Conclusion

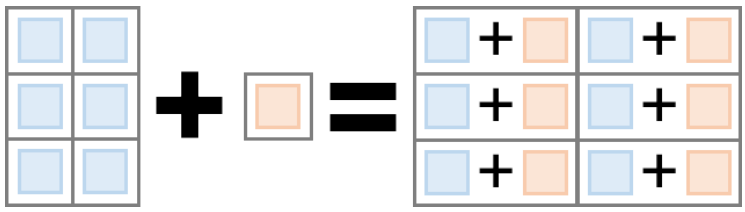
Summary

Calculations with Vectors and Matrices

Operations on Arrays

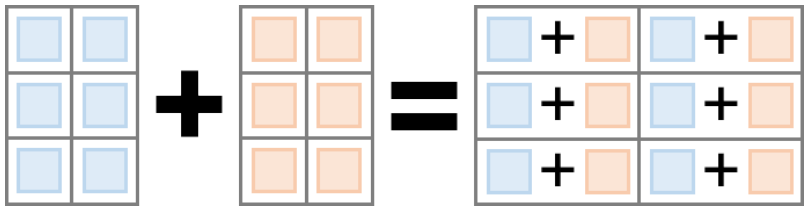
Many operators behave in an element-wise manner. That is, the operation is performed on each element of an input array individually.

Matrix Operations (Including Scalar Expansion)



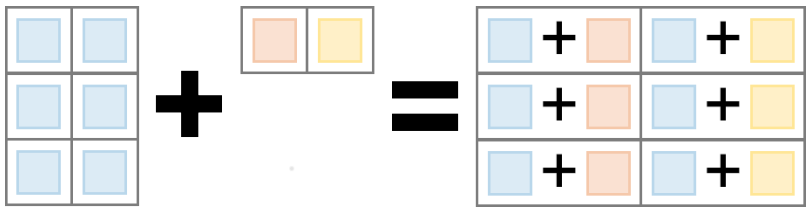
Operators	
+	Addition
-	Subtraction
*	Multiplication
/	Division
^	Exponentiation (Matrix exponentiation)

Element-Wise Operations



Operators	
+	Addition
-	Subtraction
.*	Element-wise multiplicatio
./	Element-wise division
.^	Element-wise exponentiat

Implicit Expansion



Operators	
+	Addition
-	Subtraction
.*	Element-wise multiplicatio
./	Element-wise division
.^	Element-wise exponentiat

You can perform array operations on operands of different compatible sizes. Two arrays have compatible sizes if the size o dimension is either the same or 1.

Statistical Operations on Matrices

This table includes some common mathematical functions that calculate a value for each column in a matrix.

Function	Behavior
`max`	Largest element
`min`	Smallest element
`mean`	Average or mean value
`median`	Median value
`mode`	Most frequent value
`std`	Standard deviation
`var`	Variance
`sum`	Sum of elements

```
A = [8 2 4; 3 2 6; 7 5 3; 7 10 8]
 A =

      8     2     4
      3     2     6
      7     5     3
      7    10     8

Amax = max(A)
 Amax =

      8    10     8

Astd = std(A)
 Astd =

        2.2174       3.7749       2.2174
```

`prod`	Product of elements

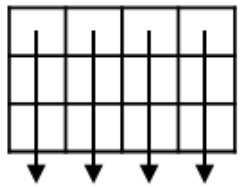
```
Asum = sum(A)
 Asum =

     25     19     21
```

Ignoring NaNs

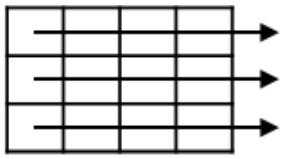
When using statistical functions, you can ignore NaN values

```
avg = mean(v,"omitnan")
```

Many statistical functions accept an optional dimensional argument that specifies whether the operation should be applied to columns independently (the default) or to rows.



Dimension = 1



Dimension = 2

```
>> M = mean(A,dim)
```

Outputs

`M`	Vector of average values along dimension `dim`

Inputs

`A`	Matrix
`dim`	Dimension across which the mean is taken: `1`: The mean of each column `2`: The mean of each row

Matrix Multiplication

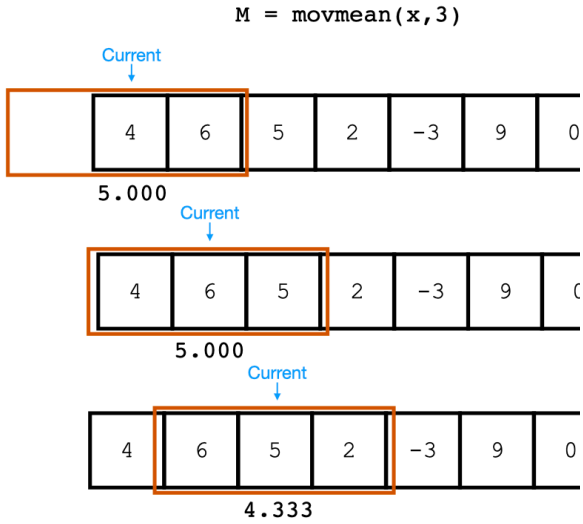
Matrix multiplication requires that the inner dimensions agree. The resulting matrix has the outer dimensions.

Moving Window Calculations

Moving window calculations can differ in how they handle endpoints of data sets.

Odd Window Sizes

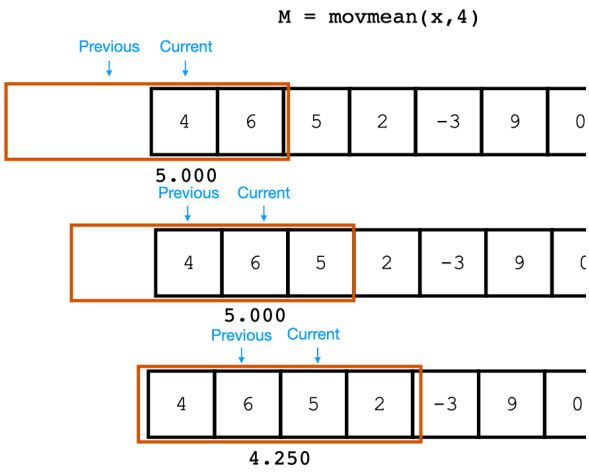
In the case where the window contains an odd number of elements, MATLAB starts by centering the window on the first element and includes only the elements that fill the window to the right of the center. As the window moves to the right, elements to the left of the window center are included in the calculation.




```
M = 1x8

    5.000  5.000  4.333  1.333  2.667  2.000
```

Even Window Sizes

If the window contains an even number of elements, MATLAB starts by centering the window between the first element and its previous element. Notice that the previous element doesn't exist in the example.




```
M = 1x8

    5.000  5.000  4.250  2.500  3.250  2.000
```