

Secant Method of Solving a Nonlinear Equation

© 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 Secant method of finding roots of a nonlinear equation.

Introduction

Secant method 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 in the Newton-Raphson Method formula 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 than Newton-Raphson method.

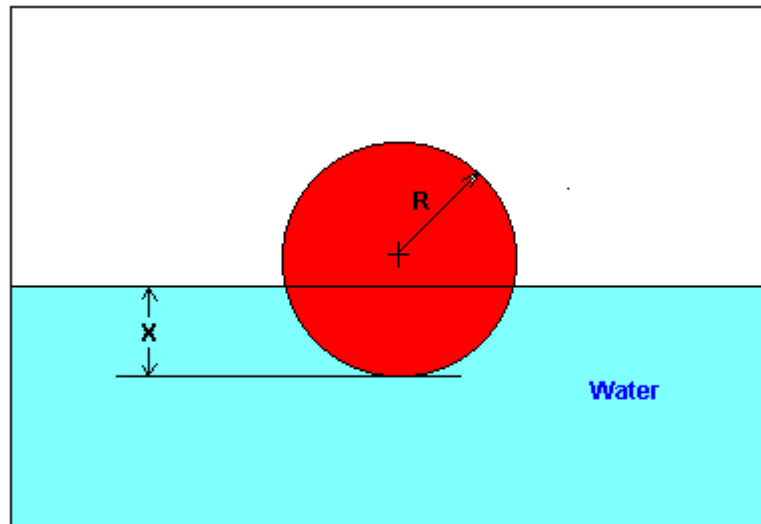
For detailed explanation of the Secant Method see [[Click here for textbook notes](#)][[Click here for PowerPoint presentation](#)]

The following simulation illustrates the Secant Method of finding roots of a nonlinear equation.

> **restart;**

Section I : Data.

You are working for 'DOWN THE TOILET COMPANY' that makes floats for ABC commodes. The ball has a specific gravity of 0.6 and has a radius of 5.5 cm.



The equation that gives the depth 'x' in meters to which the ball is submerged under water is given by $x^3 - 0.165x^2 + 3.993 \times 10^{-4} = 0$. Use Secant Method of solving nonlinear equations to find the distance to which the ball will get submerged when floating in 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
[ > l xrange:=-0.02:
```

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.

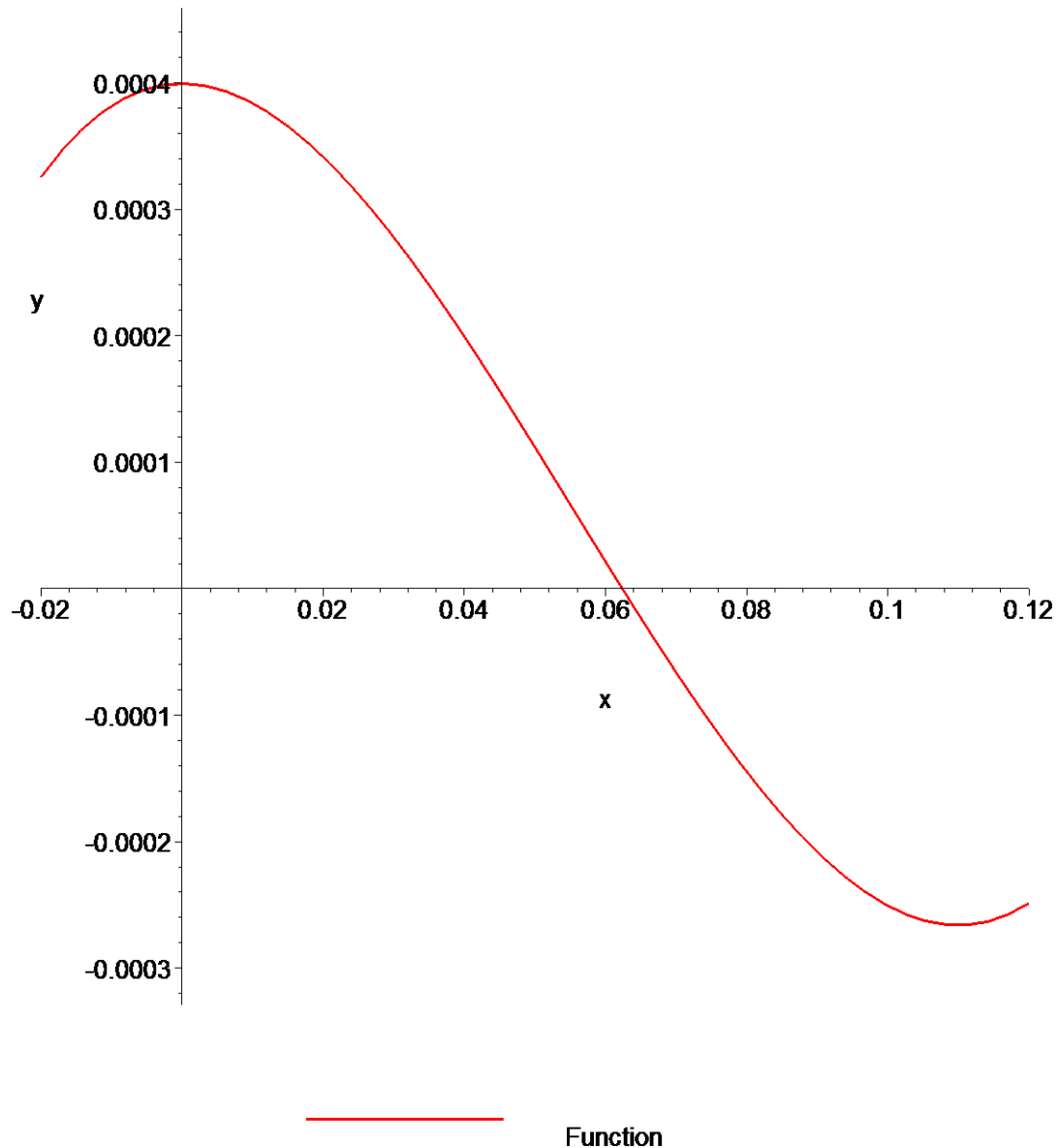
```
> yranger:=proc(uxrange,lxrange)
local i,maxi,mini,tot;
maxi:=eval(f(x),x=lxrange);
mini:=eval(f(x),x=lxrange);
for i from lxrange by (uxrange-lxrange)/10 to uxrange do
if eval(f(x),x=i)<mini then mini:=eval(f(x),x=i) end if;
if eval(f(x),x=i)>maxi then maxi:=eval(f(x),x=i) end if;
```

```
        end do;  
        tot:=maxi-mini;  
        -0.1*tot+mini..0.1*tot+maxi;  
    end proc;  
[ > xrange:=xranger(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);
```

Entered function on given interval



Section III: Iteration 1.

So, first we choose two initial guesses of the root. It should be noted that these two guesses do not have to bracket the root. We have called the two initial guesses x_{guess1} and x_{guess2} , as that will be the format for subsequent iterations. It does not matter which guess is x_{guess1} or x_{guess2} (try switching the numbers below and see what happens! You will find that one converges faster than the other). The formula mentioned in the introduction above is then applied to find the first estimate.

```
> x1:=xguess2-(eval(f(x),x=xguess2)*(xguess1-xguess2))/(eval(f(x),x=xguess1)-eval(f(x),x=xguess2));
```

```
x1 := 0.06461437908
```

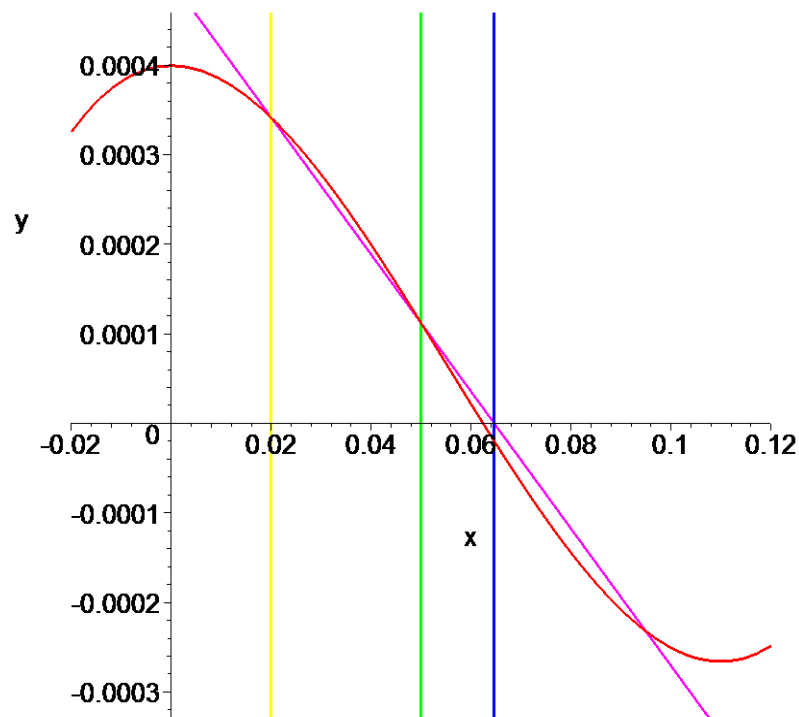
How good is that approximation? Find the absolute relative approximate error.

```
[ > epsilon:=abs ( (x1-xguess2) /x1) *100 ;  
                                     ε := 69.04713736
```

Although it is not necessary for the method, it is helpful to define the equation for the secant line passing through the two guesses. This function will be used for the graph.

```
[ > m:=(eval (f (x) ,x=xguess2) -eval (f (x) ,x=xguess1) ) / (xguess2-xguess1  
  ) :  
[ > secantline:=m*x+eval (f (x) ,x=xguess2) -m*xguess2 :  
[ > plot ([f (x) , [xguess1,t,t=yrange] , [xguess2,t,t=yrange] , [x1,t,t=yrange] ,secantline(x) ] ,x=xrange,y=yrange,title="Entered function  
  on given interval with current and next root\n and secant line  
  between two guesses",legend=["Function", "xguess1, First  
  guess", "xguess2, Second guess", "x1, New guess", "Secant  
  line"],thickness=3) ;
```

Entered function on given interval with current and next root
and secant line between two guesses



— Function
 — xguess1, First guess
 — xguess2, Second guess
 — x1, New guess
 — Secant line

Section IV: Iteration 2.

Using the same formula, calculate the next estimate of the root.

```
[ > x0:=xguess2:
```

```
[ Estimate of the root
```

```
[ > x2:=x1-(eval(f(x),x=x1)*(x0-x1))/(eval(f(x),x=x0)-eval(f(x),x=x1));
```

```
      x2 := 0.06216668835
```

```
[ Absolute relative approximate error
```

```
> epsilon:=abs((x2-x1)/x2)*100;
```

$\varepsilon := 3.937302750$

Secant line for the graph

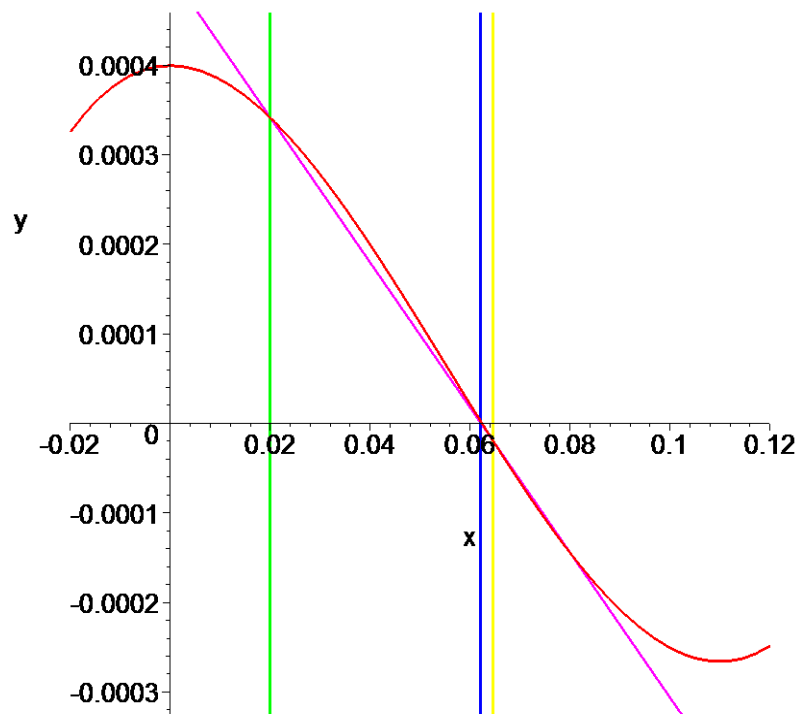
```
> m:=(eval(f(x),x=x1)-eval(f(x),x=x0))/(x1-x0):
```

```
> secantline:=m*x+eval(f(x),x=x1)-m*x1:
```

```
> plot([f(x),[x0,t,t=yrange],[x1,t,t=yrange],[x2,t,t=yrange],secantline(x)],x=xrange,y=yrange,title="Entered function on given interval with current and next root\n and secant line between two guesses",legend=["Function", "x0, First guess", "x1, Second guess", "x2, New guess", "Secant line"],thickness=3);
```

Entered function on given interval with current and next root

and secant line between two guesses



—	Function
—	x_0 , First guess
—	x_1 , Second guess
—	x_2 , New guess
—	Secant line

Section V: Iteration 3.

Using the same formula, calculate the next estimate of the root.

[Estimate of the root

```
> x3:=x2-(eval(f(x),x=x2)*(x1-x2))/(eval(f(x),x=x1)-eval(f(x),x=x2));
```

```
x3 := 0.06237886745
```

[Absolute relative approximate error

```
> epsilon:=abs((x3-x2)/x3)*100;
```

```
ε := 0.3401458037
```

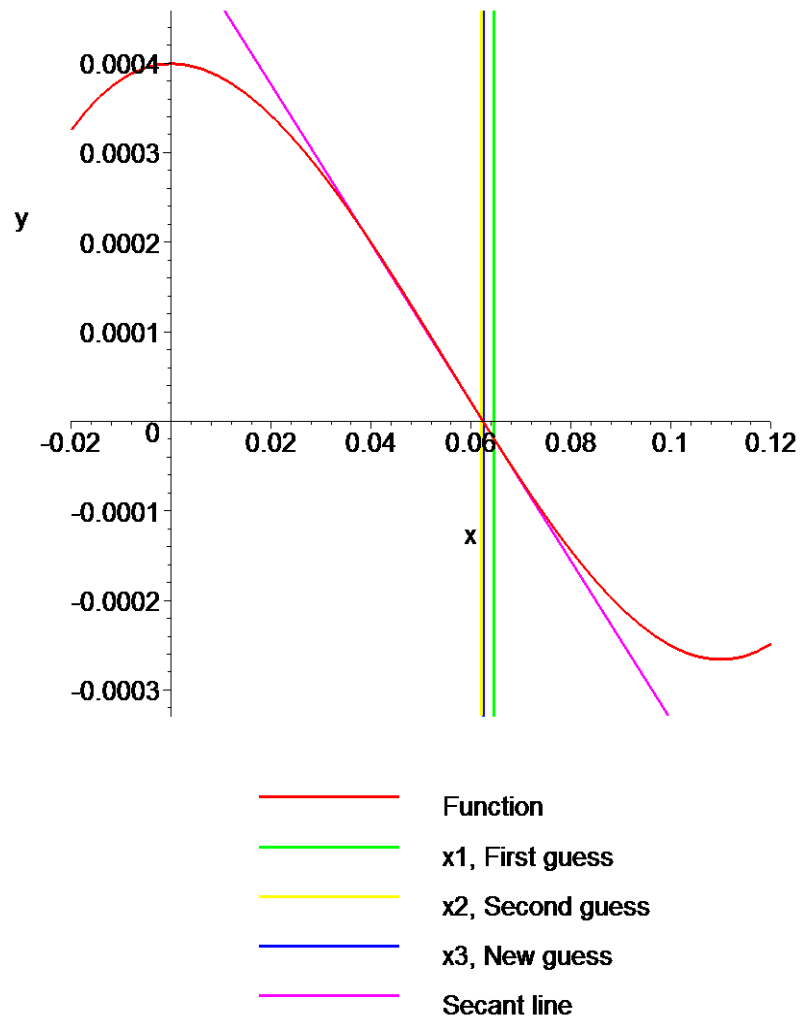
[Secant line for the graph

```
> m:=(eval(f(x),x=x2)-eval(f(x),x=x1))/(x2-x1):
```

```
> secantline:=m*x+eval(f(x),x=x2)-m*x2:
```

```
> plot([f(x),[x1,t,t=yrange],[x2,t,t=yrange],[x3,t,t=yrange],secantline(x)],x=xrange,y=yrange,title="Entered function on given interval with current and next root\n and secant line between two guesses",legend=["Function", "x1, First guess", "x2, Second guess", "x3, New guess", "Secant line"],thickness=3);
```

Entered function on given interval with current and next root
and secant line between two guesses



>

Section VI: Conclusion.

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

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.