Physics 263: MATLAB Cheatsheet V

This is a summary of the use of complex numbers in MATLAB.

1. **Entering Complex Numbers**
   
   a. **Cartesian form.** You can use either $i$ or $j$ for $\sqrt{-1}$ (since we’re in a physics class, we’ll use $i$ exclusively). If $i$ appears in a numerical expression, MATLAB will simply interpret the number as a complex number. For example, if we want to set $z$ equal to $3 + 4i$, simply type it in this way:
      
      ```matlab
      >> z = 3 + 4*i
      ``
      
      We could even omit the $*$ in this case, but it is best to leave it in, since we need it if we are using a variable such as $y$:
      
      ```matlab
      >> x = 1/sqrt(2);
      >> y = -1/sqrt(3);
      >> z = x + i*y
      ``
      
      We can use any of the standard functions (e.g., exponentials or trigonometric functions) directly. For example,
      
      ```matlab
      >> sin(1-sqrt(2)*i)
      ans = 1.8329 - 1.0455i
      ``
      
      Note that the answer comes back in standard $x + yi$ form with decimals (as opposed to fractions or explicit $\pi$’s).
   
   b. **Polar form.** If we want to enter a number in the form $re^{i\theta}$, we just use the `exp` function:
      
      ```matlab
      >> r = 1;
      >> theta = pi/6;
      >> z = r*exp(i*theta)
      z = 0.8660 + 0.5000i
      ``
      
      Note that the answer comes back in Cartesian form.

2. **Operations on Complex Numbers**

   a. **Arithmetic with complex numbers.** All of the standard operations ($+ - * /$) will work as expected (i.e., correctly) with complex numbers.

   b. **Complex conjugate.** The function `conj` takes the complex conjugate:
      
      ```matlab
      >> conj(2-3*i)
      ans = 2.0000 + 3.0000i
      ``
      
   c. **Real and imaginary parts.** The functions `real` and `imag` do the trick:
\[
\text{\texttt{>> z = (1 + i)/(2 - i);}}
\]
\[
\text{\texttt{\texttt{>> real(z)}}}
\]
\[
\text{\texttt{ans = 0.2000}}
\]
\[
\text{\texttt{>> imag(z)}}
\]
\[
\text{\texttt{ans = 0.6000}}
\]

Note that you always end up with the decimal version of numbers.

d. **Modulus and angle.** You can find \( r \) and \( \theta \) using the \texttt{abs} and \texttt{angle} functions:

\[
\text{\texttt{>> z = 2 * exp(i*pi/4)}}
\]
\[
\text{\texttt{z = 1.4142 + 1.4142i}}
\]
\[
\text{\texttt{>> abs(z)}}
\]
\[
\text{\texttt{ans = 2}}
\]
\[
\text{\texttt{>> angle(z) \% answer should be pi/5 = 0.7854}}
\]
\[
\text{\texttt{ans = 0.7854}}
\]

3. **Plotting Complex Functions**

We can use any of the plotting commands we’ve looked at before with functions of a complex variable \( z \) if we take the modulus of the function using the \texttt{abs} command. For example, suppose we want to plot the function \( f(z) = 1/(z^4 + 1) \) in the complex plane for the real and imaginary parts of \( z \) varying from \(-2\) to \(+2\). Here’s an excerpt from a script to do this.

```matlab
%% Set a grid X Y with the desired range (and 20 points on each axis)
[X Y] = meshgrid( linspace(-2,2,20), linspace(-2,2,20) );

%% Define z = x + i*y for each point in the grid
Z = X + i*Y;

%% Evaluate the function we want to study.
%% Note the use of "." before operations like "^" and "/".
\texttt{f = abs( 1 ./ (Z.^4 + 1) ); \% use "abs" to take the modulus}

figure(1); \% figure 1 will be a surface plot
surf(X,Y,f); colorbar; \% make the plot and add a color bar
xlabel('x-axis'); ylabel('y-axis'); \% add labels x and y axes

figure(2); \% figure 2 will be a contour plot
num_contours = 10; \% use \texttt{num_contours} contour lines
contourf(X,Y,f,num_contours); colorbar; \% make the plot and add a color bar
xlabel('x-axis'); ylabel('y-axis'); \% add labels
```