Root finding methods

# Bisection Method

1. Overview:

This is a bracketing method, and always converge, it works by getting the mid of an interval then substitute in the function, and if the result of this mid is multiplied value of the function at the upper bound or the lower one, and the result is positive then, that mean the mid and the bound are both under the X-axis or above it then there is no root will be found between that bound and the mid, so we update the that bound to be the mid, and the process continues until we find root.

1. Pseudo-Code:

For I = 1: numberOfIterations

mid = (upper+lower)/2

test = f(mid)

if (test <= eps)

iterations = i

root = mid

break;

end

if f(mid)\*f(upper) > 0 then

upper = mid

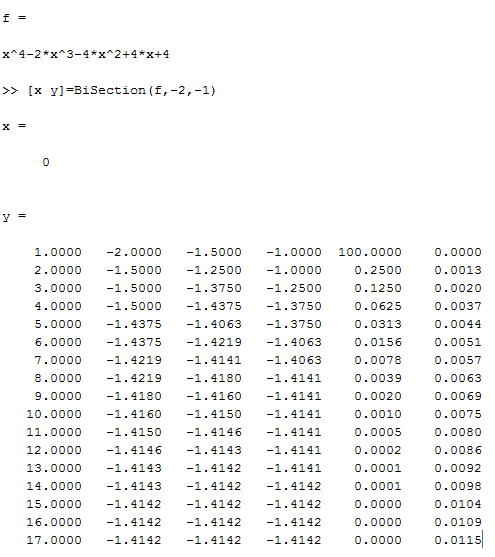
else

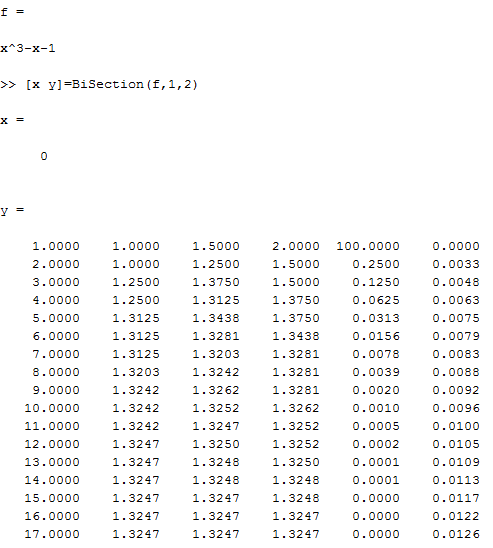
lower = mid

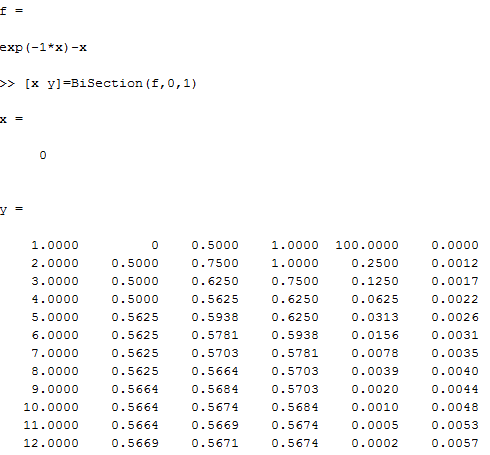
end

end

1. Pitfalls:
2. Slow.
3. Need to find initial guesses for upper and lower bound.
4. No account is taken of the fact that if f(lower bound) is closer to zero, it is likely that root is closer to lower bound.
5. Examples:







# Fixed Point Method

1. Overview:

This is an open method, that may converge or diverge, it works by generating the magic function g(x) from the input function f(x), then iterate with initial point until g(x) is equal to the input x.

1. Pseudo-Code:

For I = 1: numberOfIterations

x2 = g(x1)

if(abs(x2-x1) < eps)

I = iterations

Root = x1

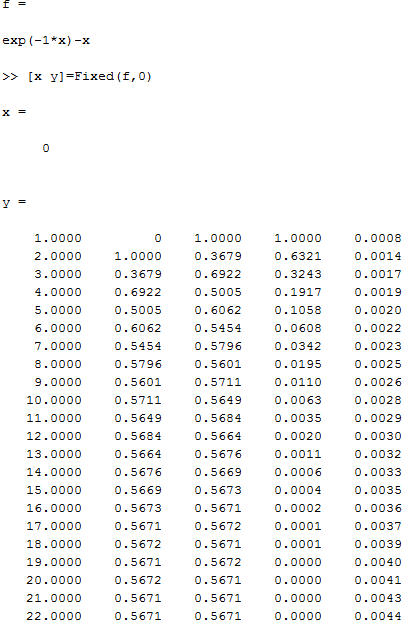
End

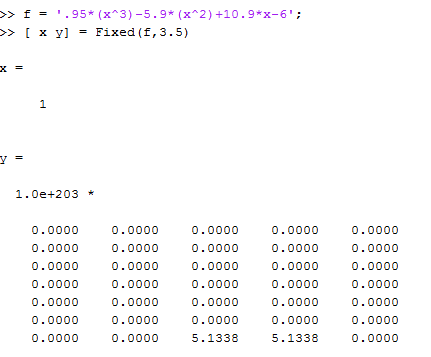
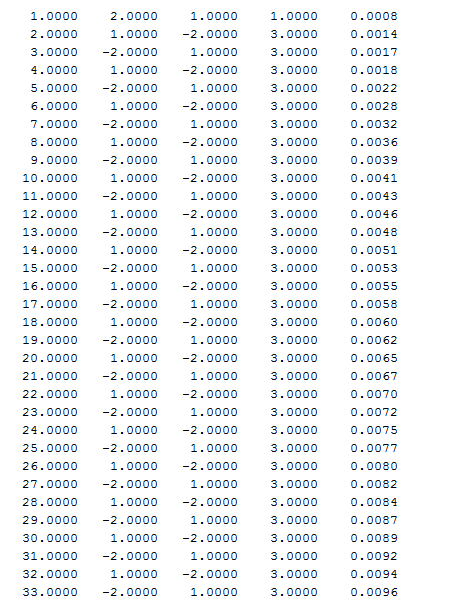
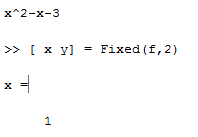
x1 = x2;

end

1. Pitfalls:
2. Guessing the initial point x0.
3. Multiple magic functions.
4. Diverge if |g’(x)| > 1
5. Number of iterations can’t be known prior.
6. Examples:

Converge:



Diverge:

# Secant Method

1. Overview:

This is an open method, that may converge or diverge, this method uses the newton method, but it approximates the derivatives by finite divided difference, so it takes 2 point as initial points for iterations.

1. Pseudo-Code:

For I = 1: numberOfIterations

x3 = x2 - ( f(x2) \* (x1-x2)) / (f(x1) - f(x2))

if(abs(x3-x2) < eps)

I = iterations

Root = x3

Break;

End

x1 = x2

x2 = x3

end

1. Pitfalls:
2. 2 points needed to start method.
3. May converge and may diverge.
4. Number of iterations can’t be known prior.
5. Division by zero if f(x1) = f(x2).
6. Work slowly with step curves.
7. Examples:

