

```

% Topic : Bisection Method - Roots of Equations
% Simulation : Pitfall - Slow Convergence of the Method
% Language : Matlab r12
% Authors : Nathan Collier, Autar Kaw
% Date : 2 September 2002
% Abstract : This simulation illustrates the slow convergence of the bisection method of
% finding roots of an equation  $f(x)=0$ .
clear all
% INPUTS: Enter the following
% Function in  $f(x)=0$ 
f = inline('x^2-1');
% Lower initial guess
xl = -1.25;
% Upper initial guess
xu = -0.5;
% Lower bound of range of 'x' to be seen
lxrange = -2;
% Upper bound of range of 'x' to be seen
uxrange = 2;
% Maximum number of iterations
nmax = 30;
% Guess of root for Matlab root function
xrguess=-0.9;
%
% SOLUTION
% The following finds the upper and lower 'y' limits for the plot based on the given
% 'x' range in the input section.
maxi = f(lxrange);
mini = f(lxrange);
for i=lxrange:(uxrange-lxrange)/10:uxrange
    if f(i) > maxi
        maxi = f(i);
    end
    if f(i) < mini
        mini = f(i);
    end
end
tot=maxi-mini;
mini=mini-0.1*tot;
maxi=maxi+0.1*tot;

% This calculates window size to be used in figures
set(0,'Units','pixels')
scnsize = get(0,'ScreenSize');
wid = round(scnsize(3));
hei = round(0.95*scnsize(4));

```

```

wind = [1, 1, wid, hei];

% This graphs the function and two lines representing the two guesses
figure('Position',wind)
clf
subplot(2,1,2),fplot(f,[lrange,urange])
hold on
plot([xl,xl],[maxi,mini],'g','linewidth',2)
plot([xu,xu],[maxi,mini],'g','linewidth',2)
plot([lrange,urange],[0,0],'k','linewidth',1)
title('Entered function on given interval with upper and lower guesses')

% This portion adds the text and math
subplot(2,1,1), text(0,0.8,['Check first if the lower and upper guesses bracket the root of
the equation'])
axis off
text(0.2,0.6,['f(xl) = ',num2str(f(xl))])
text(0.2,0.4,['f(xu) = ',num2str(f(xu))])
text(0.2,0.2,['f(xu)*f(xl) = ',num2str(f(xu)*f(xl))])
hold off

% True solution
% This is how Matlab calculates the root of a non-linear equation.
xrtrue=fzero(f,xrguess);

% Value of root by iterations
% Here the bisection method algorithm is applied to generate the values of the roots, true
error,
% absolute relative true error, approximate error, absolute relative approximate error,
and the
% number of significant digits at least correct in the estimated root as a function of
number of
% iterations.
for i=1:nmax
    xr(i)=(xl+xu)/2;
    if f(xu)*f(xr(i))<0
        xl=xr(i);
    else
        xu=xr(i);
    end
end
n=1:nmax;

% Absolute true error
for i=1:nmax
    Et(i)=abs(xrtrue-xr(i));

```

```

end

% Absolute relative true error
for i=1:nmax
    et(i)=abs(Et(i)/xrtrue)*100;
end

% Absolute approximate error
for i=1:nmax
    if i==1
        Ea(i)=0;
    else
        Ea(i)=abs(xr(i)-xr(i-1));
    end
end

% Absolute relative approximate error
for i=1:nmax
    ea(i)=abs(Ea(i)/xr(i))*100;
end

% Significant digits at least correct
for i=1:nmax
    if (ea(i)>5 | i==1)
        sigdigits(i)=0;
    else
        sigdigits(i)=floor((2-log10(ea(i)/0.5)));
    end
end

% The graphs
figure('Position',wind)
plot(n,xr,'r','linewidth',2)
title('Estimated root as a function of number of iterations')
figure('Position',wind)
subplot(2,1,1), plot(n,Et,'b','linewidth',2)
title('Absolute true error as a function of number of iterations')
subplot(2,1,2), plot(n,et,'b','linewidth',2)
title('Absolute relative true error as a function of number of iterations')
figure('Position',wind)
subplot(2,1,1), plot(n,Ea,'g','linewidth',2)
title('Absolute relative error as a function of number of iterations')
subplot(2,1,2), plot(n,ea,'g','linewidth',2)
title('Absolute relative approximate error as a function of number of iterations')
figure('Position',wind)
bar(sigdigits,'r')

```

Figure No. 1

File Edit View Insert Tools Window Help

File Edit View Insert Tools Window Help

Check first if the lower and upper guesses bracket the root of the equation

$$f(x_l) = 0.5625$$

$$f(x_u) = -0.75$$

$$f(x_l) \cdot f(x_u) = -0.42188$$

Entered function on given interval with upper and lower guesses









