

The Bisection Method

One way to define function in command window is:

```
>> f=@(x)x.^3+4*x.^2-10
f =
    @(x)x.^3+4*x.^2-10
>> f(3)
ans =
    53
```

Matlab Code: (in MATLAB editor)

```
function [x e] = mybisection(f,a,b,n)
% function [x e] = mybisection(f,a,b,n)
% Does n iterations of the bisection method for a function f
% Inputs: f -- an inline function
%         a,b -- left and right edges of the interval
%         n -- the number of bisections to do.
% Outputs: x -- the estimated solution of f(x) = 0
%         e -- an upper bound on the error
format long
c = f(a); d = f(b);
if c*d > 0.0
    error('Function has same sign at both endpoints.')
end
disp('          x          y')
for i = 1:n
    x = (a + b)/2;
    y = f(x);
    disp([    x    y])
    if y == 0.0    % solved the equation exactly
        e = 0;
        break    % jumps out of the for loop
    end
    if c*y < 0
        b=x;
    else
        a=x;
    end
end
end
```

To run this:

```
>> mybisection2(f,1,2, 13)
      x      y
1.500000000000000  2.375000000000000
1.250000000000000 -1.796875000000000
1.375000000000000  0.162109375000000
```

```
1.3125000000000000 -0.848388671875000
1.3437500000000000 -0.350982666015625
1.3593750000000000 -0.096408843994141
1.3671875000000000 0.032355785369873
1.3632812500000000 -0.032149970531464
1.3652343750000000 0.000072024762630
1.3642578125000000 -0.016046690754592
1.3647460937500000 -0.007989262812771
1.3649902343750000 -0.003959101522923
1.3651123046875000 -0.001943659010067
ans =
1.365112304687500
```