Outline

- Announcements
  - Homework I on web, due Wed. 5PM by e-mail
  - Reschedule Lecture on Fri. 9/26
  - Options:
    - before class on Wednesday (free coffee & bagels)?
    - after class on Wednesday?
    - evening 8-9?
    - Hands-on work?
- Plotting f(x)
- Dissecting plot
- Getting a handle on things
- Example: colortime.m

Plotting f(x)

- Simplest plot is an (x,y) pair
  - boring:
- Simplest interesting plot is f(x) for several x
  - plot as points points or lines:
Plotting \( f(x) \)

- Philosophical details:
  - Is plot of \( f(x) \) 1D or 2D?
  - Math: \( f \) takes one variable, so 1D
  - But, the curve is a 2D object
  - I will take a "data-oriented" view and call this 1D

Points vs. lines?
- Most of the time, we will have vectors \( x \) and \( y \) where \( y \) is \( f \) evaluated (or observed) for every \( x \)
- Plotting with dots represents EXACTLY the info we have
- Plotting with lines implies we know something about \( f(x) \) between \( x_j \) and \( x_{j+1} \)

Plotting \( f(x) \) in Matlab

- Let \( x \) be a length \( n \) vector (1D array)
  - \( x=(0:49)/49; \)
  - \( x=linspace(0,1,50); \)
- Construct \( y=f(x) \)
  - \( y=sin(2*pi*x^2)/(2*x+0.5); \)
  - Could also load \( x \) or \( y \) from a file
**Plotting f(x) in Matlab**

- Standard call: `plot(x,y,options)`
  - options control color, marker, and line style
    - ’ro’ plots in red (r) with circles at points (o) and a dotted line (:)
    - `plot(x,y)` uses default color (usually, blue)
    - `plot(y)` is `plot(1:n,y)`
    - `plot(X,Y)` (X and Y are matrices) plots one line per column in X and Y

**Output of plot**

- ”PLOT returns a column vector of handles to line objects, one handle per line”
- Huh?
  - handles?
  - line objects?

**Getting a handle on things**

- `h=plot(x,y)` will return a handle to the line—`h`
- Handles are just floating point numbers, but they function as pointers to Matlab graphics objects
- We can use them to get info about objects and to change the objects’ properties
Getting a handle on things

- Get properties with "get"
  - get(h) -- lists all of the properties of h and their values
  - get(h, property) -- returns the value of the property
- Change properties with "set"
  - set(h) -- lists all of the properties and their default values
  - set(h, property, value, property, value,...) -- changes the values of the properties
- Get is "vectorized" so you can change properties of lots of objects simultaneously

Handle Properties--ALL objects

- The last 18 properties from get(h) are properties that all objects have
- Most important:
  - Parent -- handle to parent object
  - Children -- handles to child objects
  - Type -- tells what it is (e.g. line)
  - Visible -- (on/off) can hide objects
- A few other general properties are used for GUI’s

Handle Properties--line objects

- xdata, ydata, zdata specify the points
- color describes color of the line segments
  - specify with a "colordef"
    - a special character ('r', 'g', 'b', 'k', etc.)
    - RGB vector (1-by-3 with numbers between 0 and 1)
- linestyle -- controls how line segments look
  - '-'=solid, ':'=dotted, '--'=dashed, 'none'=no lines
- linewidth -- thickness of line (a double)
Handle Properties--line objects
- marker--marker type
  - 'o'=circles, 'x'=x's, '+'=crosses,
  - 'p'=pentagrams, 's'=squares, '^'=triangles
- markerfacecolor--color of the inside of the marker
- markeredgecolor--color of the outside of the marker
- markersize--size of marker

Example--Representing time with color
- We have a function \( y=f(x,t) \) sampled at discrete times
- We want to plot \( y \) for each \( t \) as a different color
  - the colors should correspond to \( t \) and vary continuously
- We will implement this as a Matlab function "colortime.m"

Development of colortime
- 1) Identify inputs and outputs to function

<table>
<thead>
<tr>
<th>variable</th>
<th>size</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inputs:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>x</td>
<td>m-by-1</td>
<td>row labels</td>
</tr>
<tr>
<td>t</td>
<td>1-by-n</td>
<td>column labels</td>
</tr>
<tr>
<td>Y</td>
<td>m-by-n</td>
<td>data matrix s.t. ( Y(k)=f(x(i(k)),t(k)) )</td>
</tr>
<tr>
<td>Outputs:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>h</td>
<td>n-by-1</td>
<td>handles to lines representing ( Y(x,i(k)) )</td>
</tr>
</tbody>
</table>
Development of colortime

• 2) Top-down design using pseudocode
  – Like outlining a manuscript
  – First, identify key steps, describe in English
  – Then, figure out how to implement each step in code
  – Steps may be complex enough to warrant further top-down refinement (recursion)