```
%==========================================================
sec. 4.1 some common used build-in functions
% help --> MATLAB  --> mathematics --> elementary function
%==========================================================
Some commonly used elementary function
```

Exponential	
`exp(x)`	Exponential; e^x.
`sqrt(x)`	Square root; $\sqrt{x}$.
Logarithmic	
`log(x)`	Natural logarithm; $\ln x$.
`log10(x)`	Common (base-10) logarithm; $\log x = \log_{10} x$
Complex	
`abs(x)`	Absolute value; x.
`angle(x)`	Angle of a complex number x.
`conj(x)`	Complex conjugate.
`imag(x)`	Imaginary part of a complex number x.
`real(x)`	Real part of a complex number x.
Numeric	
`ceil(x)`	Round to the nearest integer toward ∞.
`fix(x)`	Round to the nearest integer toward zero.
`floor(x)`	Round to the nearest integer toward $-\infty$.
`round(x)`	Round toward the nearest integer.
`sign(x)`	Signum function: $+1$ if $x > 0$; 0 if $x = 0$; -1 if $x < 0$.

```
x = -1:.1:1;
figure, plot(x,abs(x),'o')
```

Trigonometric functions

Trigonometric*	
`cos(x)`	Cosine; $\cos x$.
`cot(x)`	Cotangent; $\cot x$.
`csc(x)`	Cosecant; $\csc x$.
`sec(x)`	Secant; $\sec x$.
`sin(x)`	Sine; $\sin x$.
`tan(x)`	Tangent; $\tan x$.
Inverse trigonometric†	
`acos(x)`	Inverse cosine; $\arccos x = \cos^{-1} x$.
`acot(x)`	Inverse cotangent; $\text{arccot}\, x = \cot^{-1} x$.
`acsc(x)`	Inverse cosecant; $\text{arccsc}\, x = \csc^{-1} x$.
`asec(x)`	Inverse secant; $\text{arcsec}\, x = \sec^{-1} x$.
`asin(x)`	Inverse sine; $\arcsin x = \sin^{-1} x$.
`atan(x)`	Inverse tangent; $\arctan x = \tan^{-1} x$.
`atan2(y,x)`	Four-quadrant inverse tangent.

*These functions accept x in radians.
†These functions return a value in radians.

Hyperbolic	
`cosh(x)`	Hyperbolic cosine; $\cosh x = (e^x + e^{-x})/2$.
`coth(x)`	Hyperbolic cotangent; $\cosh x/\sinh x$.
`csch(x)`	Hyperbolic cosecant; $1/\sinh x$.
`sech(x)`	Hyperbolic secant; $1/\cosh x$.
`sinh(x)`	Hyperbolic sine; $\sinh x = (e^x - e^{-x})/2$.
`tanh(x)`	Hyperbolic tangent; $\sinh x/\cosh x$.
Inverse hyperbolic	
`acosh(x)`	Inverse hyperbolic cosine
`acoth(x)`	Inverse hyperbolic cotangent
`acsch(x)`	Inverse hyperbolic cosecant
`asech(x)`	Inverse hyperbolic secant
`asinh(x)`	Inverse hyperbolic sine
`atanh(x)`	Inverse hyperbolic tangent

```
%% Complex number :
(1)   Definition of a complex number:
(2)   Representation with Cartesian coordinate and polar
      coordinate
(3)   Translation between these two representation.
% atan2(Y,X) is the four quadrant arctangent of the
elements
%   of X and Y. -pi <= atan2(Y,X) <= pi.
clear all;
x = [1 -1 -1 1];
y = [1 1 -1 -1];
z=x+1i.*y;
z2=x(2)+1i*y(2);

mag_z2=abs(z2);
ang_z2=atan2(y(2),x(2)) * 180/pi;

x2=mag_z2.*cosd(ang_z2);
y2=mag_z2.*sind(ang_z2);

% in radian
x2=abs(z).*cos(atan2(y,x));
y2=abs(z).*sin(atan2(y,x));

%% in radians
x2=abs(z).*cos(atan2(y,x));
y2=abs(z).*sin(atan2(y,x));
```

```
%% Exercise
```

Test Your Understanding

T3.1–3 For several values of x, con rm that $e^{ix} = \cos x + i \sin x$.

T3.1–4 For several values of x in the range $0 \le x \le 2\pi$, con rm that $\sin^{-1} x + \cos^{-1} x = \pi/2$.

```
Use (a) z1=x1+1i*y1 ; with x1=1 and y1=-2 to translate
between Cartesian coordinate and polar coordinate.
(b) add and multiple two complex numbers Z1 and Z2 in
polar coordinate
Where z2=x2+1i*y2 ; with x2=1 and y2=-1

t = clock; % given a variable of current time
fprintf( ' %02.0f:%02.0f:%02.0f\n', t(4), t(5), t(6) );

x=1:2:7;
cumsum(x')

cumsum(1:4)

date
realmax  % largest positive floating number on your
computer
realmin

rem(19, 5)
 % The following statements convert 40 inches this way by
using fix and rem commands:
feet = fix(40/12)
inches = rem(40, 12)
fprintf( ' %d feet %d inches \n', feet,inches );

% Exercise 4.6
```

%% Some useful matlab functions for the functional evalution

% % polynomial function & its integration: appendix A

See appendix A for the

polyval

Polynomial evaluation

Syntax

```
y = polyval(p,x)
[y,delta] = polyval(p,x,S)
y = polyval(p,x,[],mu)
[y,delta] = polyval(p,x,S,mu)
```

Description

`y = polyval(p,x)` evaluates the polynomial `p` at each point in `x`. The argument `p` is a vector of length `n+1` whose elements are the coefficients (in descending powers) of an nth-degree polynomial: example

$$p(x) = p_1 x^n + p_2 x^{n-1} + \ldots + p_n x + p_{n+1}.$$

The polynomial coefficients in `p` can be calculated for different purposes by functions like `polyint`, `polyder`, and `polyfit`, but you can specify any vector for the coefficients.

To evaluate a polynomial in a matrix sense, use `polyvalm` instead.

`[y,delta] = polyval(p,x,S)` uses the optional output structure `S` produced by `polyfit` to generate error estimates. `delta` is an estimate of the standard error in predicting a future observation at `x` by `p(x)`. example

`y = polyval(p,x,[],mu)` or `[y,delta] = polyval(p,x,S,mu)` use the optional output `mu` produced by `polyfit` to center and scale the data. `mu(1)` is `mean(x)`, and `mu(2)` is `std(x)`. Using these values, `polyval` centers `x` at zero and scales it to have unit standard deviation, example

$$\hat{x} = \frac{x - \bar{x}}{\sigma_x}.$$

This centering and scaling transformation improves the numerical properties of the polynomial.

Examples

collapse all

Evaluate Polynomial at Several Points

Evaluate the polynomial $p(x) = 3x^2 + 2x + 1$ at the points $x = 5, 7, 9$. The polynomial coefficients can be represented by the vector `[3 2 1]`.

Open Live Script

```
p = [3 2 1];
x = [5 7 9];
y = polyval(p,x)
```

```
y = 1×3

    86   162   262
```

Integrate Quartic Polynomial

Evaluate the definite integral

Open Live Script

$$I = \int_{-1}^{3} (3x^4 - 4x^2 + 10x - 25)dx.$$

Create a vector to represent the polynomial integrand $3x^4 - 4x^2 + 10x - 25$. The x^3 term is absent and thus has a coefficient of 0.

```
p = [3 0 -4 10 -25];
```

Use `polyint` to integrate the polynomial using a constant of integration equal to `0`.

```
q = polyint(p)
```

```
q = 1×6

    0.6000         0   -1.3333    5.0000  -25.0000         0
```

Find the value of the integral by evaluating `q` at the limits of integration.

```
a = -1;
b = 3;
I = diff(polyval(q,[a b]))
```

```
I = 49.0667
```

Linear Regression With Error Estimate

Fit a linear model to a set of data points and plot the results, including an estimate of a 95% prediction interval.

Open Live Script

Create a few vectors of sample data points *(x,y)*. Use `polyfit` to fit a first degree polynomial to the data. Specify two outputs to return the coefficients for the linear fit as well as the error estimation structure.

```
x = 1:100;
y = -0.3*x + 2*randn(1,100);
[p,S] = polyfit(x,y,1);
```

Evaluate the first-degree polynomial fit in `p` at the points in `x`. Specify the error estimation structure as the third input so that `polyval` calculates an estimate of the standard error. The standard error estimate is returned in `delta`.

```
[y_fit,delta] = polyval(p,x,S);
```

Plot the original data, linear fit, and 95% prediction interval $y \pm 2\Delta$.

```
plot(x,y,'bo')
hold on
plot(x,y_fit,'r-')
plot(x,y_fit+2*delta,'m--',x,y_fit-2*delta,'m--')
title('Linear Fit of Data with 95% Prediction Interval')
legend('Data','Linear Fit','95% Prediction Interval')
```

%% minimum values and roots of the function

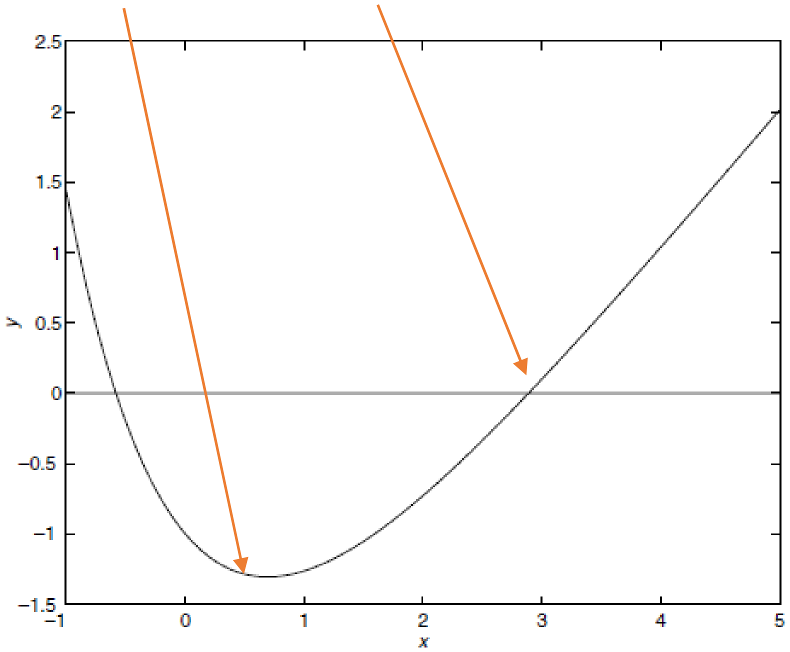
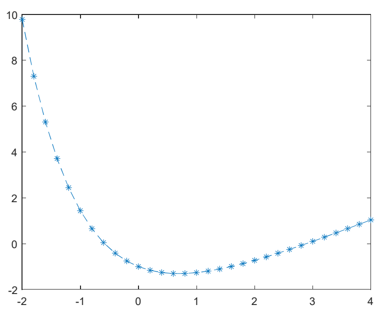


Figure 3.2–1 Plot of the function $y = x + 2e^{-x} - 3$.

Table 3.2–1 Minimization and root- nding functions

Function	Description
`fminbnd(@function,x1,x2)`	Returns a value of `x` in the interval `x1` ≤ `x` ≤ `x2` that corresponds to a minimum of the single-variable function described by the handle `@function`.
`fminsearch(@function,x0)`	Uses the starting vector `x0` to nd a minimum of the multivariable function described by the handle `@function`.
`fzero(@function,x0)`	Uses the starting value `x0` to nd a zero of the single-variable function described by the handle `@function`.

```
%% function handle
hy =@(x) x + 2*exp(-x) - 3;
x=-2:0.2:4;
% plot function
figure;plot(x,hy(x),'--*')
% minimum & root
x_min= fminbnd(hy, 0, 4);
x_root=fzero(hy, 0);
 %% the polynomial
hy2=@(x) (0.025.*x.^5-0.0625.*x.^4-0.333.*x.^3+x.^2);
x=-4:0.2:4;
```



```
figure;plot(x,hy2(x),'--*')
x_min= fminbnd(hy2, -1, 4);
x2=fzero(hy2,[-4 -1]);
```

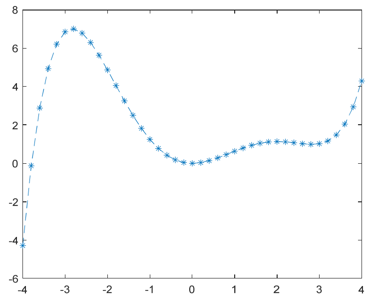


Figure 3.2–3 shows the cross section of an irrigation channel. A preliminary analysis has shown that the cross-sectional area of the channel should be 100 ft^2 to carry the desired water ow rate. To minimize the cost of concrete used to line the channel, we want to minimize the length of the channel's perimeter. Find the values of d, b, and θ that minimize this length.

■ Solution

The perimeter length L can be written in terms of the base b, depth d, and angle θ as follows:

$$L = b + \frac{2d}{\sin\theta}$$

The area of the trapezoidal cross section is

$$100 = db + \frac{d^2}{\tan\theta}$$

```
Lh=@(x) 100./x(1) - x(1)./tan(x(2)) +
2*x(1)./sin(x(2));
x0=[20,1];
x1 = fminsearch (Lh,x0,options);
% notes that x1 is a local minimum
```

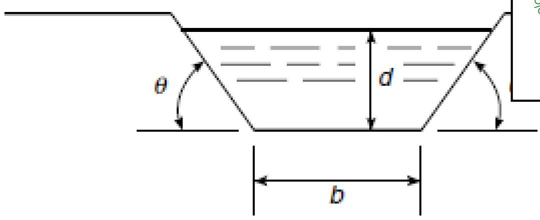


Figure 3.2–3 Cross section of an irrigation channel.

The variables to be selected are b, d, and θ. We can reduce the number of variables by solving the latter equation for b to obtain

$$b = \frac{1}{d}\left(100 - \frac{d^2}{\tan\theta}\right)$$

Substitute this expression into the equation for L. The result is

$$L = \frac{100}{d} - \frac{d}{\tan\theta} + \frac{2d}{\sin\theta}$$

We must now nd the values of d and θ to minimize L.

First de ne the function le for the perimeter length. Let the vector $\mathbf{x}$ be $[d\ \theta]$.

```
function L = channel(x)
L = 100./x(1) - x(1)./tan(x(2)) + 2*x(1)./sin(x(2));
```

Then use the `fminsearch` function. Using a guess of $d = 20$ and $\theta = 1$ rad, the session is

```
>>x = fminsearch(@channel,[20,1])
x =
   7.5984    1.0472
```

Thus the minimum perimeter length is obtained with $d = 7.5984$ ft and $\theta = 1.0472$ rad, or $\theta = 60°$. Using a different guess, $d = 1$, $\theta = 0.1$, produces the same answer. The value of the base b corresponding to these values is $b = 8.7738$.

However, using the guess $d = 20$, $\theta = 0.1$ produces the physically meaningless result $d = -781$, $\theta = 3.1416$. The guess $d = 1$, $\theta = 1.5$ produces the physically meaningless result $d = 3.6058$, $\theta = -3.1416$.

Exercise 1 :

Using estimates of rainfall, evaporation, and water consumption, the town engineer developed the following model of the water volume in the reservoir as a function of time

$$V(t) = 10^9 + 10^8(1 - e^{-t/100}) - rt$$

where V is the water volume in liters, t is time in days, and r is the town's consumption rate in liters per day. Write two user-de ned functions. The rst function should de ne the function $V(t)$ for use with the `fzero` function. The second function should use `fzero` to compute how long it will take for the water volume to decrease to x percent of its initial value of 10^9 L. The inputs to the second function should be x and r. Test your functions for the case where $x = 50$ percent and $r = 10^7$ L/day.

Exercise 2 :

The volume V and paper surface area A of a conical paper cup are given by

$$V = \frac{1}{3}\pi r^2 h \qquad A = \pi r\sqrt{r^2 + h^2}$$

where r is the radius of the base of the cone and h is the height of the cone.

a. By eliminating h, obtain the expression for A as a function of r and V.

b. Create a user-de ned function that accepts R as the only argument and computes A for a given value of V. Declare V to be global within the function.

c. For $V = 10$ in.^{3}, use the function with the `fminbnd` function to compute the value of r that minimizes the area A. What is the corresponding value of the height h? Investigate the sensitivity of the solution by plotting V versus r. How much can R vary about its optimal value before the area increases 10 percent above its minimum value?