

12.010 Computational Methods of Scientific Programming

Matlab Lecture 3

Lecturers

Thomas A Herring

Chris Hill

Summary of last class

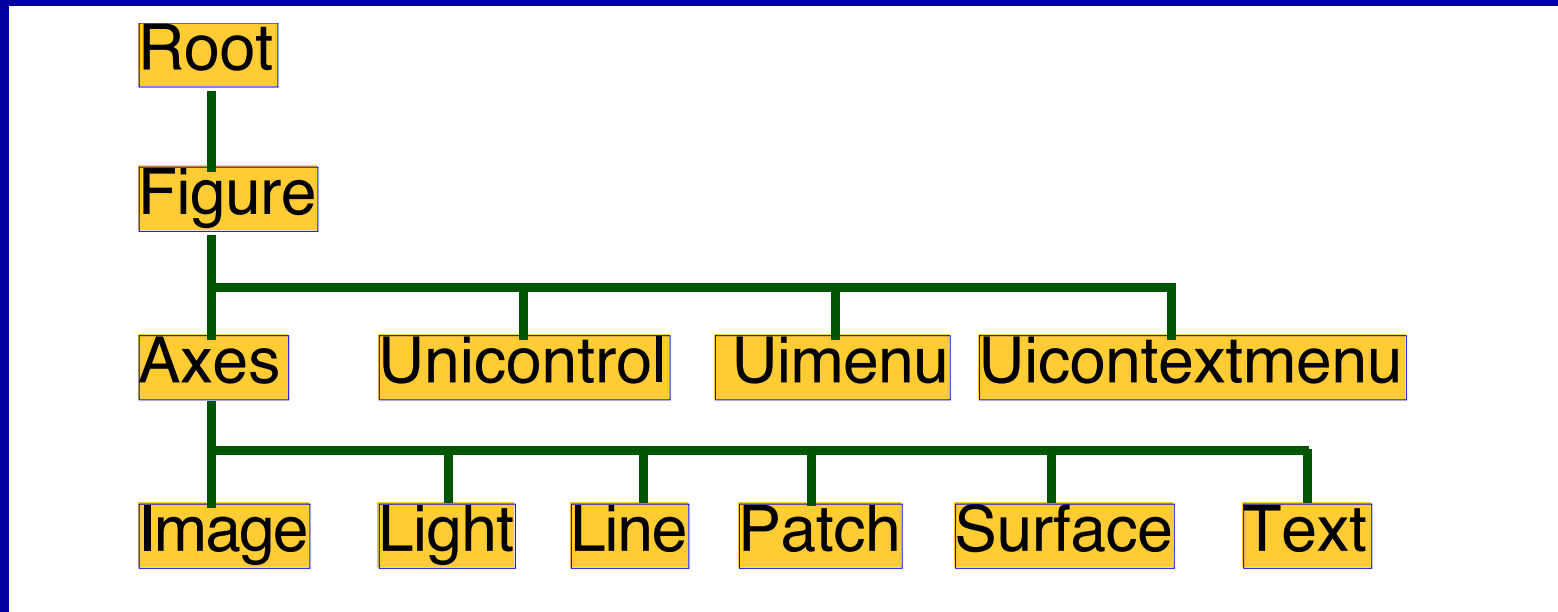
- Continued examining Matlab operations
- path and addpath commands
- Variables and constants
- IO using fopen, scanf etc.
- Formats
- Dialog boxes
- Much of the lecture is spent actually using these features in the M-files that are included with the lecture.

Summary of Today's class

- Examine graphic object handles and their use
- Look at graphics animation (example and needed for homework).
- Start the design process for a Matlab program to analyze GPS Time series results
- This lecture will be given partly by running Matlab and will look at the functions from Matlab that we could use.

Graphic Objects in Matlab

- Matlab refers to components of a graphics object by handles (see Handle Graphics Objects Browser in main Matlab help)



Basic functions of Graphics Handles

- Root: Top of hierarchy, corresponds to computer screen
- Figure: Windows on the root screen other command window (may be an actual figure but can also be a dialog/message/etc box)
- Axes: Objects that define a region of a figure and orient their children in this region
- Uicontrol: control objects that execute a function (pushbuttons, sliders etc)
- Uimenu: Interface menus that reside at the top of a figure (default is File Edit View etc)
- Uicontextmenu: Interface menus that appear when an object within axes is “right-clicked” (e.g., to change a line type)
- Matlab 7: Introduced uipanel for button panels and other user interface (ui) types such as uitoolbar, iucontainer, uitree, uitable.

Graphics Handles

- Image: Objects of a rectangular array that are displayed with a color map
- Light: Object that defines light sources
- Line: primitive graphics on 2-D plots (includes marker)
- Patch: Filled polygons with edges
- Surface: three dimensional representations of matrix data created by plotting heights above the x-y plane
- Text: Character String objects.

Graphics Handles

- The contents of current handles are obtained with the `get` command.
- The `set` command allows the object properties to be changed. Many of the options passed into e.g. the `plot` command are automatic ways of setting certain properties such as line type and color
- The root graphics handle is always 0 (try `get(0)` to see the properties of the root.
- For designing a GUI interface, the interactions between the objects need to be carefully considered.

Callbacks

- An important feature of the Matlab objects is their callback function.
- The callback defines the actions that are taken when an event occurs (such as a mouse click or window resize).
- If the callback function is simple (a single Matlab command) then it can be directly put into the callback properties
- More complex operations usually invoke a user defined function. Often this is one function which uses switch to test the case passed. This method is called switchyard programming. Call back of the form 'mygui case' is interpreted as call to mygui('case');
- Complex call back can:
 'Callback' ,{@mygui, mydata1, mydata2});
 mygui(cbo, eventdata, mydata1, mydata2) is call created.
- (See uicontextmenu menu for example of changing line styles)

Units Properties

- Getting the units properties correct is one the trickiest issues in designing interfaces
- The units property can take on the values of: pixels (default), points, inches, centimeters, normalized, characters (points and pixels are currently the same size since there are 72 of each to an inch)
- Two main issues occur here:
 - What will the interface look like on different sized monitors
 - What will happen if the figure is resized (e.g. axes units default to normalized so that they change size when the figure is scaled)
 - The ResizeFcn callback allows control over the actions when a resize happens
- Matlab help says units should always be changed back to default.

Position Property

- The Position property is measured from the lower left hand corner of the screen.
- Position is a four element array [Left Bottom width height]
- The problems with GUI's during resize is the interpretation of the position property which sets position and size.
- If its units are non-normalized, then it will stay fixed (and possible be hidden by axes or move out of the figure)
- If its units are normalized, then text usually remains the same size and may not fit in box.

Example of setting properties

- In this example we set properties to show how an graphic can be animated.
- The following links to http://geoweb.mit.edu/~tah/12.010/Matlab/Lec03_movie.m and http://geoweb.mit.edu/~tah/12.010/Matlab/per_func.m M-files give an example of an animated sequence (the per_func defines how the object moves as a function of time).
- We will play with the erase mode: Choices are `EraseMode` {normal} | none | xor | background
- Normal is the fastest in newer versions of Matlab and most accurate

Time series program design

- Display GPS time series of North, East and Height positions in a way that problem in the data and analyzed and fixed.
 - The main features we would like are:
 - Display of time series
 - Remove linear and maybe other trends from the data
 - Ability to mark times at which there are breaks in the time series
 - Ability to record times at which there are bad data in the time series. The latter may be individual points or blocks of time.
 - Concatenate files together (Pre- and post-earthquake times series)

Functions that will be needed

- Most of this lecture will be spent using Matlab to look at the types of functions that will be needed for the task and how we go about finding these functions
- Some types of functions we will need:
 - Conversion of decimal year to yr month day
 - Selection of points on a plot
 - Fitting programs that will allow polynomials and other functions to be removed from data
 - Ability to compute statistics of a plot.

Graphical interface

- We would like to implement this system into a Graphical User Interface (GUI). This aspect will be treated in Matlab Lecture 4, but we will start on the process today.
- Basic process design: Desirable features in the interface
 - Selection of time series to display
 - Once it is displayed use call backs from the graphical interface to do such operations as:
 - Detrend data
 - Select portions of the data
 - Zoom and pan image of data
 - Display statistics of data

Functions needed for interface

- We now look at a series of functions and use Matlab to see how these functions operate
- `datetime` and associated functions – conversion of different date types
- `std` standard deviation of data
- `ginput` graphic coordinates
- `polyfit` fits polynomials to data
- `polyval` evaluates a polynomial
- `logical` sets a variable or array as logical which allows data selection
- Other functions that could be useful?

Implementation

- We will now look at these features in the GPSanal.m and gpsgui.m M-files.
- GPSanal.m defines all the graphics handles that we will need to access the information in graphics. This M-file was generated by the Matlab GUI tool `guide`
- Graphics Handles used are:
 - GCF : Graphics Current Figure
 - GCA: Graphics Current Axes
 - GCBO: Graphics Call Back Object
 - GCBF: Graphics Call Back Figure
- The example M-files will show how these are used.

Summary

- Examine graphic object handles and their use
- Look at graphics animation (example and needed for homework).
- Start the design process for a Matlab program to analyze GPS Time series results
- This lecture was be given using Matlab and we looked at the functions from Matlab that we could use.

MIT OpenCourseWare
<http://ocw.mit.edu>

12.010 Computational Methods of Scientific Programming
Fall 2011

For information about citing these materials or our Terms of Use, visit: <http://ocw.mit.edu/terms>.