matlab tutorial

Kai Goebel
Bill Cheetham

goebel@cs.rpi.edu
cheetham@cs.rpi.edu

matlab

1 Start up: type “matlab”

1 Features:
   - matrix manipulation is made easy
     1 (data will be represented in matrices)
   - plotting is made very easy
   - suitable for quick prototyping
Matlab is installed on the RCS machines

1. log into an RCS account
2. or 2: "attach" to the RCS AFS directory structure
   /usr/afs/sw/bin/klog <user-id-on-RCS>
3. provide the appropriate RCS password.
4. From that point, all software available on the RCS system should be available on the local machine.
5. Note: this attachment needs only be performed once per login session (not for every command shell opened)

matlab tutorial

1. - calculations

   >> 1+1
   ans =
       2

   >> a=1
   a =
       1

   >> b=2
   b =
       2

2. - assignment of variables

3. - manipulations of variables

   >> a+b
   ans =
       3
matlab tutorial (2)

1. vectors

1. manipulation of vectors
   (addition)
1. column vector

1. transpose of vectors

```
>> c=[1 2 3]
c =
    1     2     3
>> d=[4;5;6]
d =
    4     5     6
>> c+d
ans =
    5     7     9
>> e=[7;8;9]
e =
    7     8     9
>> e' 
ans =
    7     8     9
```

matlab (3)

1. vector multiplication

```
>> a=122
ans =
    122
>> a*d
ans =
    28    35    42
    32    40    48
    36    45    54
```

1. dot product

1. caveat:
   - check inner dimensions before operation

```
>> c.*d
ans =
    4    10    10
>> c*d
??? Error using * 
Inner matrix dimensions must agree.
```

8/28/2003
matlab (4)
1. creating matrices from existing ones
2. subtracting a constant value from vectors
3. accessing particular matrix elements

matlab (5)
1. inverting matrices
2. matrix power of 2
3. square root of a matrix
matlab (6)

1. Initializing vectors and matrices
   - known length
   - unknown length

```matlab
% my_vector=zeros(1,3)
my_vector =
    0 0 0
% my_matrix=zeros(2,3)
my_matrix =
    0 0 0
    0 0 0
% my_vector=[]
my_vector =
[]
% my_vector=[my_vector,1]
my_vector =
    1
% my_vector=[my_vector,2]
my_vector =
    1 2
```

matlab (7)

1. checking your variables

1. variables are stored until reassigned or when program is terminated

1. IF THEN ELSE statements

```matlab
>> who
Your variables are:
    a    b    d    f    h
    c    e    g
>> a
a =
    1
>> b
b =
    2
>> if a<3
    b=5;
else
    b=4;
end
>> b
b =
    4
```
matlab (8)

1. FOR loops

2. WHILE loops
   (also note use of “<“)

3. negation, OR, AND operators
   (also note prompt)
matlab (10)
1 Plotting one vector vs. the other
1 …using different markers
1 …and adding a legend

```
>> a=[1 2 3] ;
>> b=[1 2 5 6 ] ;
>> c=[5 3 2 1 ] ;
>> plot(b,a,'s--')
>> hold on
>> plot(b,c,'-og')
>> legend('data a','data c')
```

matlab (11)
1 3-D plots
1 arrange viewing direction

```
>> mesh(f)
>> view(-120,30)
```
matlab (12)
1. save larger operations in text file

1. invoke by typing filename without extension

>> hello

matlab (13)
1. using functions in different files
   - for example:
     main function in create_vector.m

1. subroutine in manipulate.m
matlab (14)

1. load and save data files
   >> load iris.dat
   >> plot(iris)
   >> save kaiiris.dat iris -ascii
   >> ts iris.dat
   iris.dat
   kaiiris.dat
   >> print -deps kaiplot
   >> ts eps
   >> kaiplot.eps

1. use of system commands from within matlab

1. print plots to file (or to printer)

matlab (15)

1. HELP!

>> help mesh

MESH - 3-D mesh surface.
MESH(x,y,z) plots the colored parametric mesh defined by
four matrix arguments. The view point is specified by VIEW.
The axis labels are determined by the range of X, Y and Z,
or by the current setting of AXIS. The color scaling is determined
by the range of C, or by the current setting of CMEX. The scaled
color values are used as indices into the current COLORMAP.

MESH(x,y,z) uses c = Z, so color is proportional to mesh height.

MESH(x,y,Z) and MESH(x,y,z,C), with two vector arguments replacing
the first two matrix arguments, must have lengths(z) = n and
length(y) = m where size(x) = size(z). In this case, the vertices of
the mesh lines are the triplets (x(i), y(j), z(i,j)).
Note that x corresponds to the columns of Z and y corresponds
to the rows.

MESH(z) and MESH(z,C) use x = 1n and y = 1m. In this case,
the height, Z, is a single-valued function, defined over a
gnomically rectangular grid.

MESH returns a handle to a SURFACE object.

See also SURF, MESHC, MESHZ, MESH, SHADE and View set Figure, axes, and
surface properties which affect the display of the mesh.

See also SURF, MESHC, MESHZ, WINTERFALL.
matlab (16)

1. Look for commands relating to a keyword

   - lookfor mesh
     MESHQH Generate I and Y arrays for 3-D plots.
     MESHGRID X and Y arrays for 3-D plots.
     MESHGRID Low true if the inputs should be automatically meshgridded.
     MESHGR Convert a list of edges to a graph or matrix.
     MESH2 Mesh hidden line removal mode.
     MESH3 3-D mesh surface.
     MESH3D Easy to use 3-D mesh plotter.
     MESH3D Easy to use combination mesh/contour plotter.
     MESH3D Combination mesh/contour plot.
     MESH3D 3-D mesh with curtain.
     MESH3D Mesh/Del mesh plot.
     MESH3D Plot a mesh with colormap highlighted.
     MESH3D Mesh ordering and separators for a finite element mesh.

   }