**Assignment 4**

**Question 1 – Matlab Code**

function Xdc = dcsolvealpha(Xguess, alpha, maxerr)

% Compute dc solution using newtwon iteration for the augmented system

% G\*X + f(X) = alpha\*b

% Inputs:

% Xguess is the initial guess for Newton Iteration

% alpha is a paramter (see definition in augmented system above)

% maxerr defined the stopping criterion from newton iteration: Stop the

% iteration when norm(deltaX)<maxerr

% Oupputs:

% Xdc is a vector containing the solution of the augmented system

global G C b

delta\_x = 2147483647;

x\_test = Xguess;

% since in DC this point is always 0

x\_test\_d = zeros(size(x\_test));

% continue iterating through until the threshold of maxerr is hit

while norm(delta\_x) >= maxerr

f = f\_vector(x\_test);

phi = G\*x\_test + C\*x\_test\_d + f - alpha\*b;

% Get the Jacobian matrix

J = nlJacobian(x\_test);

% get delta\_x matrix

delta\_x = -1 \* J \ phi;

% caclulate the new point to test and get the normal of delta\_x

x\_test = x\_test + delta\_x;

end

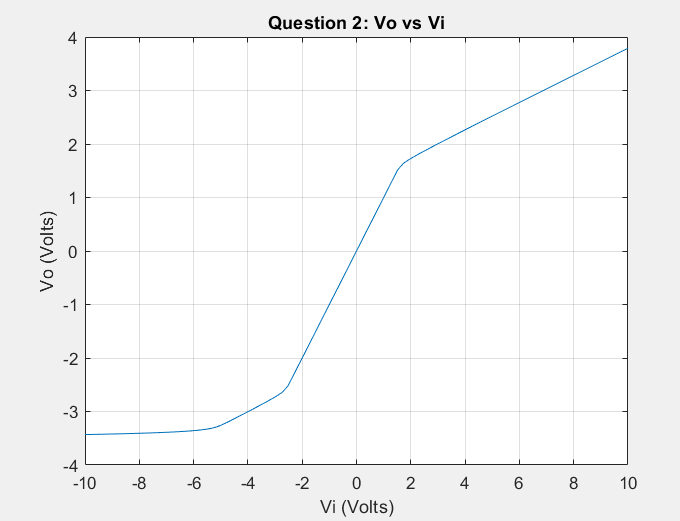
Xdc = x\_test;

**Question 2**

1. **Vo vs Vi**

|  |  |
| --- | --- |
| **Vi** | **Vo** |
| **-10 V** | **-3.4313 V** |
| **-2 V** | **-2 V** |
| **8 V** | **3.2841 V** |

**Plot for Vo vs Vi**



1. **Matlab Code**

function Xdc = dcsolvecont(n\_steps,maxerr)

% Compute dc solution using newtwon iteration and continuation method

% (power ramping approach)

% inputs:

% n\_steps is the number of continuation steps between zero and one that are

% to be taken. For the purposes of this assigments the steps should be

% linearly spaced (the matlab function "linspace" may be useful).

% maxerr is the stopping criterion for newton iteration (stop iteration

% when norm(deltaX)<maxerr

global G C b

% Vo is @ node 3

% vi is @ node 4

cont\_step = linspace(0,1,n\_steps);

% Set x\_guess to be 0 since it's the trivial solution

x\_guess = zeros(length(G), 1);

Xdc = 0;

for i = 1:n\_steps

% Xdc = Xdc + dcsolvealpha(x\_guess, cont\_step(i), maxerr);

x\_guess = dcsolvealpha(x\_guess, cont\_step(i), maxerr);

end

Xdc = x\_guess;