# LECTURE 4: 2 D GRAPHICS

One of the very useful features of MATLAB is the ease with which many different types of graphs may be drawn. Most graphs are drawn with variants of the `plot` command, which has several forms, each with many optional arguments that determine the appearance of the resulting graph. Here we introduce a few of the simplest forms. If at any time you wish to learn more, explore with MATLAB's help and doc pages.

## 1.3.1 Simple line graphs

The simplest graphs are those of functions of a single real variable; for example, the graph of the function $\sin(10/(1+x^2))$ for $0 \leq x \leq 10$ is given by the command

```
>> syms x
>> y = sin(10/(1+x^2));
>> ezplot(y)
```

You should see the graph on your screen, and it should be blue, which is the default colour. You can control the domain with

```
>> ezplot(y, [-5, 10])
```

The range of the vertical axis can also be specified:

```
>> ezplot(y, [-5 10])
>> ylim([-1.2 1.2])
```

**Exercise 1.10** Plot $\sin(10\cos(x))$ on the interval $-2\pi \leq x \leq 2\pi$. □

You can draw more than one curve, in different colours and line styles. For example:[4]

```
>> clear
>> syms x
>>
>> ezplot(cos(x))
>> hold on
>> h = ezplot(sin(x));
>> set(h, 'color', 'red')
>> set(h, 'linestyle', '--')
>> set(h, 'linewidth', 3)
>>
>> legend('cos', 'sin')
>> title('my colourful plots')
```

The use of "`hold on`" is crucial: try it without. The American spelling of "colour" is required (as is the American pronunciation of "EZ"). The variable `h` is called a "handle" and it gives you access to MATLAB "under the hood": see `get(h)` for example.

You can also clear a figure with `clf` and control which figure window you're drawing on:

```
>> figure(2); clf;
>> h = ezplot(atan(x));
>> set(h, 'linewidth', 4)
>> set(h, 'color', [0.5 0 1])
```

The colour was set using an "RGB triplet" a vector specifying the percentages of red, green and blue.

**Exercise 1.11** Draw the graphs of these five functions on the same plot:

$$y_1(x) = 1, \quad y_2(x) = 1 + x, \quad y_3(x) = 1 + x + x^2/2, \quad y_4(x) = 1 + x + x^2/2 + x^3/6,$$

and $y_5(x) = \exp(x)$, on the interval $0 \le x \le 1$. Use different colours for each. Add a legend. □

You can interact with figures using the graphical user interface. For example, zooming and panning with the mouse (try this).

**Exercise 1.12** Using a plot, find the approximate coordinates of all the real solutions of the nonlinear simultaneous equations

$$y = \sin x,$$
$$y = x^3 - 5x^2 + 4.$$

Hint: see `help ginput` for one interesting way to do this with a mouse. (You will see in Section 2.2.3 how to find numerical solutions to simultaneous equations such as these.) □

### 1.3.2 Parametric and polar plots

The command `ezplot` can also plot a function that is defined parametrically (with $x$ and $y$ as functions of $t$, say). So the ellipse $x = 2\cos t$, $y = \sin t$ is plotted with the command

```
>> syms t
>> ezplot(2*cos(t), sin(t), [0, 2*pi])
>> axis equal
```

The `axis equal` command will ensure that a circle looks like a circle.

**Exercise 1.13** The cycloid $x = t - \sin t$, $y = 1 - \cos t$ is the curve traced out by a point on a wheel as the wheel turns. Plot this curve for $0 \le t \le 6\pi$. □

Polar plots of $r = f(\theta)$ for $a \le \theta \le b$ can be achieved with `ezpolar`, for example to draw the cardioid:

```
>> syms t
>> ezpolar(1 - cos(t), [0, 2*pi])
```

**Exercise 1.14** Plot (both leaves of) the lemniscate $r^2 = \cos(2\theta)$. □

## Alternatives to the ezplot

The `ezplot` command and the other `ez` commands are useful to make quick plots, particularly of symbolic expressions. However, sometimes it might be useful to access the other MATLAB plotting commands in order to have more control over plots. We give one (non-exhaustive) example, as you will certainly encounter this style of MATLAB plotting:

```
>> syms x
>> y = sin(10/(1+x^2));
>> ym = matlabFunction(y)
>> xx = linspace(-5, 10, 512);
>> plot(xx, ym(xx))
```

The key idea here is to create a vector of `doubles` (the `linspace` command). We also convert the symbolic expression into a more traditional MATLAB function.

## Animation

We finish this chapter with an animation:

```
>> clear
>> clf
>> syms x y
>> ezsurf(real(atan(x + 1i*y)), 'circ')
>> shading flat
>> camlight left
>> material shiny
>>
>> for a = 1:90
>>   view([-30, a])
>>   pause(0.1)
>> end
```

(More about `for` loops later.)