



Arithmetic Operations with Arrays & Useful Array Functions

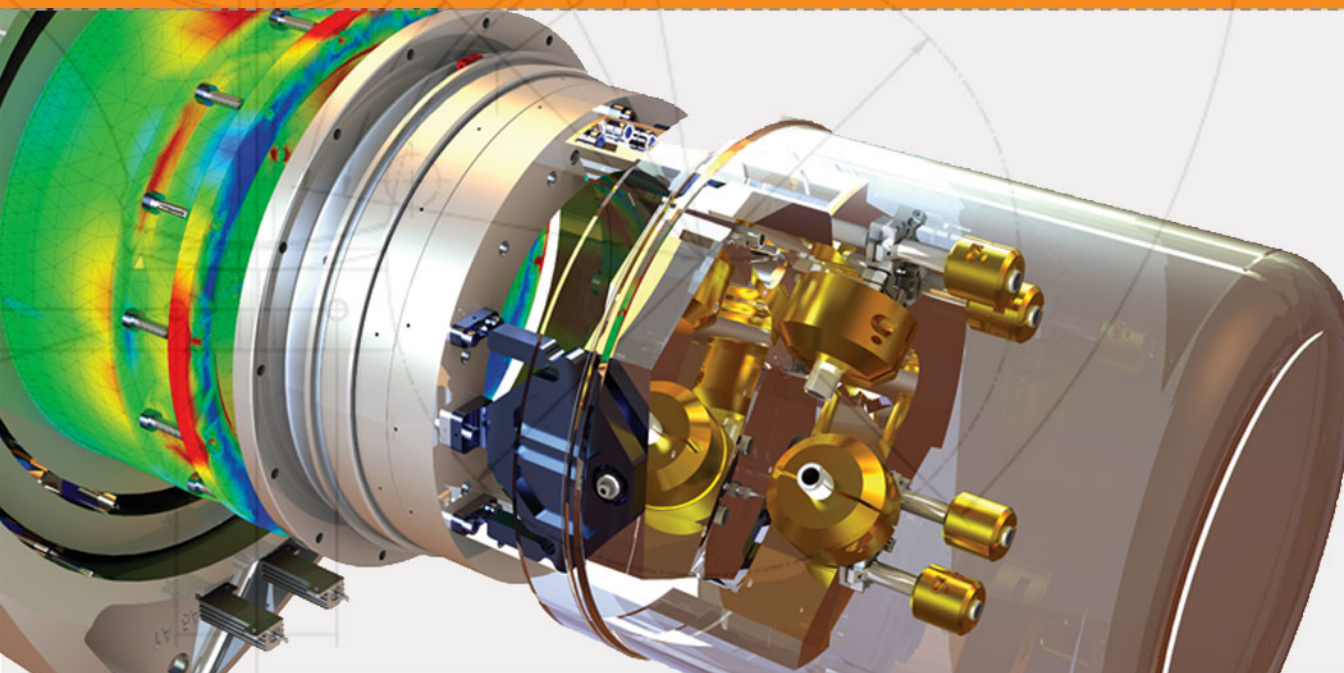


Image courtesy of National Optical Astronomy
Observatory, operated by the Association of Universities
for Research in Astronomy, under cooperative
agreement with the National Science Foundation.

Array Addition

- Two arrays can be added if and only if both arrays have exactly the same dimensions.
- Assuming the dimension requirement is satisfied, the sum of two arrays, $A + B$, is an array of the same dimensions as A and B where element (i,j) is the sum of element (i,j) of matrix A and element (i,j) of matrix B .

Example

$$\begin{pmatrix} 2 & 5 & -1 \\ 4 & 7 & 5 \end{pmatrix} + \begin{pmatrix} 1 & 6 & 8 \\ -3 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 3 & 11 & 7 \\ 1 & 10 & 9 \end{pmatrix}$$

Array Subtraction

- An array can be subtracted from another array if and only if the arrays have exactly the same dimensions.
- Assuming the dimension requirement is satisfied, the difference of two arrays, $A - B$, is an array of the same dimensions as A and B where element (i,j) is the difference of element (i,j) of matrix A and element (i,j) of matrix B .

Example

$$\begin{pmatrix} 2 & 5 & -1 \\ 4 & 7 & 5 \end{pmatrix} - \begin{pmatrix} 1 & 6 & 8 \\ -3 & 3 & 4 \end{pmatrix} = \begin{pmatrix} 1 & -1 & -9 \\ 7 & 4 & 1 \end{pmatrix}$$

Array Multiplication

- The product of an array and a scalar (constant) is simply all elements of the array multiplied by the scalar.
- Two arrays can be multiplied if and only if the number of columns in the first array is the same as the number of rows in the 2nd array (inner matrix dimensions must agree)
- Assuming the dimension requirement is satisfied, the product of two arrays, $A * B$, is an array with the same number of rows as A and the same number of columns as B.
- Element (i,j) is the sum of the products of the ith row of matrix A and the jth column of matrix B.
- Array multiplication is not commutative; that is,
 $A * B \neq B * A$ except in some special cases.
- Only 1-d and 2-d arrays can be multiplied.

Array Multiplication Example

$$\begin{pmatrix} 2 & 3 & -1 \\ 5 & 1 & 2 \end{pmatrix} * \begin{pmatrix} 3 & 5 & 4 & -2 \\ 1 & 2 & -3 & 5 \\ -1 & 1 & 0 & 2 \end{pmatrix}$$

2 x 3 times 3 x 4



Inner matrix dimensions agree so product does exist

The product will be an array of dimension 2 x 4.

Array Multiplication Example

$$\begin{pmatrix} 2 & 3 & -1 \\ 5 & 1 & 2 \end{pmatrix} * \begin{pmatrix} 3 & 5 & 4 & -2 \\ 1 & 2 & -3 & 5 \\ -1 & 1 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 10 & 15 & -1 & 9 \end{pmatrix}$$

Element (1,1) is the sum of products of row 1 of 1st matrix and column 1 of 2nd matrix: $2*3 + 3*1 + (-1)*(-1)$

Element (1,2) is the sum of products of row 1 of 1st matrix and column 2 of 2nd matrix: $2*5 + 3*2 + (-1)*1$

Element (1,3) is the sum of products of row 1 of 1st matrix and column 3 of 2nd matrix: $2*4 + 3*(-3) + (-1)*0$

Element (1,4) is the sum of products of row 1 of 1st matrix and column 4 of 2nd matrix: $2*(-2) + 3*5 + (-1)*2$

Array Multiplication Example

$$\begin{pmatrix} 2 & 3 & -1 \\ 5 & 1 & 2 \end{pmatrix} * \begin{pmatrix} 3 & 5 & 4 & -2 \\ 1 & 2 & -3 & 5 \\ -1 & 1 & 0 & 2 \end{pmatrix} = \begin{pmatrix} 10 & 15 & -1 & 9 \\ 14 & 29 & 17 & -1 \end{pmatrix}$$

Element (2,1) is the sum of products of row 2 of 1st matrix and column 1 of 2nd matrix: $5*3 + 1*1 + 2*(-1)$

Element (2,2) is the sum of products of row 2 of 1st matrix and column 2 of 2nd matrix: $5*5 + 1*2 + 2*1$

Element (2,3) is the sum of products of row 2 of 1st matrix and column 3 of 2nd matrix: $5*4 + 1*(-3) + 2*0$

Element (2,4) is the sum of products of row 2 of 1st matrix and column 4 of 2nd matrix: $5*(-2) + 1*5 + 2*2$

The Identity Matrix

- The identity matrix, I , is a square matrix (same number of rows and columns) with ones down the main diagonal and zeros everywhere else.
- The product of a matrix, A , and an identity matrix of appropriate dimension is the matrix A .

$$\begin{pmatrix} 2 & 3 & -1 \\ 5 & 1 & 2 \end{pmatrix} * \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} 2 & 3 & -1 \\ 5 & 1 & 2 \end{pmatrix} \quad \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} * \begin{pmatrix} 2 & 3 & -1 \\ 5 & 1 & 2 \end{pmatrix} = \begin{pmatrix} 2 & 3 & -1 \\ 5 & 1 & 2 \end{pmatrix}$$

Matrix Inverses

- The inverse of a matrix A is defined to be a matrix A^{-1} which satisfies:

$$A * A^{-1} = I \quad \text{and} \quad A^{-1} * A = I$$

where I is the identity matrix.

- A matrix must be square (same number of rows as columns) in order to have an inverse; however, not all square matrices have inverses.
- A square matrix has an inverse if its determinant (a scalar value) is non-zero. This is equivalent to saying that each row of the matrix is independent (not some linear combination) of the other rows in the matrix.
- Although determinants and inverses of matrices can be computed by hand, we will use MATLAB® to compute them.

Matrix Inverse Example 1

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix} \quad A^{-1} = \begin{pmatrix} 5 & -2 \\ -2 & 1 \end{pmatrix}$$

How would you verify this?

$$\begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix} * \begin{pmatrix} 5 & -2 \\ -2 & 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix} \quad \checkmark$$

Matrix Inverse Example 2

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 5 & 1 \\ 3 & 7 & 1 \end{pmatrix}$$

Does this matrix have an inverse? Explain?

No. Row 3 is the sum of Row 1 and Row 2.

The rows must be independent in order for the inverse to exist. This is not easy to verify visually. The determinant will be our numerical tool for determining whether or not the inverse exists.

Matrix Inverse Example 2

$$A = \begin{pmatrix} 1 & 2 & 0 \\ 2 & 5 & 1 \\ 3 & 7 & 1 \end{pmatrix}$$

```
>> A = [1 2 0; 2 5 1; 3 7 1]
```

```
A =
```

```
    1    2    0
    2    5    1
    3    7    1
```

```
>> det(A)
```

```
ans = 8.3267e-16
```

```
% This should be zero - Numerical Precision Problem
```

If $\det(A)$ is zero (or very small when calculated using MATLAB), the matrix does not have an inverse.

Matrix Inverse Example 3

$$A = \begin{pmatrix} 1 & 2 \\ 2 & 5 \end{pmatrix}$$

```
>> A = [1 2; 2 5];
```

```
>> det(A)
```

```
ans =      1      % Determinant is non-zero so A has inverse!
```

```
>> A_Inv = inv(A)
```

```
A_Inv =
```

```
     5     -2  
    -2      1
```

Arithmetic Operations with Arrays

OPERATION	DESCRIPTION
$A + B$	Array addition: A & B must have same dimensions unless one is a scalar.
$A - B$	Array subtraction: A & B must have same dimensions unless one is a scalar.
$A * B$	Array multiplication: inner matrix dimensions must agree unless one is a scalar.
$A .* B$	Entry by entry multiplication of A & B (i.e., element(i,j) is product of A(i,j) and B(i,j)). A & B must have same dimensions unless one is a scalar.
A^n	Array A raised to nth power. Array must be square.
$A.^n$	Each entry of A is raised to the nth power.

Arithmetic Operations with Arrays

FUNCTION	DESCRIPTION
det(A)	Computes the determinant of matrix A
inv(A)	Computes an inverse of A Caution: don't use this if det(A) is zero!
inv(A)*b	Solves the matrix equation $Ax = b$ Caution: don't use this if det(A) is zero!
A\b	Solves the matrix equation $Ax = b$ using the left divide operation. It is more efficient (faster) than $\text{inv}(A)*b$. Caution: don't use this if det(A) is zero!
A.'	Transpose of Matrix A: row i becomes column i or equivalently column i becomes row i.

To Dot or Not to Dot?

$A*B$ means we want to multiply array A by array B. This is allowable as long as the inner matrix dimensions agree.

$A.*B$ means we want to multiply each entry in A by the corresponding entry in B. This is allowable as long as A and B are both the same size.

These are two very different operations!

To Dot or Not to Dot?

Example:

$$A = \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} 4 & 5 \\ -2 & 0 \end{bmatrix}$$

$$A * B = \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix} * \begin{bmatrix} 4 & 5 \\ -2 & 0 \end{bmatrix} = \begin{bmatrix} 6 & 5 \\ 2 & 10 \end{bmatrix}$$

$$\begin{aligned} A .* B &= \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix} .* \begin{bmatrix} 4 & 5 \\ -2 & 0 \end{bmatrix} = \begin{bmatrix} 1(4) & -1(5) \\ 2(-2) & 3(0) \end{bmatrix} \\ &= \begin{bmatrix} 4 & -5 \\ -4 & 0 \end{bmatrix} \end{aligned}$$

To Dot or Not to Dot?

Example:

$$A = \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix}$$

$$A^2 = \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix} * \begin{bmatrix} 1 & -1 \\ 2 & 3 \end{bmatrix} = \begin{bmatrix} -1 & -4 \\ 8 & 7 \end{bmatrix}$$

$$A.^2 = \begin{bmatrix} (1)^2 & (-1)^2 \\ (2)^2 & (3)^2 \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 4 & 9 \end{bmatrix}$$

To Dot or Not to Dot?

Example:

$$t = [1 \quad 2 \quad 3]$$

t^2 means: $[1 \quad 2 \quad 3] * [1 \quad 2 \quad 3]$ Can't be done!
(1 x 3) (1 x 3)

$$t.^2 = [(1)^2 \quad (2)^2 \quad (3)^2] = [1 \quad 4 \quad 9]$$

To Dot

The Dot operation has proved useful when substituting a vector of x-values (entry by entry) into an equation to find corresponding y-values.

Ex. $y = t^3 - 4t^2 + 8t - 7$

```
>> t = 0:0.01:2;
```

```
>> y = t.^3 - 4*t.^2 + 8*t - 7;
```

Essentially substitutes each t-value into the equation to find corresponding y-value just like you would do by hand.

Not to Dot

The Dot operation should not be used when we want to do array multiplication.

$A*B$ as long as inner matrix dimensions agree

A^n as long as A is a square matrix

Solving system of linear equations: $Ax = b$

$$x = \text{inv}(A)*b$$

SOME USEFUL FUNCTIONS FOR ARRAYS



Min and Max

FUNCTION	DESCRIPTION
min(A)	If A is a vector, finds minimum entry. If A is a matrix, finds minimum value in each column.
max(A)	If A is a vector, finds maximum entry. If A is a matrix, finds maximum value in each column.
min(A,[],2)	If A is a vector, finds minimum entry. If A is a matrix, finds minimum value in each row.
max(A,[],2)	If A is a vector, finds maximum entry. If A is a matrix, finds maximum value in each row.
min(A,B)	Produces an array with entries equal to the smallest of a_{ij} and b_{ij} . A and B must be the same size.
max(A,B)	Produces an array with entries equal to the largest of a_{ij} and b_{ij} . A and B must be the same size.

Sum and Find

FUNCTION	DESCRIPTION
sum(A)	If A is a vector, sums the entries. If A is a matrix, sums each column.
sum(A,2)	If A is a vector, sums the entries. If A is a matrix, sums each row.
sum(sum(A))	If A is a vector, sums the entries. If A is a matrix, sums all of the entries in matrix A.
find()	Finds the location of all entries in an array that meet the conditional expression enclosed in the ()

Example of Sum for Vectors

```
>> x = [1 3 -1 4 8];
```

```
>> total = sum(x)           % Simply adds all entries in x  
total =    15
```

```
>> sum(x > 0)              % x > 0 produces a vector of logicals: [1 1 0 1 1]  
ans =     4                % so sum(x>0) shows how many entries in x exceed 0
```

```
>> HowMany = sum(0 < x & x < 4)  
HowMany =     2
```

Don't do this: `>> sum = sum(x).` This will cause sum to be redefined as a double in the MATLAB workspace and will over-ride MATLAB's sum function. If you do make this error, type `>> clear sum` to eliminate the variable sum from your workspace.



Example of Sum for Matrices

```
Matrix =  14  84  24  19  47  59  28  38  
          15  25  93  25  35  55  76  57  
          26  82  35  62  83  92  76  7
```

```
>> ColSums = sum(Matrix)      % Add each column in Matrix
```

```
ColSums =  55  191  152  106  165  206  180  102
```

```
>> RowSums = sum(Matrix,2)    % Add each row in Matrix
```

```
RowSums =  313  
          381  
          463
```

```
>> TotalSum = sum(sum(Matrix)) % Add each column then add results
```

```
TotalSum =  1157
```

Example of Max for Vectors

```
>> vector = randi([-5 10],[1,8])
```

```
vector =
```

```
-5    2    1    7    7   -3    2    2
```

```
>> highest = max(vector)
```

```
highest =    7
```

```
>> [highest location] = max(vector)
```

```
highest =    7
```

```
location =    4
```

Note: location produces only first location of maximum

Example of Max for Matrices

```
>> Matrix = randi([0 100],[3 8])
```

```
Matrix =
```

14	84	24	19	47	59	28	38
15	25	93	25	35	55	76	57
26	82	35	62	83	92	76	7

```
>> highest = max(Matrix) % Finds maximum of each column
```

```
highest =
```

26	84	93	62	83	92	76	57
----	----	----	----	----	----	----	----

```
>> HighestOfMatrix = max(max(Matrix))
```

```
HighestOfMatrix = 93
```

Find

find - goes through an array and finds the index of all entries satisfying some specified condition.

Example:

```
>> C = [4 3 2 1 8 9 1]
```

```
C = 4 3 2 1 8 9 1
```

```
>> LocationsOf3 = find(C==3)
```

```
LocationsOf3 = 2
```

```
>> LocationsOf1 = find(C==1)
```

```
LocationsOf1 = 4 7
```

```
>> GreaterThan3 = find(C > 3)
```

```
GreaterThan3 = 1 5 6
```

```
>> find(C > 2 & C < 9)
```

```
ans = 1 2 5
```

Using Max and Find

```
>> vector = randi([-5 10],[1,8])
```

```
vector =
```

```
-5    2    1    7    7   -3    2    2
```

```
>> highest = max(vector)
```

```
highest =    7
```

```
>> location = find(vector == highest)
```

```
location =
```

```
4    5
```

Note: Now we get both locations for maximum value

Your Turn

Try the following commands in MATLAB:

```
>> A = [ -1  5  7  3  6 10  5 ]
```

```
>> lowest = min(A)
```

```
>> sum(A ==5)
```

```
>> find(A == 5)
```

```
>> sum(A > 3 & A < 10)
```

```
>> find(A > 3 & A < 10)
```

Your Turn

Try the following commands in MATLAB:

```
>> A = [ -1  5  7 ; 6 10 5 ]
```

```
>> min(A)
```

```
>> min(A,[ ],2)
```

```
>> min(min(A))
```

```
>> sum(A ==5)
```

```
>> sum(A==5,2)
```

```
>> sum(sum(A==5))
```

```
>> [row col] = find(A == 5)
```

```
>> sum(sum(A > 3 & A < 10))
```

```
>> [row col] = find(A > 3 & A < 10)
```


Test Your Understanding