Introduction to matlab

Getting Started

- Log on to mentor.cc.purdue.edu
- Setting the PATH variable
- *> which matlab*
- *> export PATH=$PATH:/usr/opt/bin/matlab*
- Matlab prompt :*helpwin, helpdesk, demo ?*
- *help*
- *help [command-name] [language-construct]*

Basic arithmetic

- +,-,*,/,^
- exp()
- sin(),cos() … act on radians.
- log,log10
- E.g 11*(15/11) – 15
- Complex numbers
  - 2 + 3i,
  - angle(x)

plots

- ezplot
  - E.g ezplot(‘x^2 + y^2 – 4’,[-2.5,2.5],[-2.5,2.5]),ezplot(‘sin(t)’,’cos(t)’)
- fplot
  - E.g fplot(‘[x,x^2,x^3]’,[0,10])
- Plot
  - E.g plot(a,b)
- 3D plots
- ezplot3()
  - ezplot3(‘2*t’,’3*t’,’t’,[0,4])
- ezmesh
  - ezmesh(‘x*y’); ezmesh(‘s*cost(t)’,’s*sin(t)’,’t’);
- ezsurf
  - ezsurf(‘s*cos(t)’,’s*sin(t)’,’t’);

Plotting lines and data

- plot(x)
- plot(x,y) ; plot(x,y,’x’)
- plot(x1,y1,x2,y2)
- You can print and save graphs.

Matrices and Vectors

- Solving systems of linear equations.
- +,-,*.
- A\b. inv(A)*b
- Inv(),eig(),det(),A’
- Submatrix - A(i:j,k:l)
- Creating matrices :
  - Zeros(m.n), ones(m,n), diag(),randn(m,n),eye(m,n).
- Component wise operator – ‘.’

Writing your own functions

- .m file
- Matlab looks for the file in the current directory
- Can use *cd* at the matlab prompt
- If, for, while.
- Use *help* for details

Example function
Newton Raphson

```
function[x,k,y] = newton_raphson(fx,dfx,x0,epsilon,maxiters)
% Input - fx is the function inputted as a symbolic string 'fx'
%       - dfx is the derivation of fx also inputted as a string 'dfx'
%       - x0 is the initial approximation
%       - epsilon is the tolerance for the function
%Output - x is the newton raphson solution
%       - k is the number of iterations
%       - y = f(x)
% e.g [x,k,y] =  newton_raphson('x^3 - 0.2589*x^2 + 0.02262*x
- 0.001122','3*x^2 -% 0.5178*x + 0.02262',0,0.0001,20)
%
```

Newton – Raphson (contd.)

```
for k=1:maxiters
 x = x0;
xvals(k) = x;
 x1=x0-(eval(fx)/eval(dfx));
 x0=x1;
 x=x0;
 y=eval(fx);
 if (abs(y)<epsilon), break, end
plot(xvals,’x’)
end
```