ECSE 597

Assignment #3

Preliminaries

1. You can reuse and build upon the code you created for previous assignments assignment.
2. We provide you with the following additional functions that help handle time domain sources.
    a. *makeBt.m* function adds the stamp of time domain sinusoidal voltage and current sources. Call this function in the transient analysis methods to create the $\boldsymbol{B}(t)$ vector in MNA at a given value of time.
    b. Also, the stamps of resistors, capacitors, DC, AC, Time domain voltage sources and ideal Op Amps are provided to you in this assignment. You can also use the stamps you created in the previous assignments.
    c. Use the file InitiateCircuit.m provided with this assignment.

3. In your submission, please provide all code in a zip file in a way that allows us to run thetest benches 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 (5marks)
Write a function called *BE_method.m* with the following header:

```
function [tpoins,y]= BE_method(tEnd,h, outNode)
% This function uses BACKWARD EULER method to compute the transient reponse
% of the circuit.
%Inputs: 1. tEnd:  The simulation starts at time = 0s and ends at time =
%                  tEND s.
%        2. h: length of step size.
%        3. outNode: is the node for which the transient is required.
%Output:  1. tpoints are the time points. 2. y:  is the transient response at output
node.
```

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

1. Test your function by running the provided *Testbench_q1.m* file. This file simulates the netlists *Circuit_chebychev_filter_TD.m* (it has time domain sources) and *Circuit_chebychev_filter_freq.m* (it has frequency domain sources) in the frequency domain using your *fsolve.m* function which you developed in past assignments. In your submission, include the codefor the new function *BE_method.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(5marks)

Write a function *Trapezoidal_method.m* with the following header:

```
function [tpoints,y]= Trapezoidal_method(tEnd,h, outNode)
% This function uses Trapezoidal method to compute the transient reponse
% of the circuit.
%Inputs: 1. tEnd:  The simulation starts at time = 0s  and ends at
%                  time = tEND s.
%        2. h: length  of step size.
%        3. outNode: is the node for which the transient is required.
%Output:  1. tpoints are the time points. 2.y:  is the transient response at output
Node.
```

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

1. Test your function by running the provided *Testbench_q2.m* file. This script alsoruns 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 test bench.
2. By examining the plot, what can you deduce about the BE and TR methods?

Question III Forward Euler(5marks)

Write a function *FE_method.m* with the following header:

```
function [tpoints,y]= FE_method(tEnd,h, outNode)
% This function uses FORWARD EULER method to compute the transient reponse
% of the circuit.
%Inputs: 1. tEnd:  The simulation starts at time = 0s  and ends at
%                  time = tEND s.
%        2. h: length  of step size.
%        3. outNode: is the node for which the transient is required.
%Output:  1. tpoints are the time points. 2.y:  is the transient response at outNode.
```

Your function should use the Forward Euler method with a constant step size to compute the transient response of a linear circuit. You may assume that the C matrix of your MNA is invertible for the purposes of this assignment.

To test your code, run the provided test bench script *Testbench_q3.m* which simulates the circuit in the provided netlist Q3BEcircuit.m.

1. Examine and comment on your observations from the output files.
2. 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 Sensitivity using Perturbation method (3marks)

Write a function named *sens_perturbation_method.m* to compute the sensitivity of output voltage with respect to all the resistive and capacitive elements in the circuit using **perturbance method** (alternatively called difference method). Choose an appropriate value of the perturbance. Your function must compute the absolute and relative sensitivity of output with respect to all parameters at all frequency points. The function stub is provided below.

```
function [D,S] = sens_perturbation_method(fpoints,eleNames,outNode)
% This function uses DIFFERENCE method to compute the sensitivity of the
% output node with respect to all the parameters.
%Inputs: 1. fpoints: contains the frequency points at which the sensitivity is
%                 is required.
%        2. eleNames: is a cell array contains the names of the elements.
%        3. outNode: is the node for which the sensitivity is required.
%Output:  1.D:  is a matrix.  It should contain the ABSOLUTE sensitivity of the
%              outNode at all fpoints for all elements.
%              One way to fill store senstivity in  D is to  add sensitivity of
%              a given element in one column of D for all fpoints.
%              In this case if there are F number of frequency points and
%              P number of elements in eleNames, then the size of matrix D
%              will be FxP.
%         2. S: is  matrix. It should contain the RELATIVE sensitivity of
%               outNode for all the elements in eleNames. It can be filled
%               similar to matrix D.
```

Note: You do not have to adhere to the function stub provided above. You have the freedom to modify the function as per your preference.

Question V Sensitivity using the Differentiation method (3marks)

Write a function named *sens_differentiation_methood.m.* Your function should compute the sensitivity of output voltage for all the resistors and capacitors in the circuit using **differentiation method**. Your function must compute the absolute and relative sensitivity of output with respect to all parameters at all frequency points. The function stub is provided below.

```
function [D,S] = sens_differentiation_method(fpoints,eleNames,outNode)
% This function uses the DIFFERENTIATION method to compute the sensitivity of the
% output node with respect to all the parameters.
%Inputs: 1. fpoints: contains the frequency points at which the sensitivity is
%                 is required.
%        2. eleNames: is a cell array contains the names of the elements.
%        3. outNode: is the node for which the sensitivity is required.
%Output:  1.D:  is a matrix.  It should contain the ABSOLUTE sensitivity of the
%              outNode at all fpoints for all elements.
%              One way to fill store senstivity in  D is to  add sensitivity of
%              a given element in one column of D for all fpoints.
%              In this case if there are F number of frequency points and
%              P number of elements in eleNames, then the size of matrix D
%              will be F x P.
%         2. S: is  matrix. It should contain the RELATIVE sensitivity of
%               outNode for all the elements in eleNames. It can be filled
%               similar to matrix D.
```

Note: You do not have to adhere to the function stub provided above. You have the freedom to modify the function as per your preference.

Question VI Sensitivity using the Adjoint method (3 marks)

Write a function *sens_adjoint_method.m* to compute the sensitivity of output voltage for all the resistors and capacitors in the circuit using the **adjoint method**. Your function should evaluate the absolute and relative sensitivity of output with respect to all parameters at all frequency points. The function stub is provided below.

```
function [D,S] = sens_adjoint_method(fpoints,eleNames,outNode)
% This function uses the ADJOINT method to compute the sensitivity of the
% output node with respect to all the parameters.
%Inputs: 1. fpoints: contains the frequency points at which the sensitivity is
%                  is required.
%        2. eleNames: is a cell array contains the names of the elements.
%        3. outNode: is the node for which the sensitivity is required.
%Output:  1.D:  is a matrix.  It should contain the ABSOLUTE sensitivity of the
%              outNode at all fpoints for all elements.
%              One way to fill store sensitivity in D is to  add sensitivity of
%              a given element in one column of D for all fpoints.
%              In this case if there are F number of frequency points and
%              P number of elements in eleNames, then the size of matrix D
%              will be F x P.
%         2. S: is  matrix. It should contain the RELATIVE sensitivity of
%               outNode for all the elements in eleNames. It can be filled
 %               similar to matrix D.
```

Note: You do not have to adhere to the function stub provided above. You have the freedom to modify the function as per your preference.

Question VII (6 marks)

Test the functions written in Questions IV, V and VI by running (or modifying it according to your requirements) the code provided in *Testbench_q7.m.*

This code calls the netlist of a low-pass filter from the file *sallenkey_lp.m.*

Provide the plots of absolute and relative sensitivity obtained for R1 and C1.

For each element, plot the **adjoint method**, **differentiation method** and **perturbation method** results on the same graph.