Overview

In this chapter we start studying the many, many, mathematical functions that are included in MATLAB. Some time will be spent introducing complex variables and how MATLAB handles them. Later we will learn how to write custom (user written) functions which are a special form of m-file. Two dimensional plotting functions will be introduced. We will also explore programming constructs for flow control (if–else–elseif code blocks) and looping (for loop etc.). Next data analysis functions will be investigated, which includes sample statistics and histogram plotting functions.
Mathematical Functions

Common Math Functions

Table 3.1: Common math functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>abs(x)</td>
<td>(</td>
<td>x</td>
<td>)</td>
</tr>
<tr>
<td>round(x)</td>
<td>nearest integer</td>
<td>fix(x)</td>
<td>nearest integer</td>
</tr>
<tr>
<td>floor(x)</td>
<td>nearest integer</td>
<td>ceil(x)</td>
<td>nearest integer</td>
</tr>
<tr>
<td></td>
<td>toward (-\infty)</td>
<td></td>
<td>toward (x)</td>
</tr>
</tbody>
</table>
| sign(x)        | \(\begin{cases} 
                                -1, & x < 0 \\
                                0, & x = 0 \\
                                1, & x > 0 \end{cases}\) | rem(x, y)    | the remainder of \(x/y\) |
| exp(x)         | \(e^x\)             | log(x)       | natural log \(\ln x\) |
| log10(x)       | log base 10          |              |                 |
|                | \(\log_{10} x\)     |              |                 |

Examples:

```python
» x = [-5.5 5.5];
» round(x)
ans =   -6     6
» fix(x)
ans =    -5     5
» floor(x)
ans =    -6     5
» ceil(x)
```
Trigonometric and Hyperbolic Functions

• Unlike pocket calculators, the trigonometric functions always assume the input argument is in radians

• The inverse trigonometric functions produce outputs that are in radians

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \sin(x) )</td>
<td>( \sin(x) )</td>
<td>( \cos(x) )</td>
<td>( \cos(x) )</td>
</tr>
<tr>
<td>( \tan(x) )</td>
<td>( \tan(x) )</td>
<td>( \arcsin(x) )</td>
<td>( \sin^{-1}(x) )</td>
</tr>
<tr>
<td>( \arccos(x) )</td>
<td>( \cos^{-1}(x) )</td>
<td>( \arctan(x) )</td>
<td>( \tan^{-1}(x) )</td>
</tr>
<tr>
<td>( \text{atan2}(y, x) )</td>
<td>the inverse tangent of ( y/x ) including the correct quadrant</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Examples:

• A simple verification that \( \sin^2(x) + \cos^2(x) = 1 \)
```matlab
» x = 0:pi/10:pi;
» [x' sin(x)' cos(x)' (sin(x).^2+cos(x).^2)']
ans =
    0         0    1.0000    1.0000
  0.3142    0.3090    0.9511    1.0000
  0.6283    0.5878    0.8090    1.0000
  0.9425    0.8090    0.5878    1.0000
  1.2566    0.9511    0.3090    1.0000
  1.5708    1.0000    0.0000    1.0000
  1.8850    0.9511    -0.3090    1.0000
  2.1991    0.8090    -0.5878    1.0000
  2.5133    0.5878    -0.8090    1.0000
  2.8274    0.3090    -0.9511    1.0000
  3.1416    0.0000   -1.0000    1.0000

• Parametric plotting:
  – Verify that by plotting \( \sin \theta \) versus \( \cos \theta \) for \( 0 \leq \theta \leq 2\pi \) we obtain a circle
```
• The trig functions are what you would expect, except the features of \( \text{atan2}(y, x) \) may be unfamiliar.

\[
\begin{align*}
\phi_1 &= 0.4636 \\
\phi_2 &= -2.0344
\end{align*}
\]
» \[\text{atan}(2/4) \ \text{atan2}(2,4)\]
    \[
    \text{ans} = \ 0.4636 \ 0.4636 \ % \text{ the same}
    \]
» \[\text{atan}(-4/-2) \ \text{atan2}(-4,-2)\]
    \[
    \text{ans} = \ 1.1071 \ -2.0344 \ % \text{ different; why?}
    \]

• The hyperbolic functions are defined in terms of \(e^x\)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\sinh(x))</td>
<td>(\frac{e^x - e^{-x}}{2})</td>
<td>(\cosh(x))</td>
<td>(\frac{e^x + e^{-x}}{2})</td>
</tr>
<tr>
<td>(\tanh(x))</td>
<td>(\frac{e^x - e^{-x}}{e^x + e^{-x}})</td>
<td>(\text{asinh}(x))</td>
<td>(\ln(x + \sqrt{x^2 + 1}))</td>
</tr>
<tr>
<td>(\text{acosh}(x))</td>
<td>(\ln(x + \sqrt{x^2 - 1}))</td>
<td>(\text{atanh}(x))</td>
<td>(\ln\left(\frac{1 + x}{\sqrt{1 - x}}\right),</td>
</tr>
</tbody>
</table>

• There are no special concerns in the use of these functions except that \(\text{atanh}\) requires an argument that must not exceed an absolute value of one

**Complex Number Functions**

• Before discussing these functions we first review a few facts about complex variable theory

• In electrical engineering complex numbers appear frequently

• A complex number is an ordered pair of real numbers\(^1\)

---

denoted \((a, b)\)

- The first number, \(a\), is called the real part, while the second number, \(b\), is called the imaginary part

- For algebraic manipulation purposes we write \((a, b) = a + ib = a + jb\) where \(i = j = \sqrt{-1}\); electrical engineers typically use \(j\) since \(i\) is often used to denote current

**Note:** \(\sqrt{-1} \times \sqrt{-1} = -1 \Rightarrow j \times j = -1\)

- For complex numbers \(z_1 = a_1 + j b_1\) and \(z_2 = a_2 + j b_2\) we define/calculate

\[
\begin{align*}
z_1 + z_2 &= (a_1 + a_2) + j(b_1 + b_2) \quad \text{(sum)} \\
z_1 - z_2 &= (a_1 - a_2) + j(b_1 - b_2) \quad \text{(difference)} \\
z_1 z_2 &= (a_1 a_2 - b_1 b_2) + j(a_1 b_2 + b_1 a_2) \quad \text{(product)} \\
z_1 &= \frac{(a_1 a_2 + b_1 b_2) - j(a_1 b_2 - b_1 a_2)}{a_2^2 + b_2^2} \quad \text{(quotient)} \\
\mid z_1 \mid &= \sqrt{a_1^2 + b_1^2} \quad \text{(magnitude)} \\
\angle z_1 &= \tan^{-1}(b_1 / a_1) \quad \text{(angle)} \\
z_1^* &= a_1 - j b_1 \quad \text{(complex conjugate)}
\end{align*}
\]

- MATLAB is consistent with all of the above, starting with the fact that \(i\) and \(j\) are predefined to be \(\sqrt{-1}\)
• The **rectangular form** of a complex number is as defined above,

\[ z = (a, b) = a + jb \]

• The corresponding **polar form** is

\[ z = r \angle \theta \]

where \( r = \sqrt{a^2 + b^2} \) and \( \theta = \tan^{-1}(b/a) \)

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{conj}(x)</td>
<td>Computes the conjugate of ( z = a + jb ) which is ( z^* = a - jb )</td>
</tr>
<tr>
<td>\text{real}(x)</td>
<td>Extracts the real part of ( z = a + jb ) which is ( \text{real}(z) = a )</td>
</tr>
<tr>
<td>\text{imag}(x)</td>
<td>Extracts the imaginary part of ( z = a + jb ) which is ( \text{imag}(z) = b )</td>
</tr>
</tbody>
</table>
Table 3.4: Basic complex functions

<table>
<thead>
<tr>
<th>Function</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>\text{angle}(x)</td>
<td>computes the angle of $z = a + jb$ using \text{atan2} which is \text{atan2}(\text{imag}(z), \text{real}(z))</td>
</tr>
</tbody>
</table>

\textbf{Euler’s Formula:} A special mathematical result of, special importance to electrical engineers, is the fact that

$$e^{jb} = \cos b + j \sin b \quad (3.1)$$

• Turning (3.1) around yields

$$\sin \theta = \frac{e^{j\theta} - e^{-j\theta}}{2j} \quad (3.2)$$

$$\cos \theta = \frac{e^{j\theta} + e^{-j\theta}}{2} \quad (3.3)$$

• It also follows that

$$z = a + jb = re^{j\theta} \quad (3.4)$$

where

$$r = \sqrt{a^2 + b^2}, \ \theta = \tan^{-1} \frac{b}{a}, \ a = r \cos \theta, \ b = r \sin \theta$$
• Some examples:
  » z1 = 2+j*4; z2 = -5+j*7;
  » [z1 z2]
  ans =
  2.0000 + 4.0000i  -5.0000 + 7.0000i
  » [real(z1) imag(z1) abs(z1) angle(z1)]
  ans =
  2.0000  4.0000  4.4721  1.1071
  » [conj(z1) conj(z2)]
  ans =
  2.0000 - 4.0000i  -5.0000 - 7.0000i
  » [z1+z2 z1-z2 z1*z2 z1/z2]
  ans =
  -3.0000 +11.0000i   7.0000 - 3.0000i
  -38.0000 - 6.0000i  0.2432 - 0.4595i

**Polar Plots:** When dealing with complex numbers we often deal
with the polar form. The plot command which directly plots vec-
tors of magnitude and angle is `polar(theta,r)`

• This function is also useful for general plotting
• As an example the equation of a cardioid in polar form, with
  parameter $a$, is

\[
r = a(1 + \cos \theta), \ 0 \leq \theta \leq 2\pi
\]

» theta = 0:2*pi/100:2*pi;  % Span 0 to 2pi with 100 pts
» r = 1+cos(theta);
» polar(theta,r)