Matlab
lesson 2: data input/output and graphics

Instructor:
ir drs E.J Boks
Elecrical Engineering
Embedded Systems Engineering
phone:(026) 3658173
Room: D2.03
e-mail:ewout.boks@han.nl

Lesson 2
Graphics

Instructor:
ir drs E.J Boks
Elecrical Engineering
Embedded Systems Engineering
phone:(026) 3658173
Room: D2.03
e-mail:ewout.boks@han.nl

Contents

- Q&A about last session
- Controlling input and output formatting
- Graphics plotting and formatting
- Editing plots
- Graphics printing and handling
- GUI design with Matlab
- Animations in Matlab

Answers to last week

```
% exercises session 1

% Matlab course


% set of equations

A = [2 3 -2;6 5 3; -5 -3 -7];

b = [5 -12 4]';


% solutions for Ax = b


% solution 1 : using the inverse of the A matrix

% x = Ainv*b

Ainv = inv(A);

x1 = Ainv*b
```

Answers to last week

```
% solution 2 : using determinants (Cramer's rule )

% compute the determinant

Adet = det(A);

% first sub solution : substitute b for first column of A

% now x1a = det(Atemp)/det(A)

Atemp = [b A(:, 2:3 )];

x2(1) = det(Atemp)/Adet;

Atemp = [A(:, 1 ) b A(:, 3 )];

x2(2) = det(Atemp)/Adet;

Atemp = [A(:, 1:2 ) b ];

x2(3) = det(Atemp)/Adet;

% solution 3 using rref

x2

Aref = rref([A b]);

x3 = Aref(:, 4)
```

Answers to last week

```
 % solution 3 using rref


Aref = rref([A b]);


x3 = Aref(:, 4)
```

Numeric output formatting

- Format command can be used to format variable output:
- Format short or format short e set output format to 5 digits with or without exp format.
- Format long or format long e set output format to 15 digits, with or without exp format.
- Format hex sets output format to hexadecimal form.

Other input and output commands

- Suppressing output: add a semicolon (;) to your command.
- Add three periods (...) to your command to enable multi-line input.
- Use your cursor keys to repeat or re-edit your previous inputs.
- Use the TAB key to auto-complete a filename or command – similar to C shell command completion under Unix.

Graphics
Basic function plotting

- Matlab has a very powerful graphics engine to visualize data or function output.
- See on-line help library by typing 'help graphics'.
- Use graphics engine to present output data from your calculations. This is more accessible than raw data output in your command window.

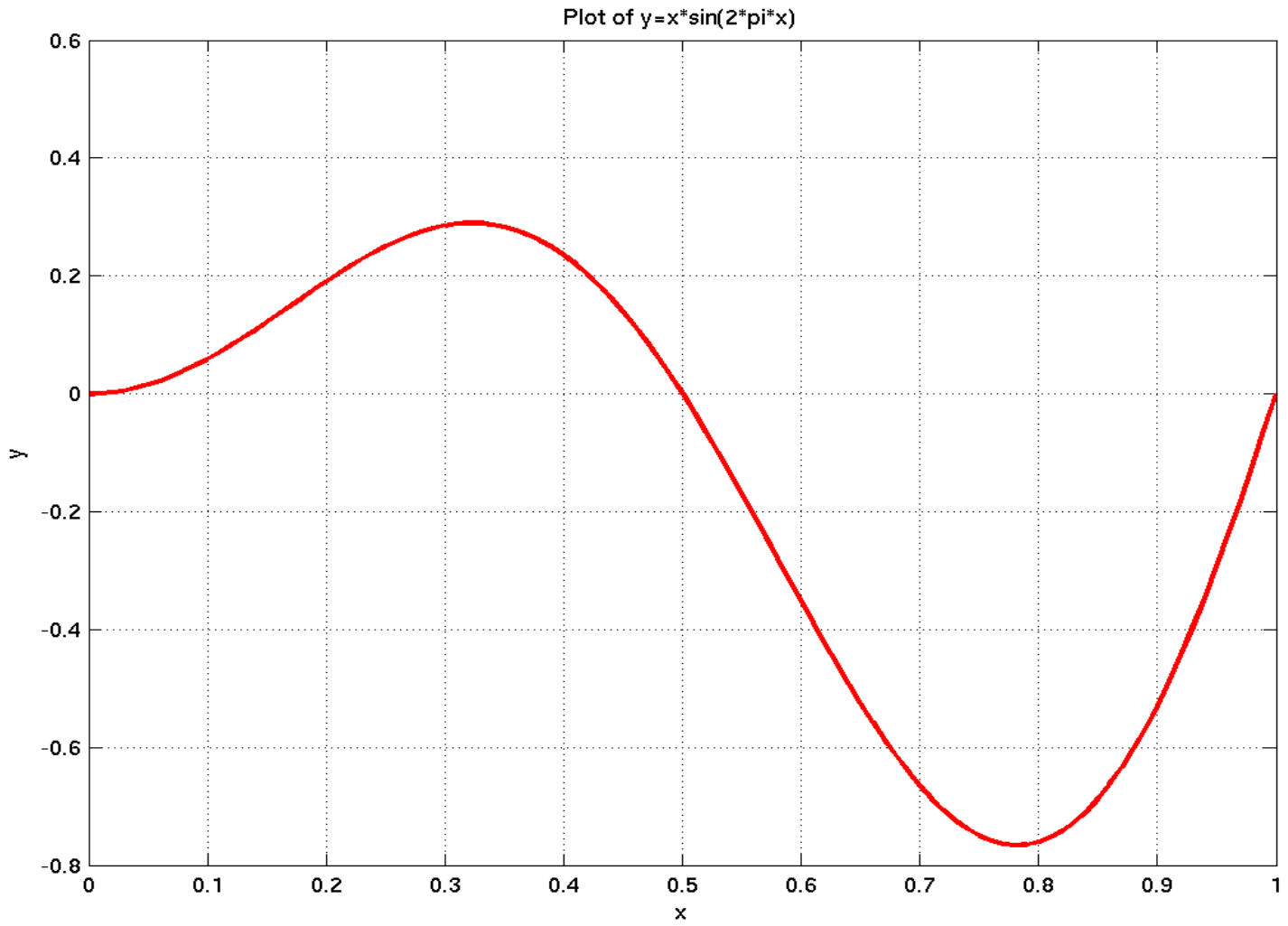
Basic plotting

- Plot function:

  plot(a_x,a_y,'colour style marker', b_x,b_y,'colour style marker',
  c_x,c_y ,'colour style marker', ..)

- An example. To plot a modified sine funtion as a red dash dotted line, enter the cmd: plot(x,x.*sin(2*pi*x),'-.r','LineWidth',2) . Do not forget to define an array x first.

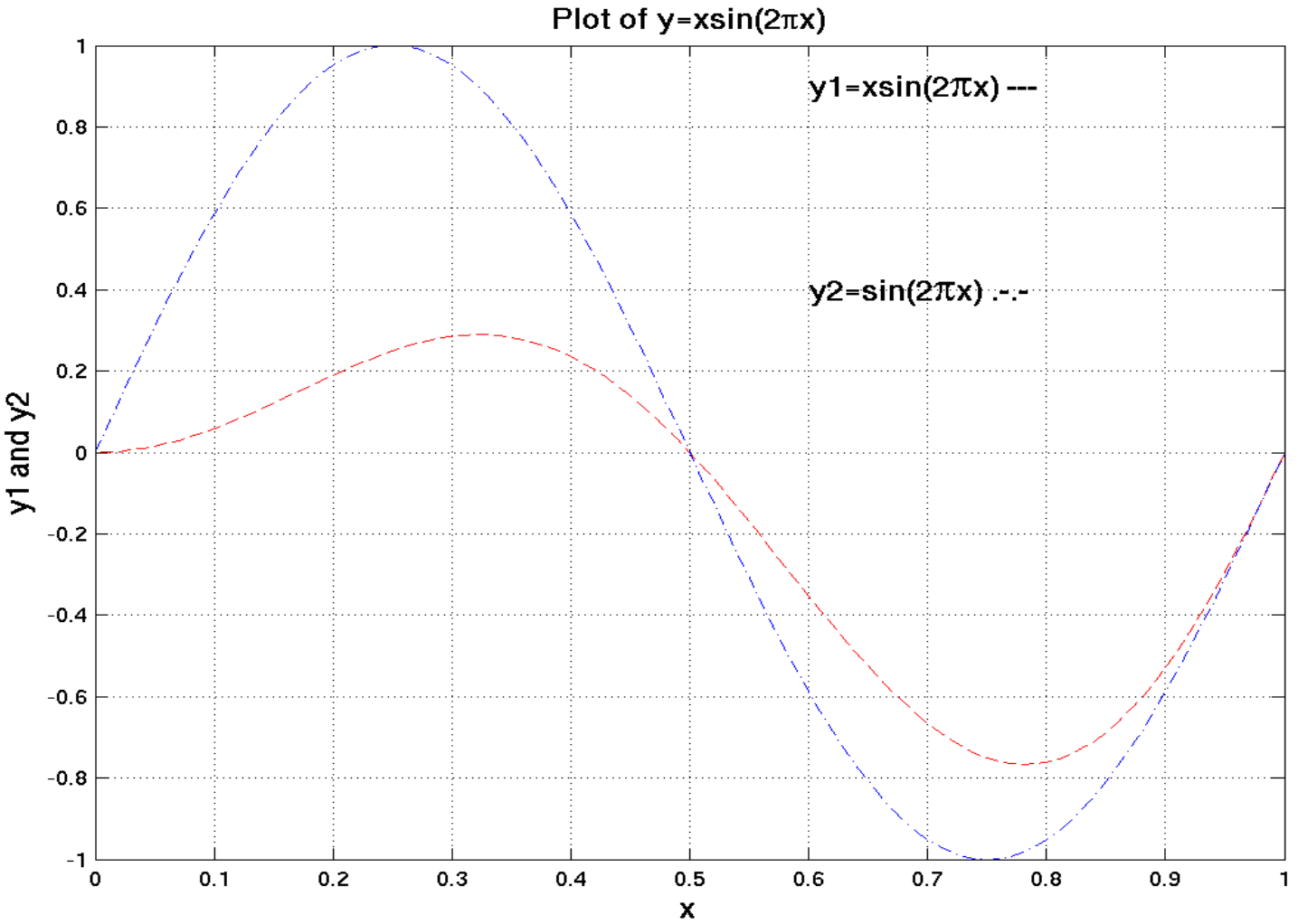
The x*sin(x) plot



Labelling and formatting

- Axis labels and titles: xlabel, ylabel and zlabel add labels to the plot axes.
- These labels are of the form label('text') or label('text', property1,property2 etc).
- For adding additional text to the figure, issue the text(x,y,'string') cmd. The parameters x and y constitute the coordinates where the text should go.
- The axis cmd controls the range of the axes.
- The grid cmd displays a grid across the plot.

A labelled Plot

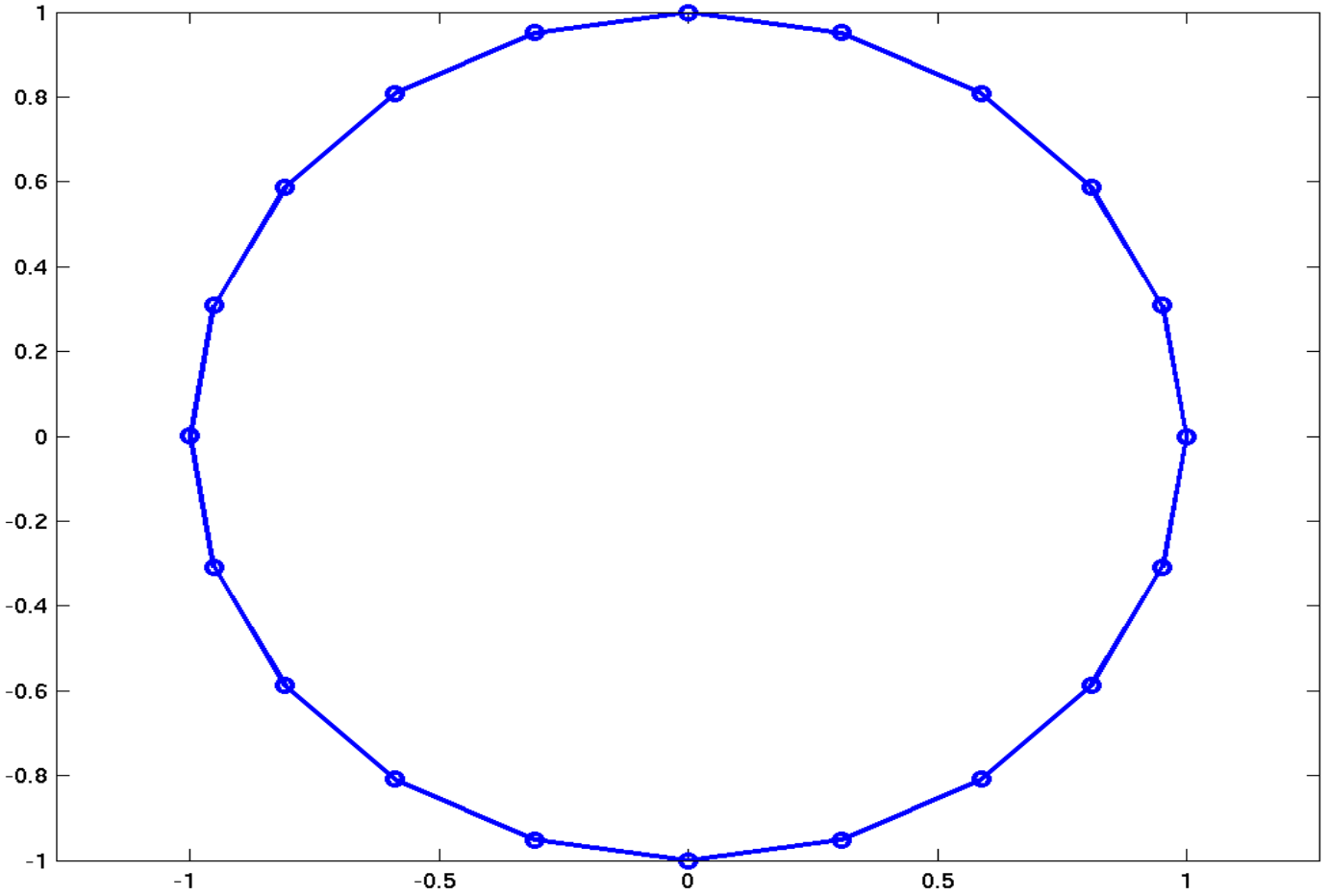


Complex data plotting

- Type 'help plot' to view a complete list of formatting options

- Plotting imaginary data: imaginary part is ignored unless argument is a single complex number, e.g plot(Z) where Z=x+jy or $Z=e^{j\omega t}$

Complex plot



logarithmic data plotting

plots can be made to use a logarithmic scale :

- semilogx( ) is similar to plot but uses a logarithmic x-axis.
- semilogy( ) is similar to plot but uses a logarithmic y-axis.
- loglog( ) is similar to plot but uses a logarithmic x and y axis.

Figures

- Adding plots to an existing graph: issue cmd hold on.

A plot resides in a so-called figure. Multiple plots may be presented using multiple figures.

- To create a figure, enter figure
- To select a figure window, type figure(n)
- Reset a graphics figure by typing clf
- To reset all figures, enter clf reset.
- For more information, type help figure

Subplots

In addition to having multiple figures, it is also possible to display multiple plots in one figure. These plots are called subplots.

- Multiple plots are possible in one figure window by using the subplot cmd in lieu of the plot cmd.
-

Discrete data plots

When displaying discrete data sets (ie sampled data), using normal plots to display the data produces mostly mixed results.

A better alternative is the stem command. stem(x,y) display a stem from (x,0) to (x,y).

Stems can be customised exactly like the plot cmd. Stems can be combined with normal linear plots in one figure.

A stairstep plot can be created using the stairs(x,y) cmd. This displays a sample and hold technique on the data.

Stairs graphics example

Example:

```
alpha = 0.01;
beta = 0.5;
t=0:10;
f = exp(-alpha*t) .* sin(beta*t);
stairs(t,f)
hold on
plot(t,f,'--*')
hold off
```

Direction and contour vector graphs

With the compass plots, a polar plot can be drawn. Using the quiver cmd, a vector is shown in a two dimensional space.

Example:

```
n=-2.0:0.2:2.0;
[x,y,z] = peaks(n);  % generate random data using peak
contour(x,y,z,10)
```

This draws a contour plot. After computing the gradient of the vectors, we can display the vectors:

```
[u,v] = gradient(z,0.2);
hold on
quiver(x,y,u,v)
hold off
```

A contour is an isoline plot. These can be produced with the contour cmd.

Additional plotting methods

In addition to normal linear and discrete plots, other plotting methods are available:

- Bar and Area graphs
- Pie charts
- Histograms
- Interactive plotting

3D plotting

Matlab offers the possibility of drawing 3 dimensional plots. This is done using the (low level) surf cmd. Surf takes 4 arguments or less which provide the position an colour information of the plot.

The 3D equivalent of plot is the plot3(x,y,z) cmd. X, y and z are vectors of equal length.

Example:

```
t=0:pi/100:15*pi;
plot3(sin(t),cos(t),t)
axis square; grid on
```

Graphics miscellaneous

- Use the *plot editor* to customise your plots. Enter the plot editor by selecting 'Tools->Edit plot' on the plot menu.
- All formatting options are available
- You can move around text and or inserted pictures.
- You can modify line colours and styles.
- Bitmap files of well-known graphics formats (*jpeg, png, tiff, xwd* etc) can be used in the following ways:
- Save your plot as a bitmap file from the editor.
- From the matlab cmd line, use the imread, imwrite and image functions to read,write and display images.

Graphics printing

- All plots can be printed through a standard printing dialogue (in the figure window) or through the cmd window.
- set(gcf, property1, value1, ..., propertyn,valuen)
- print(-device, -options, -filename)

Plot exercise

Produce two separate plots of the following data simultaneously:

1. x = {10..40}

$$y = \ln(20 * \pi * x)$$

2. f = 3 Hz, $\omega = 2\pi f$, t = { 0 .. 2 }, x= 0.5

$$y = 25 * \cos(2\pi * \omega * t - 0.5 * x)$$

Plot exercise 2

We have a baseband carrier signal with a frequency of 1 Mhz. We want to modulate the signal with an audio signal, which (for our purposes) has a frequency of 15 kHz. The modulation methods will be Amplitude Modulation (AM) and Frequency modulation (FM).

The AM modulated signal y(t) of a modulation signal x(t) is :

$$y(t)=(1+b*x(t))*CA*\cos(\omega_c t)$$

The FM modulated signal y(t) of a modulation signal x(t) is :

$$y(t)=CA\cos(\omega_c t+b*x(t))$$

In these formulae, b is the modulation index, CA is the carrier amplitude and ω_c is the carrier frequency.

In order to visualise the modulation in the time domain, we need a plot with three subplots detailing:

- The unmodulated carrier signal
- The AM modulated signal
- The FM modulated signal

Produce these plots with a title, legends etc.

GUI design

Matlab offers a complete GUI design toolkit.

Advantages of this toolkit are:

- easy creation of user friendly, self-contained presentations.
- Platform independent
- Demonstration of a concept

GUI design methodology

Matlab GUI is built using object-oriented methodology.

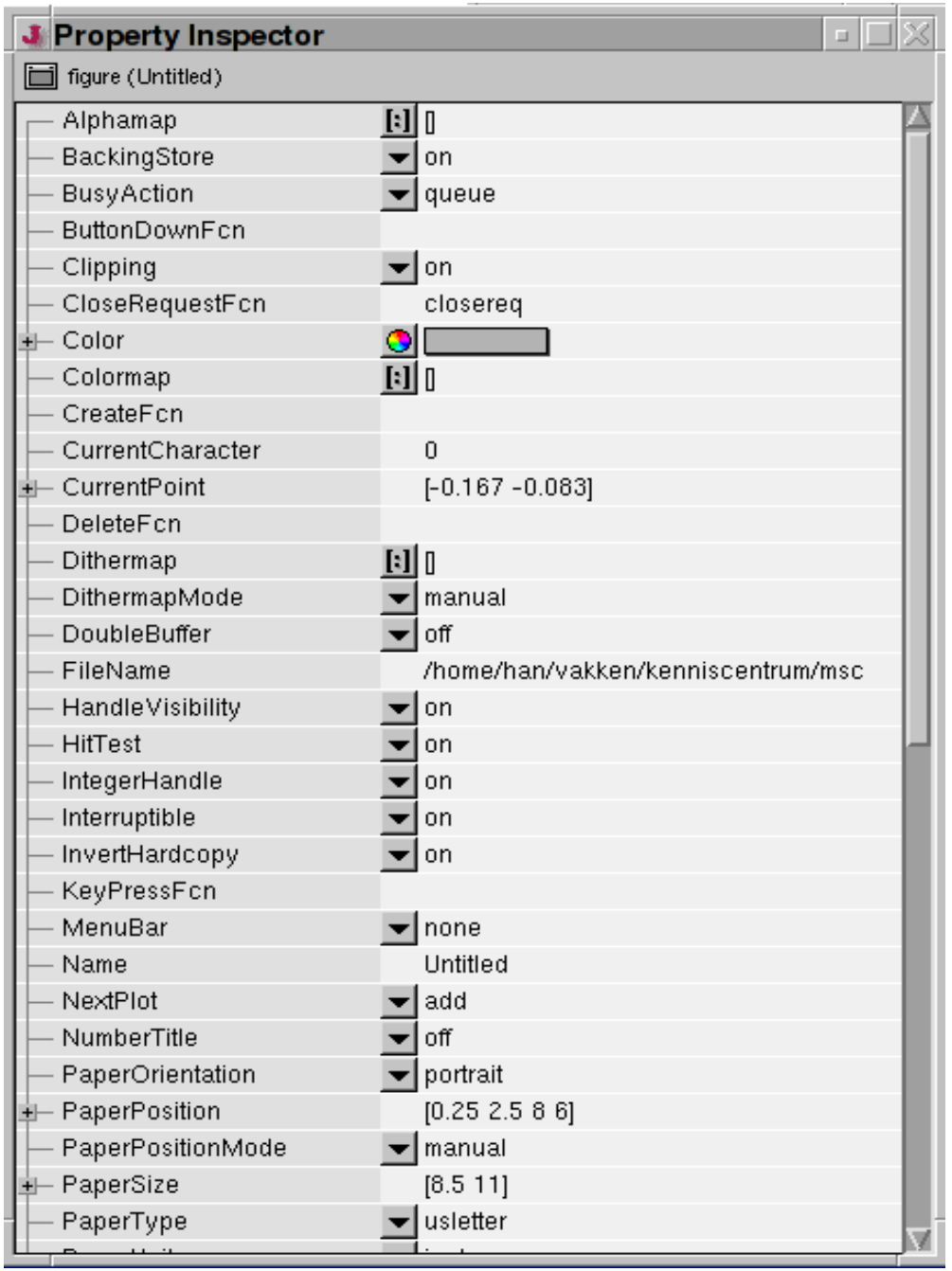
- GUI objects are part of an object tree
- GUI objects generally wait for an event and respond using a call-back mechanism.
- Once a GUI has been designed, the GUI tools write a .m file containing all the code for the GUI and a figure file containg a description of the GUI

GUI objects – uicontrol

- * Check boxes
- * Editable text fields
- * Frames
- * List boxes
- * Pop-up menus
- * Push buttons
- * Radio buttons
- * Sliders
- * Static text labels
- * Toggle buttons
-

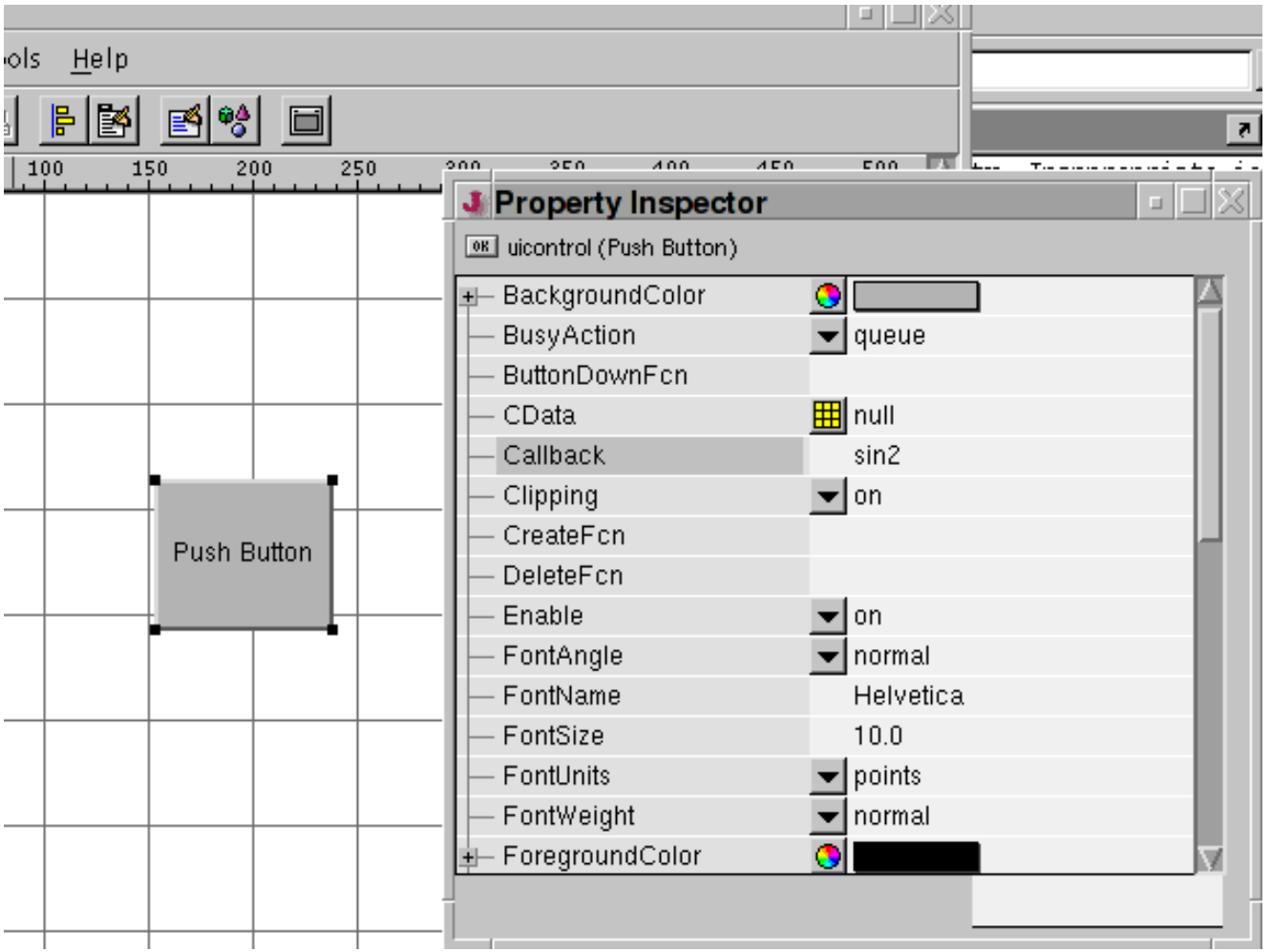
GUI property editor

By clicking on an object, the GUI property inspector appears. This allows you to customise the object to your preferences. Think of it as attributes in a C++ object.



GUI Event callback

Event callback. A routine that executes whenever you activate the uicontrol object (e.g., when you click on a push button or move a slider). Define this routine as a string that is a valid MATLAB expression or the name of an M-file. The expression executes in the MATLAB workspace.



GUI examples

To see a number of Mathworks designed GUIs, type demo at the cmd prompt.

The help documentation has an excellent example for creating a GUI.

Animations

In Matlab, movies can be created by :

- saving a number of frames which are then displayed in a sequence. This is done using the getframe cmd
- Drawing, recalculating, erasing the screen and redrawing it at a fixed rate.

Animations

An example. Enter this text into a .m file and execute it:

```
Z = peaks; surf(Z)
axis tight
set(gca,'nextplot','replacechildren');
for j = 1:20
   surf(sin(2*pi*j/20)*Z,Z)
   F(j) = getframe;
end
movie(F,20) % Play the movie twenty times
```

Animations 2

A nice matlab animation demonstration is the result: