telecommunications &
signal processing
laboratory

Matlab Plots in Microsoft Word

Peter Kabal

Department of Electrical & Computer Engineering

McGill University

McGill

March 2000

`http://www.TSP.ECE.McGill.CA`

Matlab Plots in Microsoft Word

1 Introduction

This report looks at different options for inserting plots generated from Matlab[1] into Microsoft Word[2]. For publication quality output, it is important to control the size of the graphic that will appear in the final document. If the graphic is drawn at its final size in Matlab, then no further scaling will be necessary. Further scaling is undesirable, as it not only scales the plot, but the text on the graphic as well. This report gives a procedure that can be used in Matlab to control the size of the figure. Results indicate that PostScript is the best option for good quality graphics. Graphics imported using cut and paste from Matlab (EMF format) are noticeably inferior in quality and are reproduced with a scaling error.

2 Matlab Units

Matlab plots consist of a “figure” containing “axes”. The figure are positioned with respect to the screen and the axes are positioned with respect to the figure. These rectangles are specified with four parameters: [*left bottom width height*]. The parameters of the rectangles can be expressed in several different units.

- Absolute units: These are inches, centimetres and points (72 point to an inch).
- Screen units (pixels): The screen is sized as $\begin{bmatrix}1 & 1 & x_{\mathrm{pix}} & y_{\mathrm{pix}}\end{bmatrix}$. For instance for a 1280 by 1024 pixel screen display, x_{pix} is 1280 and y_{pix} is 1024. Experiments show that pixel units at all screen resolutions is converted to absolute units using the conversion 96 pixels equal 1 inch. For a so-call 17-inch (diagonal) display (about 12.75 inches horizontally), the actual resolution is about 100 pixels/inch for a 1280×1024 resolution. Similarly for a so-called 15-inch display, the actual resolution is again near 100 pixels/inch for a 1024×768 resolution.
- Relative units: Relative values are numbers from 0 to 1. Figure dimensions are relative to the full screen size and axes dimensions are relative to the containing figure.

[1] Matlab 5.1 was used for these tests.

[2] Microsoft Word 2000 was used for these tests.

3 Matlab Plot Output Formats

Here we consider two types of output: PostScript (PS) files and Enhanced Metafile (EMF) format. The EMF format output can be copied to the Windows clipboard and then can be pasted into a Word document.

3.1 PostScript Files

For the purpose of this document, we consider encapsulated PS output with and without a preview, and the variant of PS used to create graphics that can be read by Adobe Illustrator. Encapsulated PS files (EPSF) contain a header that specifies a bounding box for the graphic. When a plain PS file is inserted into a Word document, the screen display shows only a box where the graphic will appear in the printout. When the document is printed on a PS printer, the printer interprets the PS corresponding to the plot and the plot appears on the page. When printed to a non-PS printer, the output shows a box where the graphic should appear.

Matlab optionally allows for a *tiff* preview to be included in the PS file. A low-resolution preview appears in the Word document, but the PS code for the graphic is interpreted by a PS printer to give a high-quality output. On non-PS printers, the printed document shows the preview.

3.2 Enhanced Metafile Format

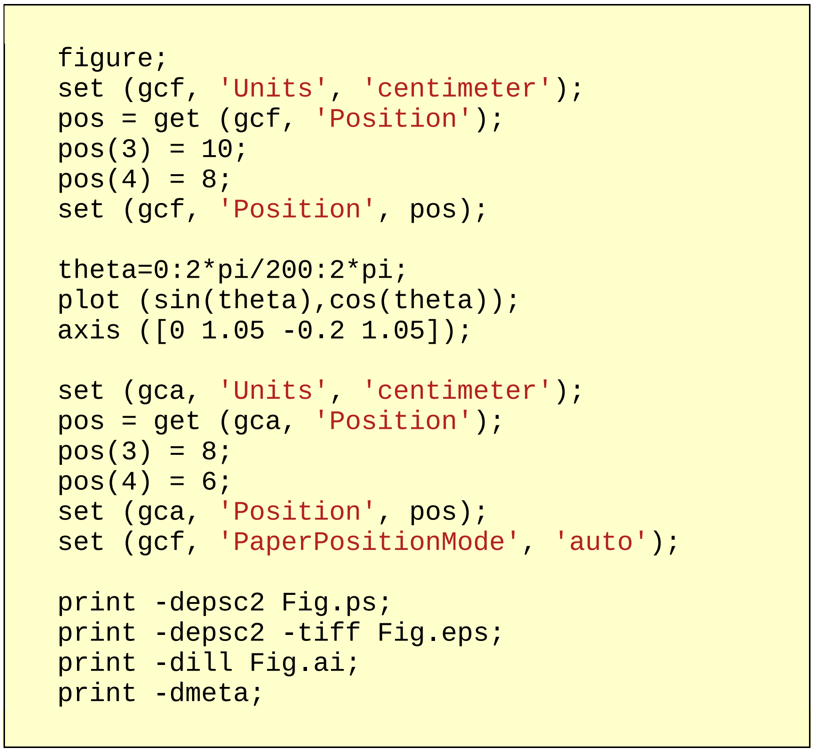
The Extended Metafile (EMF) format is used in Microsoft Windows to exchange graphics. Matlab provides for the figure to be placed on the clipboard in EMF format. The clipboard contents can then be pasted into the Word document.

3.3 Including Matlab Graphics in Word

In the case of graphics files, the *Insert Picture* command places them in the document. In the case of EMF graphics in the clipboard, the Paste Special command places them in the document.

Consider the Matlab script shown in Fig. 1. This script plots a simple function. The figure size is set to 10 cm × 8 cm and the axes in the plot are set to 8 cm × 6 cm. The *PaperPositionMode* is set to ‘*auto*’ to ensure that there is no scaling of the graphic as it is exported. This script creates three output files and leaves an EMF graphic in the clipboard.

```
% Test Matlab graphics export
set (0, 'DefaultAxesFontSize', 8, ...
        'DefaultAxesFontName', 'Times');
```



```
figure;
set (gcf, 'Units', 'centimeter');
pos = get (gcf, 'Position');
pos(3) = 10;
pos(4) = 8;
set (gcf, 'Position', pos);

theta=0:2*pi/200:2*pi;
plot (sin(theta),cos(theta));
axis ([0 1.05 -0.2 1.05]);

set (gca, 'Units', 'centimeter');
pos = get (gca, 'Position');
pos(3) = 8;
pos(4) = 6;
set (gca, 'Position', pos);
set (gcf, 'PaperPositionMode', 'auto');

print -depsc2 Fig.ps;
print -depsc2 -tiff Fig.eps;
print -dill Fig.ai;
print -dmeta;
```

Fig. 1 **Matlab script to generate graphics files.**

In this document, the imported graphics are placed in boxes with no margins. The sizes given in the captions are the sizes reported by Word in the *Format Picture* dialog.

3.3.1 PostScript, no preview

The included file Fig.ps is shown in Fig. 2. Since this file has no preview, all that appears on screen is a box showing the size and position of the graphic. A screen snapshot of the box that appears in Word is shown in Fig. 3. The axes in the graph are just about exactly the 8 cm × 6 cm specified in the Matlab script. The size of the overall graphic is just big enough to contain the plot and the axis annotations.

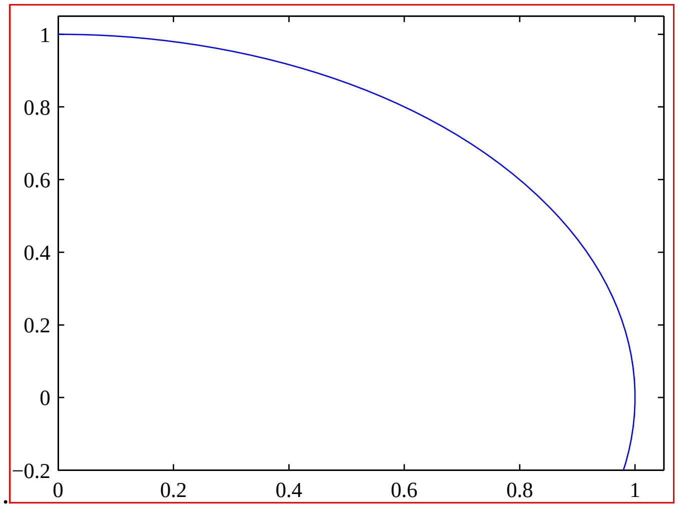
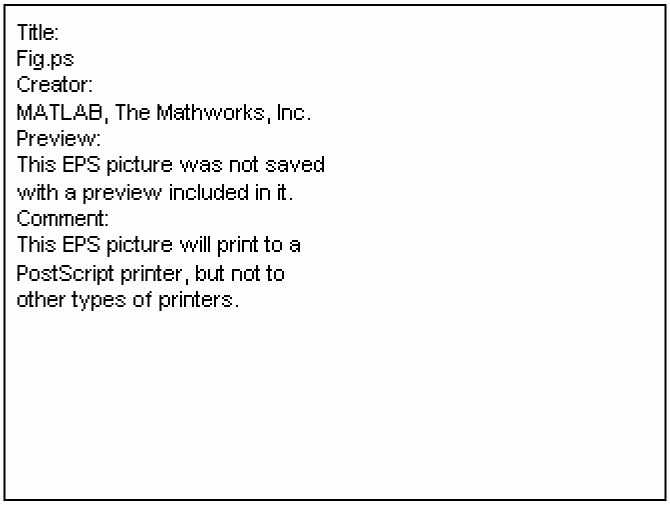


Fig. 2 **Fig.ps (6.56 cm × 8.76 cm): PostScript, no preview.**



Title:
Fig.ps
Creator:
MATLAB, The Mathworks, Inc.
Preview:
This EPS picture was not saved
with a preview included in it.
Comment:
This EPS picture will print to a
PostScript printer, but not to
other types of printers.

Fig. 3 **Screen snapshot of the figure box that appears in Word.**

3.3.2 PostScript, with preview

The included file Fig.eps is shown in Fig. 4. A coarse preview appears in Word. A screen snapshot of the preview is shown in Fig. 5. The axes in the graph are just about exactly the 8 cm × 6 cm specified in the Matlab script.

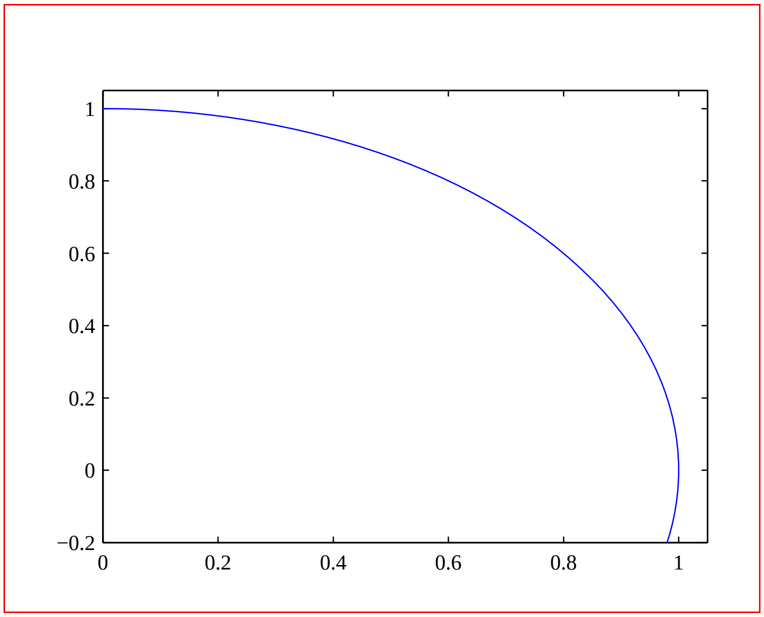


Fig. 4 **Fig.eps (8.02 cm × 9.97 cm): PostScript with preview.**

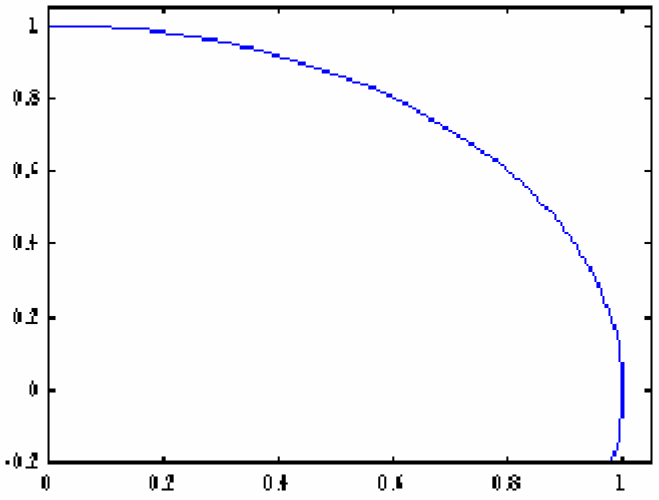


Fig. 5 **Screen snapshot of the preview of Fig.eps.**

A side effect of the inclusion of a preview is that the size of the graphic has changed from one that just encloses the drawing to one with extra white space surrounding the actual drawing. The size of the graphic as it appears in Word is the size of the figure box in Matlab.

3.3.3 EMF format from the clipboard

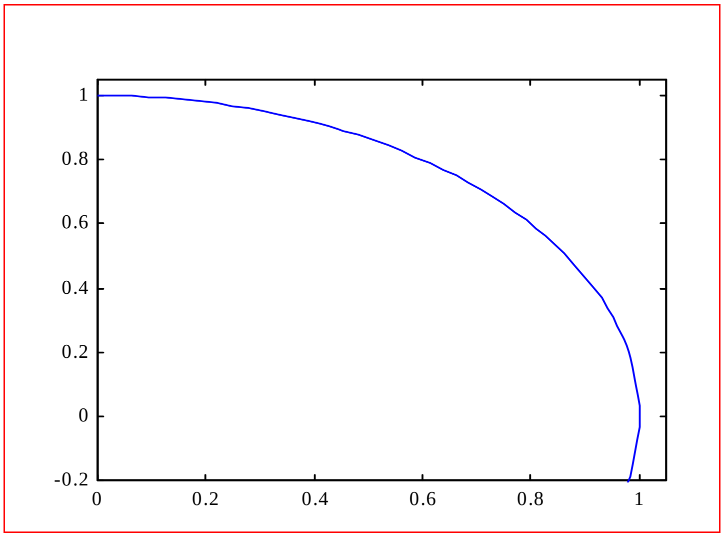


Fig. 6 **EMF graphic (9.45cm × 6.96 cm).**

The EMF graphic put on the clipboard by the Matlab script is shown in Fig. 6. The aspect ratio of the graphic above has been distorted — it is 6 % too small in width and 13 % too small in height. Closer inspection of the printed output shows that the curved line is slightly wavy and has an uneven line thickness. By comparison the PS plots (on paper) appear smooth and even.

3.3.4 EMF format from Adobe Illustrator

To investigate an alternate mechanism to produce EMF graphics, the PS files were read into Adobe Illustrator[3]. Illustrator had problems with the first two PS files (Fig.ps and Fig.eps).[4] The third PS file (Fig.ai) specifically designed for an older version of Adobe Illustrator appears properly in Illustrator. After reading the file into Illustrator, the file was exported from Illustrator as an EMF file that is included as shown in Fig. 7

The size of this graphic agrees closely with the size of the figure size specified in Matlab. The axes in the graph are just about exactly the 8 cm × 6 cm specified in the Matlab script. The EMF file produced by Illustrator has very poor rendition of the text. The curved line is also noticeably wavy, more so than for the EMF graphic produced directly by Matlab.

[3] Adobe Illustrator 8.01 was used.

[4] When saved as PS level 2, Adobe Illustrator was not able to parse the Encapsulated PS files. When saved as PS level 1, the minus sign on the y-axis label was missing in the Illustrator plot.

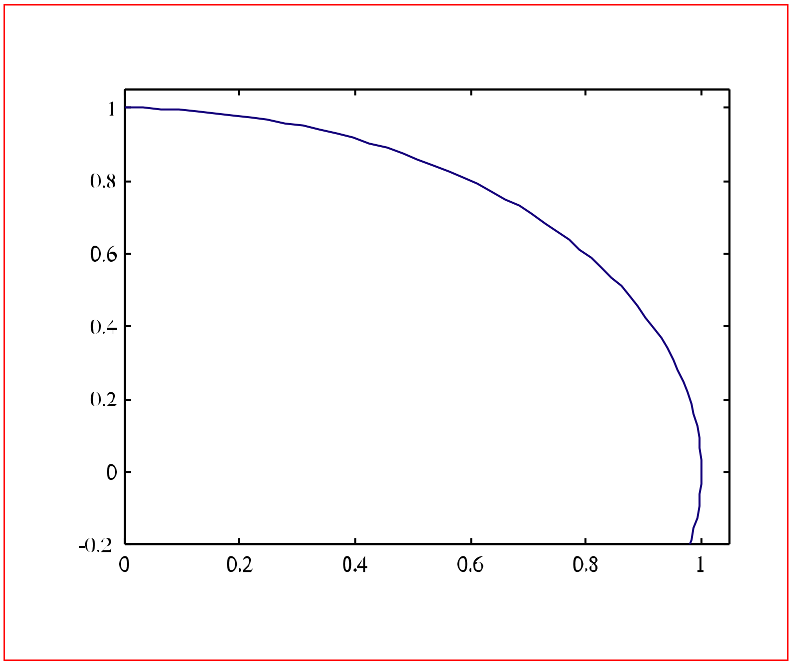


Fig. 7 **EMF graphic exported from Adobe Illustrator (10.05 cm × 8.04 cm).**

An interesting aspect of the graphics produced by Matlab is apparent when the graphic is selected in Adobe Illustrator. A screen snapshot of the selection is shown in Fig. 8.

The figure boundary (outside box) and the tighter bounding box surrounding the axes are visible in this view. We see that the graphics file includes points that fall outside the plotting area, even though they are not rendered. These invisible points are the points that fall between the axes box and the overall figure box. This means that unnecessarily large plot files may result if many points fall outside the axes.

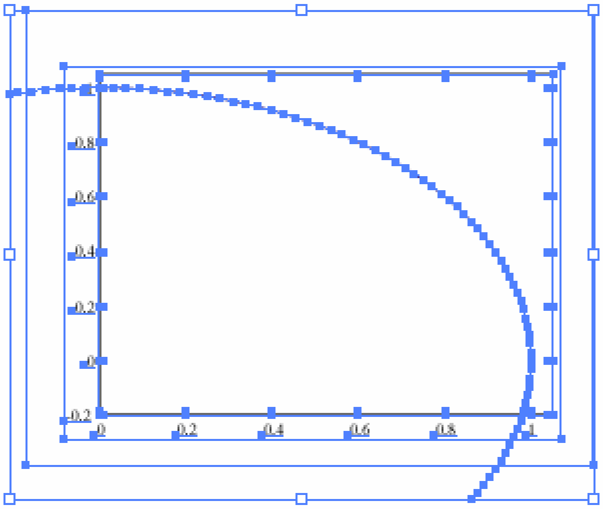


Fig. 8 **Screen snapshot of Adobe Illustrator selection for Fig.ai.**

3.4 Cropping and Scaling Matlab Graphics

Word allows for scaling and cropping of imported graphics. Fig. 9 shows the same EMF graphic that appeared in Fig. 6. It has been magnified by 6 % horizontally and 13 % vertically. Now the axes sizes are close to the values set in the Matlab script. The figure has also been cropped in Word to include less white space. Problems occur in doing so. Cropping too much at the top causes the text line above the figure to be partially cut off. In this example, the top of Fig. 9 (above the crop rectangle) overlays the line of text immediately above it.

The example above shows that cropping in Word is problematic. Fig. 10 shows an EMF graphic, again magnified by 6 % horizontally and 13 % vertically. In this case, the figure size was set in Matlab to be just big enough to enclose the axes and labels. No further cropping was done in Word. Four parameters are needed to specify the size and position of the axes. Appendix A shows the Matlab procedure *SetPlotSize* for specifying these parameters. For the graphic in Fig. 1the plot size was given as `SetPlotSize([0.7,0.6,8,6],'centimeter')`. This specification eliminates the extra white space to the left and bottom of the axes. The space to the right and above the axes was cropped in Matlab using the mouse to resize the figure box.

= =

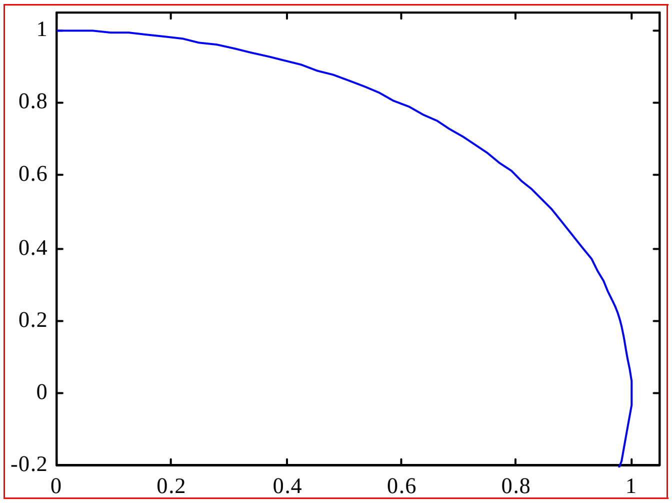


Fig. 9 **Magnified EMF graphic (cropped in Word).**

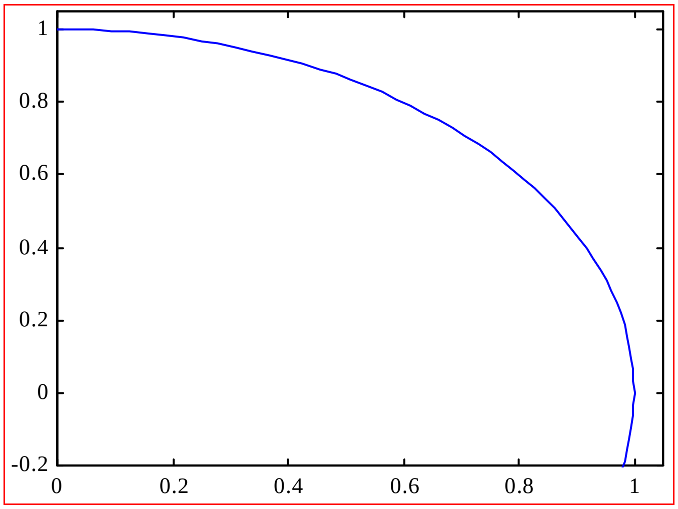


Fig. 10 **Magnified EMF graphic (cropped in Matlab).**

4 Summary

- PostScript plots are rendered well in the final printout, though the screen preview is quite coarse. The size of the included figure is the same as that specified in Matlab. The downside of PostScript is that the PS graphics do not appear on a non-PS printer. One workaround is to convert the document to format that is more portable (such as pdf).

- EMF graphics do not print with as high quality as PS graphics do. EMF produced directly from Matlab is improperly sized. However, the graphic can be scaled in Word to compensate for the changed size and changed aspect ratio. The quality of the printed output for EMF graphics is inferior to that for PostScript, but adequate for most uses. The EMF graphics produced by Adobe Illustrator are properly scaled but have very poor text rendition.

- For all graphic types except plain PS (no preview), the size of the graphic is the size of the Matlab figure. The graphic must be cropped to eliminate excess white space, but cropping in Word can be problematic. The solution is to generate the graphic in Matlab such that it has minimal white space surrounding it. The graphic can then be included in Word without further cropping.

- To guarantee that the exported PS file keeps the same axis ticks and labels as those that appear on the screen, the Matlab *XTickMode* and *YTickMode* plot options should be set to *manual*, e.g., `set(gca, ’XTickMode’, ’manual’)`. In the default automatic mode, the number of ticks and labels can change when the figure size is changed.

- Care are should be exercised to avoid generating plots with large numbers of points that fall outside the plotting axes.

5 Matlab Function *SetPlotSize*

The Matlab function *SetPlotSize* is shown below. The four dimension parameters are the position and size of the axes. The position values have to be determined somewhat by trial and error, since the space taken up by the labels depends on the font size and even the axes values themselves.

```
function SetPlotSize (pos, units)
% SetPlotSize  Set the size of the current plot
%   SetPlotSize(pos,units)
%     or
%   SetPlotSize(pos)        % default units "inches" assumed
%
% pos   - axes position [xsize, ysize] or
%                       [xleft, ybottom, xsize, ysize]
% units - units for the dimensions, default 'inches'.  The choices
%         are 'centimeters', 'pixels', 'inches', 'points', and
%         'normalized'.
% Note: The 'Units’ for the current axes are set to the units
%       specified.  If the units are absolute measurements
%       (i.e. not 'normalized'), then figure can be resized on
%       the screen using the mouse.  This feature can be used to
%       "crop" white space above and to the left of the axes.

% SetPlotSize 2000-02-08

% Notes:
% - The default axes position for Matlab is given in normalized
%   units as [0.130 0.110 0.775 0.815].  These values are used
%   to determine the space around the axes.

if (nargin <= 1)
  units = 'inches';
end

Np = length(pos);
if (Np ~= 2 & Np ~= 4)
  error ('SetPlotSize: pos must have 2 or 4 elements');
end

posnorm = [0.130 0.110 0.775 0.815];
if (Np == 2)
  figsize = pos ./ posnorm(3:4);
  axespos = [(posnorm(1:2) .* figsize) pos];
else
  figsize = (pos(1:2) + pos(3:4)) ./ (posnorm(1:2) + posnorm(3:4));
  axespos = pos;
end

% Set the figure
saveunits = get (gcf, 'Units');
set (gcf, 'Units', units);
figpos = get (gcf, 'Position');
```

```
set (gcf, 'Position', [figpos(1:2) figsize]);
set (gcf, 'Units', saveunits);

% Set the axes
set (gca, 'Units', units);
set (gca, 'Position', axespos);

disp (sprintf ('SetPlotSize: Figure size: %g x %g', figsize));
```