Семестр 2 (2021), занятие 6. Матрицы

Метод Гаусса

Пусть требуется решить систему линейных уравнений $Ax = b$, где $A = (a_{ij}) \in \mathbb{R}^{n\times n}$, $b = (b_i) \in \mathbb{R}^n$:

$$\begin{array}{ccccccccc} a_{11}x_1 & + & a_{12}x_2 & + & \dots & + & a_{1n}x_n & = & b_1 \\ a_{21}x_1 & + & a_{22}x_2 & + & \dots & + & a_{2n}x_n & = & b_2 \\ & & & & \dots & & & & \\ a_{n1}x_1 & + & a_{n2}x_2 & + & \dots & + & a_{nn}x_n & = & b_n \end{array}$$

Метод Гаусса состоит из двух частей: прямой ход (сверху вниз) и обратный ход (снизу вверх).

Прямой ход

Прямой ход метода Гаусса преобразует матрицу в левой части к треугольному виду с главной диагональю, состоящей из единиц. В результате система линейных уравнений принимает вид

$$\begin{array}{ccccccccc} x_1 & + & c_{12}x_2 & + & \dots & + & c_{1n}x_n & = & y_1 \\ & & x_2 & + & \dots & + & c_{2n}x_n & = & y_2 \\ & & & & \dots & & & & \\ & & & & & & x_n & = & y_n \end{array}$$

Первый шаг прямого хода

Предположим, что $a_{11} \neq 0$. Первый шаг прямого хода метода Гаусса преобразует систему линейных уравнений следующему к виду

$$\begin{array}{ccccccccc} x_1 & + & c_{12}x_2 & + & \dots & + & c_{1n}x_n & = & y_1 \\ & & a_{22}^{(1)}x_2 & + & \dots & + & a_{2n}^{(1)}x_n & = & b_2^{(1)} \\ & & & & \dots & & & & \\ & & a_{n2}^{(1)}x_2 & + & \dots & + & a_{nn}^{(1)}x_n & = & b_n^{(1)} \end{array}$$

Новые коэффициенты вычисляются по формулам

$$\begin{array}{rcl} c_{1j} & = & \frac{a_{1j}}{a_{11}}, \\ y_1 & = & \frac{b_1}{a_{11}}, \\ a_{ij}^{(1)} & = & a_{ij} - a_{i1}c_{1j}, \\ b_i^{(1)} & = & b_i - a_{i1}y_1, \end{array}$$

где $i, j = 2, \dots, n$.

Далее, этот процесс применяется к подматрице $A^{(1)} = (a_{ij}^{(1)})$ и вектору правой части $b = (b_i^{(1)})$.

Последующие шаги прямого хода

Предположим, что сделано $k-1$ ($k = 2, \dots, n$) шагов прямого хода. Положим

$$\begin{array}{rcl} c_{kj} & = & \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}, \\ y_k & = & \frac{b_k^{(k-1)}}{a_{kk}^{(k-1)}}, \\ a_{ij}^{(k)} & = & a_{ij}^{(k-1)} - a_{ik}^{(k-1)}c_{kj}, \\ b_i^{(k)} & = & b_i^{(k-1)} - a_{ik}^{(k-1)}y_k, \end{array}$$

где $i, j = k+1, \dots, n$.

Обратный ход

Обратный ход (снизу вверх) метода Гаусса задается следующими формулами.

$$\begin{array}{rcl} x_n & = & y_n, \\ x_i & = & y_i - \sum_{j=i+1}^n c_{ij}x_j, \quad i = n-1, \dots, 1. \end{array}$$

Свойства

Утверждение. Метод Гаусса осуществим тогда и только тогда, когда у матрицы A все главные угловые миноры отличны от нуля.

Утверждение. Метод Гаусса требует $\frac{2}{3}n^3 + O(n^2)$ арифметических операций.

Оценка точности

Учитывая, что арифметические операции выполняются с погрешностью, в результате решения системы уравнений $Ax = b$ будет получено некоторое приближенное решение $x^* \approx x$. Точность приближения оценивается с помощью величины

$$\| Ax^* - b \|,$$

где

$$\| z \| = \sqrt{\sum_{i=1}^n z_i^2} \quad (z \in \mathbb{R}^n).$$

Модификации метода Гаусса

Первый шаг прямого хода

Внесем следующее изменение в первый шаг прямого хода метода Гаусса. В самом начале найдем уравнение с максимальным по модулю коэффициентом при переменной x_1. Поменяем местами это и первое уравнение.

Аналогичные изменения внесем и в последующие шаги прямого хода метода Гаусса. Получившийся алгоритм носит название метода Гаусса с выбором главного элемента по столбцу.

Заметим, что все переменные равноправны друг с другом. Поэтому на первом шаге метода Гаусса можно найти переменную с максимальным по модулю коэффициентом в первом уравнении. Поменяем друг с другом номера этой и первой переменной.

Аналогичные изменения внесем и в последующие шаги прямого хода метода Гаусса. В результате получим метод Гаусса с выбором главного элемента по строке.

Максимальный по модулю элемент можно искать и по всей матрице. С помощью перестановки уравнений и перенумерования переменных можно добиться, чтобы этот элемент был коэффициентом при первой переменной в первом уравнении. Подобная модификация алгоритма называется методом Гаусса с выбором главного элемента по всей матрице.

Свойства

Утверждение. Метод Гаусса с выбором главного элемента по столбцу (строке, всей матрице) осуществим тогда и только тогда, когда $\det A \neq 0$.

Утверждение. Метод Гаусса с выбором главного элемента по столбцу (строке) требует $\frac{2}{3}n^3 + O(n^2)$ арифметических операций.

Метод Гаусса с выбором главного элемента по всей матрице требует $n^3 + O(n^2)$ арифметических операций.

Метод Жордана

Пусть требуется решить систему линейных уравнений $Ax = b$, где $A = (a_{ij}) \in \mathbb{R}^{n\times n}$, $b = (b_i) \in \mathbb{R}^n$.

Первый шаг

Первый шаг метода Жордана совпадает с первым шагом метода Гаусса. Предположим, что $a_{11} \neq 0$. Система линейных уравнений приводится к виду

$$\begin{array}{ccccccccc}
x_1 & + & a_{12}^{(1)}x_2 & + & \dots & + & a_{1n}^{(1)}x_n & = & b_1^{(1)} \\
 & & a_{22}^{(1)}x_2 & + & \dots & + & a_{2n}^{(1)}x_n & = & b_2^{(1)} \\
 & & & & \dots & & & & \\
 & & a_{n2}^{(1)}x_2 & + & \dots & + & a_{nn}^{(1)}x_n & = & b_n^{(1)}
\end{array}$$

Новые коэффициенты вычисляются по формулам

$$\begin{array}{rcl}
a_{1j}^{(1)} & = & \frac{a_{1j}}{a_{11}}, \\
b_1^{(1)} & = & \frac{b_1}{a_{11}}, \\
a_{ij}^{(1)} & = & a_{ij} - a_{1j}^{(1)}a_{i1}, \\
b_i^{(1)} & = & b_i - b_1^{(1)}a_{i1},
\end{array}$$

где $i, j = 2, \dots, n$.

Последующие шаги

Предположим, что сделано $k-1$ ($k = 2, \dots, n$) шагов прямого хода. Система линейных уравнений имеет вид (1).

Предположим, что $a_{k,k}^{(k-1)} \neq 0$. Новые коэффициенты вычисляются по формулам

$$\begin{array}{rcl}
a_{kj}^{(k)} & = & \frac{a_{kj}^{(k-1)}}{a_{kk}^{(k-1)}}, \\
b_k^{(k)} & = & \frac{b_k^{(k-1)}}{a_{kk}^{(k-1)}}, \\
a_{ij}^{(k)} & = & a_{ij}^{(k-1)} - a_{ik}^{(k-1)}a_{kj}^{(k)}, \\
b_i^{(k)} & = & b_i^{(k-1)} - a_{ik}^{(k-1)}b_k^{(k)},
\end{array}$$

где $i = 1, \dots, n$ ($i \neq k$); $j = k+1, \dots, n$.

Результат

После проведения вычислений матрица в левой части станет единичной, а вектор правой части будет искомым решением:

$$x_i = b_i^{(n)},$$

где $i = 1, \dots, n$.

По аналогии с методом Гаусса можно построить метод Жордана с выбором главного элемента по столбцу (строке, всей матрице).

$$\begin{array}{cccccccccccc}
x_1 & & & + & a_{1,k}^{(k-1)} & x_k & + & \dots & + & a_{1,n}^{(k-1)} & x_n & = & b_1^{(k-1)} \\
 & \ddots & & \vdots & & & \vdots & & \vdots & & & \vdots & \\
 & & x_{k-1} & + & a_{k-1,k}^{(k-1)} & x_k & + & \dots & + & a_{k-1,n}^{(k-1)} & x_n & = & b_{k-1}^{(k-1)} \\
 & & & & a_{k,k}^{(k-1)} & x_k & + & \dots & + & a_{k,n}^{(k-1)} & x_n & = & b_k^{(k-1)} \\
 & & & & & & \vdots & & \vdots & & & \vdots & \\
 & & & & a_{n,k}^{(k-1)} & x_k & + & \dots & + & a_{n,n}^{(k-1)} & x_n & = & b_n^{(k-1)}
\end{array} \qquad (1)$$

Обращение матрицы

Пусть $A = (a_{ij}) \in \mathbb{R}^{n\times n}$ – невырожденная матрица, $E \in \mathbb{R}^{n\times n}$ – единичная матрица. Требуется найти матрицу $A^{-1} \in \mathbb{R}^{n\times n}$ такую, что

$$AA^{-1} = E.$$

Алгоритм

Рассмотрим вектора

$$e_j = (\underbrace{0,\dots,0}_{j-1},1,0,\dots,0)^t.$$

Заметим что, j-й столбец обратной матрицы A^{-1} является решением системы уравнений $Ax = e_j$.

Пусть $\mathfrak{A}(A, b)$ – это алгоритм решения системы линейных уравнений $Ax = b$. Тогда для нахождения обратной матрицы достаточно с по мощью этого алгоритма провести n вычислений:

$$\mathfrak{A}(A,e_1),\mathfrak{A}(A,e_2),\dots,\mathfrak{A}(A,e_n).$$

Если $\mathfrak{A}$ является методом Гаусса или его модификацией, то эти вычисления можно осуществлять «одновременно». Матрица коэффициентов при переменных в этих вычислениях преобразуется одинаково.

Оценка точности

Учитывая, что арифметические операции выполняются с погрешностью, в результате будет получено некоторое приближенное решение $A^* \approx A^{-1}$. Точность приближения оценивается с помощью величины

$$\| AA^* - E \|,$$

где

$$\| B \| = \max_{1\leqslant j\leqslant n} \sum_{i=1}^{n} |b_{ij}| \quad (B = (b_{ij}) \in \mathbb{R}^{n\times n}).$$

Тестовые матрицы

Симметричные якобиевые матрицы порядка $n \in \mathbb{N}$ с одинаковыми диагональными элементами имеют следующий вид

$$A_n = \begin{pmatrix} d & c & 0 & 0 & \dots & 0 \\ c & d & c & 0 & \dots & 0 \\ 0 & c & d & c & \dots & 0 \\ \vdots & \vdots & \ddots & \ddots & \ddots & \vdots \\ 0 & 0 & \dots & c & d & c \\ 0 & 0 & \dots & 0 & c & d \end{pmatrix} \quad (c \neq 0).$$

Определитель

Их определители связаны соотношениями

$$\begin{aligned} \det(A_1) &= d, \\ \det(A_2) &= d^2 - c^2, \\ \det(A_n) &= d\det(A_{n-1}) - c^2\det(A_{n-2}) \quad (n > 2). \end{aligned}$$

Система уравнений

Рассмотрим n мерный вектор

$$x = (1,0,1,0,1,0,\dots)^t.$$

Тогда он является решением системы уравнений

$$A_n x = (d, 2c, d, 2c, d, 2c, \dots, d)^t,$$

если n – нечетно,

$$A_n x = (d, 2c, d, 2c, d, 2c, \dots, c)^t,$$

если n – четно.

Обращение матриц

Примеры симметричных якобиевых матриц

$$A_n^{(1)} = \begin{pmatrix} 2 & -1 & & \\ -1 & 2 & -1 & \\ & \ddots & \ddots & \ddots \end{pmatrix},$$

$$A_n^{(2)} = \begin{pmatrix} 2 & 1 & & \\ 1 & 2 & 1 & \\ & \ddots & \ddots & \ddots \end{pmatrix},$$

$$A_n^{(3)} = \begin{pmatrix} -2 & 1 & & \\ 1 & -2 & 1 & \\ & \ddots & \ddots & \ddots \end{pmatrix},$$

$$A_n^{(4)} = \begin{pmatrix} -2 & -1 & & \\ -1 & -2 & -1 & \\ & \ddots & \ddots & \ddots \end{pmatrix}.$$

Примеры обратных матриц. Случай $n = 2$:

$$A_2^{(1)^{-1}} = \frac{1}{3}\begin{pmatrix} 2 & 1 \\ 1 & 2 \end{pmatrix},$$

$$A_2^{(2)^{-1}} = \frac{1}{3}\begin{pmatrix} 2 & -1 \\ -1 & 2 \end{pmatrix}.$$

Случай $n = 3$:

$$A_3^{(1)^{-1}} = \frac{1}{4}\begin{pmatrix} 3 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 3 \end{pmatrix},$$

$$A_3^{(2)^{-1}} = \frac{1}{4}\begin{pmatrix} 3 & -2 & 1 \\ -2 & 4 & -2 \\ 1 & -2 & 3 \end{pmatrix}.$$

Случай $n = 4$:

$$A_4^{(1)^{-1}} = \frac{1}{5}\begin{pmatrix} 4 & 3 & 2 & 1 \\ 3 & 6 & 4 & 2 \\ 2 & 4 & 6 & 3 \\ 1 & 2 & 3 & 4 \end{pmatrix},$$

$$A_4^{(2)^{-1}} = \frac{1}{5}\begin{pmatrix} 4 & -3 & 2 & -1 \\ -3 & 6 & -4 & 2 \\ 2 & -4 & 6 & -3 \\ -1 & 2 & -3 & 4 \end{pmatrix}.$$

Случай $n = 5$:

$$A_5^{(1)^{-1}} = \frac{1}{6}\begin{pmatrix} 5 & 4 & 3 & 2 & 1 \\ 4 & 8 & 6 & 4 & 2 \\ 3 & 6 & 9 & 6 & 3 \\ 2 & 4 & 6 & 8 & 4 \\ 1 & 2 & 3 & 4 & 5 \end{pmatrix},$$

$$A_5^{(2)^{-1}} = \frac{1}{6}\begin{pmatrix} 5 & -4 & 3 & -2 & 1 \\ -4 & 8 & -6 & 4 & -2 \\ 3 & -6 & 9 & -6 & 3 \\ -2 & 4 & -6 & 8 & -4 \\ 1 & -2 & 3 & -4 & 5 \end{pmatrix}.$$

Программная реализация

Хранение матрицы

Для хранения матрицы $A = (a_{ij}) \in \mathbb{R}^{n\times m}$ используется одномерный массив `a` элементов типа `double` размера `n * m`. Элементу a_{ij} матрицы A сопоставляется элемент массива `a[(i)*m + j]`.

Рекомендуется для доступа к элементам матрицы использовать макрос

```
#define A(i,j) a[(i)*m + j]
```

Если предполагается в процессе работы с матрицей менять местами ее строки, то необходимо дополнительно завести массив `indi` элементов типа `int` размера `n`. В начале `indi[i] = i`.

Элементу a_{ij} матрицы A сопоставляется элемент массива `a[indi[i]*m + j]`. Соответствующий макрос принимает вид

```
#define A(i,j) a[indi[i]*m + j]
```

Для того, чтобы поменять местами строки матрицы A с номерами i_1 и i_2, достаточно поменять местами значения элементов массива `indi` с номерами i_1 и i_2.

Если предполагается менять местами и столбцы матрицы, то необходимо дополнительно завести массив `indj` элементов типа `int` размера `m`. В начале `indj[j] = j`.

Элементу a_{ij} матрицы A сопоставляется элемент массива `a[indi[i]*m + indj[j]]`. Соответствующий макрос принимает вид

```
#define A(i,j) a[indi[i]*m + indj[j]]
```

Для того, чтобы поменять местами столбцы матрицы A с номерами j_1 и j_2, достаточно поменять местами значения элементов массива `indj` с номерами j_1 и j_2.

Прототип функции

Прототип функции, реализующей решение системы линейных уравнений методом Гаусса (с выбором главного элемента по столбцу) имеет вид

```
int gauss(double *a,
          double *b,
          double *x,
          int    *indi,
          int     n);
```

В случае успеха функция возвращает `0`, а в случае ошибки (матрица A является вырожденной) возвращает `-1`.

Параметры:

- `a` – массив типа `double` размера n^2, в котором хранится матрица A;
- `b` – массив типа `double` размера n, в котором хранится вектор b;
- `x` – массив типа `double` размера n, в котором должно быть сохранено найденное решение;
- `indi` – массив типа `int` размера n, который используется для перестановки уравнений местами; этот массив инициализируется внутри функции `gauss`.
- `n` – количество уравнений (переменных).

Варианты запуска программы

Если отсутствуют аргументы командной строки, то печатается информационное сообщение о способах использования программы.

```
$ ./prog
Usage: ./prog file
Usage: ./prog c d n
```

Единственный аргумент командной строки интерпретируется как имя файла в котором содержится решаемая система линейных уравнений.

```
$ ./prog input.txt
```

Система линейных уравнений должна быть задана в файле в следующем формате.

```
a11 a12 ... a1n b1
a21 a22 ... a2n b2
...
an1 an2 ... ann bn
```

Если заданы три аргумента командной строки, то система линейных уравнений формируется на основе симметричной якобиевой матрицы. Первый аргумент задает параметр c, второй задает параметр d, а третий задает размер матрицы.

```
$ ./prog 1 -2 100
```

Приведенный пример соответствует матрице $A_{100}^{(3)}$.

Если входные параметры заданы некорректно или система уравнений не может быть решена, то в этом случае программа завершается с кодом `-1`, иначе с кодом `0`.

В стандартный поток вывода должны быть напечатаны:

- время, затраченное на решение системы линейных уравнений;
- $\| Ax^* - b \|$;
- $\| x^* - x \|$, если входные данные были сформированы на основе симметричной якобиевой матрицы.