Working with Octave

Octave online is a free coding platform, using Matlab syntax. We will be using this in Wednesday's tutorial, so please take 10 mins before the tutorial to complete the Login section, and take a look at the extras if you have time.

Contents

Login 1

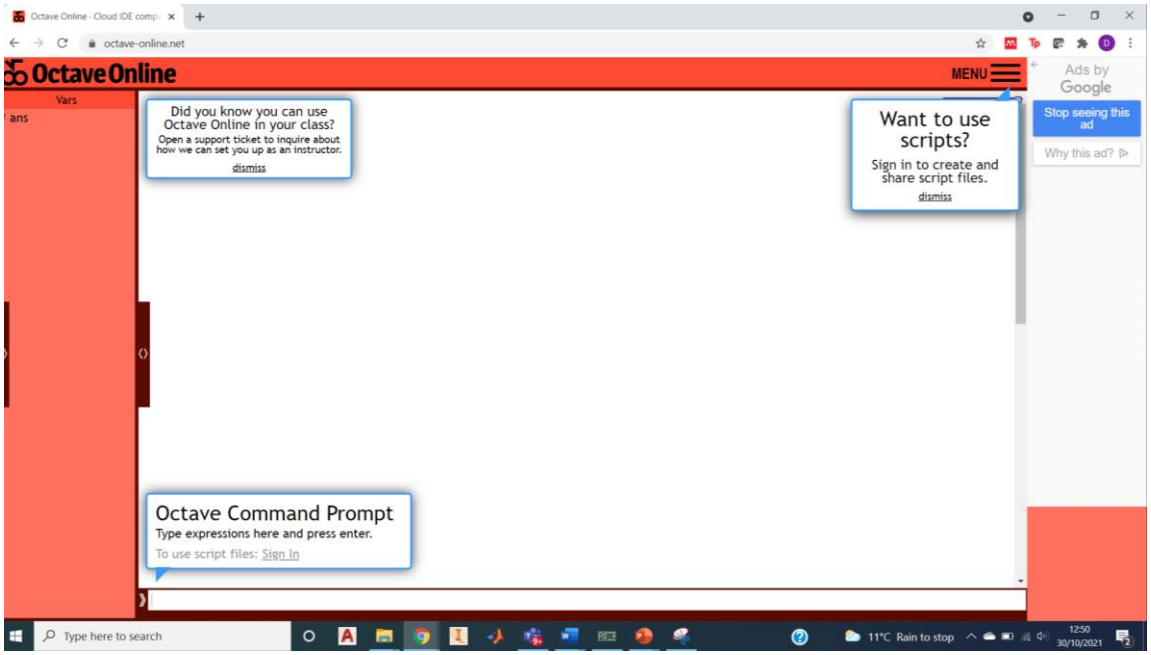
Extra 2

Plotting Sin(x) 2

Plotting something more exciting 4

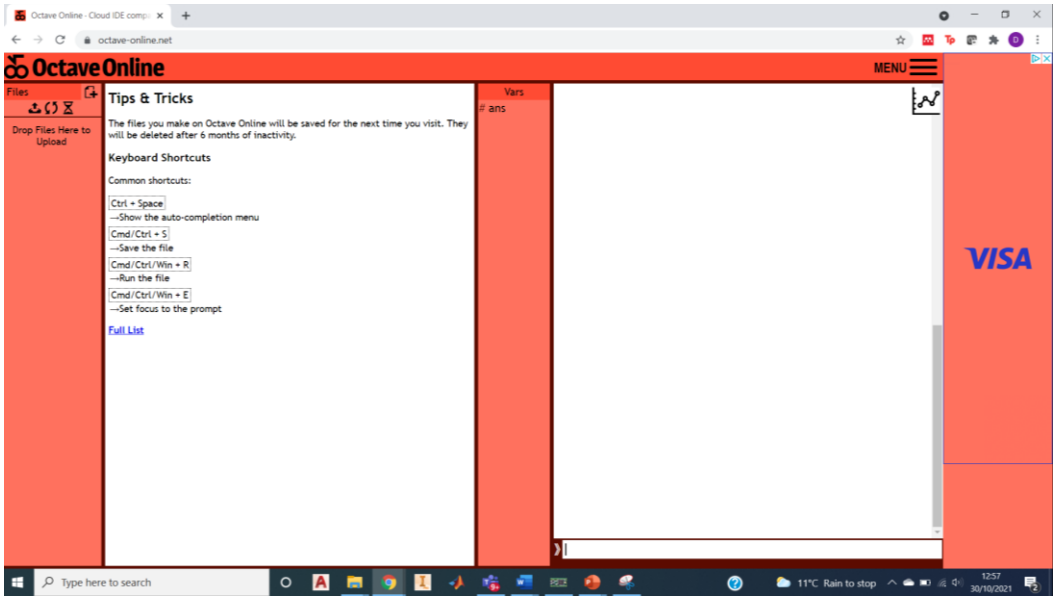
Login

To run octave search for 'octave online' or go directly to octave-online.net. This should take you to a page that looks like this:



We'll want to use scripts, so you'll need to sign in. This can be done by hitting the menu button in the top right of the screen and signing in using one of the 3 methods. I prefer using the email token method.

Now your screen should look like this:



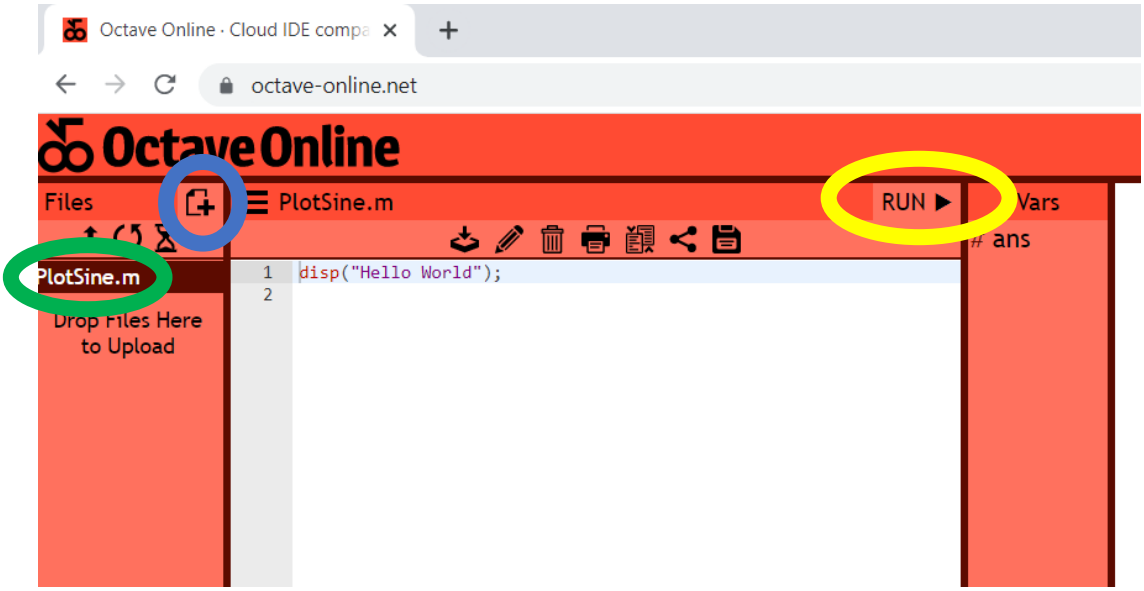
Good job! This is all you'll need before the tutorial. If you're struggling, please drop me an email at dtb26@bath.ac.uk and I'll do my best to help you out before we start.

Extra

These are optional, but would be useful to do before the tutorial.

Plotting Sin(x)

We're going to create and run a quick program to make sure everything works okay. Hit the create file button in the top left of the screen (highlighted in blue) and call the file 'PlotSine.m' – remember the '.m' at the end!



This file should now appear in the top left of the screen (highlighted green). Click on this file and now you should be able to write code into the file in the left-hand side window. As you can see, the default programme displays the phrase 'Hello World'. If you hit RUN (shown in yellow above) you should see *Hello World* appear in the command window on the right-hand side.

Copy and paste the following code into the PlotSine.m function to do exactly that! Remember, you'll need to hit save **before** running the programme.

```
clc   % Clears the command window
clear % Clears all Vars
close % Closes any figures

% Inputs:
nVals = 100;  % How many points are we
going to use

% Create places to store data
x = linspace(0,360,nVals);    %
y = zeros(size(x));           %
Initialise (create) y

% Now loop through each x value, then
solve and store y values
for i = 1:nVals
    y(i) = sind(x(i));
end

% Now do a pretty plot
figure
lineStyle = '-';
plot(x,y,lineStyle)
strTitle = ["Sin(x)"];
title(strTitle);
grid on
xlabel("X")
ylabel("Y")
```

This code can be accessed digitally at:
https://github.com/danbowen95/VilliersPark/blob/main/PlotSine.m

Can you figure out how the code works?

What happens if we change nVals to 10?

Plotting something more exciting

Now that we know how to plot, lets plot something a bit more exciting. We're going to plot the parametric function:

$$x = \cos(At) + \frac{\cos(Bt)}{2} + \frac{\sin(Ct)}{3}, y = \cos(At) + \frac{\cos(Bt)}{2} + \frac{\sin(Ct)}{3}$$

Copy and paste the code in from here:

```
clc   % Clears the command window
clear % Clears all Vars
close % Closes any figures

% Inputs:
nVals = 1000;  % How many equally spaced
increments of t shall we use
A = 8;         % Coefficient's
B = 3;
C = 5;

% Create places to store data
t = linspace(0,2*pi,nVals);
x = zeros(size(t));
y = zeros(size(t));

% Now loop through each t value, then
solve and stor x and y values
for i = 1:nVals
    x(i) = cos(A*t(i)) + (cos(B*t(i))/2)
+ (sin(C*t(i))/3);
    y(i) = sin(A*t(i)) + (sin(B*t(i))/2)
+ (cos(C*t(i))/3);
end

% Now do a pretty plot
figure
strTitle = ["A = " num2str(A)];
plot(x,y)
strTitle = ["A = " num2str(A) " , B = "
num2str(B) " , C = " num2str(C)];
title(strTitle);
```

This code can be accessed digitally at:
https://github.com/danbowen95/VilliersPark/blob/main/PlotParametric.m

What happens when we change A,B,C values?