Chapter 2 Basic Features

2.1 Simple Math

The MATLAB Command window is the primary means for interacting with MATLAB. When the Command window is active, statements are entered which can be executed immediately. For example, simple mathematical calculations can be performed and the results displayed. The results of each calculation are stored in the default MATLAB variable named 'ans' which retains its value until its changed.

Example 2.1.1

```
100+5*12
100/5 + 6^2
2*ans
2*(25+(3*exp(1)))
```

```
ans =  160
ans =   56
ans =  112
ans =   66.3097
```

Alternatively, the results can be assigned to user defined MATLAB variables.

Example 2.1.2

```
t_initial=4;
t_final=12;
t_elapsed=t_final-t_initial
rate=5
distance=rate*(t_elapsed)
```

```
t_elapsed = 8
rate = 5
distance = 40
```

Note the semicolon suppresses displaying the result of executing a MATLAB command.

2.2 The MATLAB Workspace

During a MATLAB session, the issued commands and the variables resulting from those commands are saved in the MATLAB workspace. The numerical values asssigned to variables are stored there and easily accessed by entering the variables' names at the MATLAB prompt.

Example 2.2.1

```
l=3;
w=12;
A=l*w;
A
l,w,A
```

```
A =  36
l =   3
w =  12
A =  36
```

The '`who`' command results in a list of the current MATLAB workspace variables.

Example 2.2.3

```
who
```

```
Your variables are:

A           distance    rate        t_final     w
ans         l           t_elapsed   t_initial
```

2.3 About Variables

MATLAB variable names must be a single word, no spaces, no punctuation characters, start with a letter and no more than 31 recognized characters. Several names have special meaning and should not be used for other purposes. For example,

'ans',	Default variable name used for results
'pi',	The value of π
'i' or 'j',	$\sqrt{-1}$
'eps',	Smallest number that can be added to one resulting in a number larger than one in the computer
'inf',	∞
'NaN',	Not a number, e.g. 0/0

Example 2.3.1

```
radius_of_circle_5 = 4
Area=pi*radius_of_circle_5^2

radius_of_circle_5 = 4
Area = 50.2655

eps

ans =  2.2204e-016

x=3+2*j
y=4-5i
z=x+y

x =  3.0000 + 2.0000i
y =  4.0000 - 5.0000i
z =  7.0000 - 3.0000i

a=1
b=2
c=b/(b-2*a)
d=(a*b-2) /(b-2*a)

a = 1
b = 2

Warning: Divide by zero.
c = Inf

Warning: Divide by zero.
d = NaN
```

2.4 Comments, Punctuation, and Aborting Execution

MATLAB ignores all text following a % symbol to allow commenting.

Example 2.4.1

```
int_nominal=0.08; % Loan interest rate (per year)
int_monthly=int_nominal/12; % Loan interest rate (per month)
P=270000; % Loan Amount
n_years=30; % Duration of loan (in years)
n_months=n_years*12; % Duration of loan (in months)
x=(1+int_monthly)^n_months;
A=P*(int_monthly*x)/(x-1) %  Monthly installment to repay loan
```

```
A = 1.9812e+003
```

More than one command can appear on the same line if they are separated by a comma or a semicolon.

Example 2.4.2

```
l=3; w=4; d=5;
V=l*w*d
```

```
V = 60
```

```
r=2, h=5  % Cone radius and height
V=(1/3)*pi*r^2*h  % Volume of Cone
```

```
r =  2
h =  5
V = 20.9440
```

To improve readability, long expressions or commands can be entered on multiple lines using an elipsis (...). The break cannot appear in the middle of a variable name. Comments cannot be continued, i.e. a new % is required on the next line to continue a comment.

Example 2.4.3

```
x=0.5;
sum=1/(1-x) % Computes the sum of the infinite geometric series
            % 1 + x + x$^2$ + x$^3$ + x$^4$ + x$^5$ + ......
```

```
sum = 2
```

2.5 Complex Numbers

Arithmetic operations involving complex numbers are easily handled by MATLAB. Either '`i`' or '`j`' can be used to represent the square root of minus one.

Example 2.5.1

```
Re_z=3; Im_z=4;
z1=Re_z + j*Im_z % z1 is a complex number
z2=Re_z - j*Im_z  % z2 is the complex conjugate of z1
u=sqrt(z1*z2) % Find the magnitude of z1 and its conjugate z2
v=z1/z2  %  Complex number v is the quotient of z1 and z2
w=z2*v % Complex number w is the product of z2 and v (restores z1)
y=u*(v+w)
```

```
z1 =  3.0000 + 4.0000i
z2 =  3.0000 - 4.0000i
u =   5
v =  -0.2800 + 0.9600i
w =   3.0000 + 4.0000i
y =  13.6000 +24.8000i
```

The multiplication (*) is not required when '`i`' or '`j`' is preceded by a number instead of a variable expression.

The following example illustrates converting a complex number $x+iy$ in rectangular form to polar form Re^{jq} where $R=\sqrt{x^2+y^2}$ and $q=\tan^{-1}(y/x)$ using the MATLAB functions '`abs`', and '`angle`'.

Example 2.5.2

```
x=3; % Real part of complex number
y=4; % Imaginary part of complex number
z=x+y*i  % Complex number in rectangular form
R=abs(z) % Computes the magnitude of z
theta=angle(z) % Computes the angle(rad) of z
theta_deg=180*theta/pi;  % Converts theta to degrees
```

```
z = 3.0000 + 4.0000i
R = 5
theta =  0.9273
```

In the opposite direction, i.e. converting from polar form Re^{jq} to rectangular form $x+iy$ where $x=R\cos(q)$ and $y=R\sin(q)$, the MATLAB functions '`real`' and '`imag`' are used as shown in Example 2.5.3.

Example 2.5.3

```
R=2  % Magnitude of complex number
theta=pi/4  % Angle of complex number
z=R*exp(j*theta); % Complex number in polar form
x=real(z) % Find the real part of z
y=imag(z) % Find the imaginary part of z
```

```
R = 2
theta = 0.7854
x = 1.4142
y = 1.4142
```

The function 'abs' returns the absolute value of its argument when the argument is a real number.

2.6 Mathematical Functions

MATLAB supports a variety of built-in mathematical functions. A list of mathematical functions grouped into several categories, along with other MATLAB information, appears in a Help Window by clicking on 'Help' from the menubar of the MATLAB Command window. Alternatively, the command '`helpwin`' can be typed on the command line at the MATLAB prompt. The categories of mathematical functions are:

```
matlab/elmat   - Elementary matrices and matrix manipulation
matlab/elfun   - Elementary math functions
matlab/specfun - Specialized math functions
matlab/datafun - Data analysis and Fourier transforms
matlab/polyfun - Interpolation and polynomials
matlab/funfun  - Function functions and ode solvers
matlab/sparfun - Sparse matrices
```

A complete listing of all the functions in a specific category is obtained by double clicking the category. For example, `Elementary math functions` includes all of the following:

```
  Trigonometric.
   sin         - Sine.
   sinh        - Hyperbolic sine.
   asin        - Inverse sine.
   asinh       - Inverse hyperbolic sine.
   cos         - Cosine.
   cosh        - Hyperbolic cosine.
   acos        - Inverse cosine.
   acosh       - Inverse hyperbolic cosine.
   tan         - Tangent.
   tanh        - Hyperbolic tangent.
   atan        - Inverse tangent.
   atan2       - Four quadrant inverse tangent.
   atanh       - Inverse hyperbolic tangent.
   sec         - Secant.
   sech        - Hyperbolic secant.
   asec        - Inverse secant.
   asech       - Inverse hyperbolic secant.
   csc         - Cosecant.
   csch        - Hyperbolic cosecant.
   acsc        - Inverse cosecant.
   acsch       - Inverse hyperbolic cosecant.
   cot         - Cotangent.
   coth        - Hyperbolic cotangent.
   acot        - Inverse cotangent.
   acoth       - Inverse hyperbolic cotangent.

Exponential.
   exp         - Exponential.
   log         - Natural logarithm.
   log10       - Common (base 10) logarithm.
   log2        - Base 2 logarithm.
```

```
      pow2        - Base 2 power and scale floating point number.
      sqrt        - Square root.
      nextpow2    - Next higher power of 2.

   Complex.
      abs          - Absolute value.
      angle        - Phase angle.
      complex      - Construct complex data from real and imaginary
                     parts.
      conj         - Complex conjugate.
      imag         - Complex imaginary part.
      real         - Complex real part.
      unwrap       - Unwrap phase angle.
      isreal       - True for real array.
      cplxpair     - Sort numbers into complex conjugate pairs.

    Rounding and remainder.
      fix         - Round towards zero.
      floor       - Round towards minus infinity.
      ceil        - Round towards plus infinity.
      round       - Round towards nearest integer.
      mod         - Modulus (signed remainder after division).
      rem         - Remainder after division.
      sign        - Signum.
```

Also, typing `helpwin('elfun')`in the Command window will access the MATLAB Help Window of Elementary math functions directly. Double clicking a specific function brings up detailed information about that function. For example, the '`rem`' function description is

```
REM    Remainder after division.
    REM(x,y)  is  x  -  y.*fix(x./y)  if  y  ~=  0.   By  convention,
    REM(x,0) is NaN.  The input x and y must be real arrays of
    the same size, or real scalars.

    REM(x,y) has the same sign as x while MOD(x,y) has the same
    sign as y.  REM(x,y) and MOD(x,y) are equal if x and y have
    the  same  sign,  but  differ  by  y  if  x  and  y  have  different
    signs.

    REM is a built-in function.  An M-file for REM would be the
    same as MOD with "floor" replaced by "fix".
```

Some of the elementary mathematical functions are illustrated in the following examples.

Example 2.6.1

```
% A sample of the Trigonometric Functions in MATLAB
theta1= pi/6 % Angle (in rad)
theta1_deg=180*theta1/pi % Angle (in deg)
x=sin(theta1) % Sine function
theta2=atan(1) % Inverse tangent function (in rad)
```

```
theta2_deg=180*theta2/pi % Inverse tangent function (in deg)
y=tan(theta2) % Tangent function
```

```
theta1 = 0.5236
theta1_deg = 30.0000
x = 0.5000
theta2 = 0.7854
theta2_deg = 45
y = 1.0000
```

Example 2.6.2

```
%  A sample of the Exponential Functions in MATLAB
x=0;y=1;
u0=exp(y)  % Exponential Function
u1=exp(x-y)
u2=exp(x+y)
z=log10(100*y) % Base 10 log function
a=50*y-25
w1=log(a) % Natural log function
w2=exp(w1)
w3=sqrt(a) % Square root function
w4=a^0.5
```

```
u0 =  2.7183
u1 =  0.3679
u2 =  2.7183
z =   2
a =  25
w1 =  3.2189
w2 = 25.0000
w3 =  5
w4 =  5
```

Note, the arguments of the MATLAB functions may be expressions involving previously defined or computed MATLAB variables. Not all MATLAB built-in mathematical functions have a single argument. In the next example, the function 'rem' requires two arguments.

Example 2.6.3

```
%   A sample of Rounding and Remainder Functions in MATLAB
a=9; b=5; c=-5;
x1=fix(a/b) % Round toward zero
x2=floor(a/c) % Round toward minus infinity
x3=ceil(a/c) % Round toward plus infinity
x4=rem(a,b) % Remainder after division
x5=sign(b-a) % Signum function
```

```
x1 =  1
x2 = -2
x3 = -1
x4 =  4
x5 = -1
```

Many of MATLAB's built-in functions return several outputs. In this case, the left hand side is a row vector with output variable names, separated by commas and enclosed in square brackets. The cartesian transformation function '`cart2sph`' produces 3 outputs from a set of 3 inputs as described below.

```
CART2SPH Transform Cartesian to spherical coordinates.
  [TH,PHI,R] = CART2SPH(X,Y,Z) transforms corresponding elements
   of data stored in Cartesian coordinates X,Y,Z to spherical
   coordinates (azimuth TH, elevation PHI, and radius R).  The
   arrays X,Y, and Z must be the same size (or any of them can
   be scalar). TH and PHI are returned in radians.

   TH is the counterclockwise angle in the xy plane measured
   from the positive x axis.  PHI is the elevation angle from
   the xy plane.

   See also CART2POL, SPH2CART, POL2CART.
```

Two different 2-D coordinate transformation functions are illustrated below.

Example 2.6.4

```
 % A sample of Coordinate Transformation Functions in MATLAB
x1=3,y1=4,
[th1,r1]=cart2pol(x1,y1) % Cartesian to polar
cart2pol(x1,y1) % Will return only the first output in 'ans'
th2=th1+pi
r2=r1;
[x2,y2]=pol2cart(th2,r2) % Polar to Cartesian
```

```
x1 =   3
y1 =   4
th1 =  0.9273
r1 =   5
ans =  0.9273
th2 =  4.0689
x2 =  -3.0000
y2 =  -4.0000
```