

```

% Topic    : Secant Method - Roots of Equations
% Simulation : Convergence of the Method
% Language  : Matlab r12
% Authors   : Nathan Collier, Autar Kaw
% Date      : 6 September 2002
% Abstract  : This simulation illustrates the convergence of the Secant method of
%             finding roots of an equation  $f(x)=0$ .
%
clear all
% INPUTS: Enter the following
% Function in  $f(x)=0$ 
f = inline('x^3-0.165*x^2+3.993*10^(-4)');
% Initial guess 1
xguess1 = 0.05;
% Initial guess 2
xguess2 = 0.02;
% Lower bound of range of 'x' to be seen
lrange = -0.02;
% Upper bound of range of 'x' to be seen
urange = 0.12;
% Maximum number of iterations
nmax = 7;
% Guess of root for Matlab root function
xrguess=0.01;
%
% 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(lrange);
mini = f(lrange);
for i=lrange:(urange-lrange)/10:urange
    if f(i) > maxi
        maxi = f(i);
    end
    if f(i) < mini
        mini = f(i);
    end
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
fplot(f,[lxrange,uxrange])
hold on
plot([xguess1,xguess1],[maxi,mini],'g','linewidth',2)
plot([xguess2,xguess2],[maxi,mini],'g','linewidth',2)
plot([lxrange,uxrange],[0,0],'k','linewidth',1)
title('Entered function on given interval with initial guess')

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.

%xr=zeros(nmax);
for i=1:nmax
    if i==1
        xr(i) = xguess2-(f(xguess2)*(xguess1-xguess2))/(f(xguess1)-f(xguess2));
    elseif i==2
        x0=xguess2;
        xr(i) = xr(1)-(f(xr(1))*(x0-xr(1)))/(f(x0)-f(xr(1)));
    else
        xr(i) = xr(i-1)-(f(xr(i-1))*(xr(i-2)-xr(i-1)))/(f(xr(i-2))-f(xr(i-1)));
    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)=abs(xr(i)-xguess2);
    else
        Ea(i)=abs(xr(i)-xr(i-1));
    end
end

end

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

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

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')

```









