*(currentVariableValues[i] - **previousVariableValues**[i]);  
 a[i] = error;  
 }  
 **errorList**.add(a);**iterationCounter**++;  
 **if** (error < eps) {  
 **break**;  
 }**previousVariableValues** = currentVariableValues;  
 }  
}

**Работа программы**

Ввод из файла/из строки (true/false): true

Введите название файла: lol

Вектор неизвестных:

0.4997340957751283 0.092984535209731 -0.1264021389045821 0.0791854162278084 0.05020835351119276

Количество итераций: 15

Вектор погрешностей:

0.5 0.6764705882352942 0.7291021671826625 0.8957688338493291 1.038625976706472

0.09632168656936457 0.22546808196962995 0.447641161727849 0.59296771340115 0.7461447736989533

0.13764927023440954 0.22235237748350112 0.3177883358003458 0.41792470613795457 0.5210838645381568

0.07621687765867308 0.13943522349464477 0.22202294654214957 0.29218086224224094 0.3639848355832475

0.05879657231965624 0.10160674560826835 0.15575374792522817 0.20472490123908751 0.25549621667883216

0.03985751377214525 0.07023035978856719 0.10897808171808204 0.14339892561224415 0.17876107911620126

0.028213400872346417 0.04941045771603672 0.07636063321025889 0.10041733908059874 0.1252418778846939

0.01969504236485775 0.034550439498095956 0.053472792328921714 0.0703369876646474 0.0877104727966887

0.013808901835876353 0.02421308387746085 0.03745287700156405 0.049260589911724564 0.06143093059713679

0.009667613773646333 0.016954332299659924 0.026230964884272068 0.034501501165434795 0.04302504001590752

0.0067720948787971524 0.011875542918170917 0.01837164250806446 0.02416405979718861 0.030133805467395525

0.004742710347796719 0.008317094980293177 0.012867106647046225 0.016924004676758128 0.021105077224838467

0.0033217847391123767 0.005825190214918161 0.009011853340010118 0.011853216539197867 0.014781557869091376

0.002326484618586089 0.004079823037899999 0.006311711039536275 0.008301743484520202 0.010352689486663576

0.0016294265292806642 0.002857426406389779 0.004420590032419736 0.0058143662060330115 0.007250806478009311

Process finished with exit code 0

**Вывод**

В процессе выполнения данной работы был изучен метод простых итераций, который позволяет, действительно, реализовать более простой для понимания алгоритм, по сравнению, к примеру, с методом Гаусса, но так как данный метод является итерационным, то он имеет и свои минусы, что включает лишь приближенные вычисления, а также алгоритм данного метода является сложным по сравнению с прямыми методами.