Session – 27 : More about octave

24-10-2019, 09:00 – 10:00 Hrs, RJN 302

1. Functions in octave:

One can create user defined functions in octave. If the function we want to create is "myfunc" then the file that should contain the definition of this function should be in a file named "myfunc.m".

Here is an illustration on how a function can be defined. The function in mathematical notation is as follows:

$$f(x) = exp(-x)\sin(x + c)$$

Contents of the file "`myfunc.m`"

```
function y = myfunc(x, c)
     y = exp(x) .* sin(x+c);
```

The operator "dot-star" is to ensure that when we pass an array to the function, then the multiplication is still the way we wanted, namely, element-by-element. This is how all the elementary mathematical functions such as exp, sin, cos etc., work in octave.

2. Making families of curves

Below is a piece of code that generates a family of curves for different values of the parameter c in the above defined function.

Contents of the file "`myplots.m`"

```
x=[1:0.1:5];
for c = 1:40
  c1 = c*0.1;
  y=myfunc(x,c1);
  p=plot(x,y);
  set(p,'linewidth',[2]);
  axis([1 5 -0.5 0.5]);
  xlabel('value of x')
  ylabel('value of f(x)')
  title('plot of f(x)')
  hold on
end
```

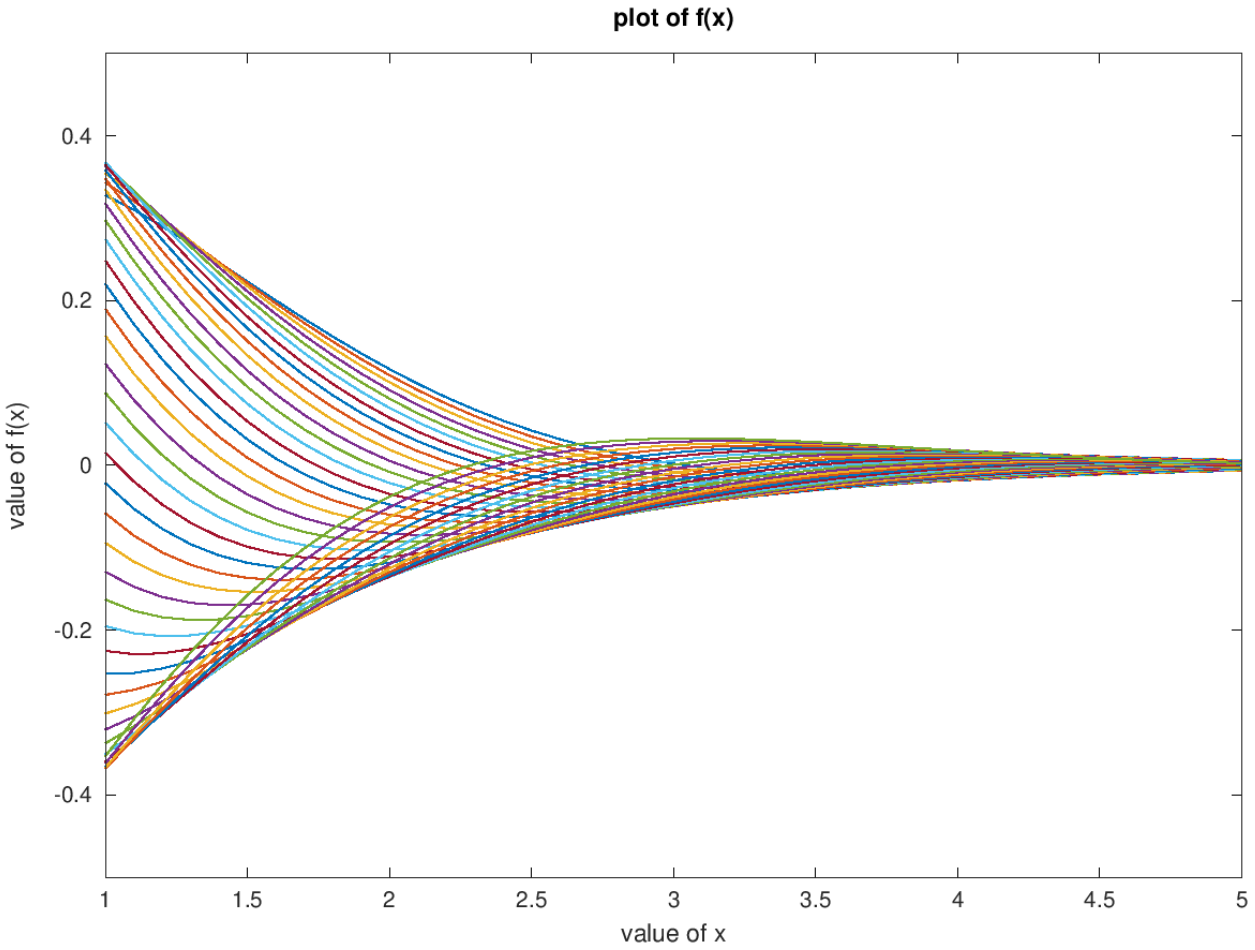
The above code can be executed by opening `octave` and at the prompt > typing the following two commands.

```
graphics_toolkit('gnuplot')
```

```
myplots
```

The first line helps octave do the plotting using the gnuplot software.
The second line executes the code in the above mentioned file. Make sure you run this when octave is opened from the shell in the same working directory as where the file "myplots.m" is located. You can navigate the directories using the "cd" command just ike in shell.

The output of the above code will display the family of curves as shown below.



You can save the image to a file using the following command at the prompt > in the octave.

```
print -dpng myplots.png
```

The string "png" after "-d" after the print command conveys that the file output should be in png format. The string that follows namely "myplots.png" conveys the name of the file to write the image to. This file will be saved to the current working directory.

3. Using the sprintf and eval functions

The syntax of the sprintf command is same as in c language. You can use it to create strings that contain dynamic information with values of certain variables inside. For example, you can title the plot to include the value of the parameter used for the family of curves.

A string can be passed on to the function eval to evaluate it. This helps in creating dynamic commands on the fly and run them.

Using these two features, we can create one image per plot in the family of curves using the following code.

Contents of the file "myplotsequence.m"

```
x=[1:0.1:5];
hold off
for c = 1:40
  c1 = c*0.1;
  y=myfunc(x,c1);
  p=plot(x,y);
  set(p,'linewidth',[2]);
  axis([1 5 -0.5 0.5]);
  xlabel('value of x')
  ylabel('value of f(x)')
  str = sprintf("f(x) at c1=%f",c1);
  title(str)
  str2 = sprintf("print -dpng plt-%d.png",c);
  eval(str2);
end
```

The above code can be executed by opening `octave` and at the prompt > typing the following two commands.

```
graphics_toolkit('gnuplot')
myplotsequence
```

You can notice that a bunch of 40 images have been created in the current working directory by the above script.

4. Making a video of an image sequence

You can use the utility imagej to create a video out of an image sequence. You can install it on ubuntu 18.04 LTS by using "`sudo apt-get install imagej`". You should already have java run time environment to have this utility running. More information on this is linked in the moodle page.

The sequence of steps is as follows:

Open imagej → File → Import → Image Sequence → click on the first image in the sequence → Open.

The imagej software should be able to identify all the plots and show you a dialogue box that indicates how many images are being loaded. Edit the fields to indicate the starting image, the increment (to skip every second image give 2, for example). You can scale the image if you need. If the file names are not easy for imagej to recognize, then you can help it by giving the regex pattern for the filenames. Click OK to make the image sequence as a video.

Once the video is ready, go to File → Save As → avi to save the image sequence as a video. You can edit the frame rate to a small number like 2 to play the video slowly if you wish.

Now your video is ready to be included in a slide presentation using LibreOffice Impress or Microsoft PowerPoint etc.,