

# Zero slope--Secant Method.

© 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 a pitfall of zero slope in the Newton-Raphson method of finding roots of  $f(x)=0$ .

## **- Introduction**

Secant method [[text notes](#)][[PPT](#)] is derived from the Newton-Raphson Method. The Secant method may or may not converge, but it converges slower than Newton-Raphson method. The same drawbacks can be seen in this method as in Newton-Raphson. One of them is Division of zero or near zero. The following simulation illustrates a pitfall of zero slope in the Secant method of finding roots of  $f(x)=0$ .

```
> restart;
```

## **- Section I : Data.**

Function in  $f(x)=0$

```
> f(x) := sin(x) :
```

Upper bound of range of 'x' that is desired

```
> xrange := 10.0 :
```

Lower bound of range of 'x' that is desired

```
> lrange := -10.0 :
```

Initial guess 1

```
> xguess1 := Pi/4 :
```

Initial guess 2

```
> xguess2 := Pi*3/4 :
```

## **- Section II: Before you start.**

Because the method uses a line tangent to the function at the initial guess, we must calculate the derivative of the function to find the slope of the line at this point. Here we will define the derivative of the function  $f(x)$  as  $g(x)$ .

```
> g(x) := diff(f(x), x) ;
```

```
g(x) := cos(x)
```

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

```

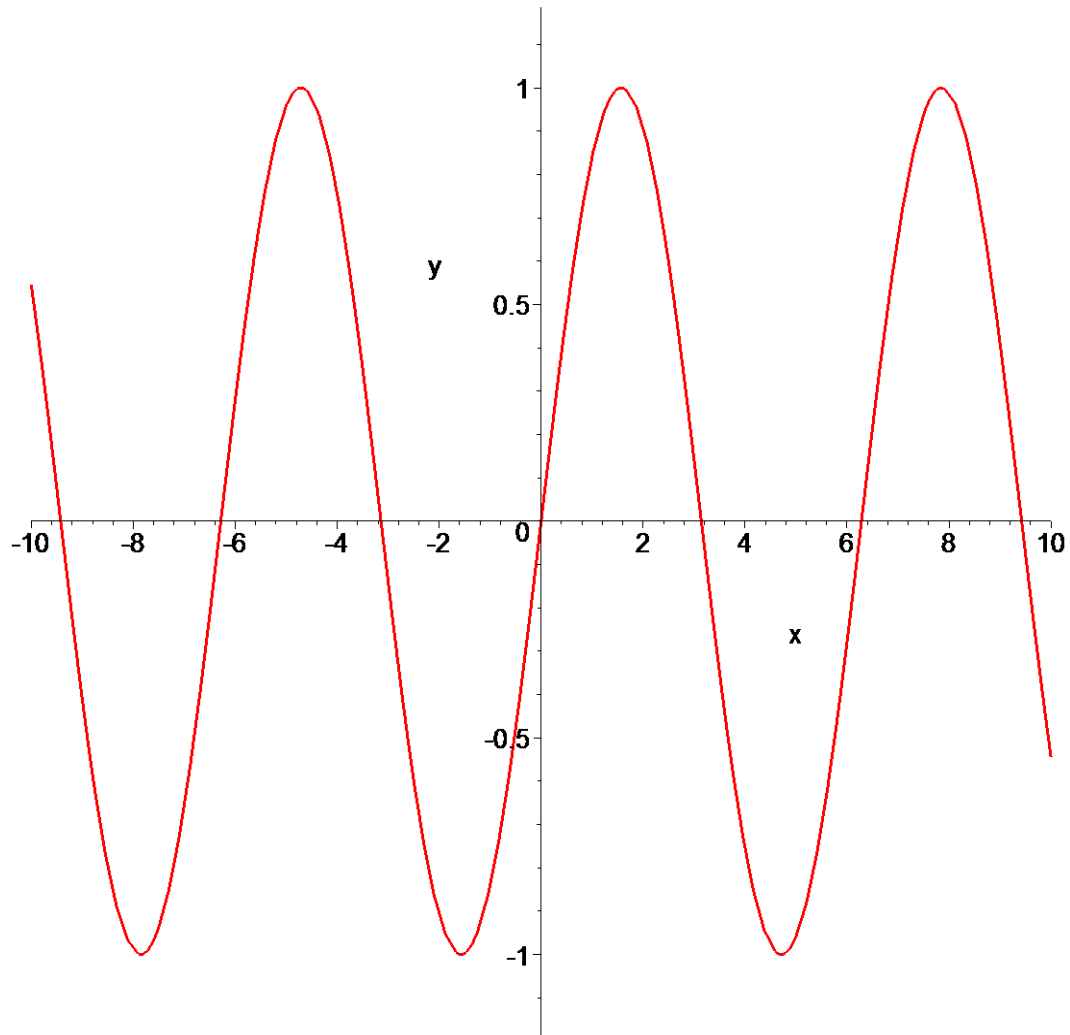
> with(plots):
Warning, the name changecoords has been redefined

> 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);

```

Entered function on given interval



Function

### **- Section III: Iteration 1.**

The Newton Raphson Method works by taking a tangent line at the value of the function at the initial guess, and seeing where that tangent line crosses the x-axis. This value will be the new estimate for the root..

The first estimate of the root is,

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

```
Error, numeric exception: division by zero
```

How good is that answer? Find the absolute relative approximate error to judge.

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

$$\varepsilon := 100 \left| \frac{x1 - \frac{3\pi}{4}}{x1} \right|$$

While not necessary to the method, for graphing purposes we define the equation of the tangent line touching  $x_0$ .

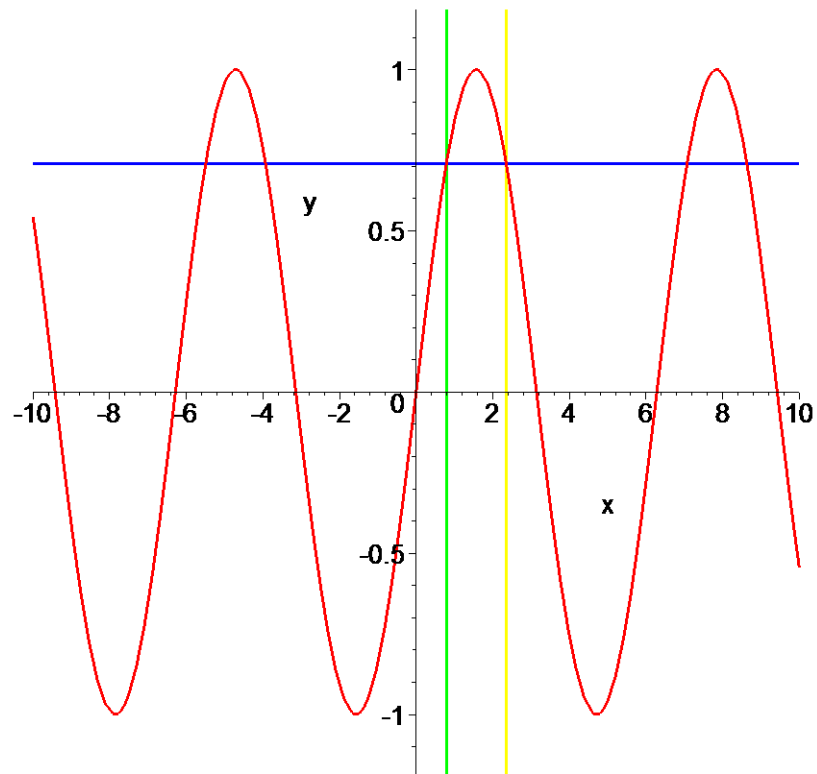
[ Secant line for the graph

```
> 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
- $x_{\text{guess1}}$ , First guess
- $x_{\text{guess2}}$ , Second guess
- Secant line

In the above example, a zero slope is encountered. As the tangent line parallels the x-axis, convergence will not occur. Notice that our first guess has a zero slope, which causes division by zero in the Secant. formula, and hence an error in our calculation (note the "division by zero" error message, and the undefined  $x_1$ ).

## **Section II: Conclusion.**

Maple helped us to apply our knowledge of numerical methods of finding roots of a nonlinear equation to understand a pitfall of zero slope in the Secant method of finding roots of  $f(x)=0$ .

## **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](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.