# Chapter 3
# Mathematical Operations with Arrays

Once variables are created in MATLAB they can be used in a wide variety of mathematical operations. In Chapter 1 the variables that were used in mathematical operations were all defined as scalars. This means that they were all $3 \times 3$ arrays (arrays with one row and one column that have only one element) and the mathematical operations were done with single numbers. Arrays, however, can be one-dimensional (arrays with one row, or with one column), two-dimensional (arrays with multiple rows and columns), and even of higher dimensions. In these cases the mathematical operations are more complex. MATLAB, as its name indicates, is designed to carry out advanced array operations that have many applications in science and engineering. This chapter presents the basic, most common mathematical operations that MATLAB performs using arrays.

Addition and subtraction are relatively simple operations and are covered first, in Section 3.1. The other basic operations—multiplication, division, and exponentiation—can be done in MATLAB in two different ways. One way, which uses the standard symbols (*, /, and ^), follows the rules of linear algebra and is presented in Sections 3.2 and 3.3. The second way, which is called element-by-element operations, is covered in Section 3.4. These operations use the symbols .*, ./, and .^ (a period is typed in front of the standard operation symbol). In addition, in both types of calculations, MATLAB has left division operators ( .\ or \ ), which are also explained in Sections 3.3 and 3.4.

**A Note to First-Time Users of MATLAB:**

Although matrix operations are presented first and element-by-element operations next, the order can be reversed since the two are independent of each other. It is expected that almost every MATLAB user has some knowledge of matrix operations and linear algebra, and thus will be able to follow the material covered in Sections 3.2 and 3.3 without any difficulty. Some readers, however, might prefer to read Section 3.4 first. MATLAB can be used with element-by-element operations in numerous applications that do not require linear algebra multiplication (or division) operations.

## 3.1 ADDITION AND SUBTRACTION

The operations + (addition) and – (subtraction) can be used to add (subtract) arrays of identical size (the same numbers of rows and columns) and to add (subtract) a scalar to an array. When two arrays are involved the sum, or the difference, of the arrays is obtained by adding, or subtracting, their corresponding elements.

In general, if $A$ and $B$ are two arrays (for example, $2 \times 3$ matrices),

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \end{bmatrix}$$

then the matrix that is obtained by adding $A$ and $B$ is:

$$\begin{bmatrix} (A_{11}+B_{11}) & (A_{12}+B_{12}) & (A_{13}+B_{13}) \\ (A_{21}+B_{21}) & (A_{22}+B_{22}) & (A_{23}+B_{23}) \end{bmatrix}$$

Examples are:

```
>> VectA=[8 5 4]; VectB=[10 2 7];
>> VectC=VectA+VectB
VectC =
    18     7    11
>> A=[5 -3 8; 9 2 10]
A =
     5    -3     8
     9     2    10
>> B=[10 7 4; -11 15 1]
B =
    10     7     4
   -11    15     1
>> A-B
ans =
     -5   -10     4
     20   -13     9
>> C=A+B
C =
    15     4    12
    -2    17    11
>> VectA+A
??? Error using ==> plus
Matrix dimensions must agree.

>>
```

Define two vectors.

Define a vector `VectC` that is equal to `VectA+VectB`.

Define two $2 \times 3$ matrices `A` and `B`.

Subtracting matrix `B` from matrix `A`.

Define a matrix `C` that is equal to `A+B`.

Trying to add arrays of different size.

An error message is displayed.

When a scalar (number) is added to (or subtracted from) an array, the scalar is added to (or subtracted from) all the elements of the array. Examples are:

```
>> VectA=[1 5 8 -10 2]                    Define a vector named VectA.
VectA =
     1     5     8   -10     2
>> VectA+4                                Add the scalar 4 to VectA.
ans =                                     4 is added to each element of VectA.
     5     9    12    -6
>> A=[6 21 -15; 0 -4 8]                   Define a 2×3 matrix A.
A =
     6    21   -15
     0    -4     8
>> A-5                                    Subtract the scalar 5 from A.
ans =
     1    16   -20
    -5    -9     3                        5 is subtracted from each element of A.
```

## 3.2 *Array Multiplication*

The multiplication operation * is executed by MATLAB according to the rules of linear algebra. This means that if $A$ and $B$ are two matrices, the operation $A*B$ can be carried out only if the number of columns in matrix $A$ is equal to the number of rows in matrix $B$. The result is a matrix that has the same number of rows as $A$ and the same number of columns as $B$. For example, if $A$ is a $4 \times 3$ matrix and $B$ is a $3 \times 2$ matrix:

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \\ A_{41} & A_{42} & A_{43} \end{bmatrix} \quad \text{and} \quad B = \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \\ B_{31} & B_{32} \end{bmatrix}$$

then the matrix that is obtained with the operation $A*B$ has dimensions $4 \times 2$ with the elements:

$$\begin{bmatrix} (A_{11}B_{11}+A_{12}B_{21}+A_{13}B_{31}) & (A_{11}B_{12}+A_{12}B_{22}+A_{13}B_{32}) \\ (A_{21}B_{11}+A_{22}B_{21}+A_{23}B_{31}) & (A_{21}B_{12}+A_{22}B_{22}+A_{23}B_{32}) \\ (A_{31}B_{11}+A_{32}B_{21}+A_{33}B_{31}) & (A_{31}B_{12}+A_{32}B_{22}+A_{33}B_{32}) \\ (A_{41}B_{11}+A_{42}B_{21}+A_{43}B_{31}) & (A_{41}B_{12}+A_{42}B_{22}+A_{43}B_{32}) \end{bmatrix}$$

A numerical example is:

$$\begin{bmatrix} 1 & 4 & 3 \\ 2 & 6 & 1 \\ 5 & 2 & 8 \end{bmatrix}\begin{bmatrix} 5 & 4 \\ 1 & 3 \\ 2 & 6 \end{bmatrix} = \begin{bmatrix} (1\cdot5+4\cdot1+3\cdot2) & (1\cdot4+4\cdot3+3\cdot6) \\ (2\cdot5+6\cdot1+1\cdot2) & (2\cdot4+6\cdot3+1\cdot6) \\ (5\cdot5+2\cdot1+8\cdot2) & (5\cdot4+2\cdot3+8\cdot6) \end{bmatrix} = \begin{bmatrix} 15 & 34 \\ 18 & 32 \\ 43 & 74 \end{bmatrix}$$

The product of the multiplication of two square matrices (they must be of the same size) is a square matrix of the same size. However, the multiplication of matrices is not commutative. This means that if $A$ and $B$ are both $n \times n$, then $A * B \neq B * A$. Also, the power operation can be executed only with a square matrix (since $A*A$ can be carried out only if the number of columns in the first matrix is equal to the number of rows in the second matrix).

Two vectors can be multiplied only if they have the same number of elements, and one is a row vector and the other is a column vector. The multiplication of a row vector by a column vector gives a $1 \times 1$ matrix, which is a scalar. This is the dot product of two vectors. (MATLAB also has a built-in function, `dot(a,b)`, that computes the dot product of two vectors.) When using the `dot` function, the vectors `a` and `b` can each be a row vector or a column vector (see Table 3-1). The multiplication of a column vector by a row vector, each with $n$ elements, gives an $n \times n$ matrix Multiplication of array is demonstrated in Tutorial 3-1.

**Tutorial 3-1: Multiplication of arrays.**

```
>> A=[1 4 2; 5 7 3; 9 1 6; 4 2 8]

A =
     1     4     2
     5     7     3
     9     1     6
     4     2     8
```

Define a $4 \times 3$ matrix `A`.

```
>> B=[6 1; 2 5; 7 3]

B =
     6     1
     2     5
     7     3
```

Define a $3 \times 2$ matrix `B`.

```
>> C=A*B

C =
    28    27
    65    49
    98    32
    84    38
```

Multiply matrix `A` by matrix `B` and assign the result to variable `C`.

```
>> D=B*A
??? Error using ==> *
Inner matrix dimensions must agree.
```

Trying to multiply `B` by `A`, `B*A`, gives an error since the number of columns in `B` is 2 and the number of rows in `A` is 4.

```
>> F=[1 3; 5 7]

F =
     1     3
     5     7

>> G=[4 2; 1 6]
```

Define two $2 \times 2$ matrices `F` and `G`.

**Tutorial 3-1: Multiplication of arrays. (Continued)**

```
G =
     4     2
     1     6
>> F*G                                        Multiply F*G
ans =
     7    20
    27    52
>> G*F                                        Multiply G*F
ans =
    14    26
    31    45
```

Note that the answer for G*F is not the same as the answer for F*G.

```
>> AV=[2 5 1]                                 Define a three-element row vector AV.
AV =
     2     5     1
>> BV=[3; 1; 4]                               Define a three-element column vector BV.
BV =
     3
     1
     4
>> AV*BV
ans =
    15
```

Multiply AV by BV. The answer is a scalar. (Dot product of two vectors.)

```
>> BV*AV
ans =
     6    15     3
     2     5     1
     8    20     4
>>
```

Multiply BV by AV. The answer is a $3\times3$ matrix.

When an array is multiplied by a number (actually a number is a $1\times1$ array), each element in the array is multiplied by the number. For example:

```
>> A=[2 5 7 0; 10 1 3 4; 6 2 11 5]            Define a 3×4 matrix A.
A =
     2     5     7     0
    10     1     3     4
     6     2    11     5
>> b=3                                        Assign the number 3 to the variable b.
b =
     3
```

```
>> b*A

ans =
     6    15    21     0
    30     3     9    12
    18     6    33    15
>> C=A*5
C =
    10    25    35     0
    50     5    15    20
    30    10    55    25
```

Multiply the matrix A by b. This can be done by either typing b*A or A*b.

Multiply the matrix A by 5 and assign the result to a new variable C. (Typing C=5*A gives the same result.)

Linear algebra rules of array multiplication provide a convenient way for writing a system of linear equations. For example, the system of three equations with three unknowns

$$A_{11}x_1 + A_{12}x_2 + A_{13}x_3 = B_1$$

$$A_{21}x_1 + A_{22}x_2 + A_{23}x_3 = B_2$$

$$A_{31}x_1 + A_{32}x_2 + A_{33}x_3 = B_3$$

can be written in a matrix form as:

$$\begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}$$

and in matrix notation as

$$AX = B \quad \text{where} \quad A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix}, \; X = \begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix}, \text{ and } \; B = \begin{bmatrix} B_1 \\ B_2 \\ B_3 \end{bmatrix}.$$

## 3.3 *Array Division*

The division operation is also associated with the rules of linear algebra. This operation is more complex, and only a brief explanation is given below. A full explanation can be found in books on linear algebra.

The division operation can be explained with the help of the identity matrix and the inverse operation.

**Identity matrix:**

The identity matrix is a square matrix in which the diagonal elements are 1s and the rest of the elements are 0s. As was shown in Section 2.2.1, an identity matrix can be created in MATLAB with the `eye` command. When the identity matrix multiplies another matrix (or vector), that matrix (or vector) is unchanged (the multiplication has to be done according to the rules of linear algebra). This is equivalent to multiplying a scalar by 1. For example:

$$\begin{bmatrix} 7 & 3 & 8 \\ 4 & 11 & 5 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 7 & 3 & 8 \\ 4 & 11 & 5 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 8 \\ 2 \\ 15 \end{bmatrix} = \begin{bmatrix} 8 \\ 2 \\ 15 \end{bmatrix} \quad \text{or}$$

$$\begin{bmatrix} 6 & 2 & 9 \\ 1 & 8 & 3 \\ 7 & 5 & 4 \end{bmatrix}\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 6 & 2 & 9 \\ 1 & 8 & 3 \\ 7 & 5 & 4 \end{bmatrix}$$

If a matrix $A$ is square, it can be multiplied by the identity matrix, $I$, from the left or from the right:

$$AI = IA = A$$

**Inverse of a matrix:**

The matrix $B$ is the inverse of the matrix $A$ if, when the two matrices are multiplied, the product is the identity matrix. Both matrices must be square, and the multiplication order can be $BA$ or $AB$.

$$BA = AB = I$$

Obviously $B$ is the inverse of $A$, and $A$ is the inverse of $B$. For example:

$$\begin{bmatrix} 2 & 1 & 4 \\ 4 & 1 & 8 \\ 2 & -1 & 3 \end{bmatrix}\begin{bmatrix} 5.5 & -3.5 & 2 \\ 2 & -1 & 0 \\ -3 & 2 & 1 \end{bmatrix} = \begin{bmatrix} 5.5 & -3.5 & 2 \\ 2 & -1 & 0 \\ -3 & 2 & 1 \end{bmatrix}\begin{bmatrix} 2 & 1 & 4 \\ 4 & 1 & 8 \\ 2 & -1 & 3 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

The inverse of a matrix $A$ is typically written as $A^{-1}$. In MATLAB the inverse of a matrix can be obtained either by raising $A$ to the power of –1, `A^-1`, or with the `inv(A)` function. Multiplying the matrices above with MATLAB is shown below.

```
>> A=[2 1 4; 4 1 8; 2 -1 3]                    Creating the matrix A.
A =
     2     1     4
     4     1     8
     2    -1     3
>> B=inv(A)                    Use the inv function to find the
                               inverse of A and assign it to B.
B =
    5.5000   -3.5000    2.0000
    2.0000   -1.0000         0
   -3.0000    2.0000   -1.0000
>> A*B           Multiplication of A and B gives the identity matrix.
ans =
     1     0     0
     0     1     0
     0     0     1
```

```
>> A*A^-1
ans =
     1     0     0
     0     1     0
     0     0     1
```

Use the power –1 to find the inverse of `A`. Multiplying it by `A` gives the identity matrix.

Not every matrix has an inverse. A matrix has an inverse only if it is square and its determinant is not equal to zero.

**Determinants:**

A determinant is a function associated with square matrices. A short review on determinants is given below. For a more detailed coverage refer to books on linear algebra.

The determinant is a function that associates with each square matrix $A$ a number, called the determinant of the matrix. The determinant is typically denoted by det($A$) or $|A|$. The determinant is calculated according to specific rules. For a second-order $2 \times 2$ matrix, the rule is:

$$|A| = \begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}, \text{ for example, } \begin{vmatrix} 6 & 5 \\ 3 & 9 \end{vmatrix} = 6 \cdot 9 - 5 \cdot 3 = 39$$

The determinant of a square matrix can be calculated with the `det` command (see Table 3-1).

**Array division:**

MATLAB has two types of array division, right division and left division.

**Left division, \ :**

Left division is used to solve the matrix equation $AX = B$. In this equation $X$ and $B$ are column vectors. This equation can be solved by multiplying, on the left, both sides by the inverse of $A$:

$$A^{-1}AX = A^{-1}B$$

The left-hand side of this equation is $X$, since

$$A^{-1}AX = IX = X$$

So the solution of $AX = B$ is:

$$X = A^{-1}B$$

In MATLAB the last equation can be written by using the left division character:

$$X = A\backslash B$$

It should be pointed out here that although the last two operations appear to give the same result, the method by which MATLAB calculates $X$ is different. In the first, MATLAB calculates $A^{-1}$ and then uses it to multiply $B$. In the second (left division), the solution $X$ is obtained numerically using a method that is based on Gauss elimination. The left division method is recommended for solving a set of linear equations, because the calculation of the inverse may be less accurate than the Gauss elimination method when large matrices are involved.

**Right division, / :**

The right division is used to solve the matrix equation $XC = D$. In this equation $X$ and $D$ are row vectors. This equation can be solved by multiplying, on the right, both sides by the inverse of $C$:

$$XCC^{-1} = DC^{-1}$$

which gives

$$X = DC^{-1}$$

In MATLAB the last equation can be written using the right division character:

$$X = D/C$$

The following example demonstrates the use of the left and right division, and the `inv` function to solve a set of linear equations.

## Sample Problem 3-1: Solving three linear equations (array division)

Use matrix operations to solve the following system of linear equations.

$$4x - 2y + 6z = 8$$
$$2x + 8y + 2z = 4$$
$$6x + 10y + 3z = 0$$

**Solution**

Using the rules of linear algebra demonstrated earlier, the above system of equations can be written in the matrix form $AX=B$ or in the form $XC=D$:

$$\begin{bmatrix} 4 & -2 & 6 \\ 2 & 8 & 2 \\ 6 & 10 & 3 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} 8 \\ 4 \\ 0 \end{bmatrix} \quad \text{or} \quad \begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} 4 & 2 & 6 \\ -2 & 8 & 10 \\ 6 & 2 & 3 \end{bmatrix} = \begin{bmatrix} 8 & 4 & 0 \end{bmatrix}$$

Solutions for both forms are shown below:

```
>> A=[4 -2 6; 2 8 2; 6 10 3];
```

Solving the form $AX = B$.

```
>> B=[8; 4; 0];
>> X=A\B
X =
   -1.8049
    0.2927
    2.6341
```

Solving by using left division: $X = A \backslash B$.

```
>> Xb=inv(A)*B
Xb =
   -1.8049
    0.2927
    2.6341
```

Solving by using the inverse of $A$: $X=A^{-1}B$.

```
>> C=[4 2 6; -2 8 10; 6 2 3];          Solving the form XC = D.
>> D=[8 4 0];
>> Xc=D/C                              Solving by using right division: X = D/C.
Xc =
   -1.8049    0.2927    2.6341
>> Xd=D*inv(C)                         Solving by using the inverse of C: X=DC^-1.
Xd =
   -1.8049    0.2927    2.6341
```

## 3.4 *Element-by-Element Operations*

In Sections 3.2 and 3.3 it was shown that when the regular symbols for multiplication and division (* and /) are used with arrays, the mathematical operations follow the rules of linear algebra. There are, however, many situations that require element-by-element operations. These operations are carried out on each of the elements of the array (or arrays). Addition and subtraction are by definition already element-by-element operations, since when two arrays are added (or subtracted) the operation is executed with the elements that are in the same position in the arrays. Element-by-element operations can be done only with arrays of the same size.

Element-by-element multiplication, division, or exponentiation of two vectors or matrices is entered in MATLAB by typing a period in front of the arithmetic operator.

| Symbol | Description | Symbol | Description |
|---|---|---|---|
| .* | Multiplication | ./ | Right division |
| .^ | Exponentiation | .\ | Left Division |

If two vectors $a$ and $b$ are $a$=[$a_1$ $a_2$ $a_3$ $a_4$] and $b$=[$b_1$ $b_2$ $b_3$ $b_4$], then element-by-element multiplication, division, and exponentiation of the two vectors gives:

$$a.*b = [a_1*b_1 \;\; a_2*b_2 \;\; a_3*b_3 \;\; a_4*b_4]$$

$$a./b = [a_1/b_1 \;\; a_2/b_2 \;\; a_3/b_3 \;\; a_4/b_4]$$

$$a.\hat{}b = \left[(a_1)^{b_1} \;\; (a_2)^{b_2} \;\; (a_3)^{b_3} \;\; (a_4)^{b_4}\right]$$

If two matrices $A$ and $B$ are

$$A = \begin{bmatrix} A_{11} & A_{12} & A_{13} \\ A_{21} & A_{22} & A_{23} \\ A_{31} & A_{32} & A_{33} \end{bmatrix} \text{ and } B = \begin{bmatrix} B_{11} & B_{12} & B_{13} \\ B_{21} & B_{22} & B_{23} \\ B_{31} & B_{32} & B_{33} \end{bmatrix}$$

then element-by-element multiplication and division of the two matrices give:

$$A\ .\!*\ B = \begin{bmatrix} A_{11}B_{11} & A_{12}B_{12} & A_{13}B_{13} \\ A_{21}B_{21} & A_{22}B_{22} & A_{23}B_{23} \\ A_{31}B_{31} & A_{32}B_{32} & A_{33}B_{33} \end{bmatrix} \qquad A\ ./\ B = \begin{bmatrix} A_{11}/B_{11} & A_{12}/B_{12} & A_{13}/B_{13} \\ A_{21}/B_{21} & A_{22}/B_{22} & A_{23}/B_{23} \\ A_{31}/B_{31} & A_{32}/B_{32} & A_{33}/B_{33} \end{bmatrix}$$

Element-by-element exponentiation of matrix $A$ gives:

$$\text{A .\^{}}\ n = \begin{bmatrix} (A_{11})^n & (A_{12})^n & (A_{13})^n \\ (A_{21})^n & (A_{22})^n & (A_{23})^n \\ (A_{31})^n & (A_{32})^n & (A_{33})^n \end{bmatrix}$$

Element-by-element multiplication, division, and exponentiation are demonstrated in Tutorial 3-2.

**Tutorial 3-2: Element-by-element operations.**

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

Define a $2 \times 3$ array `A`.

```
>> B=[1 4 10; 3 2 7]
B =
     1     4    10
     3     2     7
```

Define a $2 \times 3$ array `B`.

```
>> A.*B
ans =
     2    24    30
    15    16    28
```

Element-by-element multiplication of array `A` by `B`.

```
>> C=A./B
C =
    2.0000    1.5000    0.3000
    1.6667    4.0000    0.5714
```

Element-by-element division of array `A` by `B`. The result is assigned to variable `C`.

**Tutorial 3-2: Element-by-element operations. (Continued)**

```
>> B.^3

ans =
     1     64   1000
    27      8    343
```

Element-by-element exponentiation of array B. The result is an array in which each term is the corresponding term in B raised to the power of 3.

```
>> A*B

??? Error using ==> *
Inner matrix dimensions must agree.
```

Trying to multiply A*B gives an error, since A and B cannot be multiplied according to linear algebra rules. (The number of columns in A is not equal to the number of rows in B.)

Element-by-element calculations are very useful for calculating the value of a function at many values of its argument. This is done by first defining a vector that contains values of the independent variable, and then using this vector in element-by-element computations to create a vector in which each element is the corresponding value of the function. One example is:

```
>> x=[1:8]
x =
   1    2    3    4    5    6    7    8
>> y=x.^2-4*x
y =
  -3   -4   -3    0    5   12   21   32
>>
```

Create a vector x with eight elements.

Vector x is used in element-by-element calculations of the elements of vector y.

In the example above $y = x^2 - 4x$. Element-by-element operation is needed when $x$ is squared. Each element in the vector $y$ is the value of $y$ that is obtained when the value of the corresponding element of the vector $x$ is substituted in the equation. Another example is:

```
>> z=[1:2:11]
z =
    1    3    5    7    9   11
>> y=(z.^3 + 5*z)./(4*z.^2 - 10)

 y =
   -1.0000    1.6154    1.6667    2.0323    2.4650    2.9241
```

Create a vector z with six elements.

Vector z is used in element-by-element calculations of the elements of vector y.

In the last example $y = \frac{z^3+5z}{4z^2-10}$. Element-by-element operations are used in this example three times: to calculate $z^3$ and $z^2$, and to divide the numerator by the denominator.

## 3.5 USING ARRAYS IN MATLAB BUILT-IN MATH FUNCTIONS

The built-in functions in MATLAB are written such that when the argument (input) is an array, the operation that is defined by the function is executed on each element of the array. (One can think of the operation as element-by-element application of the function.) The result (output) from such an operation is an array in which each element is calculated by entering the corresponding element of the argument (input) array into the function. For example, if a vector with seven elements is substituted in the function `cos(x)`, the result is a vector with seven elements in which each element is the cosine of the corresponding element in `x`. This is shown below.

```
>> x=[0:pi/6:pi]
x =
     0    0.5236    1.0472    1.5708    2.0944     2.6180     3.1416
>>y=cos(x)
y =
   1.0000    0.8660    0.5000    0.0000   -0.5000   -0.8660   -
1.0000
>>
```

An example in which the argument variable is a matrix is:

```
>> d=[1 4 9; 16 25 36; 49 64 81]          Creating a 3×3 array.
d =
     1     4     9
    16    25    36
    49    64    81
>> h=sqrt(d)
h =
     1     2     3        h is a 3×3 array in which each
     4     5     6        element is the square root of the
     7     8     9        corresponding element in array d.
```

The feature of MATLAB in which arrays can be used as arguments in functions is called vectorization.

## 3.6 BUILT-IN FUNCTIONS FOR ANALYZING ARRAYS

MATLAB has many built-in functions for analyzing arrays. Table 3-1 lists some of these functions.

**Table 3-1: Built-in array functions**

| Function | Description | Example |
|---|---|---|
| `mean(A)` | If `A` is a vector, returns the mean value of the elements of the vector. | `>> A=[5 9 2 4];`<br>`>> mean(A)`<br>`ans =`<br>`5` |
| `C=max(A)`<br><br>`[d,n]=max(A)` | If `A` is a vector, `C` is the largest element in `A`. If `A` is a matrix, `C` is a row vector containing the largest element of each column of `A`.<br><br>If `A` is a vector, `d` is the largest element in `A`, and `n` is the position of the element (the first if several have the max value). | `>> A=[5 9 2 4 11 6 11 1];`<br>`>> C=max(A)`<br>`C =`<br>`11`<br><br>`>> [d,n]=max(A)`<br>`d =`<br>`11`<br>`n =`<br>`5` |
| `min(A)`<br><br>`[d,n]=min(A)` | The same as `max(A)`, but for the smallest element.<br><br>The same as `[d,n]= max(A)`, but for the smallest element. | `>> A=[5 9 2 4];`<br>`>> min(A)`<br>`ans =`<br>`2` |
| `sum(A)` | If `A` is a vector, returns the sum of the elements of the vector. | `>> A=[5 9 2 4];`<br>`>> sum(A)`<br>`ans =`<br>`20` |
| `sort(A)` | If `A` is a vector, arranges the elements of the vector in ascending order. | `>> A=[5 9 2 4];`<br>`>> sort(A)`<br>`ans =`<br>`2 4 5 9` |
| `median(A)` | If `A` is a vector, returns the median value of the elements of the vector. | `>> A=[5 9 2 4];`<br>`>> median(A)`<br>`ans =`<br>`4.5000` |

**Table 3-1: Built-in array functions (Continued)**

| Function | Description | Example |
|---|---|---|
| `std(A)` | If `A` is a vector, returns the standard deviation of the elements of the vector. | `>> A=[5 9 2 4];`<br>`>> std(A)`<br>`ans =`<br>`    2.9439` |
| `det(A)` | Returns the determinant of a square matrix `A`. | `>> A=[2 4; 3 5];`<br>`>> det(A)`<br>`ans =`<br>`    -2` |
| `dot(a,b)` | Calculates the scalar (dot) product of two vectors `a` and `b`. The vectors can each be row or column vectors. | `>> a=[1 2 3];`<br>`>> b=[3 4 5];`<br>`>> dot(a,b)`<br>`ans =`<br>`    26` |
| `cross(a,b)` | Calculates the cross product of two vectors `a` and `b`, (`axb`). The two vectors must have each three elements. | `>> a=[1 3 2];`<br>`>> b=[2 4 1];`<br>`>> cross(a,b)`<br>`ans =`<br>`    -5     3    -2` |
| `inv(A)` | Returns the inverse of a square matrix `A`. | `>> A=[2 -2 1; 3 2 -1; 2 -3 2];`<br>`>> inv(A)`<br>`ans =`<br>`    0.2000    0.2000    0`<br>`   -1.6000    0.4000    1.0000`<br>`   -2.6000    0.4000    2.0000` |

## 3.7 GENERATION OF RANDOM NUMBERS

Simulations of many physical processes and engineering applications frequently require using a number (or a set of numbers) with a random value. MATLAB has three commands—`rand`, `randn`, and `randi`—that can be used to assign random numbers to variables.

**The `rand` command:**

The `rand` command generates uniformly distributed random numbers with values between 0 and 1. The command can be used to assign these numbers to a scalar, a vector, or a matrix, as shown in Table 3-2.

**Table 3-2: The `rand` command**

| Command | Description | Example |
|---|---|---|
| `rand` | Generates a single random number between 0 and 1. | `>> rand`<br>`ans =`<br>`0.2311` |
| `rand(1,n)` | Generates an `n`-element row vector of random numbers between 0 and 1. | `>> a=rand(1,4)`<br>`a =`<br>`0.6068 0.4860 0.8913 0.7621` |
| `rand(n)` | Generates an `n`×`n` matrix with random numbers between 0 and 1. | `>> b=rand(3)`<br>`b =`<br>`0.4565 0.4447 0.9218`<br>`0.0185 0.6154 0.7382`<br>`0.8214 0.7919 0.1763` |
| `rand(m,n)` | Generates an `m`×`n` matrix with random numbers between 0 and 1. | `>> c=rand(2,4)`<br>`c =`<br>`0.4057 0.9169 0.8936 0.3529`<br>`0.9355 0.4103 0.0579 0.8132` |
| `randperm(n)` | Generates a row vector with `n` elements that are random permutation of integers 1 through `n`. | `>> randperm(8)`<br>`ans =`<br>`8 2 7 4 3 6 5 1` |

Sometimes there is a need for random numbers that are distributed in an interval other than (0,1), or for numbers that are integers only. This can be done using mathematical operations with the `rand` function. Random numbers that are distributed in a range $(a,b)$ can be obtained by multiplying `rand` by $(b-a)$ and adding the product to $a$:

$$(b-a)*\texttt{rand}+a$$

For example, a vector of 10 elements with random values between –5 and 10 can be created by ($a = -5$, $b = 10$):

```
>> v=15*rand(1,10)-5
v =
   -1.8640    0.6973    6.7499    5.2127    1.9164    3.5174
6.9132   -4.1123    4.0430   -4.2460
```

**The `randi` command:**

The `randi` command generates uniformly distributed random integer. The command can be used to assign these numbers to a scalar, a vector, or a matrix, as shown in Table 3-3.

**Table 3-3: The `randi` command**

| Command | Description | Example |
|---|---|---|
| `randi(imax)` (imax is an integer) | Generates a single random number between 1 and imax. | `>> a=randi(15)` `a =` `9` |
| `randi(imax,n)` | Generates an n×n matrix with random integers between 1 and imax. | `>> b=randi(15,3)` `b =` `4 8 11` `14 3 8` `1 15 8` |
| `randi(imax,m,n)` | Generates an m×n matrix with random integers between 1 and imax. | `>> c=randi(15,2,4)` `c =` `1 1 8 13` `11 2 2 13` |

The range of the random integers can be set to be between any two integers by typing `[imin imax]` instead of `imax`. For example, a $3 \times 4$ matrix with random integers between 50 and 90 is created by:

```
>> d=randi([50 90],3,4)
d =
    57    82    71    75
    66    52    67    61
    84    66    76    67
```

**The `randn` command:**

The `randn` command generates normally distributed numbers with mean 0 and standard deviation of 1. The command can be used to generate a single number, a vector, or a matrix in the same way as the `rand` command. For example, a $3 \times 4$ matrix is created by:

```
>> d=randn(3,4)
d =
   -0.4326    0.2877    1.1892    0.1746
   -1.6656   -1.1465   -0.0376   -0.1867
    0.1253    1.1909    0.3273    0.7258
```

The mean and standard deviation of the numbers can be changed by mathematical operations to have any values. This is done by multiplying the number generated by the `randn` function by the desired standard deviation, and adding the desired mean. For example, a vector of six numbers with a mean of 50 and standard deviation of 6 is generated by:

```
>> v=4*randn(1,6)+50
v =
   42.7785   57.4344   47.5819   50.4134   52.2527   50.4544
```

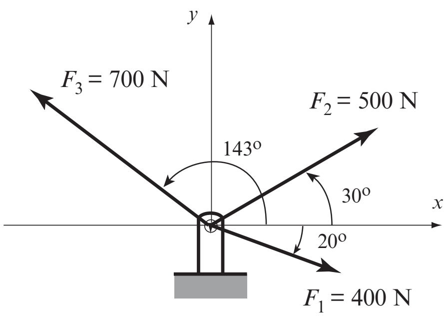
Integers of normally distributed numbers can be obtained by using the `round` function.

```
>> w=round(4*randn(1,6)+50)
w =
    51    49    46    49    50    44
```

## 3.8 EXAMPLES OF MATLAB APPLICATIONS

### Sample Problem 3-2: Equivalent force system (addition of vectors)

Three forces are applied to a bracket as shown. Determine the total (equivalent) force applied to the bracket.




**Solution**

A force is a vector (a physical quantity that has a magnitude and direction). In a Cartesian coordinate system a two-dimensional vector **F** can be written as:

$$\mathbf{F} = F_x\mathbf{i} + F_y\mathbf{j} = F\cos\theta\mathbf{i} + F\sin\theta\mathbf{j} = F(\cos\theta\mathbf{i} + \sin\theta\mathbf{j})$$

where $F$ is the magnitude of the force and θ is its angle relative to the $x$ axis, $F_x$ and $F_y$ are the components of **F** in the directions of the $x$ and $y$ axes, respectively, and **i** and **j** are unit vectors in these directions. If $F_x$ and $F_y$ are known, then $F$ and θ can be determined by:

$$F = \sqrt{F_x^2 + F_y^2} \quad \text{and} \quad \tan\theta = \frac{F_y}{F_x}$$

The total (equivalent) force applied on the bracket is obtained by adding the forces that are acting on the bracket. The MATLAB solution below follows three steps:

- Write each force as a vector with two elements, where the first element is the $x$ component of the vector and the second element is the $y$ component.
- Determine the vector form of the equivalent force by adding the vectors.
- Determine the magnitude and direction of the equivalent force.

The problem is solved in the following program, written in a script file.

```
% Sample Problem 3-2 solution (script file)
clear
F1M=400; F2M=500; F3M=700;                  Define variables with the
                                            magnitude of each vector.
Th1=-20; Th2=30; Th3=143;
                        Define variables with the angle of each vector.
F1=F1M*[cosd(Th1) sind(Th1)]
F2=F2M*[cosd(Th2) sind(Th2)]                Define the three vectors.
F3=F3M*[cosd(Th3) sind(Th3)]
Ftot=F1+F2+F3                           Calculate the total force vector.
FtotM=sqrt(Ftot(1)^2+Ftot(2)^2)             Calculate the magnitude
                                            of the total force vector.
Th=atand(Ftot(2)/Ftot(1))
                        Calculate the angle of the total force vector.
```

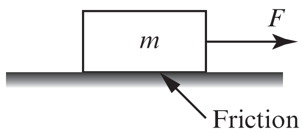
When the program is executed, the following is displayed in the Command Window:

```
F1 =
  375.8770 -136.8081                 The components of F1.
F2 =
  433.0127  250.0000                 The components of F2.
F3 =
 -559.0449  421.2705                 The components of F3.
Ftot =
  249.8449  534.4625          The components of the total force.
FtotM =
  589.9768                     The magnitude of the total force.
Th =
   64.9453              The direction of the total force in degrees.
```

The equivalent force has a magnitude of 589.98 N, and is directed 64.95° (ccw) relative to the $x$ axis. In vector notation, the force is $\mathbf{F} = 249.84\mathbf{i} + 534.46\mathbf{j}$ N.

## Sample Problem 3-3: Friction experiment (element-by-element calculations)

The coefficient of friction, μ, can be determined in an experiment by measuring the force $F$ required to move a mass $m$. When $F$ is measured and $m$ is known, the coefficient of friction can be calculated by:




$$\mu = F/(mg) \quad (g = 9.81\ \text{m/s}^2).$$

Results from measuring $F$ in six tests are given in the table below. Determine the coefficient of friction in each test, and the average from all tests.

| Test | 1 | 2 | 3 | 4 | 5 | 6 |
|---|---|---|---|---|---|---|
| Mass $m$ (kg) | 2 | 4 | 5 | 10 | 20 | 50 |
| Force $F$ (N) | 12.5 | 23.5 | 30 | 61 | 117 | 294 |

**Solution**

A solution using MATLAB commands in the Command Window is shown below.

```
>> m=[2 4 5 10 20 50];
>> F=[12.5 23.5 30 61 117 294];
>> mu=F./(m*9.81)

mu =

   0.6371   0.5989   0.6116   0.6218   0.5963   0.5994

>> mu_ave=mean(mu)

mu_ave =
    0.6109
```

Enter the values of m in a vector.

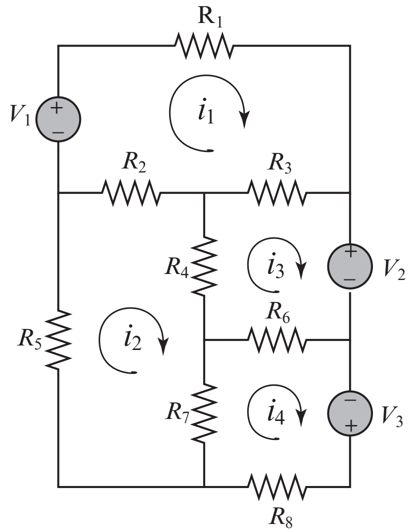
Enter the values of F in a vector.

A value for mu is calculated for each test, using element-by-element calculations.

The average of the elements in the vector mu is determined by using the function mean.

### Sample Problem 3-4: Electrical resistive network analysis (solving a system of linear equations)

The electrical circuit shown consists of resistors and voltage sources. Determine the current in each resistor using the mesh current method, which is based on Kirchhoff's voltage law.




$V_1 = 20\,\text{V},\ V_2 = 12\,\text{V},\ V_3 = 40\,\text{V}$
$R_1 = 18\,\Omega,\ R_2 = 10\,\Omega,\ R_3 = 16\,\Omega$
$R_4 = 6\,\Omega,\ R_5 = 15\,\Omega,\ R_6 = 8\,\Omega$
$R_7 = 12\,\Omega,\ R_8 = 14\,\Omega$

**Solution**

Kirchhoff's voltage law states that the sum of the voltage around a closed circuit is zero. In the mesh current method a current is first assigned for each mesh ($i_1$, $i_2$, $i_3$, $i_4$ in the figure). Then Kirchhoff's voltage law is applied for each mesh. This results in a system of linear equations for the currents (in this case four equations). The solution gives the values of the mesh currents. The current in a resistor that belongs to two meshes is the sum of the currents in the corresponding meshes. It is convenient to assume that all the currents are in the same direction (clockwise in this case). In the equation for each mesh, the voltage source is positive if the current flows to the – pole, and the voltage of a resistor is negative for current in the direction of the mesh current.

The equations for the four meshes in the current problem are:

$$V_1 - R_1 i_1 - R_3(i_1 - i_3) - R_2(i_1 - i_2) = 0$$

$$-R_5 i_2 - R_2(i_2 - i_1) - R_4(i_2 - i_3) - R_7(i_2 - i_4) = 0$$

$$-V_2 - R_6(i_3 - i_4) - R_4(i_3 - i_2) - R_3(i_3 - i_1) = 0$$

$$V_3 - R_8 i_4 - R_7(i_4 - i_2) - R_6(i_4 - i_3) = 0$$

The four equations can be rewritten in matrix form $[A][x] = [B]$:

$$\begin{bmatrix} -(R_1+R_2+R_3) & R_2 & R_3 & 0 \\ R_2 & -(R_2+R_4+R_5+R_7) & R_4 & R_7 \\ R_3 & R_4 & -(R_3+R_4+R_6) & R_6 \\ 0 & R_7 & R_6 & -(R_6+R_7+R_8) \end{bmatrix} \begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix} = \begin{bmatrix} -V_1 \\ 0 \\ V_2 \\ -V_3 \end{bmatrix}$$

The problem is solved in the following program, written in a script file:

```
V1=20; V2=12; V3=40;
R1=18; R2=10; R3=16; R4=6;
R5=15; R6=8; R7=12; R8=14;
A=[-(R1+R2+R3) R2 R3 0
R2 -(R2+R4+R5+R7) R4 R7
R3 R4 -(R3+R4+R6) R6
0 R7 R6 -(R6+R7+R8)]
>> B=[-V1; 0; V2; -V3]
>> I=A\B
```

Define variables with the values of the V's and R's.

Create the matrix A.

Create the vector B.

Solve for the currents by using left division.

When the script file is executed, the following is displayed in the Command Window:

```
A =
   -44    10    16     0
    10   -43     6    12
    16     6   -30     8
     0    12     8   -34
B =
   -20
     0
    12
   -40
I =
    0.8411
    0.7206
    0.6127
    1.5750
>>
```

The numerical value of the matrix A.

The numerical value of the vector B.

$\begin{bmatrix} i_1 \\ i_2 \\ i_3 \\ i_4 \end{bmatrix}$

The solution.

The last column vector gives the current in each mesh. The currents in the resistors $R_1$, $R_5$, and $R_8$ are $i_1 = 0.8411$ A, $i_2 = 0.7206$ A, and $i_4 = 1.5750$ A, respectively. The other resistors belong to two meshes and their current is the sum of the currents in the meshes.

The current in resistor $R_2$ is $i_1 - i_2 = 0.1205$ A.

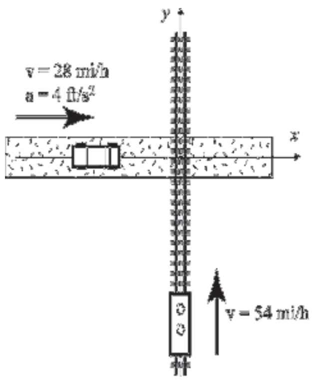
The current in resistor $R_3$ is $i_1 - i_3 = 0.2284$ A.

The current in resistor $R_4$ is $i_2 - i_3 = 0.1079$ A.

The current in resistor $R_6$ is $i_4 - i_3 = 0.9623$ A.

The current in resistor $R_7$ is $i_4 - i_2 = 0.8544$ A.

## Sample Problem 3-5: Motion of two particles




A train and a car are approaching a road crossing. At time $t = 0$ the train is 400 ft south of the crossing traveling north at a constant speed of 54 mi/h. At the same time the car is 200 ft west of the crossing traveling east at a speed of 28 mi/h and accelerating at 4 ft/s$^2$. Determine the positions of the train and the car, the distance between them, and the speed of the train relative to the car every second for the next 10 seconds.

To show the results, create an $11 \times 6$ matrix in which each row has the time in the first column and the train position, car position, distance between the train and the car, car speed, and the speed of the train relative to the car in the next five columns, respectively.

**Solution**

The position of an object that moves along a straight line at a constant acceleration is given by $s = s_0 + v_0 t + \frac{1}{2}at^2$ where $s_0$ and $v_0$ are the position and velocity at $t = 0$, and $a$ is the acceleration. Applying this equation to the train and the car gives:

$$y = -400 + v_{0train} t \quad \text{(train)}$$

$$x = -200 + v_{0car} t + \frac{1}{2} a_{car} t^2 \quad \text{(car)}$$

The distance between the car and the train is: $d = \sqrt{x^2 + y^2}$. The velocity of the train is constant and in vector notation is given by $\mathbf{v}_{train} = v_{otrain}\mathbf{j}$. The car is accelerating and its velocity at time $t$ is given by $\mathbf{v}_{car} = (v_{ocar} + a_{car}t)\mathbf{i}$. The velocity of the train relative to the car, $\mathbf{v}_{t/c}$, is given by $\mathbf{v}_{t/c} = \mathbf{v}_{train} - \mathbf{v}_{car} = -(v_{0car} + a_{car}t)\mathbf{i} + v_{0train}\mathbf{j}$. The magnitude (speed) of this velocity is the length of the vector.

The problem is solved in the following program, written in a script file. First a vector `t` with 11 elements for the time from 0 to 10 s is created, then the positions of the train and the car, the distance between them, and the speed of the train relative to the car at each time element are calculated.

```
v0train=54*5280/3600; v0car=28*5280/3600; acar=4;
```
Create variables for the initial velocities (in ft/s) and the acceleration.

```
t=0:10;
```
Create the vector `t`.

```
y=-400+v0train*t;
x=-200+v0car*t+0.5*acar*t.^2;
```
Calculate the train and car positions.

```
d=sqrt(x.^2+y.^2);
```
Calculate the distance between the train and car.

```
vcar=v0car+acar*t;                        Calculate the car's velocity.
speed_trainRcar=sqrt(vcar.^2+v0train^2);
                          Calculate the speed of the train relative to the car.
table=[t' y' x' d' vcar' speed_trainRcar']
                                    Create a table (see note below).
```

**Note:** In the commands above, `table` is the name of the variable that is a matrix containing the data to be displayed.

When the script file is executed, the following is displayed in the Command Window:

```
table =
        0 -400.0000 -200.0000  447.2136   41.0667   89.2139
   1.0000 -320.8000 -156.9333  357.1284   45.0667   91.1243
   2.0000 -241.6000 -109.8667  265.4077   49.0667   93.1675
   3.0000 -162.4000  -58.8000  172.7171   53.0667   95.3347
   4.0000  -83.2000   -3.7333   83.2837   57.0667   97.6178
   5.0000   -4.0000   55.3333   55.4777   61.0667  100.0089
   6.0000   75.2000  118.4000  140.2626   65.0667  102.5003
   7.0000  154.4000  185.4667  241.3239   69.0667  105.0849
   8.0000  233.6000  256.5333  346.9558   73.0667  107.7561
   9.0000  312.8000  331.6000  455.8535   77.0667  110.5075
  10.0000  392.0000  410.6667  567.7245   81.0667  113.3333

   Time      Train      Car       Car-train   Car      Train speed
   (s)       position   position  distance    speed    relative to
             (ft)       (ft)      (ft)        (ft/s)   the car (ft/s)
```

In this problem the results (numbers) are displayed by MATLAB without any text. Instructions on how to add text to output generated by MATLAB are presented in Chapter 4.

## 3.9 PROBLEMS

***Note:*** *Additional problems for practicing mathematical operations with arrays are provided at the end of Chapter 4.*

1. For the function $y = x^2 - \frac{x}{x+3}$, calculate the value of $y$ for the following values of $x$ using element-by-element operations: $0, 1, 2, 3, 4, 5, 6, 7$ .

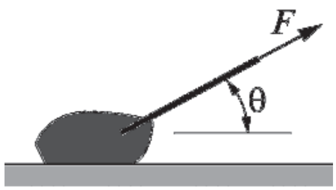
2. For the function $y = x^4 e^{-x}$, calculate the value of $y$ for the following values of $x$ using element-by-element operations: $1.5, 2, 2.5, 3, 3.5, 4$ .

3. For the function $y = (x + x\sqrt{x+3})(1+2x^2) - x^3$, calculate the value of $y$ for the following values of $x$ using element-by-element operations: –2, –1.5, –1, –0.5, 0, 0.5, 1, 1.5, 2.

4. For the function $y = \frac{4\sin x + 6}{(\cos^2 x + \sin x)^2}$, calculate the value of $y$ for the following values of $x$ using element-by-element operations: 15°, 25°, 35°, 45°, 55°, 65°.

5. The radius, $r$, of a sphere can be calculated from its volume, $V$, by:
$$r = \sqrt[3]{\frac{3V}{4\pi}}$$
The surface area of a sphere, $S$, is given by:
$$S = 4\pi r^2$$
Determine the radius and surface area of spheres with volumes of 4,000, 3,500, 3,000, 2,500, 2,000, 1,500, and 1,000 in.$^3$. Display the results in a three-column table where the values of $r$, $V$, and $S$ are displayed in the first, second, and third columns, respectively. The values of $r$ and $S$ that are displayed in the table should be rounded to the nearest tenth of an inch.

6. A 70 lb-bag of rice is being pulled by a person by applying a force $F$ at an angle $\theta$ as shown. The force required to drag the bag is given by:




$$F(\theta) = \frac{70\mu}{\mu\sin\theta + \cos\theta}$$
where $\mu = 0.35$ is the friction coefficient.
(*a*) Determine $F(\theta)$ for $\theta = 5°, 10°, 15°, 20°, 25°, 30°$, and $35°$.
(*b*) Determine the angle $\theta$ where $F$ is minimum. Do it by creating a vector $\theta$ with elements ranging from $5°$ to $35°$ and spacing of 0.01. Calculate $F$ for each value of $\theta$ and then find the maximum $F$ and associated $\theta$ with MATLAB's built-in function `max`.

7. The remaining loan balance, $B$, of a fixed payment $n$ years mortgage after $x$ years is given by:
$$B = \frac{L\left[\left(1+\frac{r}{12}\right)^{12n} - \left(1+\frac{r}{12}\right)^{12x}\right]}{\left(1+\frac{r}{12}\right)^{12n} - 1}$$
where $L$ is the loan amount, and $r$ is the annual interest rate. Calculate the balance of a 30-year, \$100,000 mortgage, with annual interest rate of 6% (use 0.06 in the equation) after 0, 5, 10, 15, 20, 25, and 30 years. Create a seven-element vector for $x$ and use element-by-element operations. Display the results in a two-row table where the values of years and balance are displayed in the first and second rows, respectively.

8. The length $|\mathbf{u}|$ (magnitude) of a vector $\mathbf{u} = x\mathbf{i} + y\mathbf{j} - z\mathbf{k}$ is given by $|\mathbf{u}| = \sqrt{x^2 + y^2 + z^2}$. Given the vector $\mathbf{u} = -5.6\,\mathbf{i} + 11\,\mathbf{j} - 14\mathbf{k}$, determine its length by writing one MATLAB command in which the vector is multiplied by itself using element-by-element operation and the MATLAB built-in functions `sum` and `sqrt` are used.

9. A unit vector $\mathbf{u}_n$ in the direction of the vector $\mathbf{u} = x\mathbf{i} + y\mathbf{j} + z\mathbf{k}$ is given by $\mathbf{u}_n = \mathbf{u}/|\mathbf{u}|$ where $|\mathbf{u}|$ is the length (magnitude) of the vector, given by $|\mathbf{u}| = \sqrt{x^2 + y^2 + z^2}$. Given the vector $\mathbf{u} = 4\mathbf{i} + 13\mathbf{j} - 7\mathbf{k}$, determine the unit vector in the direction of $\mathbf{u}$ using the following steps:
(*a*) Assign the vector to a variable `u`.
(*b*) Using element-by-element operation and the MATLAB built-in functions `sum` and `sqrt` calculate the length of $\mathbf{u}$ and assign it to the variable `Lu`.
(*c*) Use the variables from parts (*a*) and (*b*) to calculate $\mathbf{u}_n$.
(*d*) Verify that the length of $\mathbf{u}_n$ is 1 using the same operations as in part (*b*).

10. The angle between two vectors $\mathbf{u}_1 = x_1\mathbf{i} + y_1\mathbf{j} + z_1\mathbf{k}$ and $\mathbf{u}_2 = x_2\mathbf{i} + y_2\mathbf{j} + z_2\mathbf{k}$ can be determined by $\cos\theta = \dfrac{x_1x_2 + y_1y_2 + z_1z_2}{|\mathbf{u}_1||\mathbf{u}_2|}$, where $|\mathbf{u}_i| = \sqrt{x_i^2 + y_i^2 + z_i^2}$. Given the vectors $\mathbf{u}_1 = 3.2\mathbf{i} - 6.8\mathbf{j} + 9\mathbf{k}$ and $\mathbf{u}_2 = -4\mathbf{i} + 2\mathbf{j} + 7\mathbf{k}$, determine the angle between them (in degrees) by writing one MATLAB command that uses element-by-element multiplication and the MATLAB built-in functions `acosd`, `sum`, and `sqrt`.

11. The following vector is defined in MATLAB:

```
d=[2 4 3]
```

By hand (pencil and paper) write what will be displayed if the following commands are executed by MATLAB. Check your answers by executing the commands with MATLAB.
(*a*) `d+d` (*b*) `d.^d` (*c*) `d.*d` (*d*) `d.^2`

12. The following two vectors are defined in MATLAB:

```
v=[3 -1 2] , u=[6 4 -3]
```

By hand (pencil and paper) write what will be displayed if the following commands are executed by MATLAB. Check your answers by executing the commands with MATLAB.
(*a*) `v.*u` (*b*) `v.^u` (*c*) `v*u'`

13. Define the vector $v = [1\ 3\ 5\ 7]$. Then use the vector in a mathematical expression to create the following vectors:
(*a*) $a = [3\ 9\ 15\ 21]$ (*b*) $b = [1\ 9\ 25\ 49]$

(*c*) $c = [1\ 1\ 1\ 1]$ (*d*) $d = [6\ \ 6\ \ 6\ \ 6]$

14. Define the vector $v = [5\ \ 4\ \ 3\ \ 2]$. Then use the vector in a mathematical expression to create the following vectors:

(*a*) $a = \left[\frac{1}{5+5}\ \ \frac{1}{4+4}\ \ \frac{1}{3+3}\ \ \frac{1}{2+2}\right]$ (*b*) $b = [5^5\ 4^4\ 3^3\ 2^2]$

(*c*) $c = \left[\frac{5}{\sqrt{5}}\ \ \frac{4}{\sqrt{4}}\ \ \frac{3}{\sqrt{3}}\ \ \frac{2}{\sqrt{2}}\right]$ (*d*) $d = \left[\frac{5^2}{5^5}\ \ \frac{4^2}{4^4}\ \ \frac{3^2}{3^3}\ \ \frac{2^2}{2^2}\right]$

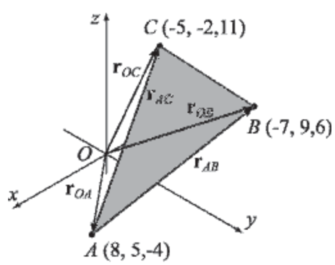
15. Define $x$ and $y$ as the vectors $x = [0.5,\ 1,\ 1.5,\ 2,\ 2.5]$ and $y = [0.8,\ 1.6,\ 2.4,\ 3.2,\ 4.0]$. Then use them in the following expressions to calculate $z$ using element-by-element calculations.

(*a*) $z = x^2 + 2xy$ (*b*) $z = xye^{y/x} - \sqrt[3]{x^4y^3 + 8.5}$

16. Define $r$ and $s$ as scalars $r = 1.6 \times 10^3$ and $s = 14.2$, and, $t$, $x$, and $y$ as vectors $t = [1,\ 2,\ 3,\ 4,\ 5]$, $x = [2,\ 4,\ 6,\ 8,\ 10]$, and $y = [3,\ 6,\ 9,\ 12,\ 15]$. Then use these variables to calculate the following expressions using element-by-element calculations for the vectors.

(*a*) $G = xt + \frac{r}{s^2}(y^2 - x)t$ (*b*) $R = \frac{r(-xt + yt^2)}{15} - s^2(y - 0.5x^2)t$

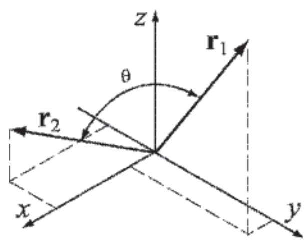
17. The area of a triangle $ABC$ can be calculated by $|\mathbf{r}_{AB} \times \mathbf{r}_{AC}|/2$, where $\mathbf{r}_{AB}$ and $\mathbf{r}_{AC}$ are vectors connecting the vertices $A$ and $B$, and $A$ and $C$, respectively. Determine the area of the triangle shown in the figure. Use the following steps in a script file to calculate the area. First, define the vectors $\mathbf{r}_{OA}$, $\mathbf{r}_{OB}$, and $\mathbf{r}_{OC}$ from knowing the coordinates of points $A$, $B$, and $C$. Then determine the vectors $\mathbf{r}_{AB}$ and $\mathbf{r}_{AC}$ from $\mathbf{r}_{OA}$, $\mathbf{r}_{OB}$, and $\mathbf{r}_{OC}$. Finally, determine the area by using MATLAB's built-in functions `cross`, `sum`, and `sqrt`.



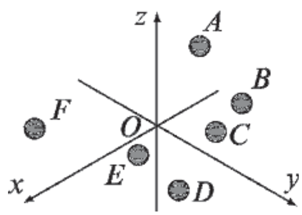

18. The cross product of two vectors can be used for determining the angle between two vectors:

$$\theta = \sin^{-1}\left(\frac{|\mathbf{r}_1 \times \mathbf{r}_2|}{|\mathbf{r}_1||\mathbf{r}_2|}\right)$$

Use MATLAB's built-in functions `asind`, `cross`, `sqrt`, and `dot` to find the angle (in degrees) between $\mathbf{r}_1 = 2.5\mathbf{i} + 8\mathbf{j} - 5\mathbf{k}$ and $\mathbf{r}_2 = -1\mathbf{i} + 6\mathbf{j} + 3\mathbf{k}$. Recall that $|\mathbf{r}| = \sqrt{\mathbf{r} \cdot \mathbf{r}}$.



19. The center of mass, $(\bar{x}, \bar{y}, \bar{z})$, of $n$ particles can be calculated by:




$$\bar{x} = \frac{\sum_{i=1}^{i=n} m_i x_i}{\sum_{i=1}^{i=n} m_i}, \quad \bar{y} = \frac{\sum_{i=1}^{i=n} m_i y_i}{\sum_{i=1}^{i=n} m_i}, \quad \bar{z} = \frac{\sum_{i=1}^{i=n} m_i z_i}{\sum_{i=1}^{i=n} m_i}$$

where $x_i$, $y_i$, and $z_i$ and $m_i$ are the coordinates and the mass of particle $i$, respectively. The coordinates and mass of six particles are listed in the following table. Calculate the center of mass of the particles.

| Particle | Mass (kg) | Coordinate $x$ (mm) | Coordinate $y$ (mm) | Coordinate $z$ (mm) |
|---|---|---|---|---|
| *A* | 0.5 | -10 | 8 | 32 |
| *B* | 0.8 | -18 | 6 | 19 |
| *C* | 0.2 | -7 | 11 | 2 |
| *D* | 1.1 | 5 | 12 | -9 |
| *E* | 0.4 | 0 | -8 | -6 |
| *F* | 0.9 | 25 | -20 | 8 |

20. Define the vectors:

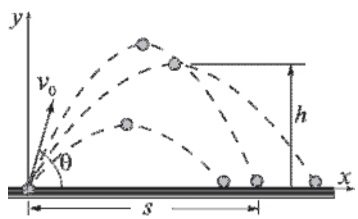
$$\mathbf{a} = 7\mathbf{i} - 4\mathbf{j} + 6\mathbf{k}, \quad \mathbf{b} = -4\mathbf{i} + 7\mathbf{j} + 5\mathbf{k}, \text{ and } \mathbf{c} = 5\mathbf{i} - 6\mathbf{j} + 8\mathbf{k}$$

Use the vectors to verify the identity:

$$\mathbf{a} \times (\mathbf{b} \times \mathbf{c}) = \mathbf{b}(\mathbf{a} \cdot \mathbf{c}) - \mathbf{c}(\mathbf{a} \cdot \mathbf{b})$$

Using MATLAB's built-in functions `cross` and `dot`, calculate the value of the left and right sides of the identity.

21. The maximum distance $s$ and the maximum height $h$ that a projectile shot at an angle $\theta$ are given by:




$$s = \frac{v_0^2}{g}\sin 2\theta \quad \text{and} \quad h = \frac{v_0^2 \sin^2\theta}{2g}$$

where $v_0$ is the shooting velocity and $g = 9.81$ m/s$^2$. Determine $s(\theta)$ and $h(\theta)$ for $\theta = 15°, 25°, 35°, 45°, 55°, 65°, 75°$ if $v_0 = 260$ m/s.

22. Use MATLAB to show that the sum of the infinite series $\sum_{n=1}^{\infty} \frac{1}{n^2}$ converges to $\pi^2/6$. Do this by computing the sum for:

(*a*) $n = 5$, (*b*) $n = 50$, (*c*) $n = 5000$

For each part create a vector $n$ in which the first element is 1, the increment is 1 and the last term is 5, 50, or 5,000. Then use element-by-element calculations to create a vector in which the elements are $\frac{1}{n^2}$. Finally, use MAT-

LAB's built-in function `sum` to sum the series. Compare the values to $\pi^2/6$. Use `format long` to display the numbers.

23. Use MATLAB to show that the sum of the infinite series $\sum_{n=1}^{\infty} \frac{n^2}{2^n}$ converges to 6. Do this by computing the sum for
(*a*) $n = 5$, (*b*) $n = 15$, (*c*) $n = 30$
For each part, create a vector `n` in which the first element is 1, the increment is 1 and the last term is 5, 15, or 30. Then use element-by-element calculations to create a vector in which the elements are $\frac{n^2}{2^n}$. Finally, use MATLAB's built-in function `sum` to sum the series. Use `format long` to display the numbers.

24. The natural exponential function can be expressed by $e^x = \sum_{n=0}^{\infty} \frac{x^n}{n!}$. Determine $e^2$ by calculating the sum of the series for:
(*a*) $n = 5$, (*b*) $n = 15$, (*c*) $n = 25$
For each part create a vector `n` in which the first element is 0, the increment is 1, and the last term is 5, 15, or 25. Then use element-by-element calculations to create a vector in which the elements are $\frac{x^n}{n!}$. Finally, use the MATLAB built-in function `sum` to add the terms of the series. Compare the values obtained in parts (*a*), (*b*), and (*c*) with the value of $e^2$ calculated by MATLAB.

25. Show that $\lim_{x \to \pi/3} \frac{\sin(x - \pi/3)}{4\cos^2 x - 1} = \frac{-\sqrt{3}}{6}$. Do this by first creating a vector $x$ that has the elements $\pi/3 - 0.1$, $\pi/3 - 0.01$, $\pi/3 - 0.0001$, $\pi/3 + 0.0001$, $\pi/3 + 0.01$, and $\pi/3 + 0.1$. Then, create a new vector $y$ in which each element is determined from the elements of $x$ by $\frac{\sin(x - \pi/3)}{4\cos^2 x - 1}$. Compare the elements of $y$ with the value $\frac{-\sqrt{3}}{6}$. Use `format long` to display the numbers.

26. Show that $\lim_{x \to 0} \frac{5\sin(6x)}{4x} = 7.5$. Do this by first creating a vector $x$ that has the elements 1.0, 0.1, 0.01, 0.001, and 0.0001. Then, create a new vector $y$ in which each element is determined from the elements of $x$ by $\frac{5\sin(6x)}{4x}$. Compare the elements of $y$ with the value 7.5. Use `format long` to display the numbers.

27. The Hazen Williams equation can be used to calculate the pressure drop, $P_d$ (psi/ft of pipe) in pipes due to friction:

$$P_d = 4.52Q^{1.85} / (C^{1.85} d^{4.87})$$

where $Q$ is the flow rate (gpm), $C$ is a design coefficient determined by the type of pipe, and $d$ is pipe diameter in inches. Consider a 3.5-in.-diameter steel pipe with $C = 120$. Calculate the pressure drop in a 1000-ft-long pipe for flow rates of 250, 300, 350, 400, and 450 gpm. To carry out the calculation, first create a five-element vector with the values of the flow rates (250, 300, ...). Then use the vector in the formula using element-by-element operations.

28. The monthly lease payment, $Pmt$, of a new car can be calculated by:

$$Pmt = \frac{\left[Pv - \dfrac{Fv}{(1+i/12)^N}\right]}{\dfrac{1 - \dfrac{1}{1+(1+i/12)^N}}{i/12}}$$

where $Pv$ and $Fv$ are the present value and the future value (at the end of the lease) of the car, respectively. $N$ is the duration of the lease in months, and $i$ is the interest rate per year. Consider a 36-months-lease of a car with a present value of \$38,000 and a future value of \$23,400. Calculate the monthly payments if the yearly interest rates are 3, 4, 5, 6, 7, and 8%. To carry out the calculation, first create a five-element vector with the values of the interest rates (0.03, 0.04, ...). Then use the vector in the formula using element-by-element operations.

29. Create the following three matrices:

$$A = \begin{bmatrix} 5 & -3 & 7 \\ 1 & 0 & -6 \\ -4 & 8 & 9 \end{bmatrix}, \quad B = \begin{bmatrix} 3 & 2 & -1 \\ 6 & 8 & -7 \\ 4 & 4 & 0 \end{bmatrix}, \quad C = \begin{bmatrix} -9 & 8 & 3 \\ 1 & 7 & -5 \\ 3 & 3 & 6 \end{bmatrix}$$

(*a*) Calculate $A + B$ and $B + A$ to show that addition of matrices is commutative.

(*b*) Calculate $A*(B*C)$ and $(A*B)*C$ to show that multiplication of matrices is associative.

(*c*) Calculate $5(B+C)$ and $5B+5C$ to show that, when matrices are multiplied by a scalar, the multiplication is distributive.

(*d*) Calculate $(A+B)*C$ and $A*C+B*C$ to show that matrix multiplication is distributive.

30. Use the matrices $A$, $B$, and $C$ from the previous problem to answer the following:

(*a*) Does $A*B = B*A$? (*b*) Does $(B*C)^{-1} = B^{-1}*C^{-1}$?

(*c*) Does $(A^{-1})^t = (A^t)^{-1}$? ($^t$ means transpose) (*d*) Does $(A + B)^t = A^t + B^t$?

31. Create a $3 \times 3$ matrix `A` having random integer values between 1 and 5. Call the matrix $A$ and, using MATLAB, perform the following operations. For each part explain the operation.

(*a*) `A.^A` (*b*) `A.*A` (*c*) `A*A-1`

(*d*) `A./A` (*e*) `det(A)` (*f*) `inv(A)`

32. The magic square is an arrangement of numbers in a square grid in such a way that the sum of the numbers in each row, and in each column, and in each diagonal is the same. MATLAB has a built-in function `magic(n)` that returns an $n \times n$ magic square. In a script file create a $(5 \times 5)$ magic square, and then test the properties of the resulting matrix by finding the sum of the elements in each row, in each column and in both diagonals. In each case, use MATLAB's built-in function `sum`. (Other functions that can be useful are `diag` and `fliplr`.)

33. Solve the following system of three linear equations:

$$\begin{aligned} -2x + 5y + 7z &= -17.5 \\ 3x - 6y + 2z &= 40.6 \\ 9x - 3y + 8z &= 56.2 \end{aligned}$$

34. Solve the following system of six linear equations:

$$\begin{aligned} 2a - 4b + 5c - 3.5d + 1.8e + 4f &= 52.52 \\ -1.5a + 3b + 4c - d - 2e + 5f &= -21.1 \\ 5a + b - 6c + 3d - 2e + 2f &= -27.6 \\ 1.2a - 2b + 3c + 4d - e + 4f &= 9.16 \\ 4a + b - 2c - 3d - 4e + 1.5f &= -17.9 \\ 3a + b - c + 4d - 2e - 4f &= -16.2 \end{aligned}$$

35. A football stadium has 100,000 seats. In a game with full capacity people with the following ticket and associated cost attended the game:

| | Student | Alumni | Faculty | Public | Veterans | Guests |
|---|---|---|---|---|---|---|
| Cost | \$25 | \$40 | \$60 | \$70 | \$32 | \$0 |

Determine the number of people that attended the game in each cost category if the total revenue was \$4,897,000, there were 11,000 more alumni than faculty, the number of public plus alumni together was 10 times the number of veterans, the number of faculty plus alumni together was the same as the number of students, and the number of faculty plus students together was four times larger than the number of guests and veterans together.

36. A food company manufactures five types of 8-oz trail mix packages using different mixtures of peanuts, almonds, walnuts, raisins, and M&Ms. The mixtures have the following compositions:

| | Peanuts (oz) | Almonds (oz) | Walnuts (oz) | Raisins (oz) | M&Ms (oz) |
|---|---|---|---|---|---|
| Mix 1 | 3 | 1 | 1 | 2 | 1 |
| Mix 2 | 1 | 2 | 1 | 3 | 1 |
| Mix 3 | 1 | 1 | 0 | 3 | 3 |
| Mix 4 | 2 | 0 | 3 | 1 | 2 |
| Mix 5 | 1 | 2 | 3 | 0 | 2 |

How many packages of each mix can be manufactured if 105 lb of peanuts, 74 lb of almonds, 102 lb of walnuts, 118 lb of raisins, and 121 lb of M&Ms are available? Write a system of linear equations and solve.

37. The electrical circuit shown consists of resistors and voltage sources. Determine $i_1, i_2, i_3$ and $i_4$, using the mesh current method based on Kirchhoff's voltage law (see Sample Problem 3-4).

$V_1 = 28$ V, $V_2 = 36$ V, $V_3 = 42$ V
$R_1 = 16\,\Omega$, $R_2 = 10\,\Omega$, $R_3 = 6\,\Omega$
$R_4 = 12\,\Omega$, $R_5 = 8\,\Omega$, $R_6 = 14\,\Omega$
$R_7 = 4\,\Omega$, $R_8 = 5\,\Omega$.