

## Figures & Axes, Printing & Saving



---

---

---

---

---

---

---

---

## Outline

- Announcements
  - Homework I due Wed. 5PM by e-mail
  - No lecture on Fri. 10/26, rescheduled to Wed. 10/31 at 8AM (free caffeine & carbohydrates)
- Printing and saving
- Summary so far
- More 1D functions
- bar: 1D function, 2D objects

---

---

---

---

---

---

---

---

## Printing and Saving

- Can save figure to a .fig file from the GUI
  - Opening the file (from GUI) will recreate the figure
  - The figure will contain same objects as before
    - can add to the figure or edit objects
- Print through GUI or command line
  - print by itself will send gcf to default printer

---

---

---

---

---

---

---

---

## Exporting graphics

- Can save figures to several standard graphics formats using print
  - print -djpeg fname.jpg will save(gcf) to a JPEG file
    - JPEG (Joint Photographic Experts Group) file is a standard raster file
    - a raster file is a matrix of pixels
    - This means that they have a fixed resolution
      - if you blow up a JPEG, the quality will decline (you will begin to see the pixels)
      - can control the resolution using -r<pixels/inch>
    - JPEGs are extremely portable (can view them in a web browser) and compact
    - Good if your figure is very complex (lots of 2D objects and color)

---

---

---

---

---

---

---

---

## Exporting Graphics

- print -depsc fname.eps will save(gcf) to an EPS file
  - EPS (encapsulated post script) is standard format for saving vector graphics
  - Vector graphics are made up of mathematical objects--lines, Bezier curves, polygons, text.
  - The objects have properties such as line weights, fonts, & colors
  - Because the objects are represented mathematically, EPS files can be scaled without losing resolution
  - They are less portable than JPEGs (need special software like Illustrator, or ghostscript)
  - However, you can edit the file easily

---

---

---

---

---

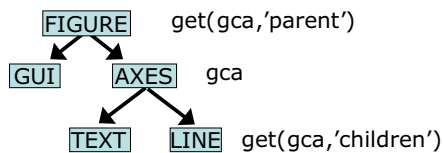
---

---

---

## Handle Graphics Summary

- We've only learned about 3 graphics objects
- But, we now know how Matlab's graphics are organized and how to manipulate them:



---

---

---

---

---

---


---

---

## Handle Graphics Summary

- Objects have properties (like fields in a database or a Java object)
- Each object has a handle (like a name or pointer)
- We can use the handle to examine properties and change them using set and get
- Other objects have new properties, but how we work with them is the same

handle=h



type	line
color	[0 0 1]
marker	none
linestyle	:

---

---

---

---

---

---

---

---

## A Demonstration

- To prove that we understand handle graphics, I will show some specialized 1D plots, and we will try to figure out how they are implemented in Matlab
- For each one, we will answer:
  - What objects are created?
  - What are their properties?

---

---

---

---

---

---

---

---

## Demo

Name	Descrip.	Objects Created	Properties
spy	matrix structure		
semilogy	Log-scale		
polar	polar coordinates		
plotyy	plot against 2 y-scales		
bar	bar plots		

---

---

---

---

---

---

---

---

### What about bar?

- bar represents a 1D function using 2D objects--rectangles
- the rectangles are represented in Matlab as a patch object
  - Patches are polygons
  - Patches can have complicated colors
  - Patches (or related surface objects) are used by all higher-order functions

---

---

---

---

---

---

---

---

### Key properties of patch objects

- edgecolor--color of the edges
- facecolor--color inside the the patch
- Both of these can be set to a specific color (or none)
- Or, we can prescribe another dimension of data at each vertex and let it control the color

---

---

---

---

---

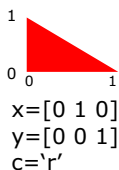
---

---

---

### Drawing patches

- Lots of functions produce patches
- patch is the lowest level functioned (followed closely by fill)
  - patch(x,y,c)--x and y specify vertex coordinates, c controls the color
  - patch(X,Y,C)--Each column of X, Y, and C is a separate patch



---

---

---

---

---

---

---

---