Матрицы. Определитель

Матрицы

Матрица — математический объект, записываемый в виде прямоугольной таблицы чисел, которая представляет собой совокупность строк и столбцов, на пересечении которых находятся её элементы. Количество строк и столбцов матрицы задают размер матрицы. Если у матрицы количество строк m совпадает с количеством столбцов n, то такая матрица называется квадратной, а число m = n называется размером квадратной матрицы или её порядком.

Задание матрицы в MATLAB:

Основной способ: с помощью оператора квадратные скобки: []

Примеры:

```
>> A = [1 2; 3 4]
A =
     1     2
     3     4

>> B = [1:3;4:6;7:9]
B =
     1     2     3
     4     5     6
     7     8     9

>> C = [B(1:2,:); A(1:3)]
C =
     1     2     3
     4     5     6
     1     3     2
```

Особые матрицы:

Матрица из единиц: `ones(…)`

Матрица из нулей: `zeros(…)`

Единичная матрица: `eye(…)`

Матрица из одной строки называется вектор строкой, из одного столбца – вектор столбцом.

Операции над матрицами

Умножение матрицы на число

Умножение матрицы A на число λ заключается в построении матрицы B, элементы которой получены путём умножения каждого элемента матрицы A на это число, то есть каждый элемент матрицы B равен

$$b_{ij} = \lambda a_{ij}$$

Сложение матриц

Сложение матриц $A + B$ есть операция нахождения матрицы C, все элементы которой равны попарной сумме всех соответствующих элементов матриц A и B, то есть каждый элемент матрицы C равен

$$c_{ij} = a_{ij} + b_{ij}$$

Умножение матриц

Умножение матриц – есть операция вычисления матрицы C, элементы которой равны сумме произведений элементов в соответствующей строке первого множителя и столбце второго.

Пусть даны две прямоугольные матрицы размера mxn и nxq соответственно:

$$A = \begin{pmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{pmatrix}, \qquad B = \begin{pmatrix} b_{11} & b_{12} & \dots & b_{1q} \\ b_{21} & b_{22} & \dots & b_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ b_{n1} & b_{n2} & \dots & b_{nq} \end{pmatrix}$$

Тогда матрица C размерностью mxq называется их произведением:

$$\mathrm{C} = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1q} \\ c_{21} & c_{22} & \dots & c_{2q} \\ \vdots & \vdots & \ddots & \vdots \\ c_{m1} & c_{m2} & \dots & c_{mq} \end{pmatrix}$$

Где:

$$c_{ij} = \sum_{k=1}^{n} a_{ik} b_{kj} \quad i = 1, \dots, m; \quad j = 1,2, \dots, q$$

Количество столбцов в матрице A должно совпадать с количеством строк в матрице B. Если матрица A имеет размерность $m \times n$, B —$n \times p$, то размерность их произведения $AB = C$ есть $m \times p$.

Операция умножения двух матриц выполнима только в том случае, если число столбцов в первом сомножителе равно числу строк во втором; в этом случае говорят, что форма матриц согласована. В частности, умножение всегда выполнимо, если оба сомножителя — квадратные матрицы одного и того же порядка.

Следует заметить, что из существования произведения AB вовсе не следует существование произведения BA.

Комплексное сопряжение

Если элементами матрицы $A = (a_{ij})$ являются комплексные числа, то комплексно сопряжённая матрица равна $\bar{A} = (\bar{a}_{ij})$. Здесь — число $\bar{a}$, комплексно сопряжённое к $\bar{a}$.

Транспонирование и эрмитово сопряжение

Транспонирование: $A^T = (a_{ji})$

Эрмитово сопряжение – комплексное сопряжение и транспонирование: $A^* = \mathrm{A}^\dagger = \bar{A}^T$

В MATLAB :

```
B = 2*A   % умножение на число
C = A+B   % сложение матриц
C = A*B   % умножение матриц
C = A.*B % поэлементное умножение матриц
```

```
B = A^2  % возведение матрицы A в степень, то же что и B=A*A
B = A.^2 % возведение каждого элемента матрицы A в степень
B = A'   % эрмитово сопряжение
B = A.'  % транспонирование
```

Свойства операций

1) Сложение и вычитание допускается только для матриц одинакового размера.
2) Существует нулевая матрица Θ такая, что её прибавление к другой матрице A не изменяет A, то есть $A + \Theta = A$

   Все элементы нулевой матрицы равны нулю.
3) Возводить в степень можно только квадратные матрицы.
4) Ассоциативность сложения: $A + (B + C) = (A + B) + C$.
5) Коммутативность сложения: $A + B = B + A$.
6) Ассоциативность умножения: $A(BC) = (AB)C$.
7) Вообще говоря, умножение матриц некоммутативно: $AB \neq BA$
8) Дистрибутивность умножения относительно сложения:

   $A(B + C) = AB + AC$;

   $(B + C)A = BA + CA$.
9) Свойства операции транспонирования матриц:

   $(A^T)^T = A$

   $(AB)^T = B^T A^T$

   $(A^{-1})^T = (A^T)^{-1}$, если обратная матрица A^{-1} существует.

   $(A + B)^T = A^T + B^T$

   $detA = detA^T$
10) Для квадратных матриц существует единичная матрица E (аналог единицы для операции умножения чисел) такая, что умножение любой матрицы на неё не влияет на результат, а именно:

   $EA = AE = A$

Элементарные преобразования матриц

Элементарными преобразованиями строк матрицы называются следующие преобразования:

1) Умножение строки на число отличное от нуля,
2) Прибавление одной строки, умноженной на число, к другой строке,
3) Перестановка местами двух строк.

Элементарные преобразование столбцов матрицы определяются аналогично. При элементарных преобразованиях ранг матрицы не меняется.

Определитель

Определитель (или детерминант) — одно из основных понятий линейной алгебры. Определитель матрицы является многочленом от элементов квадратной матрицы (то есть такой, у которой количество строк и столбцов равны).

Определитель матрицы А обозначается как: $det(A)$, $|\mathrm{A}|$ или $\Delta(A)$.

Определение через перестановки

Для матрицы $n \times n$ справедлива формула:

$$\Delta = \sum_{\alpha_1,\alpha_2,\dots,\alpha_n} (-1)^{N(\alpha_1,\alpha_2,\dots,\alpha_n)} \cdot a_{\alpha_1 1} \cdot \dots \cdot a_{\alpha_n n}$$

где $\alpha_1, \alpha_2, \dots, \alpha_n$ — перестановка чисел от 1 до n, $N(\alpha_1, \alpha_2, \dots, \alpha_n)$ — число инверсий в перестановке, суммирование идёт по всем возможным перестановкам порядка n. Таким образом, в определитель войдёт $n!$ слагаемых, которые также называют «членами определителя».

Определение через разложение по первой строке

Для матрицы первого порядка детерминантом является сам единственный элемент этой матрицы:

$$\Delta = |a_{11}| = a_{11}$$

Для матрицы детерминант определяется как

$$\Delta = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

Для матрицы определитель задаётся рекурсивно:

$$\Delta = \sum_{j=1}^{n} (-1)^{1+j} a_{1j} \bar{M}_j^1,$$

где $\bar{M}_j^1$ – дополнительный минор к элементу a_{1j} (дополнительный минор $M_{j_1 \dots j_k}^{i_1 \dots i_k}$ квадратной матрицы A порядка $n (k \leq n)$ — определитель матрицы, полученной из исходной вычеркиванием $i_1 \dots i_k$ строк и $j_1 \dots j_k$ столбцов). Эта формула называется разложением по строке.

Легко доказать, что при транспонировании определитель матрицы не изменяется (иными словами, аналогичное разложение по первому столбцу также справедливо, то есть даёт такой же результат, как и разложение по первой строке):

$$\Delta = \sum_{i=1}^{n} (-1)^{1+j} a_{i1} \bar{M}_1^i,$$

Обобщением вышеуказанных формул является разложение детерминанта по Лапласу, дающее возможность вычислять определитель по любым k строкам (столбцам):

$$\Delta = \sum_{1 \leq j_1 < \cdots < j_k \leq n} (-1)^{i_1 + \cdots + i_k + j_1 + \cdots + j_k} M_{j_1 \dots j_k}^{i_1 \dots i_k} \overline{M}_{j_1 \dots j_k}^{i_1 \dots i_k}$$

Свойства определителя

1) При добавлении к любой строке (столбцу) линейной комбинации других строк (столбцов) определитель не изменится.
2) Если две строки (столбца) матрицы совпадают, то её определитель равен нулю.
3) Если две (или несколько) строки (столбца) матрицы линейно зависимы, то её определитель равен нулю.
4) Если переставить две строки (столбца) матрицы, то её определитель умножается на (-1).
5) Общий множитель элементов какого-либо ряда определителя можно вынести за знак определителя.
6) Если хотя бы одна строка (столбец) матрицы нулевая, то определитель равен нулю.
7) Сумма произведений всех элементов любой строки на их алгебраические дополнения равна определителю.
8) Сумма произведений всех элементов любого ряда на алгебраические дополнения соответствующих элементов параллельного ряда равна нулю.
9) Определитель произведения квадратных матриц одинакового порядка равен произведению их определителей

Задание:

1) Проверить на примерах 10 свойств операций над матрицами
2) Написать функцию для вычисления определителя матрицы **до 3-го порядка** через подстановки
3) Проверить на примерах 9 свойств определителя
4) Реализовать рекурсивное вычисление факториала
5) Реализовать рекурсивное вычисление определителя матрицы произвольного порядка через разложение по строке (столбцу)

Все задания должны быть выполнены в m-файлах.

Указание к заданию 3 и 4

В программировании рекурсия — вызов функции (процедуры) из неё же самой, непосредственно (простая рекурсия) или через другие функции (сложная или косвенная рекурсия), например, функция A вызывает функцию B, а функция B — функцию A. Количество вложенных вызовов функции или процедуры называется глубиной рекурсии.

Реализация рекурсивных вызовов функций в практически применяемых языках и средах программирования, как правило, опирается на механизм стека вызовов — адрес возврата и локальные переменные функции записываются в стек, благодаря чему каждый следующий рекурсивный вызов этой функции пользуется своим набором локальных переменных и за счёт этого работает корректно. Оборотной стороной этого довольно простого по структуре механизма является то, что на каждый рекурсивный вызов требуется некоторое количество оперативной памяти компьютера, и при чрезмерно большой глубине рекурсии может наступить переполнение стека вызовов. Вследствие этого, обычно рекомендуется избегать рекурсивных программ, которые приводят (или в некоторых условиях могут приводить) к слишком большой глубине рекурсии.

Впрочем, имеется специальный тип рекурсии, называемый «хвостовой рекурсией». Интерпретаторы и компиляторы функциональных языков программирования, поддерживающие оптимизацию кода (исходного и/или исполняемого), автоматически преобразуют хвостовую рекурсию к итерации (циклу), благодаря чему обеспечивается выполнение алгоритмов с хвостовой рекурсией в ограниченном объёме памяти. Такие рекурсивные вычисления, даже если они формально бесконечны (например, когда с помощью рекурсии организуется работа командного интерпретатора, принимающего команды пользователя), никогда не приводят к исчерпанию памяти.

Любую рекурсивную функцию можно заменить циклом и стеком.

Примеры рекурсивных функций:

1) Напечатать n раз «привет»

```
function print_hello(n)
if n~=0
   disp 'привет'
   print_hello(n-1)
end
```

2) Напечатать n-ое число Фибоначчи

```
function d = fib(n)
if n<=2 && n>0
    d = n-1;
elseif n>1
    d = fib(n-1) + fib(n-2);
end
```

Примечание: данный метод вычисления чисел Фибоначчи имеет экспоненциальную сложность, т.к. для вычисления fib(n) нужно вычислить функцию fib(…) 2 раза - fib(n-1) и fib(n-2), для вычисления которых потребуется вычислить fib уже 4 раза и т.д.
Пример приведен здесь лишь в качестве иллюстрации рекурсивной функции и не может использоваться на практике.
Итеративное вычисление чисел Фибоначчи имеет линейную сложность.