# Тема №7 «Матричные вычисления»

## Лекция 2 «Операции с матрицами»

### Векторизация массивов

Векторная алгебра Mathcad включает несколько необычный оператор, который называется оператором векторизации (**vectorize operator**). Этот оператор предназначен, как правило, для работы с массивами. Он позволяет провести однотипную операцию над всеми элементами массива (т. е. матрицы или вектора), упрощая тем самым программирование циклов. Например, иногда требуется умножить каждый элемент одного вектора на соответствующий элемент другого вектора. Непосредственно такой операции в Mathcad нет, но ее легко осуществить с помощью векторизации (листинг 1 9.16).

**Листинг 1**. Использование векторизации для перемножения элементов вектора:

$$v := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

$$v \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} = 14$$

$$\overrightarrow{\left[ v \cdot \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \right]} = \begin{pmatrix} 1 \\ 4 \\ 9 \end{pmatrix}$$

Для этого:

- Введите векторное выражение, как показано во второй строчке листинга (обратите внимание, что в таком виде символ умножения обозначает оператор скалярного произведения векторов).
- Переместите курсор таким образом, чтобы линии ввода выделяли все выражение, которое требуется подвергнуть векторизации (рис. 1.1).
- Введите оператор векторизации, нажав кнопку **Vectorize** (Векторизация) на панели **Matrix** (Матрица) (рис.1.1), или сочетанием клавиш **CTRL + -**.
- Введите =, чтобы получить результат.

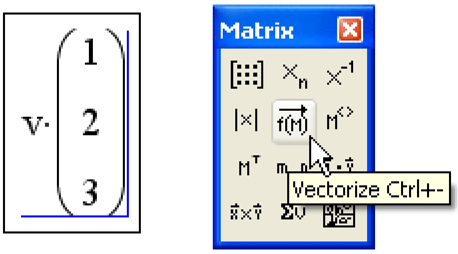


**Рис. 1.1**. Оператор векторизации

Оператор векторизации можно использовать только с векторами и матрицами одинакового размера.

## Символьные операции с матрицами

Все матричные и векторные операторы, о которых шла речь выше, допустимо использовать в символьных вычислениях. Мощь символьных операций заключается в возможности проводить их не только над конкретными числами, но и над переменными. Несколько примеров приведены в листинге 2.

**Листинг 2** . Примеры символьных операций над векторами и матрицами:

$$\begin{pmatrix} a & b \\ c & d \\ f & g \end{pmatrix} \cdot \begin{pmatrix} o & p & q \\ r & s & t \end{pmatrix} \rightarrow \begin{pmatrix} a\cdot o + b\cdot r & a\cdot p + b\cdot s & a\cdot q + b\cdot t \\ c\cdot o + d\cdot r & c\cdot p + d\cdot s & c\cdot q + d\cdot t \\ f\cdot o + g\cdot r & f\cdot p + g\cdot s & f\cdot q + g\cdot t \end{pmatrix}$$

$$\left| \begin{pmatrix} a & 0 & 0 \\ 0 & b & 1 \\ 0 & 1 & c \end{pmatrix} \right| \rightarrow a\cdot(b\cdot c - 1)$$

$$\begin{pmatrix} a & b \\ c & d \end{pmatrix}^{-1} \rightarrow \frac{1}{a\cdot d - b\cdot c} \cdot \begin{pmatrix} d & -b \\ -c & a \end{pmatrix}$$

$$\begin{pmatrix} a \\ b \\ c \end{pmatrix} \times \begin{pmatrix} r \\ s \\ t \end{pmatrix} \rightarrow \begin{pmatrix} b\cdot t - c\cdot s \\ c\cdot r - a\cdot t \\ a\cdot s - b\cdot r \end{pmatrix}$$

$$\sum \begin{pmatrix} a \\ b \\ c \end{pmatrix} \rightarrow a + b + c$$

Смело используйте символьный процессор в качестве мощного математического справочника. Например, когда Вы хотите вспомнить какоелибо определение из области линейной алгебры (так, правила перемножения и обращения матриц показаны в первых строках листинга 2).

## Матричные функции. Функции создания матриц.

Перечислим основные встроенные функции, предназначенные для облегчения работы с векторами и матрицами. Они нужны для создания матриц, слияния и выделения части матриц, получения основных свойств матриц и т.п.

Самым наглядным способом создания матрицы или вектора является применение первой кнопки панели инструментов Matrix (Матрицы). Однако в большинстве случаев, в частности при программировании сложных проектов, удобнее бывает создавать массивы с помощью встроенных функций.

### Определение элементов матрицы через функцию

- **matrix(M,N,f)** – создание матрицы размера M x N, каждый *i,j* элемент которой есть f (i, j) (листинг 3 );
  - *м* – количество строк;
  - *N* – количество столбцов;
  - *f (i, j)* – функция.

**Листинг 3**. Создание матрицы:

$$f(i,j) := i + 0.5\cdot j$$

$$A := \mathrm{matrix}(2,3,f)$$

$$A = \begin{pmatrix} 0 & 0.5 & 1 \\ 1 & 1.5 & 2 \end{pmatrix}$$

Для создания матриц имеются еще две специфические функции, применяемые, в основном, для быстрого и эффектного представления каких-либо зависимостей в виде трехмерных графиков (типа поверхности или пространственной кривой). Все их аргументы, кроме первого (функции), необязательны.

## Слияние и разбиение матриц

Из матрицы или вектора можно выделить либо подматрицу, либо вектор-столбец, либо отдельный элемент. И обратно, можно "склеить" несколько матриц в одну.

### Выделение части матрицы

Часть матрицы выделяется одним из следующих способов:

- для выделения одного элемента предназначен оператор нижнего индекса. Оператор вводится нажатием кнопки **Subscript** (Нижний индекс) со значком $x_n$ на панели **Matrix** (Матрица), либо нажатием клавиши **[** (листинг 4 , вторая строка сверху);

- для выделения из матрицы столбца примените оператор выделения столбца нажатием кнопки **Matrix Column** с изображением угловых скобок <> на панели **Matrix**, либо сочетанием клавиш **CTRL + 6** (листинг 4 ). Этот оператор называют еще, по аналогии с предыдущим, оператором верхнего индекса;
- чтобы выделить из матрицы строку, примените тот же оператор <> к транспонированной матрице (листинг 4 , снизу);
- для выделения подматрицы используйте встроенную функцию **submatrix**(*A/ir/jr,ic, jc*), возвращающую часть матрицы *A*, находящуюся между строками *ir, jr* и столбцами *ic, jc* включительно (листинг 5 ).

Выделить из матрицы один столбец или строку можно и с помощью функции **submatrix**.

**Листинг 4**. Доступ к отдельным элементам, столбцам и строкам матрицы:

$$A := \begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}$$

$$A_{0,1} = 2 \qquad A_{1,1} = 5$$

$$A^{\langle 2 \rangle} = \begin{pmatrix} 3 \\ 6 \end{pmatrix} \qquad A^{\langle 0 \rangle} = \begin{pmatrix} 1 \\ 4 \end{pmatrix} \qquad A^{\langle 1 \rangle} = \begin{pmatrix} 2 \\ 5 \end{pmatrix}$$

$$\left(A^T\right)^{\langle 0 \rangle} = \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix} \qquad \left(A^T\right)^{\langle 0 \rangle^T} = \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$$

**Листинг 5** . Выделение подматрицы:

$$\mathrm{submatrix}\left[\begin{pmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{pmatrix}, 0, 1, 0, 1\right] = \begin{pmatrix} 1 & 2 \\ 4 & 5 \end{pmatrix}$$

$$\mathrm{submatrix}\left[\begin{pmatrix} 1 & 0 & 0 \\ -1 & -3 & -4 \end{pmatrix}, 0, 0, 0, 1\right] = \begin{pmatrix} 1 & 0 \end{pmatrix}$$

Те же операции применимы к матрицам-векторам и матрицам-строкам. Следует помнить только, что размер их составляет *N* x *1* и *1* x *N*, соответственно (листинг 6 ).

**Листинг 6** . Выделение частей из векторов и строк:

$$(1\ \ 2\ \ 3)^{\langle 0\rangle} = (1) \qquad \left(\begin{pmatrix}1\\2\\3\end{pmatrix}^{T}\right)^{\langle 1\rangle} = (2)$$

$$\text{submatrix}\left[\begin{pmatrix}1\\2\\3\end{pmatrix},0,1,0,0\right] = \begin{pmatrix}1\\2\end{pmatrix}$$

**Слияние матриц**

Для того чтобы составить из двух или более матриц одну, в Mathcad предусмотрены две матричные функции (листинг 7):

- augment (*A, в, c,…*) – матрица, сформированная слиянием матриц-аргументов слева направо;
- stack (*A, в, c,…*) – матрица, сформированная слиянием матриц-аргументов сверху вниз;
  - *A,B,C,…* – векторы или матрицы соответствующего размера.

**Листинг 7**. Примеры слияния матриц:

$$A := \begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\end{pmatrix} \qquad B := \begin{pmatrix}0 & 0 & 0\\0 & 0 & 0\end{pmatrix}$$

$$\text{stack}(A,B) = \begin{pmatrix}1 & 2 & 3\\4 & 5 & 6\\0 & 0 & 0\\0 & 0 & 0\end{pmatrix}$$

$$\text{augment}(A,B) = \begin{pmatrix}1 & 2 & 3 & 0 & 0 & 0\\4 & 5 & 6 & 0 & 0 & 0\end{pmatrix}$$

$$\text{augment}\left[\begin{pmatrix}1\\1\end{pmatrix},A,B,\begin{pmatrix}3\\3\end{pmatrix}\right] = \begin{pmatrix}1 & 1 & 2 & 3 & 0 & 0 & 0 & 3\\1 & 4 & 5 & 6 & 0 & 0 & 0 & 3\end{pmatrix}$$

## Вывод размера матриц

Для получения сведений о характеристиках матриц или векторов предусмотрены следующие встроенные функции (листинг 8):

- rows (*A*) – число строк;
- cols (*A*) – число столбцов;
- length (*v)* – число элементов вектора;
- last (*v*) – индекс последнего элемента вектора;
  - *A* – матрица или вектор;
  - *v* – вектор.

Число элементов вектора и индекс его последнего элемента совпадают, если индексы нумеруются с 1, т. е. системная константа ORIGIN равна 1.

**Листинг 8**. Размер матриц и векторов:

$$w := \begin{pmatrix} 1 & 2 & 3 \end{pmatrix}$$

$$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} \qquad v := \begin{pmatrix} 1 \\ 2 \\ 3 \end{pmatrix}$$

$$\mathrm{rows}(A) = 3 \qquad \mathrm{rows}(v) = 3 \qquad \mathrm{rows}(w) = 1$$

$$\mathrm{cols}(A) = 2 \qquad \mathrm{cols}(v) = 1 \qquad \mathrm{cols}(w) = 3$$

$$\mathrm{length}(v) = 3$$

$$\mathrm{last}(v) = 2$$

## Сортировка матриц

Часто бывает нужно переставить элементы матрицы или вектора, расположив их в определенной строке или столбце в порядке возрастания или убывания. Для этого имеются несколько встроенных функций, которые позволяют гибко управлять сортировкой матриц:

- sort(v) – сортировка элементов вектора в порядке возрастания (листинг 9 );
- csort(*A*,*i*) – сортировка строк матрицы выстраиванием элементов 1-го столбца в порядке возрастания (листинг 10);
- rsort(*A*,*i*) – сортировка столбцов матрицы выстраиванием элементов *i*-й строки в порядке возрастания (листинг 11);
- reverse (*v)* – перестановка элементов вектора в обратном порядке (листинг 9);
  - *v* – вектор;

- *A* – матрица;
- *i* – индекс строки или столбца.

Если элементы матриц или векторов комплексные, то сортировка ведется по действительной части, а мнимая часть игнорируется.

**Листинг 9**. Сортировка векторов:

$$v := \begin{pmatrix} 3 \\ 1 \\ 4 \\ 2 \end{pmatrix} \qquad \mathrm{sort}(v) = \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \end{pmatrix} \qquad \mathrm{reverse}(v) = \begin{pmatrix} 2 \\ 4 \\ 1 \\ 3 \end{pmatrix}$$

**Листинг 10**. Сортировка матриц по столбцу:

$$A := \begin{pmatrix} 1 & 9 \\ 3 & 0 \\ 2 & 8 \end{pmatrix}$$

$$\mathrm{csort}(A,1) = \begin{pmatrix} 3 & 0 \\ 2 & 8 \\ 1 & 9 \end{pmatrix} \qquad \mathrm{csort}(A,0) = \begin{pmatrix} 1 & 9 \\ 2 & 8 \\ 3 & 0 \end{pmatrix}$$

**Листинг 11**. Сортировка матриц по строке (матрица A из листинга 9.28):

$$\mathrm{rsort}(A,1) = \begin{pmatrix} 9 & 1 \\ 0 & 3 \\ 8 & 2 \end{pmatrix} \qquad \mathrm{rsort}(A,2) = \begin{pmatrix} 1 & 9 \\ 3 & 0 \\ 2 & 8 \end{pmatrix}$$

## Норма квадратной матрицы

В линейной алгебре используются различные матричные нормы (**norm**), которые ставят в соответствие матрице некоторую скалярную числовую характеристику. Норма матрицы отражает порядок величины матричных элементов. В разных специфических задачах линейной алгебры применяются различные виды норм. Mathcad имеет четыре встроенные функции для расчета разных норм квадратных матриц:

- norm1 (A) – норма в пространстве L1;
- norm2 (A) – норма в пространстве L2;
- norme (A) – евклидова норма (**euclidean norm**);
- normi (A) – max-норма, или норма (**infinity norm**);
  - A – квадратная матрица.

Примеры расчета различных норм двух матриц A и в с различающимися на два порядка элементам! приведены в листинге 12. В последней строке этого листинга пояснен) определение евклидовой нормы, которое похоже на определение длины вектора.

**Листинг 12**. Нормы матриц:

$$A := \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} \qquad B := \begin{pmatrix} 100 & 200 \\ 300 & 400 \end{pmatrix}$$

$$\mathrm{norm1}(A) = 6 \qquad \mathrm{norm1}(B) = 600$$

$$\mathrm{norm2}(A) = 5.465 \qquad \mathrm{norm2}(B) = 546.499$$

$$\mathrm{normi}(A) = 7 \qquad \mathrm{normi}(B) = 700$$

$$\mathrm{norme}(A) = 5.477 \qquad \mathrm{norme}(B) = 547.723$$

$$\sqrt{1^2 + 2^2 + 3^2 + 4^2} = 5.477$$

В большинстве задач неважно, какую норму использовать. Как видно, в обычных случаях разные нормы дают примерно одинаковые значения, хорошо отражая порядок величины матричных элементов. Определение остальных норм заинтересованный метатель отыщет в справочниках по линейной алгебре или в справочной системе Mathcad (раздел Mathcad Resources).

## Ранг матрицы

Рангом (**rank**) матрицы называют наибольшее натуральное число *k*, для которого существует не равный нулю определитель *k*-го порядка подматрицы, составленной из любого пересечения *k* столбцов и *k* строк матрицы.

Для вычисления ранга в Mathcad предназначена функция **rank**.

- rank (A) – ранг матрицы;
  - A – матрица.

**Листинг 12**. Ранг матрицы:

$$\mathrm{rank}\begin{pmatrix} 1 & 2 \\ 3 & 6 \end{pmatrix} = 1 \qquad \mathrm{rank}\begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix} = 2$$

$$\mathrm{rank}\begin{pmatrix} 1 & 2 \\ 3 & 6 \\ 1 & 2 \end{pmatrix} = 1 \qquad \mathrm{rank}\begin{pmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{pmatrix} = 2$$

## Системы линейных алгебраических уравнений

Центральным вопросом вычислительной линейной алгебры является решение систем линейных алгебраических уравнений (**СЛАУ**), т. е. систем уравнений вида:

$$a_{i1}x_1+a_{i2}x_2+\ldots+a_{in}x_n=b_i \qquad (1)$$

В матричной форме СЛАУ записывается в эквивалентном виде:

$$Ax = b, \qquad (2)$$

где $A$ – матрица коэффициентов СЛАУ размерности $N$x$N$, $x$ – вектор неизвестных, $b$– вектор правых частей уравнений.

К системам линейных уравнений сводится множество, если не сказать большинство, задач вычислительной математики.

СЛАУ имеет единственное решение, если матрица $A$ является невырожденной, или, по-другому, несингулярной, т. е. ее определитель не равен нулю. С вычислительной точки зрения, решение СЛАУ не представляет трудностей, если матрица $A$ не очень велика. С большой матрицей проблем также не возникнет, если она не очень плохо обусловлена. В Mathcad СЛАУ можно решить как в более наглядной форме (1), так и в более удобной для записи форме (2). Для первого способа следует использовать вычислительный блок **Given/Find** , а для второго – встроенную функцию **lsoive**.

- lsoive ($A$, $b$) – решение системы линейных уравнений;
- $A$ – матрица коэффициентов системы;
- $b$ – вектор правых частей.

Применение функции **lsoive** показано в листинге 13. При этом матрица $A$ может быть определена любым из способов, необязательно явно, как во всех примерах этого раздела. Встроенную функцию **lsoive** допускается применять и при символьном решении СЛАУ (листинг 14).

Соответствующая матрице $A$ и вектору $b$ система уравнений выписана явно в листинге 15.

**Листинг 13**. Решение СЛАУ:

$$A := \begin{pmatrix} 1 & 5 & 2 \\ 0.7 & 12 & 5 \\ 3 & 0 & 4 \end{pmatrix} \qquad b := \begin{pmatrix} 1 \\ 2.9 \\ 3.1 \end{pmatrix}$$

$$\mathrm{lsolve}(A,b) = \begin{pmatrix} -0.186 \\ -0.129 \\ 0.915 \end{pmatrix}$$

**Листинг 14**. Символьное решение СЛАУ (продолжение листинга 13):

$$\mathrm{lsolve}(A,b) = \begin{pmatrix} -.186486486486486 \\ -.128648648648649 \\ .914864864864865 \end{pmatrix}$$

В некоторых случаях, для большей наглядности представления СЛАУ, его можно решить точно так же, как систему нелинейных уравнений. Пример численного решения СЛАУ из предыдущих листингов показан в листинге 15. Не забывайте, что при численном решении всем неизвестным требуется присвоить начальные значения (это сделано в первой строке листинга 15). Они могут быть произвольными, т. к. решение СЛАУ с невырожденной матрицей единственно.

При решении СЛАУ с помощью функции **Find** **Mathcad** автоматически выбирает линейный численный алгоритм, в чем можно убедиться, вызывая на имени **Find** контекстное меню.

**Листинг 15**. Решение СЛАУ с помощью вычислительного блока:

$$x := 0 \qquad y := 0 \qquad z := 0$$

Given

$$1x + 5y + 2z = 1$$

$$0.7x + 12y + 5z = 2.9$$

$$3x + 0y + 4z = 3.1$$

$$\mathrm{find}(x,y,z) = \begin{pmatrix} -0.186 \\ -0.129 \\ 0.915 \end{pmatrix}$$