ECSE597/ECSE472
Assignment #5 part b

Preliminaries
1. You can reuse and build upon your previous code for this assignment.
2. We provide you with the following additional functions that help handle time domain sources.
   a. volsine.m and cursine.m are two functions that add time domain sinusoidal voltage and current sources respectively. They are used in the netlist. The netlists used in the questions below should provide examples. Also read the comments in the functions for the required syntax.
   b. BTime.m is a function that evaluates and returns the time domain value of the right hand side vector b(t) as a function of time.
3. In your submission please provide all code in a zip file in a way that allows us to run the testbenches ourselves (include all code, not just the recent one).
4. Also submit a pdf file containing the answers to the questions, the output plots and the code for functions you have written for this assignment.

Question I: Backward Euler
Write a function called transient_beuler.m with the following header:

```
function [tpoints,r] = transient_beuler(t1,t2,h,out)
% [tpoints,r] = beuler(t1,t2,h,out)
% Perform transient analysis for LINEAR Circuits using Backward Euler
% assume zero initial condition.
% Inputs:  t1 = starting time point (typically 0)
%          t2 = ending time point
%          h  = step size
%          out = output node
% Outputs  tpoints = are the time points at which the output
%                    was evaluated
%          r       = value of the response at above time points
% plot(tpoints,r) should produce a plot of the transient response
```

Your function should use Backward Euler with a constant step size to compute the transient response of a <u>linear circuit</u>.

1. Test your function by running the provided Testbench_Question1.m file. This file simulates the netlists Circuit_chebychev_filter_TD.m (provided in the assignment) in the time domain (transient) and Circuit_chebychev_filter (also provided – this is the same circuit you used before to test your fsolve) in the frequency domain using your fsolve.m function which you developed in past assignments. In your submission, include the code for the new function transient_beuler.m as well as the output plot of the testbench function.
2. Explain the relation between the two plots in the output figure.

Question II Trapezoidal Rule

Write a function transient_trapez.m with the following header:

```
function [tpoints,r] = transient_trapez(t1,t2,h,out)
% [tpoints,r] = Transient_trapez(t1,t2,h,out)
% Perform Transient Analysis using the Trapezoidal Rule for LINEAR
% circuits.
% assume zero initial condition.
% Inputs:   t1 = starting time point (typically 0)
%           t2 = ending time point
%           h  = step size
%           out = output node
% Outputs   tpoints = are the time points at which the output
%                     was evaluated
%           r       = value of the response at above time points
% plot(tpoints,r) should produce a plot of the transient response
```

Your function should use Trapezoidal Rule with a constant step size to compute the transient response of a <u>linear circuit</u>.

1. Test your function by running the provided Testbench_Question2.m file. This scripts also runs your code from Question 1 and compares BE to TR. In your submission, provide the code for the function you wrote as well as the plot from the testbench.
2. By examining the plot, what can you deduce about the BE and TR methods?

Question III Forward Euler

Write a function transient_feuler.m with the following header:

```
function [tpoints,r] = transient_feuler(t1,t2,h,out)
% [tpoints,r] = Transient_feuler(t1,t2,h,out)
% Perform Transient analysis for LINEAR circuit using Forward Euler
% This function assumes the C matrix is invertible and will not work
% for circuits where C is not invertible.
% assume zero initial condition.
% Inputs:   t1 = starting time point (typically 0)
%           t2 = ending time point
%           h  = step size
%           out = output node
% Outputs   tpoints = are the time points at which the output
%                     was evaluated
%           r       = value of the response at above time points
% plot(tpoints,r) should produce a plot of the transient response
```

Your function should use the Forward Euler method with a constant step size to compute the transient response of a <u>linear circuit</u>. You may assume that the C matrix of your MNA is invetible for the purposes of this assignment (this means that your function will not work for most circuits).

1. In order to test your code, run the provided test bench script Testbench_Question3.m which simulates the circuit in the provided netlist Q3BEcircuit.m.
2. Examine and comment on what you learn from the output files.
3. It can be shown that, when the C matrix is invertible, the poles of the circuit are the eigenvalues of the matrix $-C^{-1}G$ (note the negative sign). Determine the stability condition of the forward euler method for this circuit and experimentally verify your results by running simulations (note the eig function in matlab computes the eigenvalues of a matrix).

Question IV Nonlinear Circuits

Write a function nl_transient_beuler.m with the following header:

```
function [tpoints,r] = nl_transient_beuler(t1,t2,h,out)
% [tpoints,r] = beuler(t1,t2,h,out)
% Perform transient analysis for NONLINEAR Circuits using Backward Euler
% Assume zero initial condition.
% Inputs:   t1 = starting time point (typically 0)
%           t2 = ending time point
%           h  = step size
%           out = output node
% Outputs   tpoints = are the time points at which the output
%                     was evaluated
%           r       = value of the response at above time points
% plot(tpoints,r) should produce a plot of the transient response
```

Your function should use Backward Euler with a constant step size to compute the transient response of a <u>nonlinear circuit</u>. For now, you only need to support diodes and you can reuse the code you wrote before and that we have supplied to you.

Test your circuit by running the provided script Testbench_Question4 which simulates the rectifier circuit in netlist Circuit_Rectifier.m also provided.