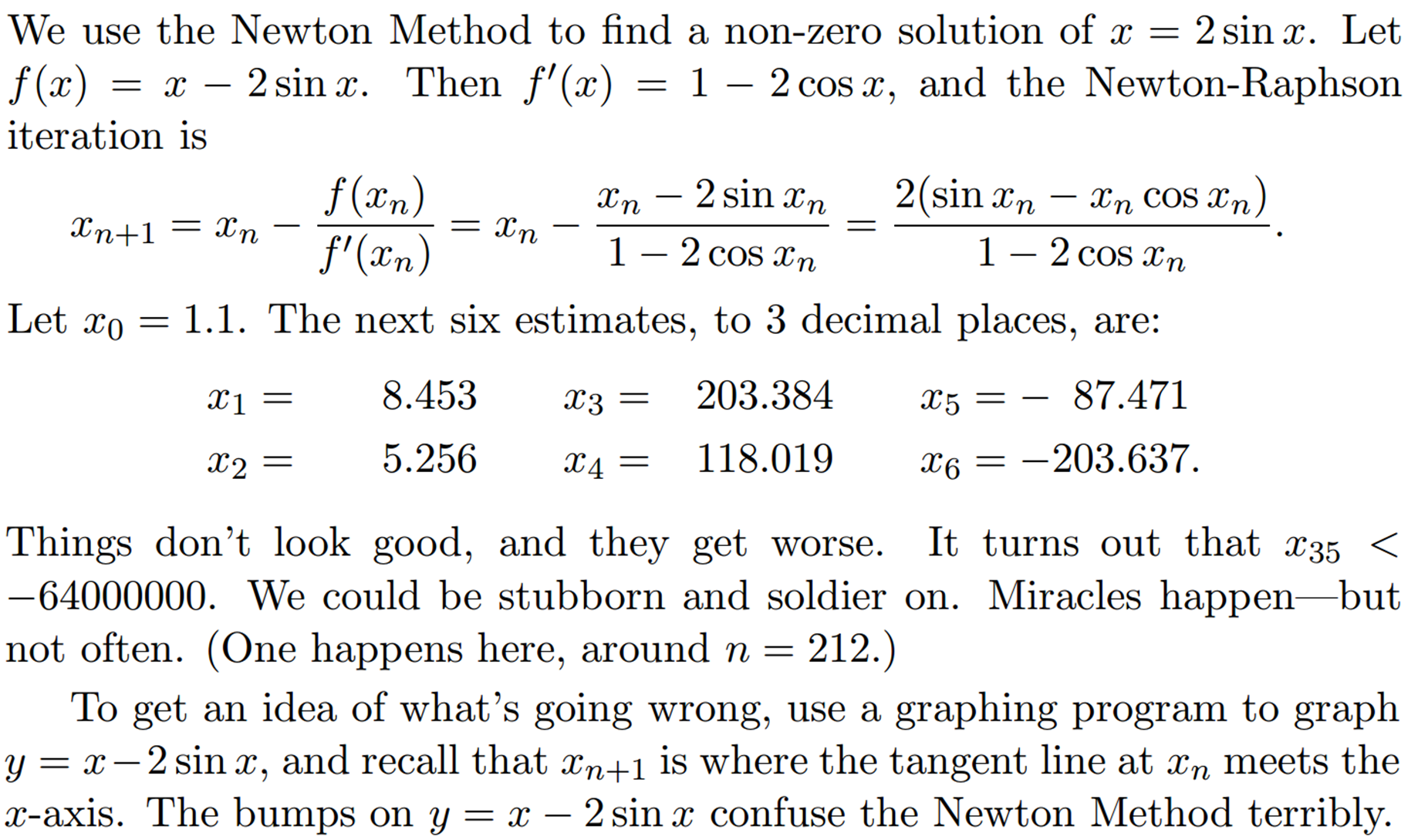
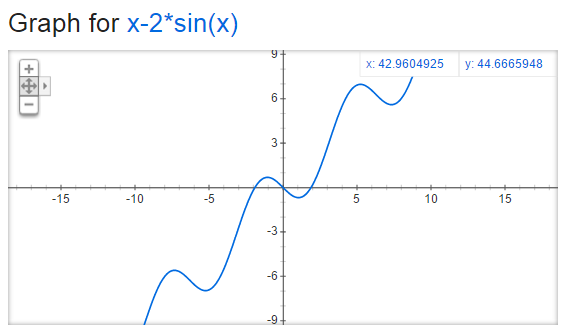
**Practical 5:**

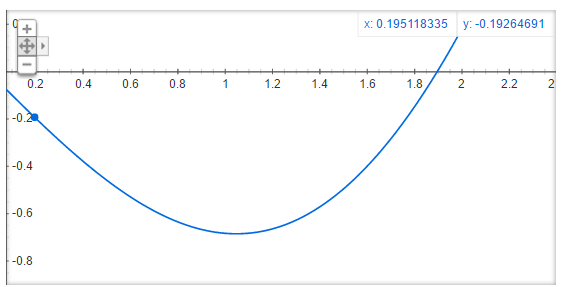
**Objective**: Finding roots of equations: Newton-Raphson and Secant Methods

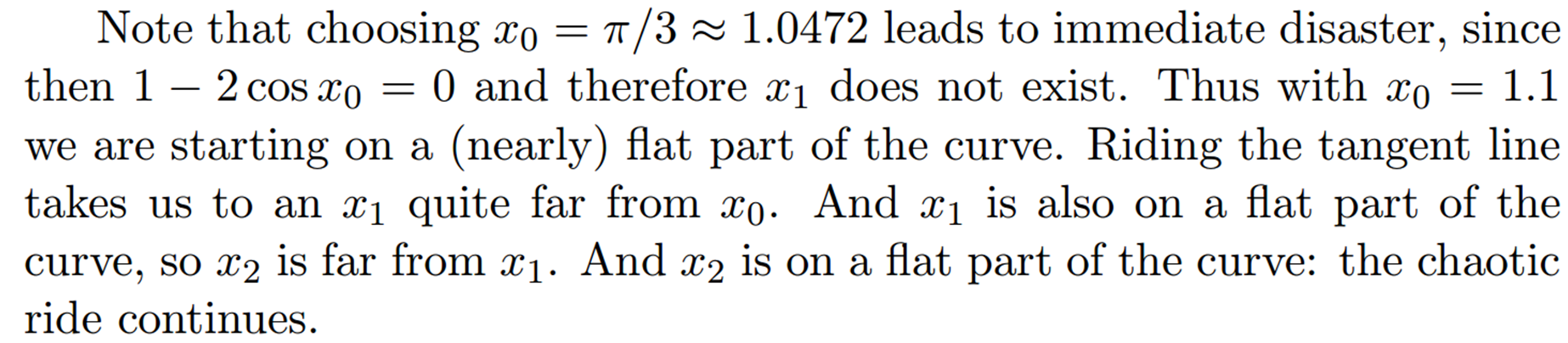
**Newton-Raphson Idea:**

1. Guess an  close to the root of interest.
2. Start Iteration: Approximate the function at that point by a straight line. The obvious choice is the line tangent to (in the direction of) the function's graph at that point.
3. Notice that the slope of the required tangent is the derivative of the function, so the line we want has that slope and goes through the point  .
4. This tangent line goes through the -axis at a point , which is easy to calculate and which we bet is nearer to the root than  is.
5. Compute  and , and we're ready to go to Start Iteration and repeat the process until for some , we find a  close enough to zero for our purposes.

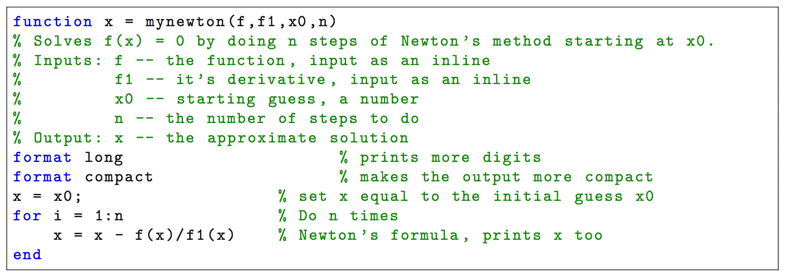


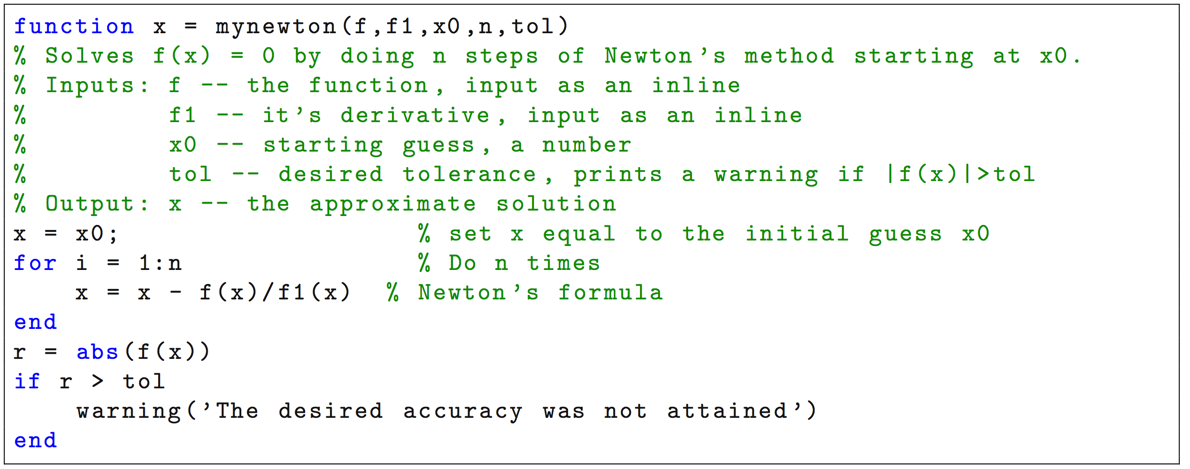


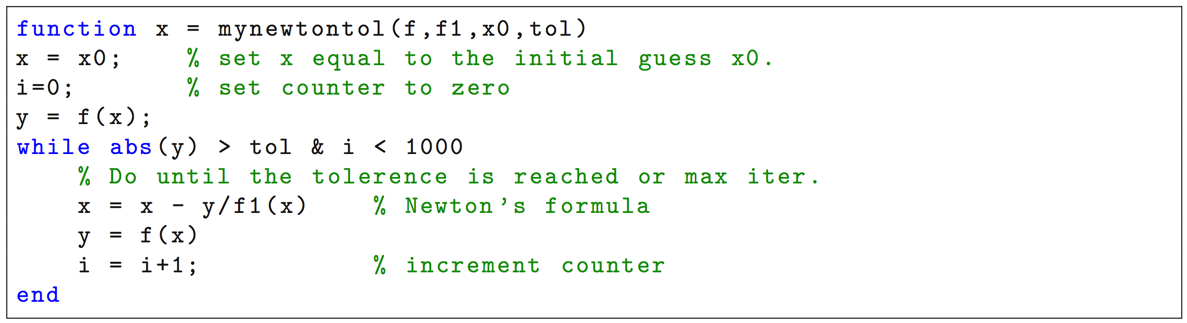




**Sample Code (MATLAB): Newton-Raphson**

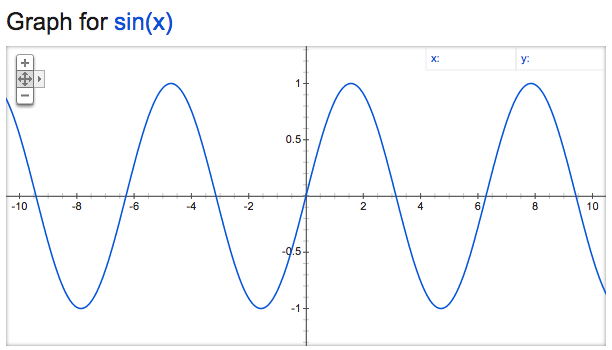






**Part A: Root of Nonlinear Function (Matlab Examples)**

1. Calculate by finding the zero of the sine function near 3.



fun=@sin

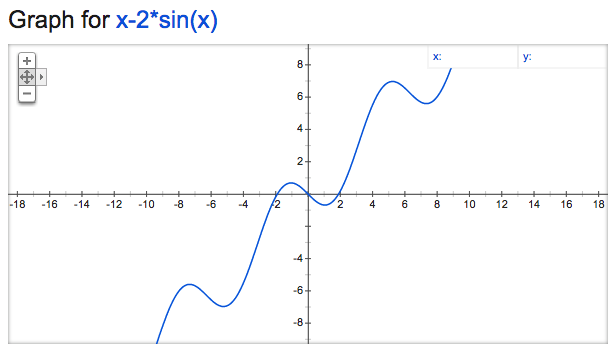
x0=3

format long

fzero(fun,x0)

result : x = 3.14159265358979

1. Find a non-zero solution of near 1.1



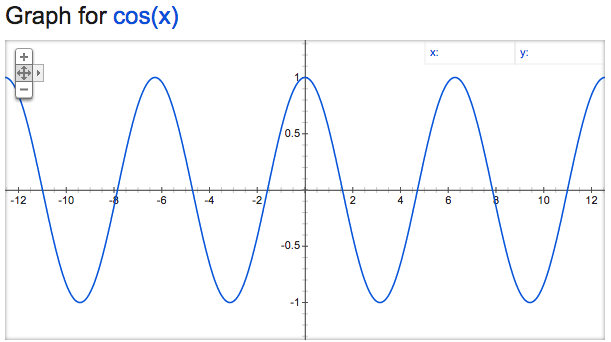
fun=@(x)x-2\*sin(x)

x0=1.1

x=fzero(fun,x0)

result : x = 1.89549426703398

1. Find the zero of cosine between 1 and 2.



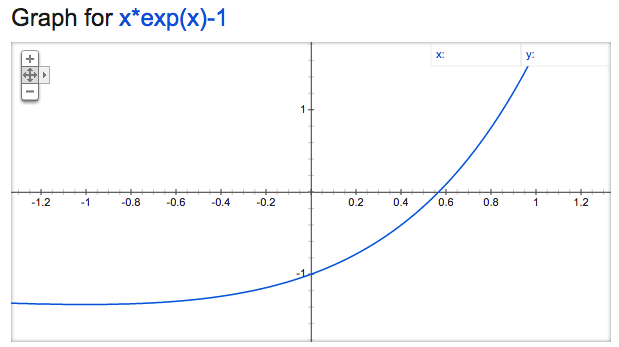
fun=@cos

x0=[1 2]

x=fzero(fun,x0)

result : x = 1.57079632679490

1. Find the root of in the interval [0,1], correct to 6 decimal places.



fun=@(x)x\*exp(x)-1

x0=[0 1]

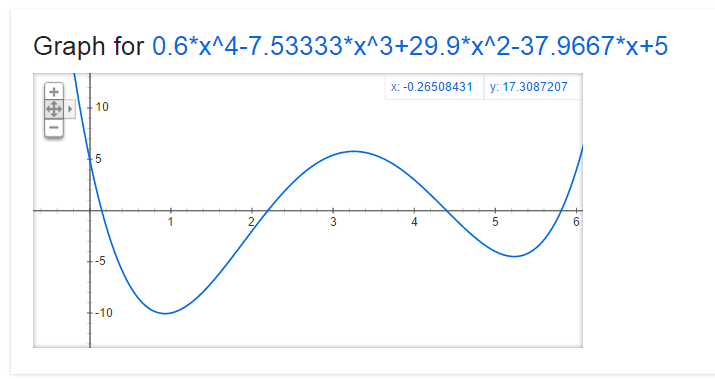
x=fzero(fun,x0)

result : x = 0.567143

**Part B: Newton Raphson Method**

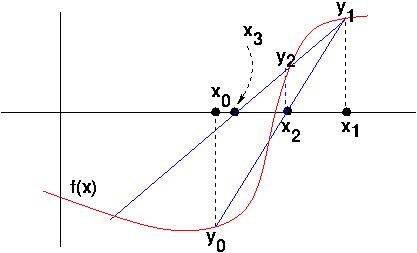
Write a program to find the root of

by using , and stopping when .



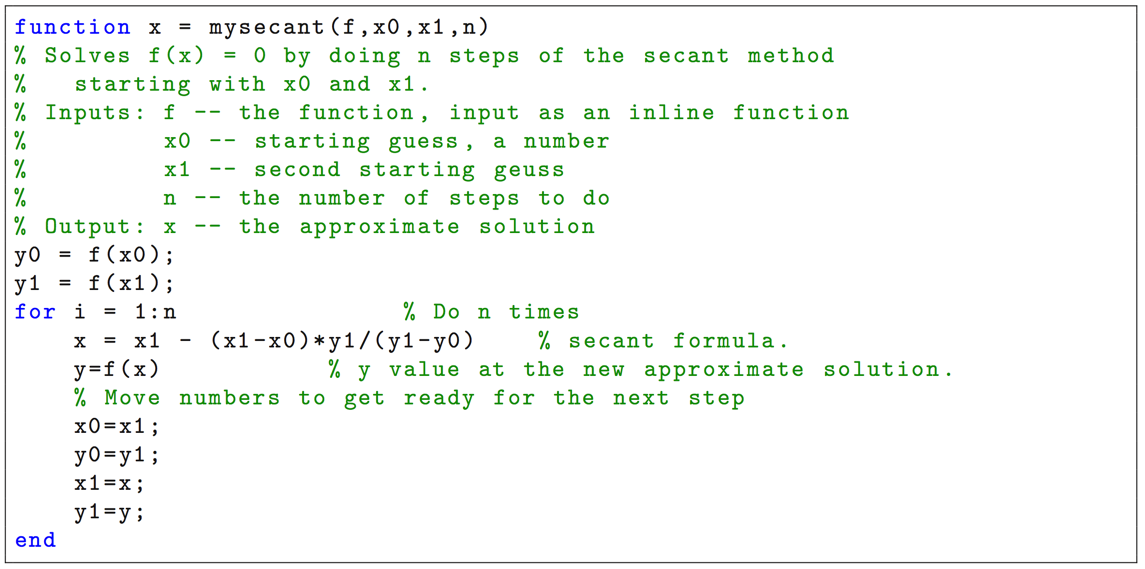
Root of x = 4.397102

**Secant Idea:**



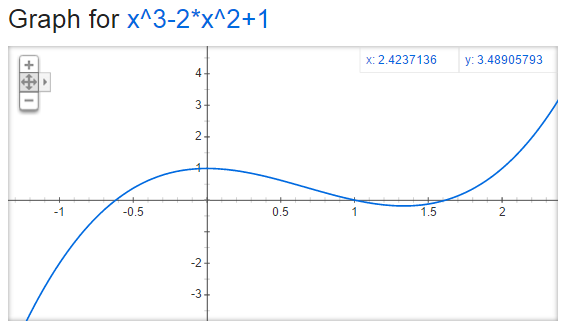
1. Pick two initial values of , close to the desired root. Call them , . Evaluate  and .
2. As with the tangent line in Newton's method, produce the (secant) line through () and (), and compute where it crosses the -axis, and call that point . Get .
3. Bootstrap along: replace (), () with (), () and repeat.
4. Keep this process up: derive () from () and () until  meets the error criterion.

**Sample Code (MATLAB): Secant**



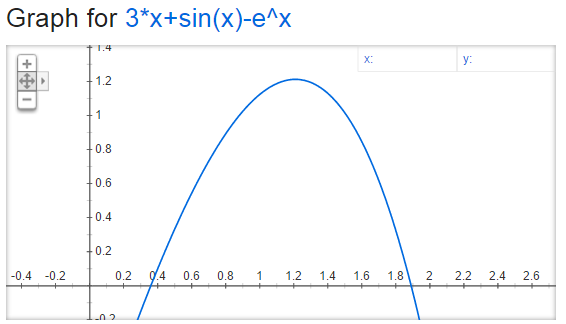
**Part C: Secant Method**

1. Write a program to find the negative root of by taking , and .



The root of x = -0.61803

1. Write a program to find a positive root in with tolerance for the following initial points:



* 1. and : The root of x = 1.89003
  2. and : The root of x = 1.89003
  3. and : The root of x = 1.89003
  4. and : The root of x = 1.89003
  5. and : The root of x = 0.36042
  6. and : The root of x = 0.36042
  7. and : The root of x = 3.48920