Санкт-Петербургский национальный исследовательский университет ИТМО

Вычислительная математика

Лабораторная работа №1
Решение системы линейных алгебраических уравнений СЛАУ

Автор:

Ненов Владислав Александрович

Вариант 4

Группа №P32082

Преподаватель:

Екатерина Алексеевна Машина

Санкт-Петербург
2023

Задание

Разработать программу для подсчета корней СЛАУ.

Для итерационных методов 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: $x1$, $x2$, … , xn
• Вывод количества итераций, за которое было найдено решение.
• Вывод вектора погрешностей.

Выполнение

Метод простых итераций - численный метод решения системы линейных алгебраических уравнений. Суть метода заключается в нахождении по приближённому значению величины следующего приближения, являющегося более точным.

Метод позволяет получить значения корней системы с заданной точностью в виде предела последовательности некоторых векторов (в результате итерационного процесса).

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

Достаточным условием сходимости итерационного процесса к решению системы при любом начальном векторе является выполнение условия преобладания диагональных элементов.

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

Листинг программы

```
public boolean compute() {
    if (!diagonalDominance())
        return false;
    b.divide(a.get(0, 0));
    iterationsCount = 0;

    results = b.asPlainArray();
    double error = Double.MAX_VALUE;
    double[] diff = null;
    while (error > precision) {
        diff = results.clone();
        results = computeIteration(results);
        for (int i = 0; i < results.length; i++) {
            diff[i] = Math.abs(results[i] - diff[i]);
        }
        error = Arrays.stream(diff).max().orElseThrow();
        iterationsCount++;
    }
    resultErrors = diff;
    return true;
}
```

```
    private double[] computeIteration(double[] results) {
        double[] newResult = new double[results.length];

        for (int i = 0; i < a.size(); i++) {
            double temp = 0;
            for (int j = 0; j < a.size(); j++) {
                if (j == i)
                    continue;
                temp += (a.get(i, j) / a.get(i ,i)) *
results[j]; }
            newResult[i] = b.get(i, 0) - temp;
        }
        return newResult;
    }
```

```
    private static boolean checkDiagDominance(QuadMatrix matrix)
{       boolean success = true;
        boolean anyStrict = false;

        for (int i = 0; i < matrix.size(); i++) {
            double sum = matrix.sumModules(i) - matrix.get(i, i);
            double value = Math.abs(matrix.get(i, i));
            if (value < sum) {
                success = false;
                break;
            }
            anyStrict = anyStrict || value > sum;
        }
        return success && anyStrict;
    }
```

```
    private boolean permutations(QuadMatrix coeffs, Matrix numbs, int ind)
{       // if branch ended
        if (ind == coeffs.size() - 1) {
            boolean result = checkDiagDominance(coeffs);
            if (result) {
                a = coeffs;
                b = numbs;
            }
            return result;
        }
        // else
        for (int i = ind; i < coeffs.size(); i++)
        {
            coeffs.replaceLines(i, ind);
            numbs.replaceLines(i, ind);
            if (permutations(coeffs, numbs, ind + 1))
                return true;
            coeffs.replaceLines(i, ind);
            numbs.replaceLines(i, ind);
        }
        return false;
    }
```

OBJ

Пример работы

2 Входные данные:
3 0.01
2 2 10 14
10 1 1 12
2 10 1 13

Результат вычислений:
Результат: 0,9996; 0,9995; 0,9993;
Вектор погрешностей: 0,0019; 0,0025; 0,0031;
Итераций: 5

1 Входные данные:
3 0.01
1 3 5 1
1 3 6 8
9 8 5 2

Результат вычислений:
Не обнаружено диагонального преобладания!

Вывод

Во время выполнения лабораторной работы я изучил работу метода простых итераций. Основными недостатками можно назвать медленную сходимость, а также трудоемкость получения диагонального вида матрицы для получения достаточного условия сходимости.