Handle Graphics and 1D Primitives

Some Handel graphics

Outline

- Announcements
  - Homework 1 on web, due Wed. 5PM by e-mail
- 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:

![Plot of function](image1.png)  ![Plot of function](image2.png)
Plotting $f(x)$

- 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)
  - $x=(0:49)/49$;
  - $x=\text{linspace}(0,1,50)$;
- Construct $y=f(x)$
  - $y=\sin(2\pi x^2)/(2x+0.5)$;
  - 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 (blue)
  - plot($y$) is plot(1:length($y$),$y$)
  - 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

  • 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
      • types vary with property (some are texts, some are arrays)
  • 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
  • set is "vectorized" so you can change properties of lots of objects simultaneously
Getting a handle on things

- Example:
  - `h=plot(x,y);`
  - `get(h,'color')`
    - `ans = [0 0 1]`
    - [ red, green, blue]
  - `set(h,'color','r')` or
    - `set(h,'color',[1 0 0])`
    - changes color of line to red

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>Inputs:</th>
<th>variable</th>
<th>size</th>
<th>description</th>
</tr>
</thead>
<tbody>
<tr>
<td>x</td>
<td>m-by-1</td>
<td></td>
<td>row labels</td>
</tr>
<tr>
<td>t</td>
<td>1-by-n</td>
<td></td>
<td>column labels</td>
</tr>
<tr>
<td>Y</td>
<td>m-by-n</td>
<td></td>
<td>data matrix s. t. $Y(k)=f(x(k),t(k))$</td>
</tr>
</tbody>
</table>

| Outputs: | h        | n-by-1 | handles to lines representing $Y(x(k),t(k))$ |
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)

Development of colortime

• 1) Check that inputs are OK
• 2) Plot the lines with the appropriate colors

Development of colortime

• 1) Check that inputs are OK
• 2) Plot the lines with the appropriate colors
  – Plot the lines
  – Change the colors
Development of colortime

1) Check that inputs are OK
2) Plot the lines with the appropriate colors
   - Plot the lines
     ```matlab
     h=plot(x,Y);%matlab plots a line for each column in Y
     ```
   - Change the colors
     ```matlab
     for j=1:n
       set(h,'color',RGB(j,:));
     end
     ```