function all\_roots=incremental\_search(f,xmin,xmax,Nx)

x=linspace(xmin,xmax,Nx);

all\_roots=[];

for i=1:length(x)-1

if f(x(i))\*f(x(i+1))<0

xr=(x(i)+x(i+1))/2;

root=fzero(f,xr);

disp(root)

disp(f)

disp(xr)

all\_roots=[all\_roots root];

end

end

end

function [x,e] = mybisect(f,a,b,n)

% function [x e] = mybisect(f,a,b,n)

% Does n iterations of the bisection method for a function f

% Inputs: f -- an inline function

% a,b -- left and right edges of the interval

% n -- the number of bisections to do.

% Outputs: x -- the estimated solution of f(x) = 0

% e -- an upper bound on the error

format long

c = f(a);

d = f(b);

if (c\*d > 0.0)

error('Function has same sign at both endpoints.')

end

disp(' x y ')

for i = 1:n

x = (a + b)/2;

y = f(x);

disp ([ x y])

if (y == 0.0) % solved the equation exactly

e = 0;

break % jumps out of the for loop

end

if (c\*y) < 0

b=x;

else

a=x;

end

end

x = (a + b)/2;

e = (b-a)/2;

end

function better\_plot(x,y,LineSpec,LineThickness,xlab,ylab,PlotType,FontSize,gridon)

if (PlotType=="plot")

plot(x,y,LineSpec,'LineWidth',LineThickness);

end

if (PlotType=="semilogx")

semilogx(x,y,LineSpec,'LineWidth',LineThickness);

end

if(PlotType=="semilogy")

loglog(x,y,LineSpec,'LineWidth',LineThickness);

end

if(PlotType=="loglog")

loglog(x,y,LineSpec,'LineWidth',LineThickness);

end

xlabel(xlab);

ylabel(ylab);

set(gca,'FontSize',FontSize);

if gridon

grid on

end

end