Университет ИТМО

Факультет программной инженерии и компьютерной техники
Направление подготовки 09.03.01 Информатика и вычислительная техника

Лабораторная работа №1
по дисциплине «Вычислительная математика»
Метод простой итерации

Выполнил:
Студент группы P32301
Овчаренко А.А.

Преподаватель:
Перл Ольга Вячеславовна

г. Санкт-Петербург
2023

Table of Contents

Описание метода

Метод простой итерации относится к итерационным методам решения линейных систем. Он дает возможность получить решение для системы с заданной точностью через последовательное вычисление вектора решения.

Важным условием работы метода простой итерации является сходимость последовательности векторов решения $x^{(0)}$, $x^{(1)}$, … , $x^{(k)}$. Таким образом перед применением метода простой итерации лучше проверить условие сходимости метода.

Теорема: для сходимости итерационного метода при любом начальном векторе $x^{(0)}$ к решению системы достаточно наличия у матрицы системы линейных уравнений диагонального преобладания.

Этапы использования метода простой итерации:

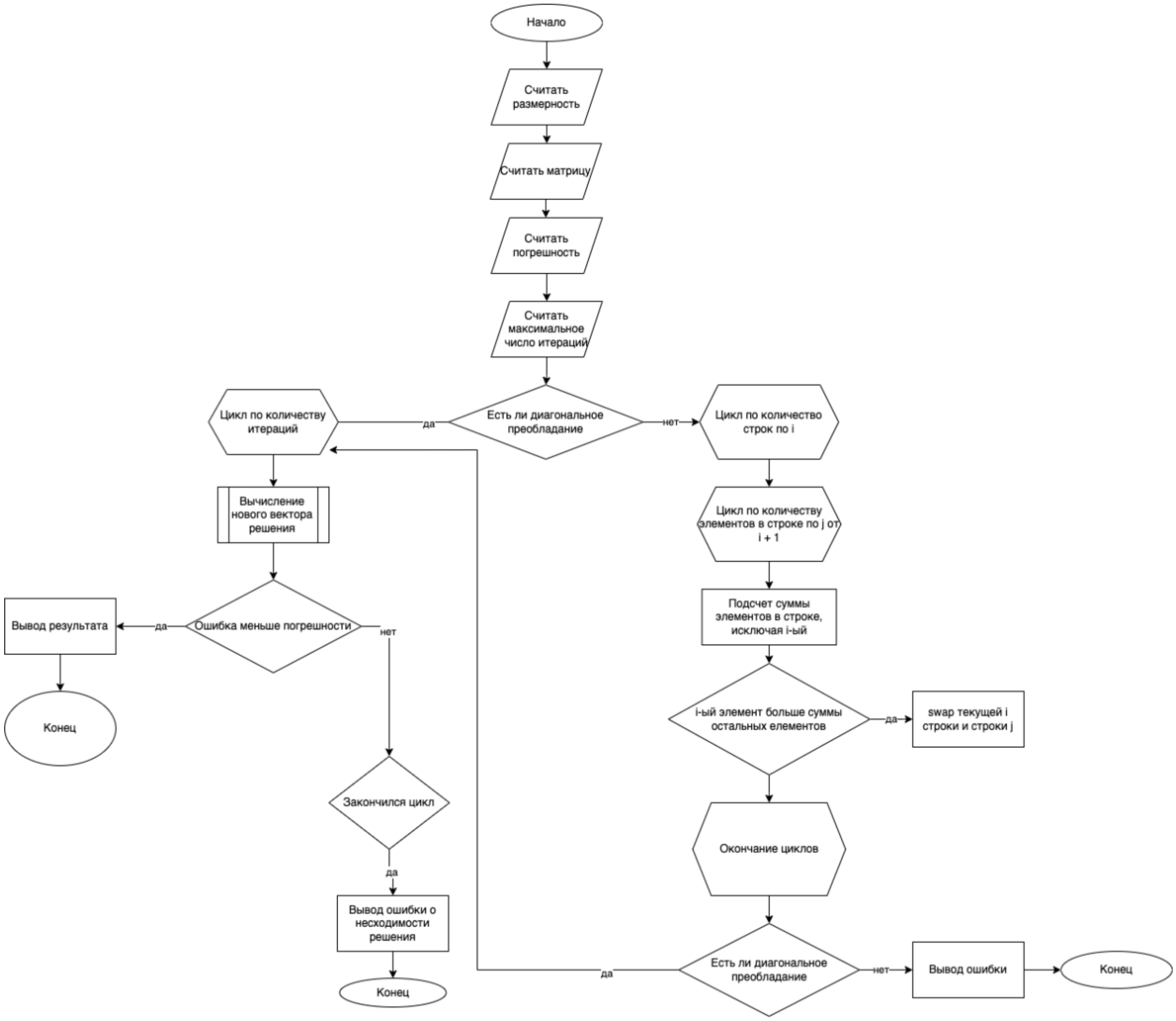
1. Проверка диагонального преобладания. Если матрица имеет преобладание, то переходим на следующий этап. Если нет, то с помощью элементарных преобразований приводим матрицу к диагональному преобладанию. Если это сделать не получается, то применяем другой метод для решения системы.
2. Имеем систему линейных уравнений с матрицей диагонального преобладания к виду:

$$\begin{cases} x_1 = \dfrac{b_1}{a_{11}} - \dfrac{a_{12}}{a_{11}}x_2 - \cdots - \dfrac{a_{1n}}{a_{11}}x_n \\ \vdots \\ x_n = \dfrac{b_n}{a_{nn}} - \dfrac{a_{n1}}{a_{nn}}x_1 - \dfrac{a_{n2}}{a_{nn}}x_2 - \cdots - \dfrac{a_{n,n-1}}{a_{nn}}x_{n-1} \end{cases}$$

   Таким образом исходная система Ax = B преобразуется к виду x = b + a x.

3. В качестве нулевого приближения используем $x^{(0)}$ = b. Итеративно находим (k+1)ое приближение через k: $x^{(k+1)}$ = b + a $x^{(k)}$.
4. Оценку погрешности можно производить, используя три критерия:
   a. Критерий по абсолютным отклонениям – максимальная по модулю разница между соответствующими координатами текущего решения и предыдущего.
   b. Критерий по относительным разностям – максимальная по модулю разница между соответствующими координатами текущего решения и предыдущего, деленную на соответствующую координату текущего решения.
   c. Критерий по невязке

Блок-схема



Начало
Считать размерность
Считать матрицу
Считать погрешность
Считать максимальное число итераций
Есть ли диагональное преобладание
да
нет
Цикл по количеству итераций
Вычисление нового вектора решения
Ошибка меньше погрешности
да
нет
Вывод результата
Конец
Закончился цикл
да
Вывод ошибки о несходимости решения
Конец
Цикл по количество строк по i
Цикл по количеству элементов в строке по j от i + 1
Подсчет суммы элементов в строке, исключая i-ый
i-ый элемент больше суммы остальных елементов
да
swap текущей i строки и строки j
Окончание циклов
Есть ли диагональное преобладание
да
нет
Вывод ошибки
Конец

Листинг

Код программы доступен по ссылке – github.com

Примеры работы программы

- Ввод пользователя

```
Enter dimension
6
Enter rows
1 2 3 4 5 19 2
6 4 6 25 3 5 5
1 1 2 5 23 2 7
22 6 7 4 2 1 1
0 9 0 0 0 0 0
1 4 15 2 3 3 3
Enter accuracy
0.0001
Enter maximum number of iterations
1000
Success
Number of iterations: 17
Answer is
-0.0476695782206128
0.0
0.1332520954818635
0.1506745673688162
0.26328859327225557
-0.014322792773670306
Error rates
7.306442027546489E-5
0.0
6.784720013888745E-5
8.812152507450821E-5
5.520530344510499E-5
7.556212617081635E-5
```

- Генерация матрицы

```
Enter dimension
5
Enter accuracy
0.00001
Enter maximum number of iterations
1000
3.3759081912440045 0.7353197002234411 0.05927991055627424 0.9845997038881132 0.8307977729477823

0.7569555751107776 2.4452380330700514 0.8251321446873494 0.4583061944670568 0.44955260716100465

0.7812713612220277 0.46703287823212425 2.2450984811499404 0.5901064875183443 0.5793500721935458

0.7624144455047076 0.4321399830508086 0.45088207290706706 3.011808362151462 0.9492660660256603

0.745284751991128 0.009862489989521106 0.528700761128547 0.10674317460555605 3.5125795043859935

8.474880635006386 3.5498840647041807 9.192436181065132 6.574889493135342 8.718603532559717

Success
Number of iterations: 51
Answer is
1.1226158352274558
0.26500409056725616
3.442507333050185
1.0540759606195365
1.3300094260347377
Error rates
9.6468342509759E-6
7.728176087129501E-6
8.71516555012164E-6
8.128535924578628E-6
8.838203933825284E-6
```

- Чтение из файла

Данные:

5

10.782 -3.37 -1.689 -0.141 -2.091 123.81377

-0.041 -8.639 -0.256 -1.705 -2.365 -146.44079

-1.067 0.065 9.923 2.125 -0.559 -207.77786

1.442 -0.398 -1.634 9.801 -2.002 -326.33278

0.394 -1.758 -0.9 0.887 -8.993 173.40532

0.000001

1000

Результат:

```
Success
Number of iterations: 15
Answer is
13.94081309274644
33.17256648602075
-12.290572778279772
-41.77791867642831
-28.046870696403747
Error rates
4.987252459898173E-7
7.410160662857379E-7
3.274976787537298E-7
3.5592690039720765E-7
6.170996620369351E-7
```

Размерность матрицы	Точность	Количество итераций	Время в милисекудах
10	0.0000000001	100	2
100	0.0000000001	100	14
1000	0.0000000001	100	152
10000	0.0000000001	100	14886
10	0.0000000001	200	3
100	0.0000000001	1000	38
1000	0.0000000001	1000	1158
10000	0.0000000001	1000	136480

Вывод

Метод Гаусса для решения СЛАУ крайне удобен, так как прост в реализации и дает аналитически точный ответ. В отличии от итерационных методов, точность которых зависит еще и от количества итераций, метод Гаусса дает не 100% точный ответ только из-за погрешности float арифметики. Сложность алгоритма составляет O(n^3) из-за трех вложенных циклов до n, где n – размер матрицы. Также метод Гаусса корректно работает для любых матриц, не накладывая ограничений на нее. Еще хочется отметить, что после применения метода Гаусса очень легко вычислить определитель, так как во время работы алгоритма матрица триангулируется. Метод Гаусса-Зейделя схож с методом простой итерации. Он будет работать быстрее, но при этом более сложен в реализации.

Метод простой итерации для решения СЛАУ прост в реализации на ЭВМ, но имеет ряд существенных ограничений.

Достоинством данного метода является экономия памяти, так как не нужно хранить всю матрицу, нужно только одна строка на каждой итерации. Также его просто реализовывать на ЭВМ в силу простоты самого алгоритма.

К недостаткам можно отнести требование на сходимость последовательности решений, метод Гаусса таких ограничений не имеет. Также низкая скорость работы: чтобы получить результат с большой точностью нужно много раз пересчитывать все значений.

Время вычислений зависит от размера матрицы и от количества итераций. Теоретическая сложность данного метода $O(n^2*maxIter)$. По предоставленным наблюдается немного другая зависимость: время зависит не от квадрата размера матрицы, а от чуть меньшей степени, зависимость от количества итераций линейная.