Bryan Jay
260738764

# 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;
```

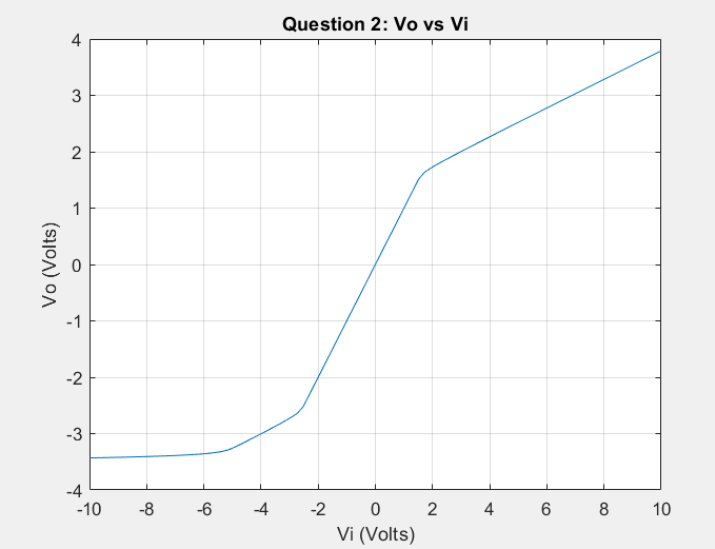
## Question 2

### a) $V_o$ vs $V_i$

| $V_i$ | $V_o$ |
|---|---|
| **-10 V** | **-3.4313 V** |
| **-2 V** | **-2 V** |
| **8 V** | **3.2841 V** |

Bryan Jay
260738764

**Plot for $V_o$ vs $V_i$**




**b) 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;
```