Adding elements to a vector:

Elements can be added to an existing vector by assigning values to the new elements. For example, if a vector has 4 elements, the vector can be made longer by assigning values to elements 5, 6, and so on. If a vector has n elements and a new value is assigned to an element with address of $n+2$ or larger, MATLAB assigns zeros to the elements that are between the last original element and the new element. Examples:

```
>> DF=1:4                      Define vector DF with 4 elements.
DF =
     1     2     3     4
>> DF(5:10)=10:5:35            Adding 6 elements starting with the 5th.
DF =
     1     2     3     4    10    15    20    25    30    35
>> AD=[5  7  2]                Define vector AD with 3 elements.
AD =
     5     7     2
>> AD(8)=4                     Assign a value to the 8th element.
AD =                           MATLAB assigns zeros to
     5   7   2   0   0   0   0   4    the 4th through 7th elements.
>> AR(5)=24                    Assign a value to the 5th element of a new vector.
AR =                           MATLAB assigns zeros to the
     0     0     0     0    24       1st through 4th elements.
>>
```

Adding elements to a matrix:

Rows and/or columns can be added to an existing matrix by assigning values to the new rows or columns. This can be done by assigning new values, or by appending existing variables. This must be done carefully since the size of the added rows or columns must fit the existing matrix. Examples are:

```
>> E=[1 2 3 4; 5 6 7 8]
E =
     1     2     3     4
     5     6     7     8
>> E(3,:)=[10:4:22]
E =
     1     2     3     4
     5     6     7     8
    10    14    18    22
>> K=eye(3)
K =
     1     0     0
     0     1     0
     0     0     1
>> G=[E  K]
G =
     1     2     3     4     1     0     0
     5     6     7     8     0     1     0
    10    14    18    22     0     0     1
```

Define a 2×4 matrix E.

Add the vector 10 14 18 22 as the third row of E.

Define 3×3 matrix K.

Append the matrix K to matrix E. The number of rows in E and K must be the same.

```
>> AW=[3 6 9; 8 5 11]
AW =
     3     6     9
     8     5    11
>> AW(4,5)=17
AW =
     3     6     9     0     0
     8     5    11     0     0
     0     0     0     0     0
     0     0     0     0    17
>> BG(3,4)=15
BG =
     0     0     0     0
     0     0     0     0
     0     0     0    15
>>
```

Define a 2×3 matrix.

Assign a value to the (4,5) element.

MATLAB changes the matrix size to 4×5, and assigns zeros to the new elements.

Assign a value to the (3,4) element of a new matrix.

MATLAB creates a 3×4 matrix and assigns zeros to all the elements except BG(3,4).

2.8 DELETING ELEMENTS

An element, or a range of elements, of an existing variable can be deleted by reassigning nothing to these elements. This is done by using square brackets with nothing typed in between them. By deleting elements a vector can be made shorter and a matrix can be made to have a smaller size. Examples are:

```
>> kt=[2 8 40 65 3 55 23 15 75 80]          Define a vector
                                            with 10 elements.
kt =
     2     8    40    65     3    55    23    15    75    80
>> kt(6)=[]                  Eliminate the sixth element.
kt =
     2    8   40   65    3   23   15   75   80     The vector now
                                                   has 9 elements.
>> kt(3:6)=[]                Eliminate elements 3 through 6.
kt =
     2     8    15    75    80       The vector now has 5 elements.
>> mtr=[5 78 4 24 9; 4 0 36 60 12; 56 13 5 89 3]
mtr =
      5     78      4     24      9         Define a 3 × 5 matrix.
      4      0     36     60     12
     56     13      5     89      3
>> mtr(:,2:4)=[]             Eliminate all the rows of
                             columns 2 through 4.
```

3.4 ELEMENT-BY-ELEMENT OPERATIONS

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

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 gives:

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

Element-by-element exponentiation of matrix A gives:

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

```
>> A=[2 6 3; 5 8 4]
A =
     2     6     3
     5     8     4
>> B=[1 4 10; 3 2 7]
B =
     1     4    10
     3     2     7
>> A.*B
ans =
     2    24    30
    15    16    28
>> C=A./B
C =
    2.0000    1.5000    0.3000
    1.6667    4.0000    0.5714
```

Define a 2×3 array A.

Define a 2×3 array B.

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

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

```
>> B.^3

ans =
     1    64   1000
    27     8    343

>> A*B

??? Error using ==> *
Inner matrix dimensions must a
```

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.

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.)

Ex:

$$y = \frac{z^3 + 5z}{4z^2 - 10}.$$

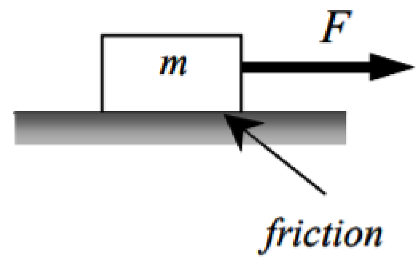
```
>> z=[1:2:11]
z =
    1    3    5    7    9   11
>> y=(z.^3 + 5*z)./(4*z.^2 - 10)

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

Create a vector z with eight elements.

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

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];                 Enter the values of m in a vector.
>> F=[12.5 23.5 30 61 117 294];        Enter the values of F in a vector.
>> mu=F./(m*9.81)
                                        A value for mu is calculated for each test,
mu =                                    using element-by-element calculations.

   0.6371    0.5989    0.6116    0.6218    0.5963    0.5994

>> mu_ave=mean(mu)
                                        The average of the elements in the vector mu
mu_ave =                                is determined by using the function mean.
    0.6109
```

Polynomials

$$f(x) = a_n x^n + a_{n-1} x^{n-1} + \ldots + a_1 x + a_0$$

Polynomial	**MATLAB representation**
$8x + 5$	p = [8 5]
$2x^2 - 4x + 10$	d = [2 –4 10]
$6x^2 - 150$, MATLAB form: $6x^2 + 0x - 150$	h = [6 0 –150]
$5x^5 + 6x^2 - 7x$, MATLAB form: $5x^5 + 0x^4 + 0x^3 + 6x^2 - 7x + 0$	c = [5 0 0 6 –7 0]

8.1.1 Value of a Polynomial

The value of a polynomial at a point x can be calculated with the function `polyval` that has the form:

```
polyval(p,x)
```

`p` is a vector with the coefficients of the polynomial.

`x` is a number, or a variable that has an assigned value, or a computable expression.

Sample Problem 8-1: Calculating polynomials with MATLAB

For the polynomial: $f(x) = x^5 - 12.1x^4 + 40.59x^3 - 17.015x^2 - 71.95x + 35.88$

a) Calculate $f(9)$.

b) Plot the polynomial for $-1.5 \le x \le 6.7$.

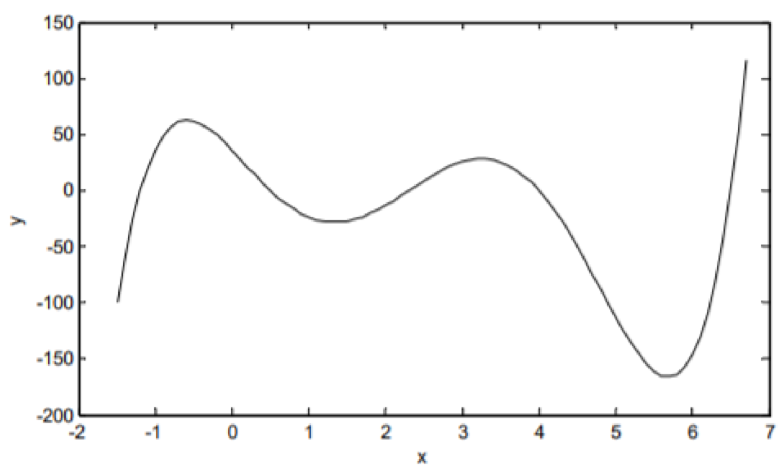
```
>> p = [1 -12.1 40.59 -17.015 -71.95 35.88];
>> polyval(p,9)
ans =
  7.2611e+003
```

b) To plot the polynomial, a vector x is first defined with elements ranging from –1.5 to 6.7. Then a vector y is created with the values of the polynomial for every element of x. Finally, a plot of y vs. x is made.

```
>> x=-1.5:0.1:6.7;
>> y=polyval(p,x);
>> plot(x,y)
```

Calculating the value of the polynomial for each element of the vector x.

The plot created by MATLAB is (axes labels were added with the Plot Editor):



Roots of a Polynomial

```
r = roots(p)
```

`r` is a column vector with the roots of the polynomial.

`p` is a row vector with the coefficients of the polynomial.

```
>> p= 1 -12.1 40.59 -17.015 -71.95 35.88];
>> r=roots(p)
r =
    6.5000
    4.0000
    2.3000
   -1.2000
    0.5000
```

When the roots are known, the polynomial can actually be written as:
$f(x) = (x+1.2)(x-0.5)(x-2.3)(x-4)(x-6.5)$

8.1.3 Addition, Multiplication, and Division of Polynomials

$$f_1(x) = 3x^6 + 15x^5 - 10x^3 - 3x^2 + 15x - 40$$

$$f_2(x) = 3x^3 - 2x - 6$$

Addition:

```
>> p1=[3 15 0 -10 -3 15 -40];
>> p2=[3 0 -2 -6];
>> p=p1+[0 0 0 p2]
p =
     3    15     0    -7    -3    13   -46
```

Three 0's are added in front of `p2`, since the order of `p1` is 6 and the order of `p2` is 3.

Multiplication:

Two polynomials can be multiplied with the MATLAB built-in function `conv` which has the form:

```
c = conv(a,b)
```

`c` is a vector of the coefficients of the polynomial that is the product of the multiplication.

`a` and `b` are the vectors of the coefficients of the polynomials that are being multiplied.

For example, multiplication of the polynomials $f_1(x)$ and $f_2(x)$ above gives:

```
>> pm=conv(p1,p2)
pm =
     9    45    -6   -78   -99    65   -54   -12   -10   240
```

which means that the answer is:

$9x^9 + 45x^8 - 6x^7 - 78x^6 - 99x^5 + 65x^4 - 54x^3 - 12x^2 - 10x + 240$

Division:

A polynomial can be divided by another polynomial with the MATLAB built-in function `deconv` which has the form:

```
[q,r] = deconv(u,v)
```

`q` is a vector with the coefficients of the quotient polynomial.
`r` is a vector with the coefficients of the remainder polynomial.

`u` is a vector with the coefficients of the numerator polynomial.
`v` is a vector with the coefficients of the denominator polynomial.

An example of division that gives a remainder is $2x^6 - 13x^5 + 75x^3 + 2x^2 - 60$ divided by $x^2 - 5$:

```
>> w=[2 -13 0 75 2 0 -60];
>> z=[1 0 -5];
>> [g h]=deconv(w,z)
g =
    2   -13    10    10    52
h =
    0     0     0     0     0    50    200
```

The quotient is: $2x^4 - 13x^3 + 10x^2 + 10x + 52$.

The remainder is: $50x + 200$.

The answer is: $2x^4 - 13x^3 + 10x^2 + 10x + 52 + \dfrac{50x + 200}{x^2 - 5}$.

Derivatives of Polynomials

For example, if $f_1(x) = 3x^2 - 2x + 4$, and $f_2(x) = x^2 + 5$, the derivatives of $3x^2 - 2x + 4$, $(3x^2 - 2x + 4)(x^2 + 5)$, and $\frac{3x^2 - 2x + 4}{x^2 + 5}$ can be determined by:

```
>> f1= 3 -2 4];
>> f2=[1 0 5];
```

Creating the vectors coefficients of f_1 and f_2.

```
>> k=polyder(f1)
k =
     6    -2
```

The derivative of f_1 is: $6x - 2$.

```
>> d=polyder(f1,f2)
d =
    12    -6    38   -10
```

The derivative of f_1*f_2 is: $12x^3 - 6x^2 + 38x - 10$.

```
>> [n d]=polyder(f1,f2)
n =
     2    22   -10
d =
     1     0    10     0    25
```

The derivative of $\frac{3x^2 - 2x + 4}{x^2 + 5}$ is: $\frac{2x^2 + 22x - 10}{x^4 + 10x^2 + 25}$.