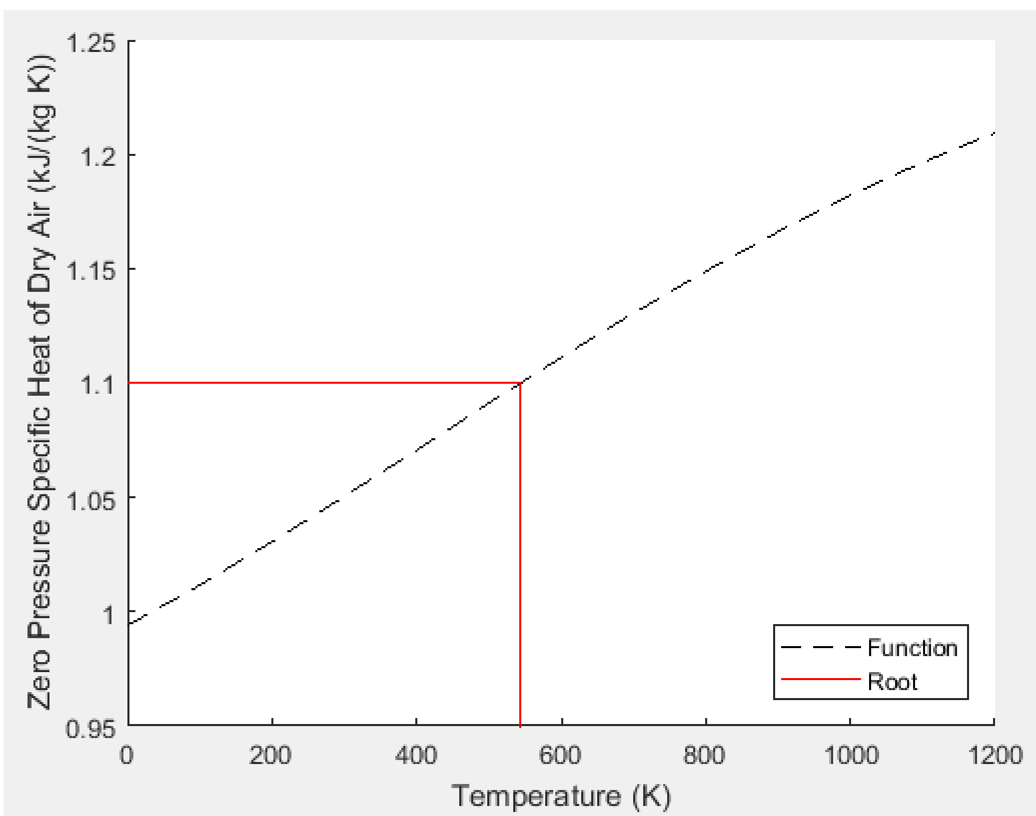
1.)

Name	Value
cd	1x31 double
error	5.6353e-09
f	1x31 double
g	9.8100
iteration	26
m	95
t	9
v	46
xl	0.3966
xm	0.3966
xm_prev	0.3966
xu	0.3966

Name	Value
cd	1x31 double
error	9.1837e-09
f	1x31 double
g	9.8100
iteration	13
m	95
t	9
v	46
xl	0.2000
xm	0.3966
xm_prev	0.3966
xu	0.3966

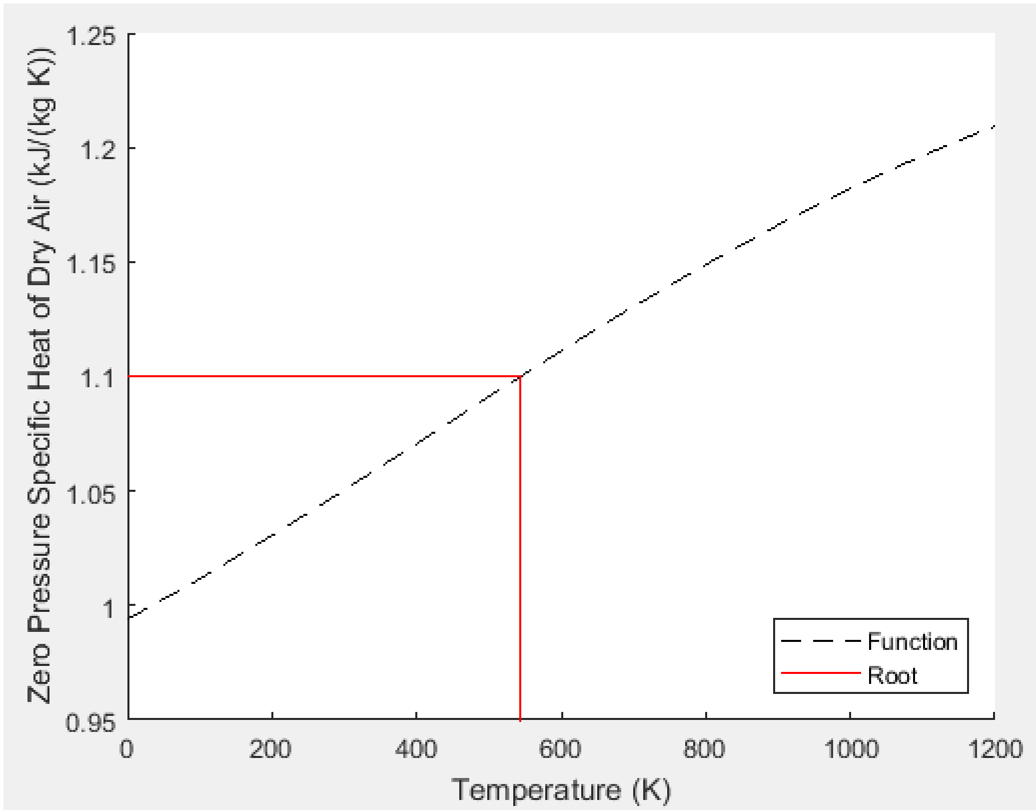
On the left is my script using the bisection method, and on the right is my script using the false position method. Variable xm represents the root and variable iteration represents how many loops it took to get below 10^-8 error. As shown, the false position method took half as many iterations to get the same root value, meaning it was twice as efficient.

2.)



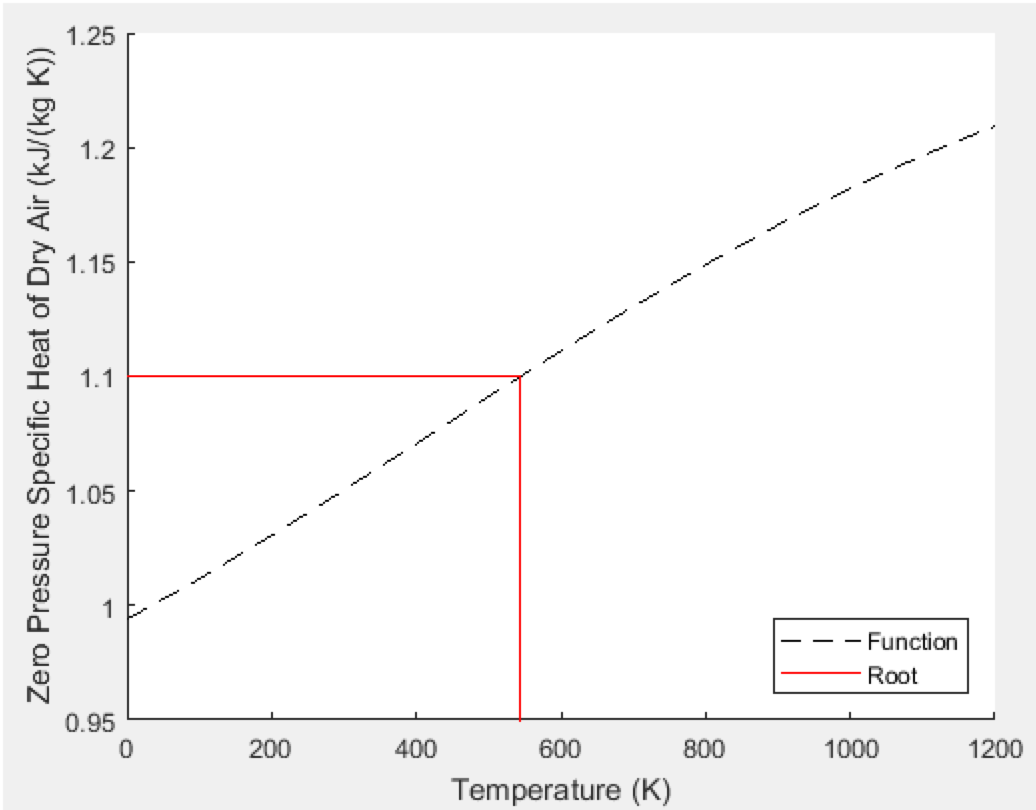
Using the bisection method I was able to find the root at x = 544.0875, but it took 27 iterations. Even though it converged, that's a lot of iterations; not very efficient.

3.)



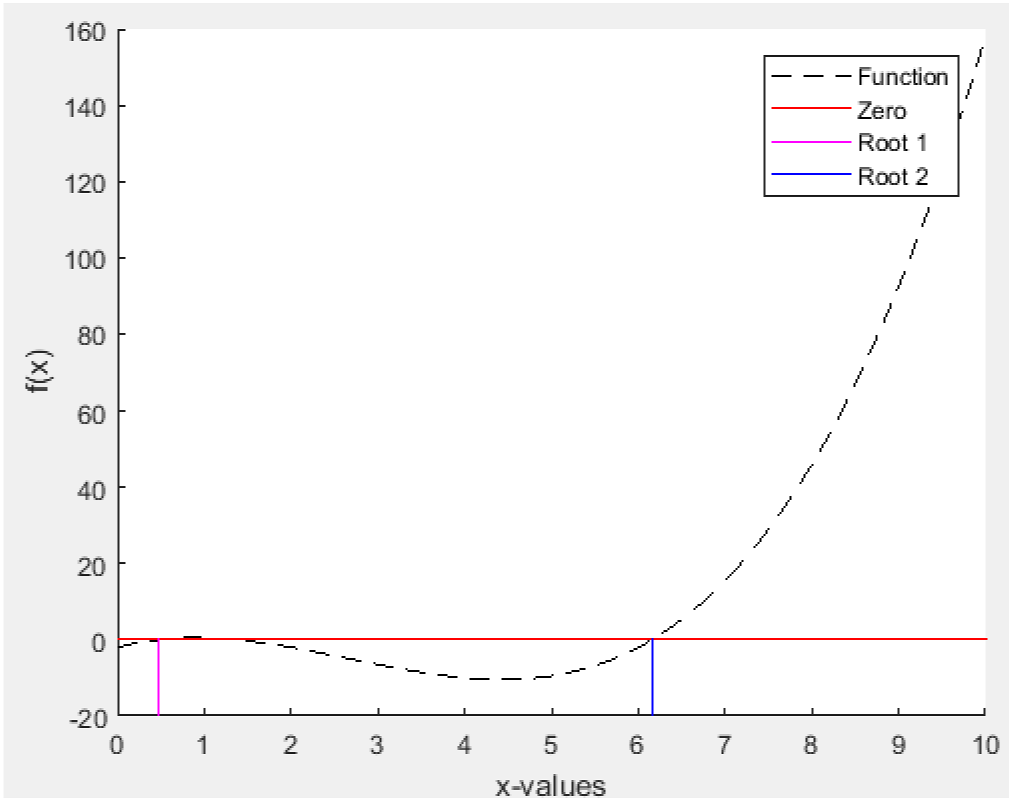
Using the false position method I was able to find the root at x = 544.0875, same as using the bisection method, but it took only 4 iterations rather than 27. This means that, at least in this instance and using the starting interval [.2 .5], the false position method was almost 7x more efficient than the bisection method.

4.)



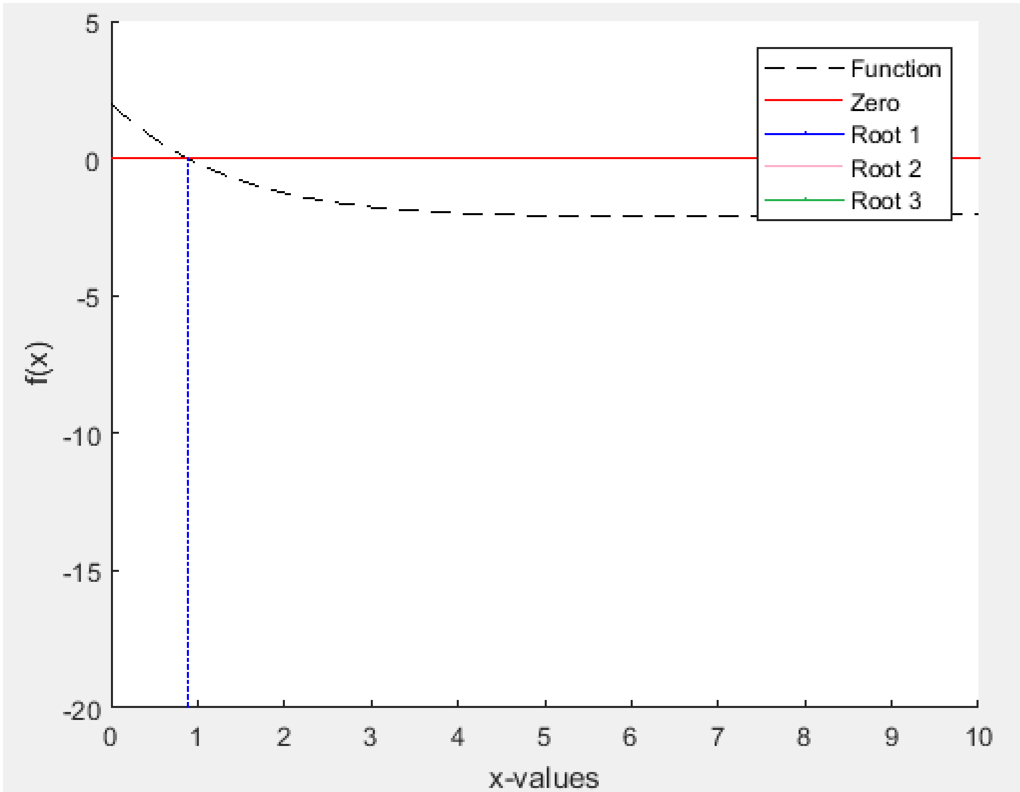
Using the Newton-Raphson method I was able to find the root at x = 544.0875, same as the bisection method and false position method, taking 5 iterations to converge. This is slightly slower than the false position methods 4 iterations and much faster than the bisection methods 27 iterations. Between the three methods used, I would use the false position method if I knew the interval the root is on, and the Newton-Raphson method if I did not know the interval.

5.)



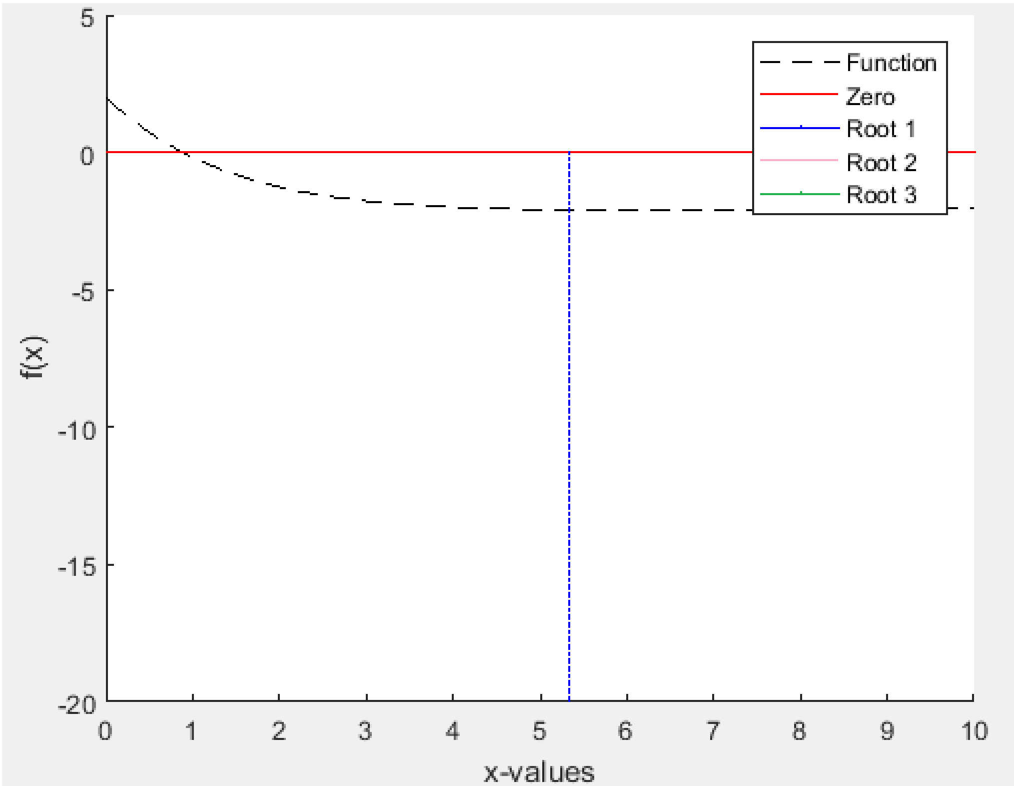
Using the Newton-Raphson method, roots were found at x = 6.1563 and x = .4746. The different initial guesses provided different roots (the roots closest to the guess). However, one root (at approximately x = 1.5) was not found because no initial guesses were close enough. This is a flaw with the Newton-Raphson method; multiple guesses are required to find multiple roots.

6.)



Using the Newton-Raphson method, of the different initial guesses, only x = 2 was able to converge to the root x = .8857. x = 6 and x = 8 were initial guesses that began on a horizontal (or near-horizontal) line and as such were not able to converge to the root.

7.)



Using the secant method, I was not able to converge to the root at x = .8857. Instead, it converged to x = 5.3226. This illustrates the problem with open methods like the secant method; they don't always converge.