Publication quality figures using matlab: Why I do as much as possible within matlab

Richard E. Turner (rt60@nyu.edu)

Umesh Rajashekar

June 7th, 2011

My Desiderata

- Produce figure for a paper (e.g. for latex)
- Fonts, line widths, etc. to be a specified size (e.g. 12pts)
- Fonts and symbols identical in figure and text
- Automate process as much as possible: figure regeneration common
  - Collect additional data
  - Tweak simulations
  - Reuse the figure in a new publication, talk, grant etc.
  - Minor cosmetic alterations (require small amount of work)
- Reuse visualisation code

My Approach

Don’t

- Export figures using the “export” menu function
- Modify figure properties using the mouse

Avoid

- Using third party graphics manipulation programs where possible

Do

- Use functions and scripts to generate plots: **Reuseability**
- Specify fonts, line styles, axis positions, figure sizes as variables: **Modifiability**
- Export using the print command: **Controllability**

Which format? Vector versus raster graphics

	vector graphics or **line art**	**raster graphics** or **bitmap**
Format	object properties	pixel values
Resolution limited by	screen/printer	image pixel size
File size for line art	small	large
File size for images	massive	large
Editing	illustrator, inkscape, intaglio	photoshop, gimp
File formats	eps, pdf, emf	png, jpeg, emf

Conclusion: Export line art components in EPS or PDF

Demonstrations: Focus on exporting

- `demoFig.m` progressively improves a simple figure in matlab for demo purposes. The outputs are collected into the pdf `demoFig.pdf`

- `Plot1By1.m`, `Plot1By2.m` etc. are shells of functions that I make a copy of and fill in when making a new figure

PSfrag - Latex symbols in figures

- Latex method for replacing text in eps files with latex text
- Add the psfrag package to the latex document and use:

```
\psfrag{text in eps to replace}[l][l]{text in latex to use}
```

- **TIP**: to find the exact text string in the eps file, open it in a text editor and search for it.

`http://en.wikipedia.org/wiki/PSfrag`

Other useful functions and information

- `exportfig.m`

  `www.mathworks.com/company/newsletters/digest/june00/export/`
  `www.mathworks.com/company/newsletters/digest/december00/export.html`

- `export_fig.m`

  `http://sites.google.com/site/oliverwoodford/software/export_fig`

These are two functions that handle exporting in a more WYSIWYG manner, also taking care to export properly. I prefer my method as it is more explicit and gives me more control. But these methods may be better for figures containing mixtures of bitmap and vector components.

- **basic beautification of figures:**
  `http://blogs.mathworks.com/loren/2007/12/11/making-pretty-graphs/`

Other useful plotting functions and options in matlab

- `patch.m`

  - 2D polygon plotting, useful for plotting error bars
- `plotyy.m`

  - different y axes on each side of the plot
- `annotation.m`

  - add arrows, textboxes etc. to your figure
- `legend.m, nudgeLegend.m`

  - add legend to a plot, and tweak its appearance (my code)
- `set(gca,’box’,’off’)`

  - turns the figure bounding box off
- `set(gca,’layer’,’top’)`

  - bring the axis to the top to stop stuff being plotted over the black edges and tick marks
- `set(gca,’TickDir’,’out’)`

  - set the direction of the tick marks to point outwards

-
  ```
  pu = get(gcf,'PaperUnits');
  pp = get(gcf,'PaperPosition');
  set(gcf,'Units',pu,'Position',pp)
  ```
  To set the screen size to be the same as the papersize - more WYSIWYG
- for mixtures of vector and bitmaped graphics - export each part separately and overlay e.g. within latex itself
- transparent background in eps file - comment out lines in the eps file that read: "X X X PR" or "X X X X MP", where X is some number.
- Latex trick: use the layout package to find out how wide the page is for setting figure dimensions.