

Arithmetic Operators + - * / \ ^ '

Matrix and array arithmetic

Syntax

A+B
A-B
A*B
A.*B
A/B
A./B
A\B
A.\B
A^B
A.^B
A'
A.'

Description

MATLAB has two different types of arithmetic operations. Matrix arithmetic operations are defined by the rules of linear algebra. Array arithmetic operations are carried out element by element, and can be used with multidimensional arrays. The period character (.) distinguishes the array operations from the matrix operations. However, since the matrix and array operations are the same for addition and subtraction, the character pairs `.+` and `.-` are not used.

- + Addition or unary plus. `A+B` adds `A` and `B`. `A` and `B` must have the same size, unless one is a scalar. A scalar can be added to a matrix of any size.
- Subtraction or unary minus. `A-B` subtracts `B` from `A`. `A` and `B` must have the same size, unless one is a scalar. A scalar can be subtracted from a matrix of any size.
- * Matrix multiplication. `C = A*B` is the linear algebraic product of the matrices `A` and `B`. More precisely,



For nonscalar `A` and `B`, the number of columns of `A` must equal the number of rows of `B`. A scalar can multiply a matrix of any size.

- .* Array multiplication. $A.*B$ is the element-by-element product of the arrays A and B . A and B must have the same size, unless one of them is a scalar.
- / Slash or matrix right division. B/A is roughly the same as $B*inv(A)$. More precisely, $B/A = (A' \setminus B')$. See the reference page for [mrdivide](#) for more information.
- ./ Array right division. $A./B$ is the matrix with elements $A(i,j)/B(i,j)$. A and B must have the same size, unless one of them is a scalar.
- \ Backslash or matrix left division. If A is a square matrix, $A \setminus B$ is roughly the same as $inv(A)*B$, except it is computed in a different way. If A is an n -by- n matrix and B is a column vector with n components, or a matrix with several such columns, then $x = A \setminus B$ is the solution to the equation $AX = B$ computed by Gaussian elimination. A warning message is displayed if A is badly scaled or nearly singular. See the reference page for [mldivide](#) for more information.

If A is an m -by- n matrix with $m \sim n$ and B is a column vector with m components, or a matrix with several such columns, then $x = A \setminus B$ is the solution in the least squares sense to the under- or overdetermined system of equations $AX = B$. The effective rank, k , of A is determined from the QR decomposition with pivoting (see [Algorithm](#) for details). A solution x is computed that has at most k nonzero components per column. If $k < n$, this is usually not the same solution as $pinv(A)*B$, which is the least squares solution with the smallest norm \boxtimes .
- .\ Array left division. $A.\setminus B$ is the matrix with elements $B(i,j)/A(i,j)$. A and B must have the same size, unless one of them is a scalar.
- ^ Matrix power. x^p is x to the power p , if p is a scalar. If p is an integer, the power is computed by repeated squaring. If the integer is negative, x is inverted first. For other values of p , the calculation involves eigenvalues and eigenvectors, such that if $[V,D] = eig(x)$, then $x^p = V*D.^p/V$.

If x is a scalar and P is a matrix, x^P is x raised to the matrix power P using eigenvalues and eigenvectors. x^P , where x and P are both matrices, is an error.
- .^ Array power. $A.^B$ is the matrix with elements $A(i,j)$ to the $B(i,j)$ power. A and B must have the same size, unless one of them is a scalar.

- ' Matrix transpose. A' is the linear algebraic transpose of A . For complex matrices, this is the complex conjugate transpose.
- .' Array transpose. $A.'$ is the array transpose of A . For complex matrices, this does not involve conjugation.

Nondouble Data Type Support

This section describes the arithmetic operators' support for data types other than `double`.

Data Type `single`

You can apply any of the arithmetic operators to arrays of type `single` and MATLAB returns an answer of type `single`. You can also combine an array of type `double` with an array of type `single`, and the result has type `single`.

Integer Data Types

You can apply most of the arithmetic operators to real arrays of the following integer data types:

- `int8` and `uint8`
- `int16` and `uint16`
- `int32` and `uint32`

All operands must have the same integer data type and MATLAB returns an answer of that type.

Note The arithmetic operators do not support operations on the data types `int64` or `uint64`. Except for the unary operators `+A` and `A.'`, the arithmetic operators do not support operations on complex arrays of any integer data type.

For example,

```
x = int8(3) + int8(4);  
class(x)  
  
ans =  
  
int8
```

The following table lists the binary arithmetic operators that you can apply to arrays of the same integer data type. In the table, *A* and *B* are arrays of the same integer data type and *c* is a scalar of type `double` or the same type as *A* and *B*.

Operation	Support when A and B Have Same Integer Type
$+A, -A$	Yes
$A+B, A+c, c+B$	Yes
$A-B, A-c, c-B$	Yes
$A.*B$	Yes
$A*c, c*B$	Yes
$A*B$	No
$A/c, c/B$	Yes
$A.\backslash B, A./B$	Yes
$A\backslash B, A/B$	No
$A.^B$	Yes, if <i>B</i> has nonnegative integer values.
c^k	Yes, for a scalar <i>c</i> and a nonnegative scalar integer <i>k</i> , which have the same integer data type or one of which has type <code>double</code>
$A.', A'$	Yes

Combining Integer Data Types with Type Double

For the operations that support integer data types, you can combine a scalar or array of an integer data type with a scalar, but not an array, of type `double` and the result has the same integer data type as the input of integer type. For example,

```
y = 5 + int32(7);
class(y)

ans =

int32
```

However, you cannot combine an array of an integer data type with either of the following:

- A scalar or array of a different integer data type
- A scalar or array of type `single`

The section [Numeric Types](#), under [Data Types](#) in the MATLAB Programming documentation, provides more information about operations on nondouble data types.

Remarks

The arithmetic operators have M-file function equivalents, as shown:

Binary addition	$A+B$	<code>plus(A,B)</code>
Unary plus	$+A$	<code>uplus(A)</code>
Binary subtraction	$A-B$	<code>minus(A,B)</code>
Unary minus	$-A$	<code>uminus(A)</code>
Matrix multiplication	$A*B$	<code>mtimes(A,B)</code>
Arraywise multiplication	$A.*B$	<code>times(A,B)</code>
Matrix right division	A/B	<code>mrdivide(A,B)</code>
Arraywise right division	$A./B$	<code>rdivide(A,B)</code>
Matrix left division	$A\B$	<code>mldivide(A,B)</code>
Arraywise left division	$A.\B$	<code>ldivide(A,B)</code>
Matrix power	A^B	<code>mpower(A,B)</code>
Arraywise power	$A.^B$	<code>power(A,B)</code>
Complex transpose	A'	<code>ctranspose(A)</code>

Matrix `A.'` `transpose(A)`
transpose

Note For some toolboxes, the arithmetic operators are overloaded, that is, they perform differently in the context of that toolbox. To see the toolboxes that overload a given operator, type `help` followed by the operator name. For example, type `help plus`. The toolboxes that overload `plus (+)` are listed. For information about using the operator in that toolbox, see the documentation for the toolbox.

Examples

Here are two vectors, and the results of various matrix and array operations on them, printed with `format rat`.

Matrix Operations		Array Operations	
<code>x</code>	1 2 3	<code>y</code>	4 5 6
<code>x'</code>	1 2 3	<code>y'</code>	4 5 6
<code>x+y</code>	5 7 9	<code>x-y</code>	-3 -3 -3
<code>x + 2</code>	3 4 5	<code>x-2</code>	-1 0 1
<code>x * y</code>	Error	<code>x.*y</code>	4 10 18
<code>x'*y</code>	32	<code>x' .* y</code>	Error

$x*y'$	4 5 6 8 10 12 12 15 18	$x.*y'$	Error
$x*2$	2 4 6	$x.*2$	2 4 6
$x\backslash y$	16/7	$x.\backslash y$	4 5/2 2
$2\backslash x$	1/2 1 3/2	$2./x$	2 1 2/3
x/y	0 0 1/6 0 0 1/3 0 0 1/2	$x./y$	1/4 2/5 1/2
$x/2$	1/2 1 3/2	$x./2$	1/2 1 3/2
x^y	Error	$x.^y$	1 32 729
x^2	Error	$x.^2$	1 4 9
2^x	Error	$2.^x$	2 4 8

$(x+iy)'$	$\begin{matrix} 1 & - & 4i & 2 \\ - & 5i & 3 & - \\ & & & 6i \end{matrix}$		
$(x+iy) \cdot'$	$\begin{matrix} 1 & + & 4i & 2 \\ + & 5i & 3 & + \\ & & & 6i \end{matrix}$		

Diagnostics

- From matrix division, if a square A is singular,

Warning: Matrix is singular to working precision.

- From elementwise division, if the divisor has zero elements,

Warning: Divide by zero.

Matrix division and elementwise division can produce NaNs or Infs where appropriate.

- If the inverse was found, but is not reliable,

Warning: Matrix is close to singular or badly scaled.
Results may be inaccurate. RCOND = xxx

- From matrix division, if a nonsquare A is rank deficient,

Warning: Rank deficient, rank = xxx tol = xxx

See Also

[mldivide](#), [mrdivide](#), [chol](#), [det](#), [inv](#), [lu](#), [orth](#), [permute](#), [ipermute](#), [qr](#), [rref](#)

References

[1] Anderson, E., Z. Bai, C. Bischof, S. Blackford, J. Demmel, J. Dongarra, J. Du Croz, A. Greenbaum, S. Hammarling, A. McKenney, and D. Sorensen, *LAPACK User's Guide* (http://www.netlib.org/lapack/lug/lapack_lug.html), Third Edition, SIAM, Philadelphia, 1999.

[2] Davis, T.A., *UMFPACK Version 4.6 User Guide*

(<http://www.cise.ufl.edu/research/sparse/umfpack>), Dept. of Computer and Information Science and Engineering, Univ. of Florida, Gainesville, FL, 2002.

[3] Davis, T. A., *CHOLMOD Version 1.0 User Guide*

(<http://www.cise.ufl.edu/research/sparse/cholmod>), Dept. of Computer and Information Science and Engineering, Univ. of Florida, Gainesville, FL, 2005.

◀ Functions — Alphabetical List Relational Operators < > <= >= == ~= ▶

© 1984–2006 The MathWorks, Inc. · [Terms of Use](#) · [Patents](#) · [Trademarks](#) · [Acknowledgments](#)