Matlab Programming
Plots [1] [2]

Mili I. Shah

September 17, 2009

[1] *Matlab, An Introduction with Applications, 2^{nd} ed.* by Amos Gilat
[2] *Matlab Guide, 2^{nd} ed.* by D. J. Higham and N. J. Higham

2D Plots

Many different type of 2D plots

- `plot`: Basic 2D plot (x-y plane)
- `loglog`: Plot with logarithmically scaled axes
- `semilogx`: Plot with logarithmically scaled x-axis
- `semilogy`: Plot with logarithmically scaled y-axis
- `bar`: Bar graph
- `hist`: Histogram
- `polar`: Polar graph

Basic Plots

- Basic connect-the-dots command

  plot(x,y,*string*)

  ```
  >> x = linspace(-2,2);
  >> y = x.^2;
  >> plot(x,y)
  ```

  Now for a red curve with the markers as stars

  ```
  >> plot(x,y,'r*')
  ```

  For a full list of options type `help plot`
- `loglog` is the same as `plot` except the axes are logarithmic
- `semilogx` is the same as `plot` except the x-axis is logarithmic
- `semilogy` is the same as `plot` except the y-axis is logarithmic
- `hold on` Holds the graph so you can add more
- `xlabel,ylabel,title,legend` Labels for the graph
- `axis auto, axis equal, axis square, axis tight`
- `axis([xmin xmax ymin ymax])`

(Not so) Basic Plots

```
% This produces one of those killer spirographs
% Adapted from:
%  http://www.starchamber.com/paracelsus/matlab/
r1 = rand;
r2 = rand;
r3 = rand;


a1= 0:0.15:314;
a2 = a1 * r1/r2;


x = (r1 - r2) * cos(a1) - cos(a2) * r3;
y = (r1 - r2) * sin(a1) - sin(a2) * r3;


plot(x,y)
```

Try `axis equal`, `axis square`, `grid on`, `axis off`, `hold`

Bar Graph

Creating a bar graph

bar(x,y,*string*)

```
>> year = [1988:1994];
>> sales = round(50*rand(7,1));
>> bar(year,sales,'r')
```

Histograms

Creating a histogram

hist(y,*string*)

```
>> grades = round(100*rand(100,1));
>> subplot(2,1,1), plot(grades)
>> subplot(2,1,2), hist(grades)
```

Polar Coordinates

polar(theta,radius,*string*)

```
>> theta=pi/2:4/5*pi:4.8*pi;
>> r=ones(1,6);
>> polar(theta,r)
```

Basic 3D Plots

- `plot3`: Basic 3D Connect-the-Dots

  plot3(x,y,z,*string*)

- `mesh`: Plots $z = f(x, y)$ with lines that connect the points

  x = stX:incX:endX; y = stY:incY:endY;
  [X,Y]=meshgrid(x,y); mesh(X,Y,Z)

- `surf`: Plots $z = f(x, y)$ similar to `mesh` but the area is colored

  x = stX:incX:endX; y = stY:incY:endY;
  [X,Y]=meshgrid(x,y); surf(X,Y,Z)

plot3

- Similar to the 2D `plot`

```
>> t = 0:0.1:6*pi;
>> x = sqrt(t).*sin(2*t);
>> y = sqrt(t).*cos(2*t);
>> z = 0.5*t;
>> plot3(x,y,z,'r*-')
>> grid on
```

mesh and surf

3D Plots of the form $z = f(x, y)$

- x, y are independent
- z is dependent

Methodology

1. Create a grid in the $x - y$ plane that covers domain

   ```
   x = stX:incX:endX;
   y = stY:incY:endY;
   [X,Y]=meshgrid(x,y);
   ```

2. Calculate the value of z at each point of the grid

   Don't forget these are being done element-wise so use your periods (.)

3. Create the plot

   ```
   mesh(X,Y,Z)
   surf(X,Y,Z)
   ```

mesh

Plots $z = f(x, y)$ with lines that connect the points

```
>> x = -1:3;
>> y = 1:4;
>> [X,Y]=meshgrid(x,y);
>> Z = X.*Y.∧2./(X.∧2+Y.∧2);
>> subplot(2,1,1);
>> mesh(X,Y,Z)
```

surf

Plots $z = f(x, y)$ with lines that connect the points and the area colored

```
>> x = -1:3;
>> y = 1:4;
>> [X,Y]=meshgrid(x,y);
>> Z = X.*Y.^2./(X.^2+Y.^2);
>> subplot(2,1,2);
>> surf(X,Y,Z)
```