# Guidelines for plotting in Matlab

J. Rabault

31st January 2016

## Matlab / Octave / Python

You can use either Matlab, Octave (free copy of Matlab, most of your Matlab codes will run without any modification in Octave and vice versa), Python or any other tool you want. However, here are a few guidelines you should absolutely follow when generating pictures for a report, thesis etc. I give functions examples that correspond to Matlab. Figures that do not follow those guidelines strongly loose in impact. It is a very bad choice not to spend a few minutes to make you figures clear and lisible when you have spent hours to obtain and process the data that is presented.

The Matlab scripts for generating good looking pictures is available through my UiO webpage: http://folk.uio.no/jeanra/Teaching/MEK4600/MEK4600Main.html

#### Labels

All your plots must have accurate labels on the axis (functions *xlabel()* and *ylabel()*). Units must be included. Plots without labels and labels do not have scientific value.

#### Legend and multiple curves

If you have more than one curve on a plot, you must put a legend (function *legend()*). In order to be able to distinguish between different curves, you must set the line style (options example: - -*r* in the plot function).

#### Characters size and lines thickness

For your plots to be easily readable, you must take care of the characters size and line thickness (example: *'LineWidth',Lwdth* option). I advice you to do it through variables (*Lwdth*) to be able to change all your figures quickly at once.

## Scripting

I advise you to script the generation of your figures. You will save a lot of time. Here is an example of how to do it.

## Formatting scripts

Copy this code in a file called ScriptAspectFigures_Pre.m and put it in the folder in which you want to generate your nice looking pictures:

```matlab
%% units conversion cm / pixel
cm2p=0.0282;      % centimetres to pixels

%% aspect ration of phi look best
height=1/1.618;
width=1;

%% select one or two columns size
% one column is 20/39 ratio, two column is 39/39
%
if (nColumnsPlotParam==1)
sml = 20/39;
elseif (nColumnsPlotParam==2)
sml = 39/39;
end
%
scale=16.38/cm2p*2/nColumnsPlotParam; % scale for two-column in pixels
xpos=50;
ypos=500;

%% visual details
axes('FontName','Times New Roman') % Set axis font style
box('on');                         % Define box around whole figure
set(gcf,'Position',[xpos ypos scale*width*sml scale*height*sml])
```

Copy this code in a file called ScriptAspectFigures_Post.m and put it in the folder in which you want to generate your nice looking pictures:

```matlab
%% update the figure
pause(0.5);
drawnow;
pause(0.5);

%% set the font size
set(gca,'FontSize',FtSize);

%% set the size of the figure paper
set(gcf,'PaperUnits','centimeters');
set(gcf,'PaperSize',[width height].*16.38*sml);
set(gcf,'PaperPositionMode','manual');
set(gcf,'PaperPosition',[0 0 width height].*16.38*sml);

%% take away margins
ti = get(gca,'TightInset');
set(gca,'Position',[ti(1) ti(2) 1-ti(3)-ti(1) 1-ti(4)-ti(2)]);

%% use the right units
set(gca,'units','centimeters');
pos = get(gca,'Position');
ti = get(gca,'TightInset');%

```

```
set(gcf, 'PaperUnits','centimeters');
set(gcf, 'PaperSize', [pos(3)+ti(1)+ti(3) pos(4)+ti(2)+ti(4)]);
set(gcf, 'PaperPositionMode', 'manual');
set(gcf, 'PaperPosition',[0 0 pos(3)+ti(1)+ti(3) pos(4)+ti(2)+ti(4)]);

%% choose vector or bitmap
% set(gcf,'Renderer','OpenGL');        % bitmap
set(gcf,'Renderer','Painters');        % vector format
set(gcf,'RendererMode','manual');      % so that the Renderer is manual

%% save the plot
% generate the name with extension
if (nColumnsPlotParam==1)
figName = strcat(figName,'_OneColumn');
elseif (nColumnsPlotParam==2)
figName = strcat(figName,'_TwoColumn');
end
% plot
if saveFig
    if saveFig > 1
       print(saveFig,'-dpdf','-r400',figName);
    else
        print(gcf,'-dpdf','-r400',figName);
    end
end
```

## Example of use of the scripts

You can now use those scripts for generating a figure. If you apply all the guidelines, you should get a code that looks like this:

```
clear all;
close all;

%% plot a dummy figure to illustrate all commands

% data to plot
x = 0:0.01:2*pi;
sinx = sin(x);
cosx = cos(x);

% how to plot a clean figure
%
% open a new figure --------------------------------------------------------
figure(1)
% parameters for the plotting ----------------------------------------------
nColumnsPlotParam = 1;
FtSize            = 12;
Lwdth             = 2;
% call the pre script ------------------------------------------------------
ScriptAspectFigures_Pre;
%
% plot the data ------------------------------------------------------------
plot(x,sinx,'--r','LineWidth',Lwdth);
% put several curves together
hold on;
```

```
plot(x,cosx,'.-b','LineWidth',Lwdth);
% add the labels ------------------------------------------------------------
xlabel('x quantity (unit)');
ylabel('y quantity (unit)');
% add the limit for the plot domain ---------------------------------------
xlim([0 2*pi]);
ylim([-1.2 1.2]);
% add the legend ------------------------------------------------------------
legend('curve_1','curve_2','Location','SouthWest');
% parameters for saving the figure ----------------------------------------
figName = 'InterestingFigure';
saveFig = true;
% call the post script -----------------------------------------------------
ScriptAspectFigures_Post;
```

The result you get should look something like that:

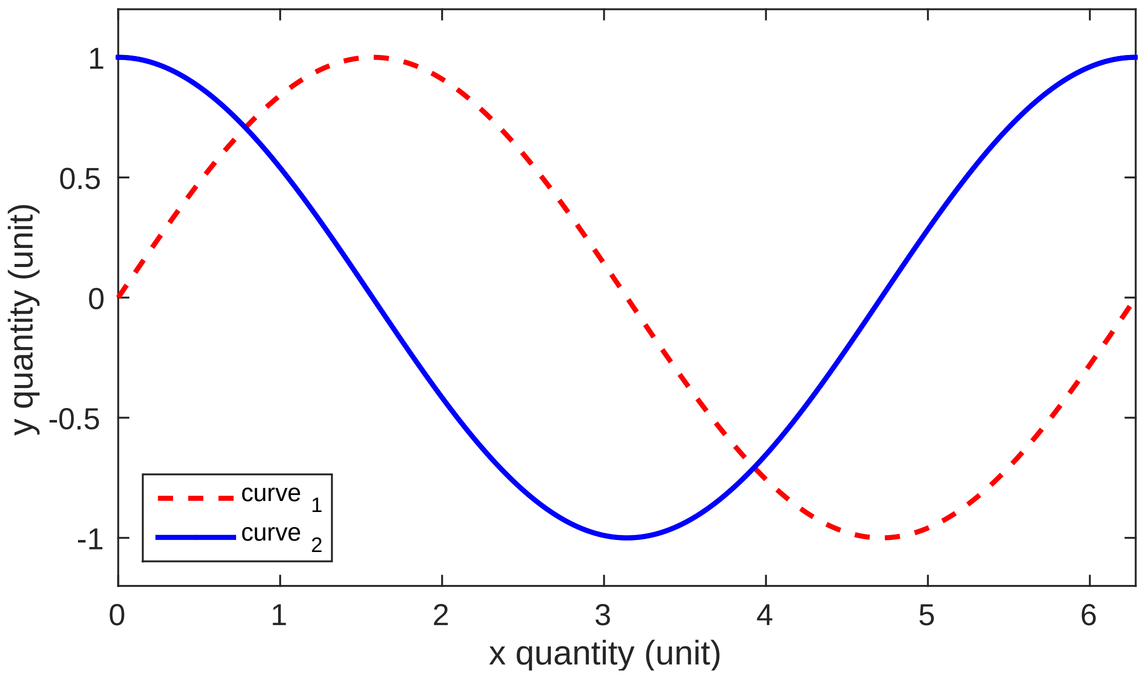

**Figure 1:** An example of plot with good labels, legend and char size.

## Acknowledgment

The code shared by Dr. Graig Sutherland has been a great source of inspiration for writing those scripts.