Secant Method--Convergence of the Roots.

© 2003 Nathan Collier, Autar Kaw, Jai Paul , Michael Keteltas, University of South Florida ,
kaw@eng.usf.edu , http://numericalmethods.eng.usf.edu/mws

NOTE: This worksheet demonstrates the use of Maple to illustrate the convergence of the roots using the Secant method of finding roots of a nonlinear equation.

Introduction

Secant method [text notes][PPT] is derived from the Newton-Raphson Method. Sometimes evaluating the derivative of the function could be very tedious and cumbersome. To overcome this, the derivative is approximated and the next estimate is given as

$$x_{i+1} = x_i - \frac{f(x_i)(x_i - x_{i-1})}{f(x_i) - f(x_{i-1})}$$

This method now requires two initial guesses, but unlike the bisection method, the two initial guesses do not need to bracket the root of the equation. The Secant method may or may not converge, but when it converges, it converges faster than the Bisection method. However, since the derivative is approximated, it converges slower then Newton-Raphson method. The following simulation illustrates the convergence of the roots using the secant method of finding roots of a nonlinear equation.

```
> restart;
```

Section I : Data.

The following is the data that is used to solve the nonlinear equation which is obtained from the floating ball problem from the General Engineering to find the depth 'x' to which the ball is submerged under water

Function in f(x)=0

```
> f(x):=x^3-0.165*x^2+3.993*10^(-4):
```

Initial guess 1

```
> xguess1:=0.05:
```

Initial guess 2

```
> xguess2:=0.02:
```

Upper bound of range of 'x' that is desired

```
> uxrange:=0.12:
```

Lower bound of range of 'x' that is desired

```
> lxrange:=-0.02:
```

Maximum number of iterations

```
> nmax:=7:
```

Enter the umber of the root desired

```
> rootnumber:=1:
```

Section II: Plotting the Data.

We now plot the data. The following function determines the upper and lower ranges on the Y-axis. This is done using the upper and lower ranges of the X-axis specified, and the value of the original functional at these values.

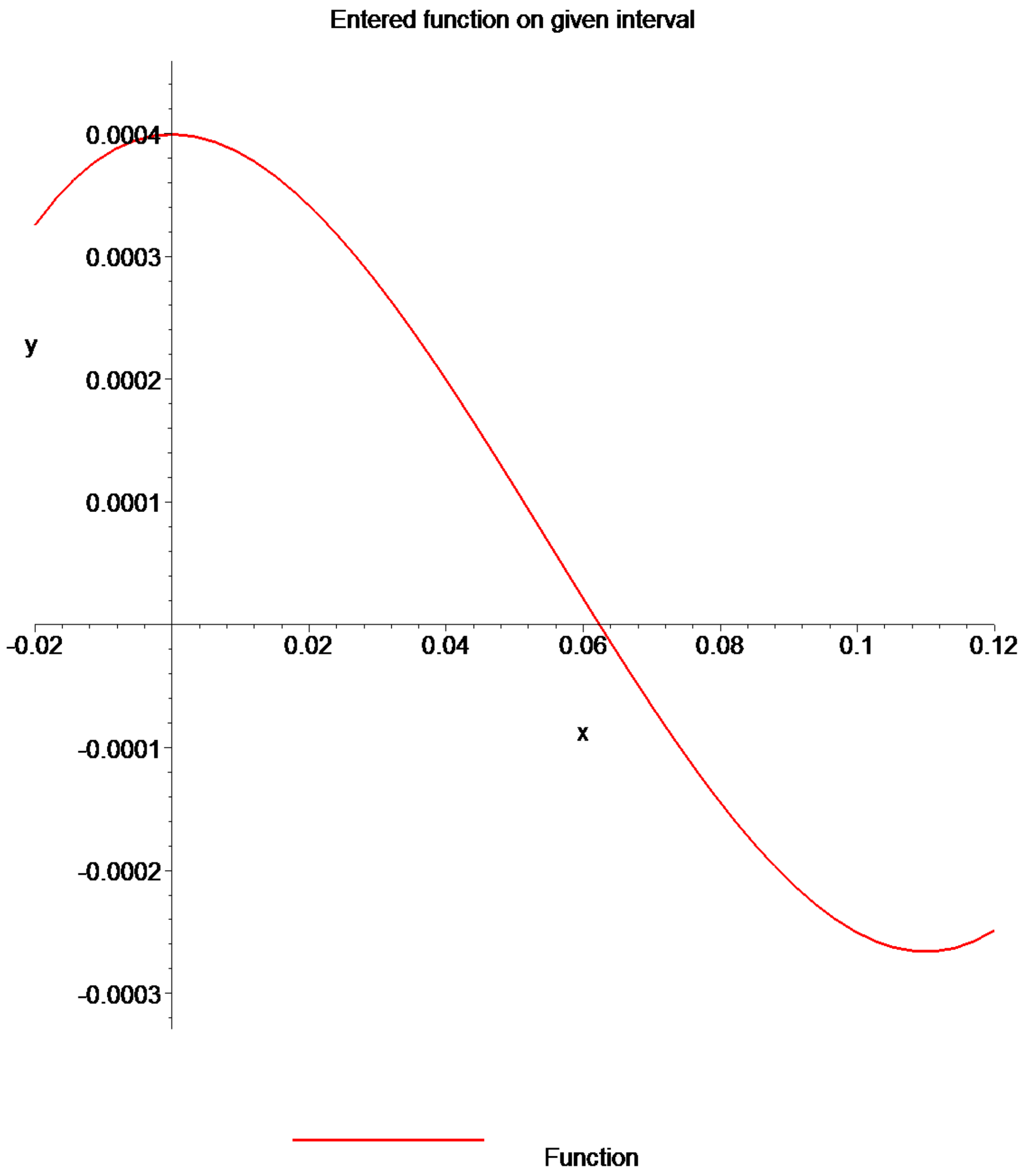
```
> yranger:=proc(uxrange,lxrange)
  local i,maxi,mini,tot;
  maxi:=eval(f(x),x=lxrange);
  mini:=eval(f(x),x=lxrange);
  for i from lxrange by (uxrange-lxrange)/10 to uxrange do
  if eval(f(x),x=i)<mini then mini:=eval(f(x),x=i) end if;
  if eval(f(x),x=i)>maxi then maxi:=eval(f(x),x=i) end if;
  end do;
  tot:=maxi-mini;
  -0.1*tot+mini..0.1*tot+maxi;
  end proc:
> yrange:=yranger(uxrange,lxrange):
> xrange:=lxrange..uxrange:
```

The following calls are needed to use the plot function

```
> with(plots):
Warning, the name changecoords has been redefined
```

```
> with(plottools):
Warning, the name arrow has been redefined
```

```
> plot(f(x),x=xrange,y=yrange,title="Entered function on given
  interval",legend=["Function"],thickness=3);
```



Section III: True Value.

The "true" solution is taken as the solution that Maple's numerical root solver obtains. This is a decent assumption because their subroutines have been professionally written. You must take caution, however, because Maple's "RootOf" function might be finding another of the function's roots. For the rest of the sheet to be correct, you need to ensure that xrtrue is the root that you are attempting to find. This can be altered by changing the value of 'rootnumber' above.

```
> xrtrue:=RootOf(f(x),x,index=rootnumber):
> xrtrue:=evalf(xrtrue);
```

```
                         xrtrue := 0.06237758151
```

Section IV: Value of root as a function of iterations.

Here the Secant 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:=proc(n)
       local p, i, x1, x2;
       x1:=xguess1;
       x2:=xguess2;

       for i from 1 to n do

  p:=x2-(eval(f(x),x=x2)*(x1-x2))/(eval(f(x),x=x1)-eval(f(x),x=x2
  ));
            x1:=x2;
            x2:=p
       end do;

       p;
  end proc:
>
> nrange:=1..nmax:
```

Absolute true error

```
> Et:=proc(n)
     abs(xrtrue-xr(n));
  end proc:
```

Absolute relative true error

```
> et:=proc(n)
     abs(Et(n)/xrtrue)*100;
  end proc:
```

Absolute approximate error

```
> Ea:=proc(n)
  local p,xq;
  xq:=xguess2;
     if n<2 then
        p:=abs(xr(1)-xq);
     else
        p:=abs(xr(n)-xr(n-1));
     end if;
  p;
  end proc:
```

Absolute relative approximate error

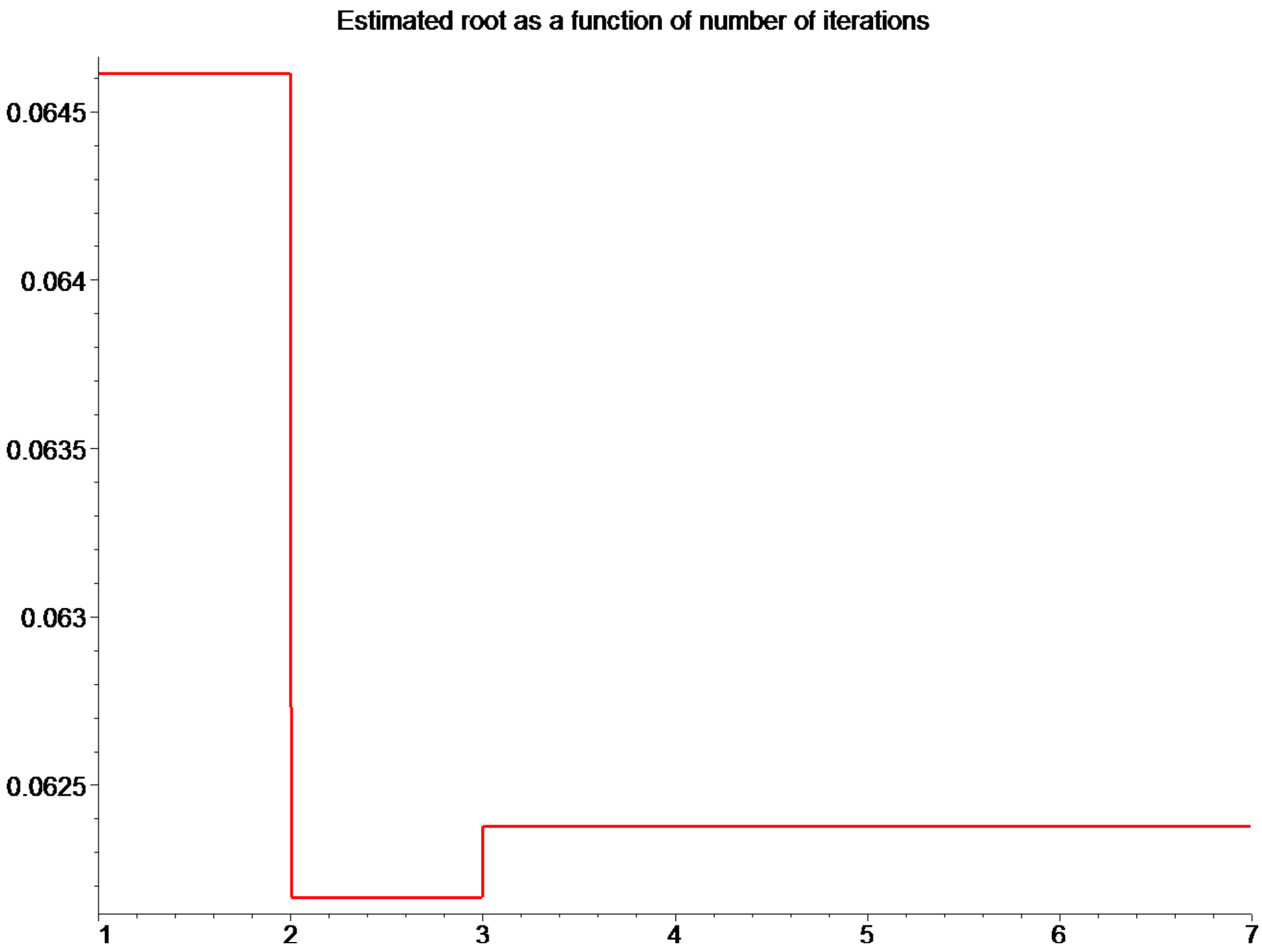
```
> ea:=proc(n)
     abs(Ea(n)/xr(n))*100;
  end proc:
```

Significant digits at least correct

```
> sigdigits:=proc(n)
     local p;
     p:=floor((2-log10(ea(n)/0.5)));
     if p<0 then p:=0 end if;
     p;
  end proc:
```

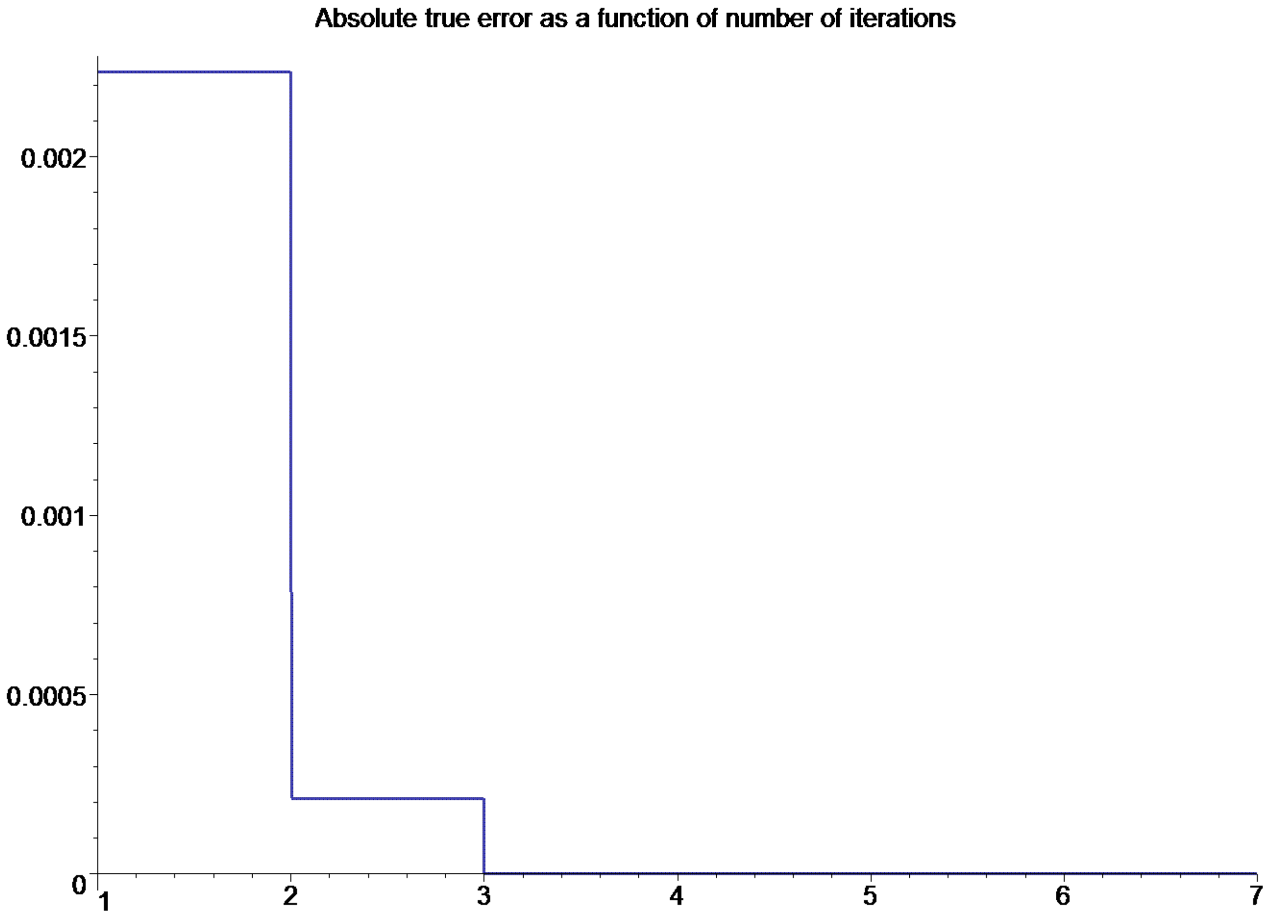
Section V: Graphs of Results.

```
> plot(xr,nrange,title="Estimated root as a function of number of
  iterations",thickness=3,color=red);
```

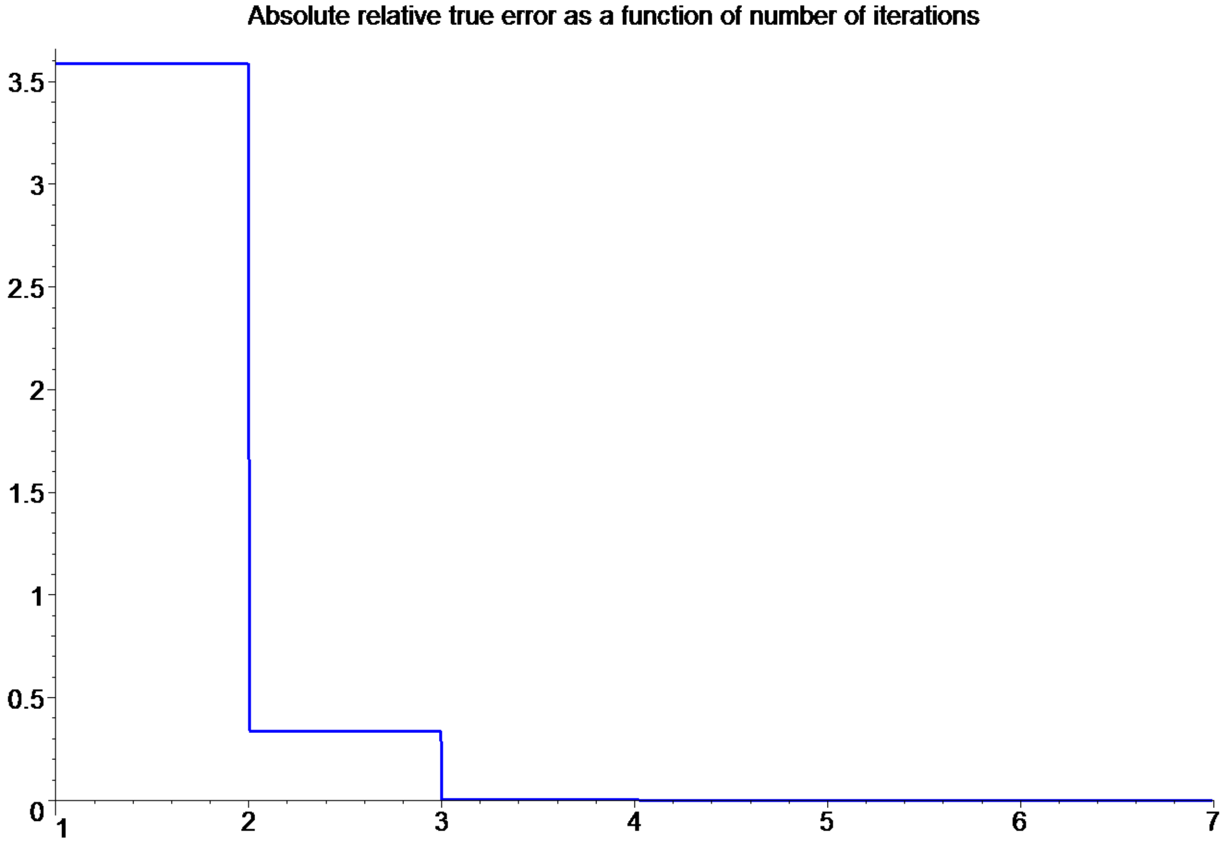


Estimated root as a function of number of iterations

```
> plot(Et,nrange,title="Absolute true error as a function of
  number of iterations",thickness=3,color=navy);
```

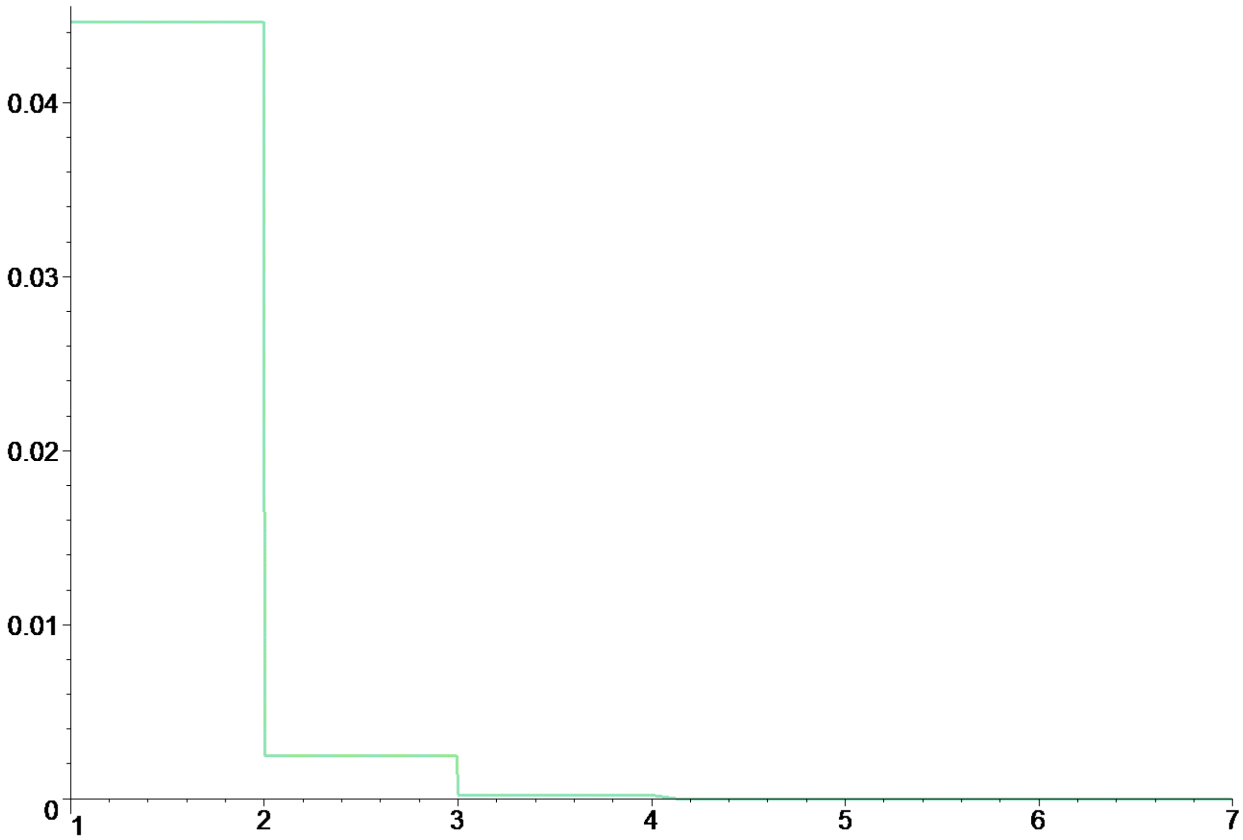


```
> plot(et,nrange,title="Absolute relative true error as a
  function of number of iterations",thickness=3,color=blue);
```

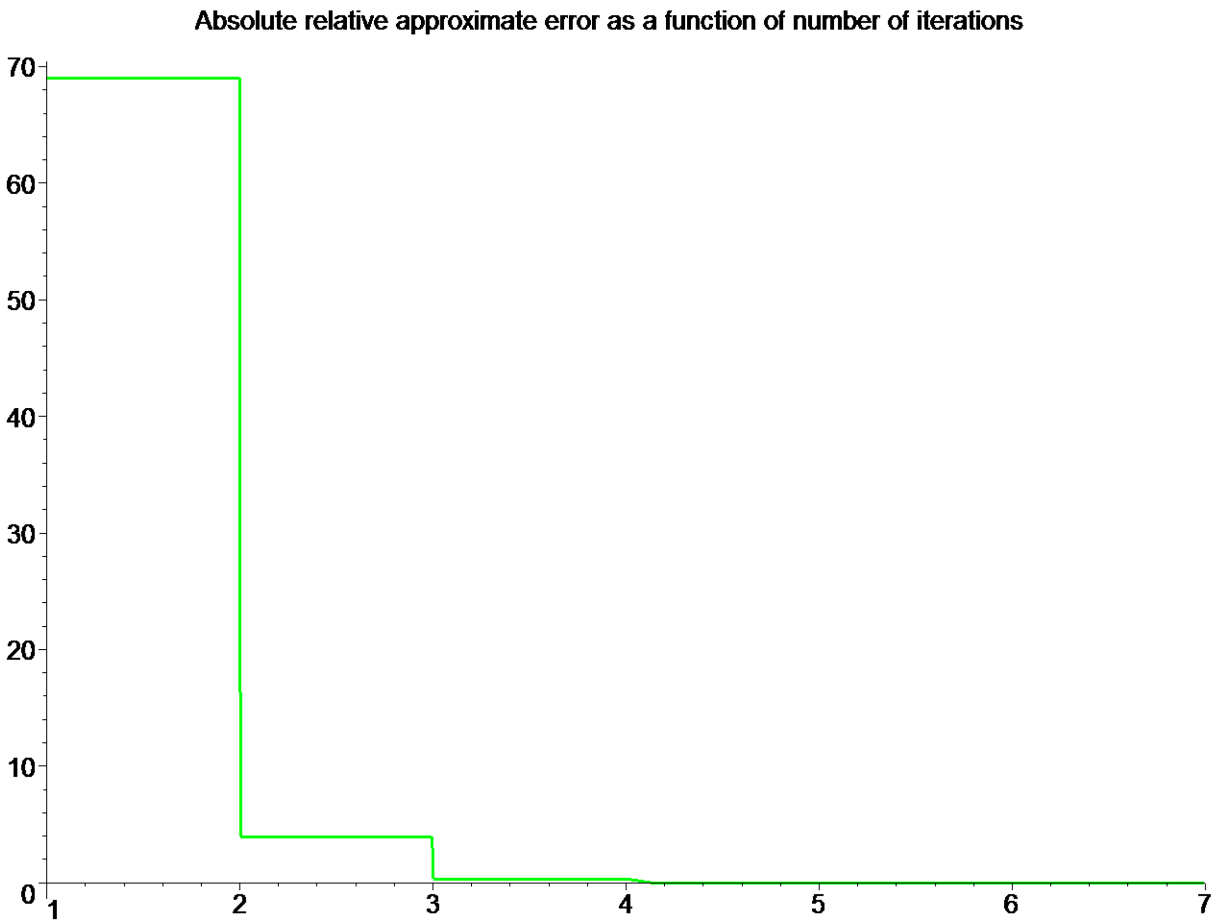


```
> plot(Ea,nrange,title="Absolute approximate error as a function
  of number of iterations",thickness=3,color=aquamarine);
```

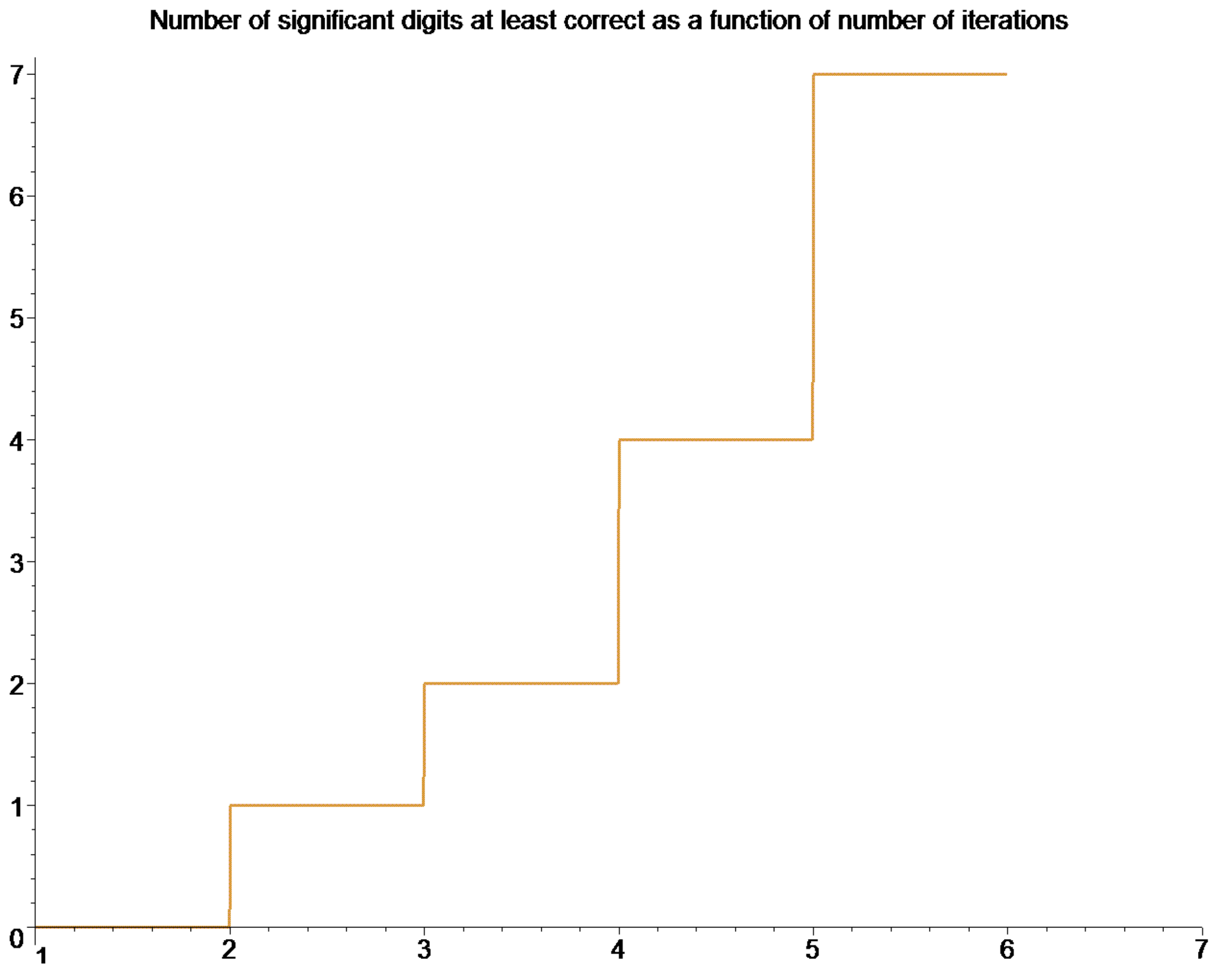
Absolute approximate error as a function of number of iterations



```
> plot(ea,nrange,title="Absolute relative approximate error as a
  function of number of iterations",thickness=3,color=green);
```



```
> plot(sigdigits,nrange,title="Number of significant digits at
  least correct as a function of number of
  iterations",thickness=3,color=gold);
```



Section VI: Conclusion.

Maple helped us to apply our knowledge of numerical methods of finding roots of a nonlinear equation using the secant method to simulate the convergence of the root of the given nonlinear equation.

References

[1] *Nathan Collier, Autar Kaw, Jai Paul , Michael Keteltas, Holistic Numerical Methods Institute, See http://numericalmethods.eng.usf.edu/mws/gen/03nle/mws_gen_nle_txt_secant.pdf*

Disclaimer*:* While every effort has been made to validate the solutions in this worksheet, University of South Florida and the contributors are not responsible for any errors contained and are not liable for any damages resulting from the use of this material.