# Code

## Newton Function

function [root] = Newton(x0,max\_Step)

%% NEWTON Determines root using Newton's method

%% Requires initial value x0 and max\_step; returns root.

% Print header (Each col is 4 space wide)

fprintf('%s %s \n', 'step', 'x');

% Set x = initial x, i.e. x0

x =x0 ;

% Run loop for 1 to max\_step

for i=1:max\_Step

% Evaluate fx = x^2 - 3x + 2

fx = x^2 - 3\*x + 2;

% Evaluate dfx

dfx = 2\*x - 3;

% Find x\_new from x, fx, dfx using the

x\_new = x - fx/dfx;

% Update x with x\_new

x =x\_new ;

% Print step number (i) and x.

% Col1: integer, 4 spaces wide

% Col2: float, 8 spaces wide with 4 decimal places

fprintf('%d \t%4.4f\n', i,x);

end

root = x;

fprintf('\n\n');

end

## HW7 file code

clc; clear;

max\_step = 5;

x0 = 1; % set starting value to 1

disp(['x0=',num2str(x0),' and max\_step=',num2str(max\_step)])

root = Newton(x0,max\_step);

% Try again, this time with 2

x0 = 3;

disp(['x0=',num2str(x0),' and max\_step=',num2str(max\_step)])

root = Newton(x0,max\_step);

% when we use x0=2 we get the root is 2 in all step ecause exect root is 2

% And try again, this time with x0 = 300. Did you get the

% result in 5 steps? If not, then what different

% can you do to get either 1 or 2 as the estimated root?

x0 = 300;

% max\_step = ??;

disp(['x0=',num2str(x0),' and max\_step=',num2str(max\_step)])

root=Newton(x0,max\_step);

% no by using 5 step we not get the result

% initial value is large and it reduce in every step almost half

x0 = 300;

max\_step = 15;

disp(['x0=',num2str(x0),' and max\_step=',num2str(max\_step)])

root=Newton(x0,max\_step);

# Output





