Национальный исследовательский университет ИТМО

Факультет ПиИКТ

Лабораторная работа № 1

по «Вычислительной математике»

Решение системы линейных алгебраических уравнений СЛАУ

Работу выполнил: Велюс Арина

Группа: P32151

Преподаватель: Машина Екатерина Александровка

Санкт-Петербург

2023 г.

Цель работы:

Используя известные методы вычислительной математики, написать программный код, осуществляющий решение СЛАУ. Проанализировать полученные результаты, оценить погрешность.

Вариант №2 (Метод простых итераций)

Задание лабораторной работы:

1. № варианта определяется как номер в списке группы согласно ИСУ.
2. В программе численный метод должен быть реализован в виде отдельной подпрограммы/метода/класса, в который исходные/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).

<u>Для итерационных методов должно быть реализовано:</u>

- 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: $x_1, x_2, x_3, \dots, x_n$
- Вывод количества итераций, за которое было найдено решение.
- Вывод вектора погрешностей: $\left| x_i^{(k)} - x_i^{(k-1)} \right|$

Описание метода, расчетные формулы:

Суть методы состоит в построении последовательности векторов решений $\vec{x}$, которая стремится к точному решению.

Рассмотрим систему линейных уравнений с невырожденной матрицей ($\det A \neq 0$):

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ a_{21}x_1 + a_{22}x_2 + \dots + a_{2n}x_n = b_2 \\ \dots \dots \\ a_{n1}x_1 + a_{n2}x_2 + \dots + a_{nn}x_n = b_n \end{cases} \qquad (5)$$

Приведем систему уравнений к виду (6), выразив неизвестные x_1, x_2, x_3 соответственно из первого, второго и третьего уравнений системы (5):

$$\begin{cases} x_1 = c_{11}x_1 + c_{12}x_2 + \dots + c_{1n}x_n + d_1 \\ x_2 = c_{21}x_1 + c_{22}x_2 + \dots + c_{2n}x_n + d_2 \\ \dots \dots \\ x_n = c_{n1}x_1 + c_{n2}x_2 + \dots + c_{nn}x_n + d_n \end{cases} \qquad (6)$$

Или в векторно-матричном виде: $\boldsymbol{x = Cx + D}$, где x – вектор неизвестных, C – матрица коэффициентов преобразованной системы размерности $n*n$, D – вектор правых частей преобразованной системы.

Систему (2) представим в сокращенном виде:

$$\boldsymbol{x_i = \sum_{j=1}^{n} c_{ij}x_j + d_i} \ , \qquad i = 1, 2, \dots, n$$

$$c_{ij} = \begin{cases} 0, \ \text{при } i = j \\ -\frac{a_{ij}}{a_{ii}}, \ \text{при } i \neq j \end{cases} \qquad d_i = \frac{b_i}{a_{ii}} \quad i = 1, 2, \dots, n$$

Рабочая формула метода простой итерации:

$$\boldsymbol{x_i^{(k+1)} = \frac{b_i}{a_{ii}} - \sum_{\substack{j=1 \\ j\neq i}}^{n} \frac{a_{ij}}{a_{ii}} x_j^k} \ , \qquad \boldsymbol{i = 1, 2, \dots, n}$$

где k – номер итерации.

За начальное (нулевое) приближение выбирают вектор свободных членов: $x^{(0)} = D$ или нулевой вектор: $x^{(0)} = 0$

Следующее приближение: $\vec{x}^{(1)} = c\vec{x}^{(0)} + \vec{d}, \ \ \vec{x}^{(2)} = c\vec{x}^{(1)} + \vec{d}$ ...

Теорема. Достаточным условием сходимости *итерационного процесса* к решению системы при любом начальном векторе $x_i^{(0)}$ является выполнение условия преобладания диагональных элементов или доминирование диагонали:

$$|a_{ii}| \geq \sum_{j\neq i} |a_{ij}| \ , \qquad i = 1, 2, \dots, n$$

При этом хотя бы для одного уравнения неравенство должно выполняться строго. Эти условия являются достаточными для сходимости метода, но они не являются необходимыми, т. е. для некоторых систем итерации сходятся и при нарушении этого условия.

Теорема. Достаточным условием сходимости итерационного метода к решению системы при любом начальном векторе $x_i^{(0)}$ является требование к норме матрицы С:

$$\|C\| < 1$$

$$\|C\| = \max_{1\leq i\leq n} \sum_{j=1}^{n} |c_{ij}| < 1$$

$$\|C\| = \max_{1\leq j\leq n} \sum_{i=1}^{n} |c_{ij}| < 1$$

$$\|C\| = \sqrt{\sum_{i=1}^{n} \sum_{j=1}^{n} c_{ij}^2 < 1}$$

Условие сходимости $\|C\| < 1$ в этом методе равносильно условию диагонального преобладания.

Листинг программы:

<u>Процедура, приводящая матрицу к виду 6</u>

```
private void modifyMatrix() {
   for (int i = 0; i < matrix.length; i++) {
      int currentIndex = i;
      matrix[i] = Arrays.stream(matrix[i]).
            map(x -> -x / matrix[currentIndex][currentIndex]).
            toArray();
      matrix[i][i] = 0;
      matrix[i][n] *= -1;
   }
}
```

<u>Метод итераций</u>

```
private int iterate() {
   int currentIter = 0;
   double maxEps = Double.MAX_VALUE;
   error = new double[solution.length];
   while (currentIter < MAX_ITERATION && maxEps >= eps) {
      double currentMaxEps = 0;
      double[] currentSolution = Arrays.copyOf(solution, solution.length);
      for (int i = 0; i < currentSolution.length; i++) {
         double newValue = 0;
         for (int j = 0; j < currentSolution.length; j++) {
            if (i == j) continue;
            newValue += matrix[i][j]*currentSolution[j];
         }
         newValue += matrix[i][matrix.length];
         error[i] = Math.abs(newValue - solution[i]);
         if (error[i] > currentMaxEps) currentMaxEps = error[i];
         solution[i] = newValue;
      }
      maxEps = currentMaxEps;
      currentIter++;
   }
   return currentIter;
}
```

Вывод:
В ходе работы реализован метод простых итераций, позволяющий решать СЛАУ с высокой точностью. Метод несложен в программном изложении и довольно быстро справляется с небольшими по размеру матрицами.