

Sheet 1

Some syntax trouble spots:

| | |
|----------------|----------------------------|
| multiplication | <code>3*t</code> for $3t$ |
| powers | <code>x^2</code> for x^2 |
| number π | <code>pi</code> |
| division | <code>/</code> |

| | | |
|----------------------|-----|------------|
| <code>sin(x)</code> | for | $\sin x$ |
| <code>abs(x)</code> | for | $ x $ |
| <code>cos(x)</code> | for | $\cos x$ |
| <code>sqrt(x)</code> | for | \sqrt{x} |
| <code>tan(x)</code> | for | $\tan x$ |
| <code>log(x)</code> | for | $\ln x$ |
| <code>exp(x)</code> | for | e^x |

Create a directory named `sheet01` and make it your working directory.

Example 1. Matlab will always ignore a line that starts with a `%` sign, so we'll often put comments after it. We plot the function $y = t \sin(3t)$ for $0 \leq t \leq 2\pi$. First save a blank sheet as `ex1.m` and enter the following:

```
t = linspace(0,2*pi,20);
y = t.*sin(3*t);
plot(t,y)
```

Then hit the evaluate button: it is the first button from the left in the second row of the toolbar in the Editor window.

Notes.

1. The semicolon `;` tells Matlab to suppress the output. Try running the file without it in the `linspace` line.
2. The `linspace(0,2*pi,20)` tells Matlab to generate a list of points from 0 to 2π ; the 20 tells it to generate 20 equally spaced points. You can leave off the 20 or change it—the default is 100. Try it—note the more points used, the smoother the plot.
3. Since `t` is an *array* of numbers, when defining `y`, if you had used `*`, Matlab would see the multiplication as that for arrays—“matrix multiplication.” This isn't what you want; you need to make sure Matlab computes an array of numbers corresponding to the `t` values. This is why you must use the `.*` instead of just `*`. See what happens if you leave off the period. So when multiplying, dividing or exponentiating arrays to be used in plotting, always use `.*`, `.^`, `./` in place of `*`, `^`, `/`.
4. You can specify the color you want. For instance to make the curve black, use `plot(t,y,'k')` in place of `plot(t,y)`. Here are some color specifiers:

| | | | |
|-------|---|---------|---|
| red | r | magenta | m |
| green | g | yellow | y |
| blue | b | black | k |
| cyan | c | white | w |

Example 2. Save your ex1.m file one last time and repeat this procedure for each example. Now plot the functions $y = \cos(2t)$ and $y = \cos(3t)$ on the same axes. Here's how (first open a new sheet and save it at ex2.m):

```
t = linspace(0,2*pi);
y1 = cos(2*t);
y2 = cos(3*t);
plot(t,y1,t,y2)
```

To assign the color red to $y = \cos(2t)$ and the color blue to $y = \cos(3t)$, use `plot(t,y1,'r',t,y2,'b')`.

Notes. The following are all optional; you can use any or all.

1. To label the axes, include the statements `xlabel('time (seconds)');`
`ylabel('millivolts');` after the plot statement.
2. To label the curves, include the statement `legend('y = cos 2t', 'y = cos 3t');` after the plot statement.
3. To include a grid on the plot include the statement `grid on;` after the plot statement.

```
plot(t,y1,'r',t,y2,'b')
xlabel('time (seconds)'); ylabel('millivolts');
legend('y = cos 2t', 'y = cos 3t'); grid on;
```

Example 3. Plot the function $y = 2 \sin(1.9t) \sin(0.1t)$ for $0 \leq t \leq 100$. You might need to play around with `linspace` to get enough points so the plot doesn't look too jagged.

Then for $0 \leq t \leq 100$, plot all the functions $y = 2 \sin(1.9t) \sin(0.1t)$, $y = 2 \sin(0.1t)$ and $y = -2 \sin(0.1t)$ on the same graph. Make the first red and the other two green.

You can lengthen the axes in the plot by placing the line `axis([-0.5, 100.5, -2.3, 2.3]);` somewhere after the `plot` line. The first two numbers say how long to make the x -axis and the second pair is for the y -axis.

Example 4. Plot the function $f(t) = 0.01t^4 - 0.45t^2 + 0.5t - 2$. Play around with the plot range in `linspace` statement until you get a good plot.

Example 5. Plot the function $f(x) = \frac{5}{1 + e^{5.5-1.5x}} - \frac{x^2}{20}$ for $-10 \leq x \leq 10$.

Example 6. Plot the function $f(x) = \frac{x^2 - 4x - 5}{x - 2}$ for $-4 \leq x \leq 8$. Notice there is a vertical asymptote at $x = 2$. Matlab doesn't do such a great job of plotting the function. Here is another way to get a better looking plot. Plot the function by making two `linspace` statements for the domain of f , say `x1` for $-4 \leq x \leq 1.7$ and `x2` for $2.3 \leq x \leq 8$. Then make corresponding `y1` and `y2` and plot them all on the same graph, making both the same color. You might need to adjust the 1.7 and 2.3 to get a good plot.