

DIS 004

Raehyun Kim*

1 Bisection

Code

```
1 % Bisection method
2 % Fun : function handle
3 % x0, x1 : interval
4 % N : MaxIt
5 % tol : desired accuracy
6 Fun = @(x) x*x-1; % root is +1 / -1
7 x0 = 0; x1 = 0.5;
8 % check the condition
9 fprintf('Fun(x0)*Fun(x1) = %.5f x0 is %e x1 is %e \n', Fun(x0)*Fun(x1), x0, x1);
10 N = 100;
11 tol = 1e-5; % 10**(-5) 10^(-5) pow(10,-5)
12
13 [fun, x, flg] = BisectionMethod(Fun, x0, x1, N, tol);
14 if flg == 0
15     %success
16     fprintf('BisectionMethod successfully finds root x = %f \n', x(end));
17 else
18     %failed
19     fprintf('BisectionMethod failed to find root \n');
20 end
```

*Department of Mathematics, University of California at Berkeley

2 Newton's Method

Code

```
1 % NewtonMethod
2 % Fun : function handle
3 % dFun : deriv of function (function handle)
4 % x0 : initial guess (starting point)
5 % params : object includes information to determine convergence
6 % params.MaxIt : maximal iteration / params.tol : accuracy
7 Fun = @(x) x*x-1;
8 dFun = @(x) 2*x;
9 % Fun = @(x) x.^3-5*x;
10 % dFun = @(x) 3*x.^2 - 5;
11 x0 = 10000000;
12 params.MaxIt = 100; params.tol = 1e-10;
13
14 [fun, dfun, x, out] = NewtonMethod(Fun, dFun, x0, params);
15 if out.flg == 0
16     %success
17     fprintf('NewtonMethod successfully finds root x = %f at iteration %d \n', x(end), out.it);
18 else
19     %failed
20     fprintf('NewtonMethod failed to find root \n');
21 end
```