Contents


```
% Ali Heydari
% Math 231, hw3
% Newton's Method

% Problem 3)part a) second root x = 1

% a = input('Please enter a value for the lower bound a: ');
% b = input('Please enter a value for the upper bound (b) : ');

x_k = ones(1,10);
error = zeros(1,10);
e_n = zeros(1,10);
```

Interactive Interface (with user input)

get initial conditions when we want it interactive :

```
% x_k(1) = input('Please enter the initial guess x_0: ');
% delta = input('Please enter the desired tolerance: ');
% f = input('Please enter f(x)?(type @(x) [then the function] ');
% f_prime = input('Please enter d/dx(f(x)) (derivative)?(type @(x) [then the function] ');
```

Non Interactive (without user input)

```
x_k(1) = 1.2;
f = @(x) x^3 - 3*x + 2;
delta = 10^(-6);
f_prime =@(x) 3*x^2 - 3;


xk = x_k(1);
fx = f(x_k(1));  % evaluate function at x_o: f(x_o)

counter = 1;      % counter
```

Method

```
% keep finding the root until f(x_k)~~ 0
```

```
while abs(f(x_k(counter))) > 2 * delta

    % Neton's method formula
    x_k(counter+1) = x_k(counter) - (f(x_k(counter)) / f_prime(x_k(counter)));

    % display xk
    xk = x_k(counter+1);
    % Evaluating the function at new x_k
    fx = f(x_k(counter+1));



    % no real zero, so we keep this as an arbitrary bound
   e_n(counter) = abs(xk - (-2));

     if fx <= 1 * 10 ^-6

            root = xk;
     end


     % find the error

     error(counter) = abs(x_k(counter+1) - x_k(counter));

     counter = counter + 1;
    % Update the counter
end
```

Output Formatting

```
disp(" ");
disp(" ");
fprintf('The root of the function is at x = %i \n', root);
fprintf('Number of iterations: %i \n', counter);
disp(" ");
disp(" ");

disp("        pn              |p_{n+1} - p_n|       e_n = |pn - p| ")
for i= 1 : counter

fprintf("%i   %i         %i         %i\n",i ,x_k(i),error(i),e_n(i));

end
```

Outputs

```
The root of the function is at x = 1.000416e+00
Number of iterations: 10


        pn              |p_{n+1} - p_n|       e_n = |pn - p|
1   1.200000e+00         9.696970e-02         3.103030e+00
2   1.103030e+00         5.067389e-02         3.052356e+00
3   1.052356e+00         2.595560e-02         3.026401e+00
4   1.026401e+00         1.314308e-02         3.013258e+00
5   1.013258e+00         6.614316e-03         3.006643e+00
6   1.006643e+00         3.318043e-03         3.003325e+00
7   1.003325e+00         1.661767e-03         3.001664e+00
8   1.001664e+00         8.315732e-04         3.000832e+00
9   1.000832e+00         4.159594e-04         3.000416e+00
10   1.000416e+00         0         0
```