SPICE

Kurupati Sai Pruthvi Teja

Problem Statement

- To implement as SPICE program for simulation of network containing R,L,G,C elements.

Matlab files and description

- CAD_HW1 – main file for parsing and operations
- pulse_wave – generation of pulse wave from given parameters
- pwl_wave – generation of pwl wave from given parameters
- sin_wave – generation of sine wave from given paramet
- stamp_dc – generates stamp for dc operating point
- str2unitConv – conversion of given string into corresponding numeric
- String_split_comma – splits a given string into two separated by comma
- trans_Sim – transient simulation function for dc voltage
- trans_Sim2 – transient simulation function for alternate voltage sources

Approach to the SPICE Program : Steps Involved

- Reading file
- Parsing Netlist and Simulation Commands
- Making data structures for individual components
- Constructing A and b matrices using data structures formed
- Performing DC operating point analysis $[A^{-1}b]$
- Transient analysis
- Generation of alternate sources (sin, pulse, pwl) from given parameters
- DC operating point for alternate sources
- Transient analysis for alternate sources

Approach to the SPICE Program : Parsing

- Parsing of SPICE file is done by reading the given spice code line by line.
- During parsing the components of the circuit are identified and corresponding matrices are formed for each individual type element. Each row of formed matrix contains nodes and parameters for that component.
- Simulation commands are parsed and the corresponding parameters such as time step, end time, alternate sources parameters are stored.
- For unit conversion of parameters containing ‘ms’ , ‘kHz’, ‘k’, ‘Meg’ etc.. a function named **string2unitConv** is written.

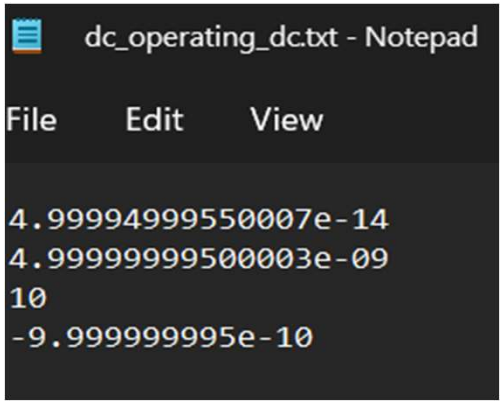
Key functions used : strcmp, str2double, feof, fopen, fgetl, split

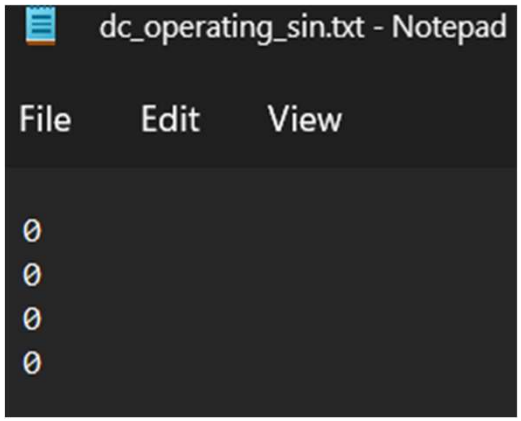
Approach to the SPICE Program : DC and Transient analysis

- From the parsed data A and b matrices are formed by combining individual stamps of components. The matrices are formed according to modified node analysis (MNA).
- DC operating point is calculated where the ac source is shorted and capacitors are opened. The obtained operating point is exported into text file.
- The transient analysis is done considering initial conditions as zero i.e. capacitor is uncharged.
- For transient analysis b matrix is updated for every time step and $A^{-1}b$ is calculated.
- The functions written for transient analysis is ***trans_Sim***, ***trans_Sim2***
- The alternate sources are generated by using given parameters. The functions written for this are ***sin_wave, pwl_wave, pulse_wave***.

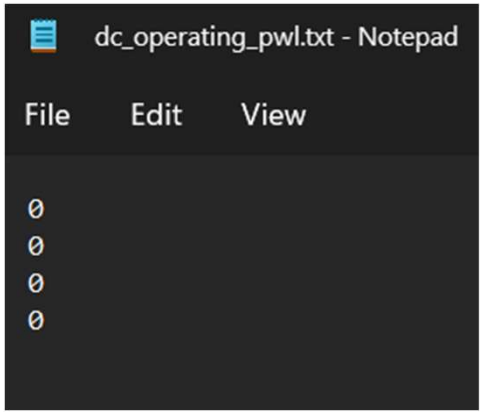
Key functions used : linsolve

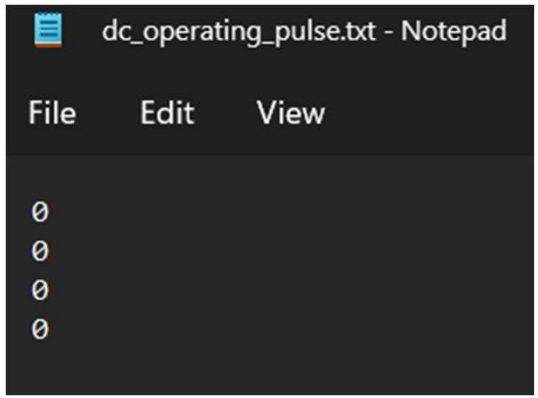
Results : P2test1.sp – DC Operating Point



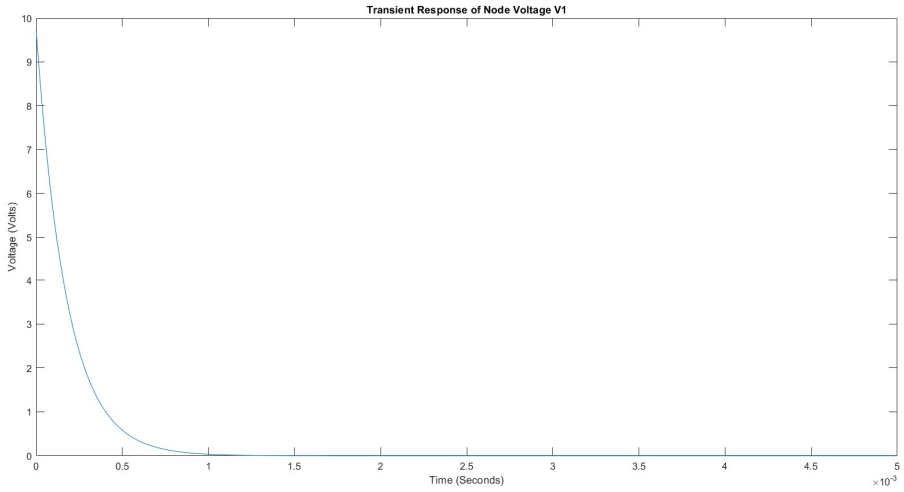


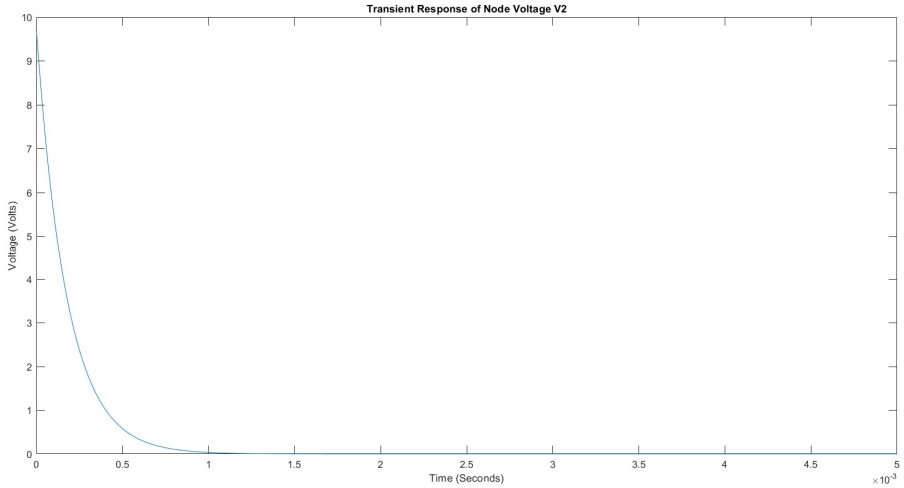
Order of Values
Node Voltage (V1)
Node Voltage (V2)
Node Voltage (V3)
Branch Current (Is)

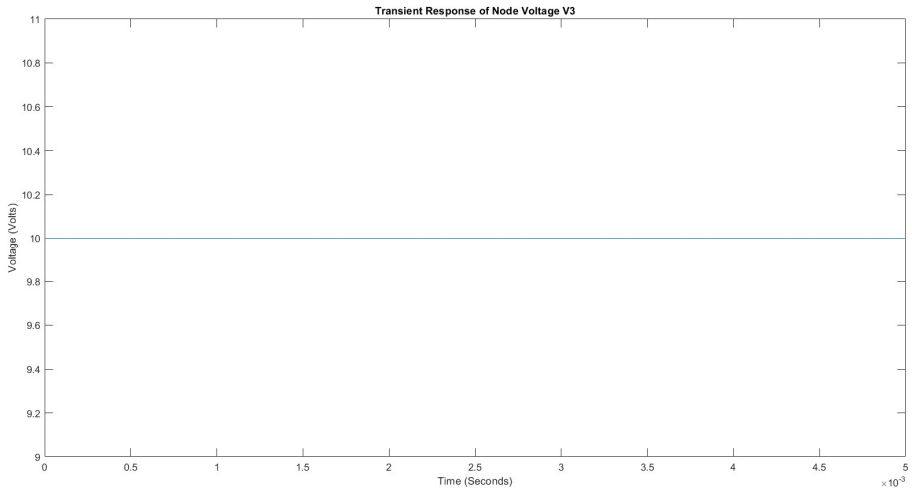




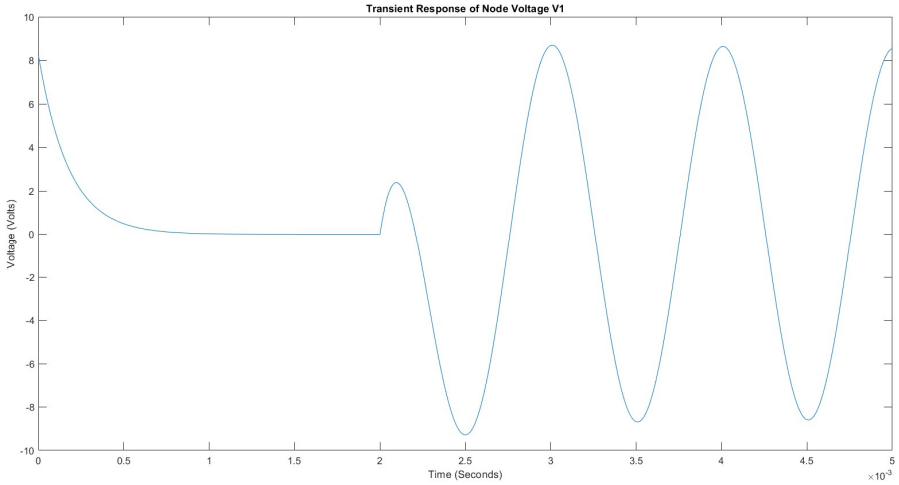
Results : P2test1.sp – DC Voltage

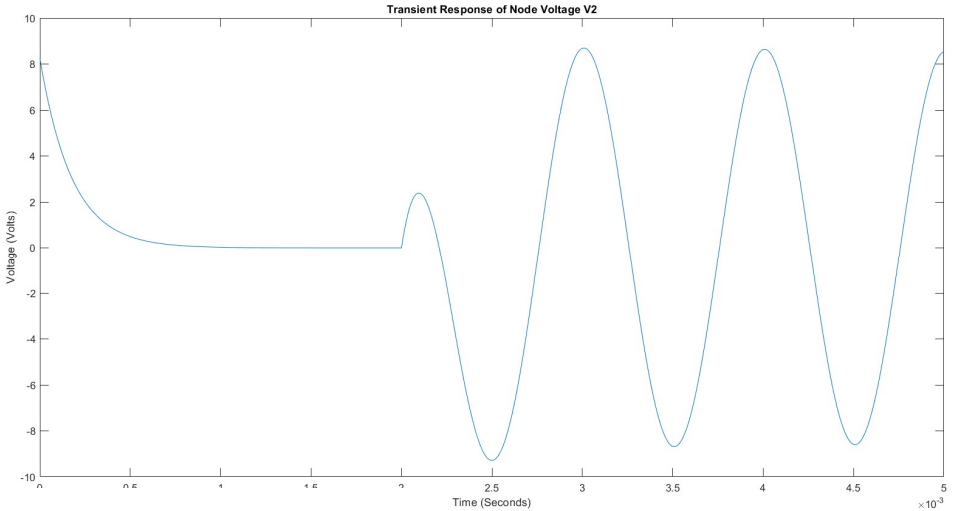


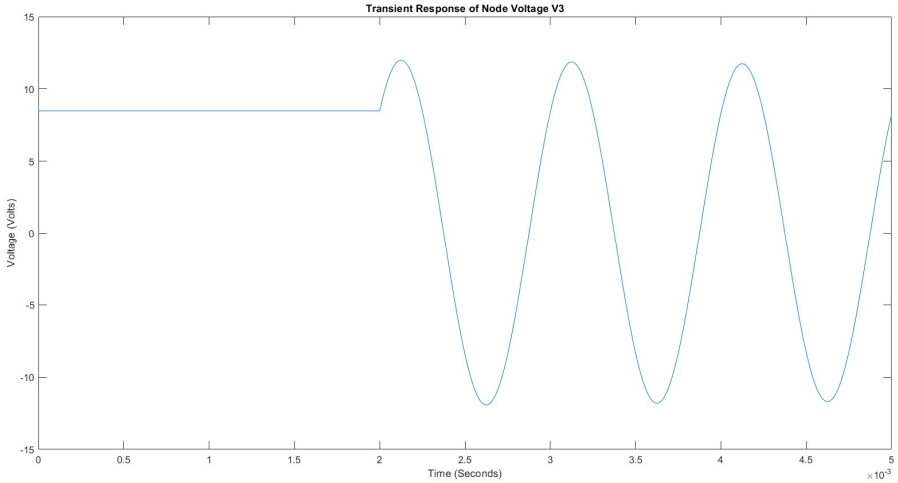




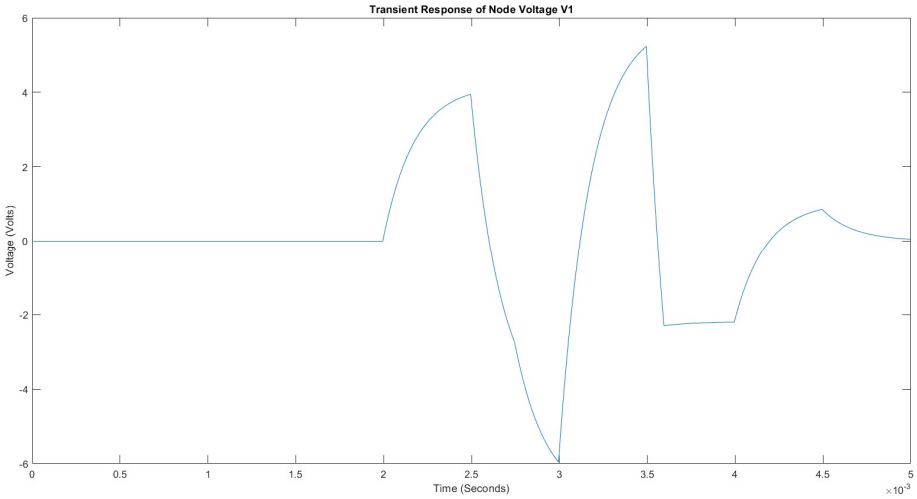
Results : P2test1.sp – Sine Wave

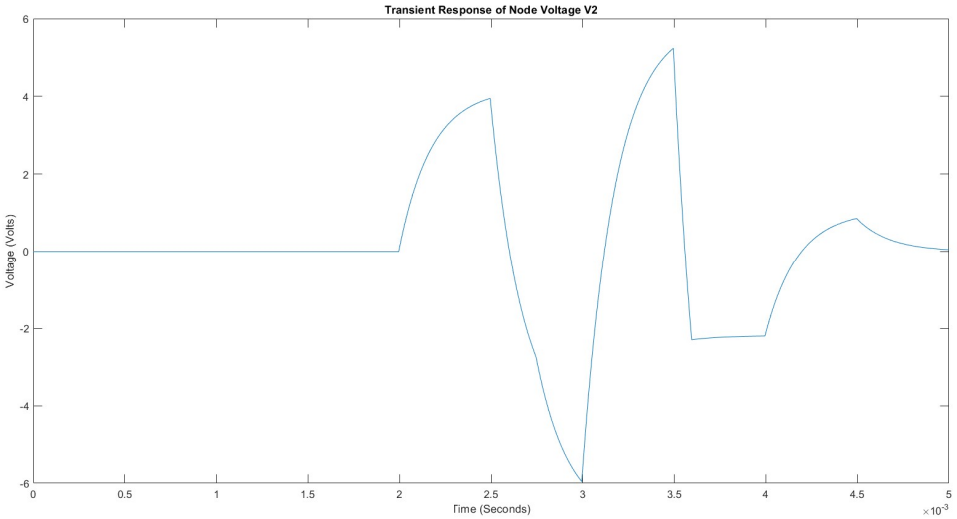


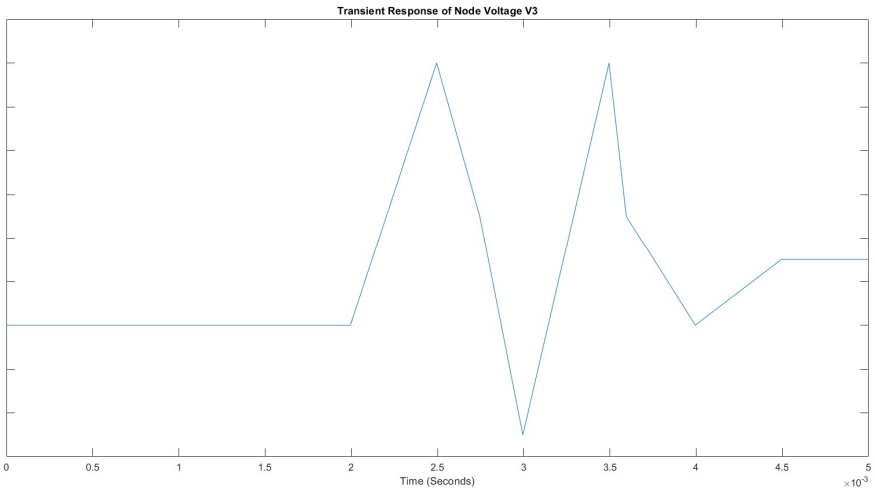




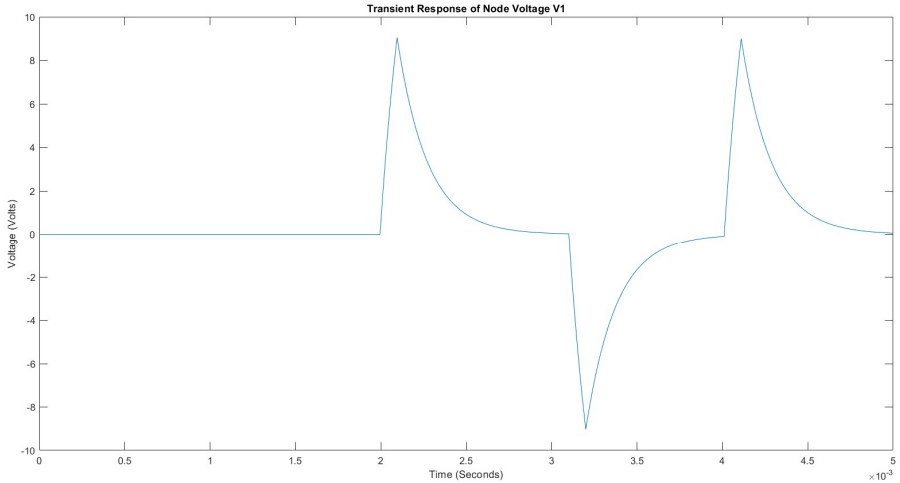
Results : P2test1.sp – Piece Wise Linear Wave

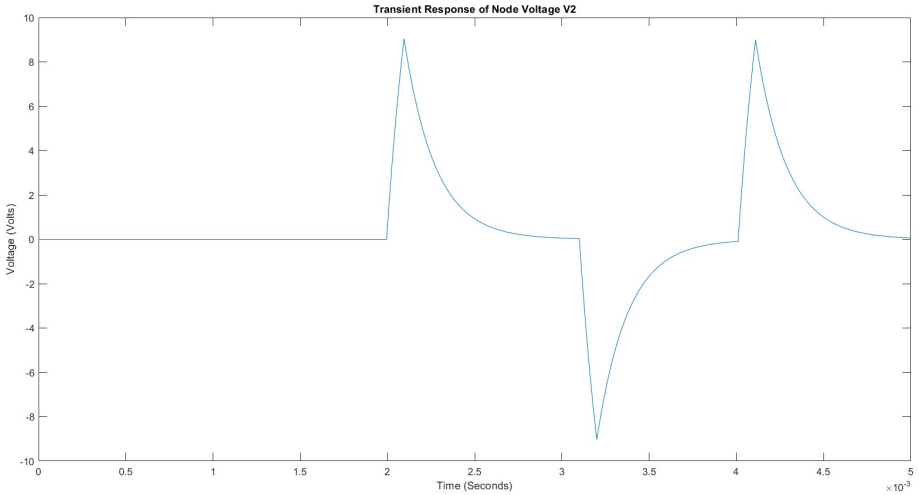


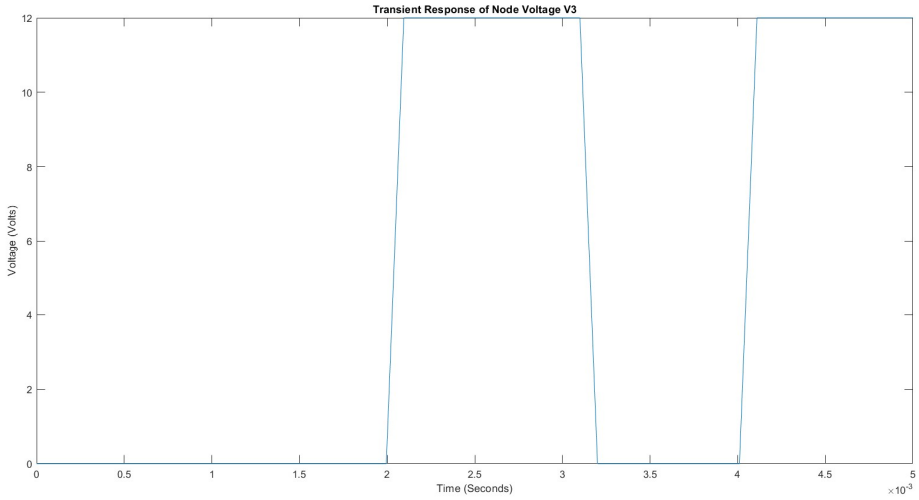




Results : P2test1.sp – Pulse Wave







Results : P2test2.sp – DC Operating Point

dc_operating_dc.txt - Notepad

File Edit View

```
10
9.99999500000319
9.99999460000345
9.99999360000408
-9.99999362222975e-10
```

dc_operating_pwl.txt - Notepad

File Edit View

```
2.999
2.99899850050096
2.99899838054103
2.99899808064123
-2.9989980876324e-10
```

Order of Values

Node Voltage (V1)
Node Voltage (V2)
Node Voltage (V3)
Node Voltage (V4)
Branch Current (Is)

dc_operating_pulse.txt - Notepad

File Edit View

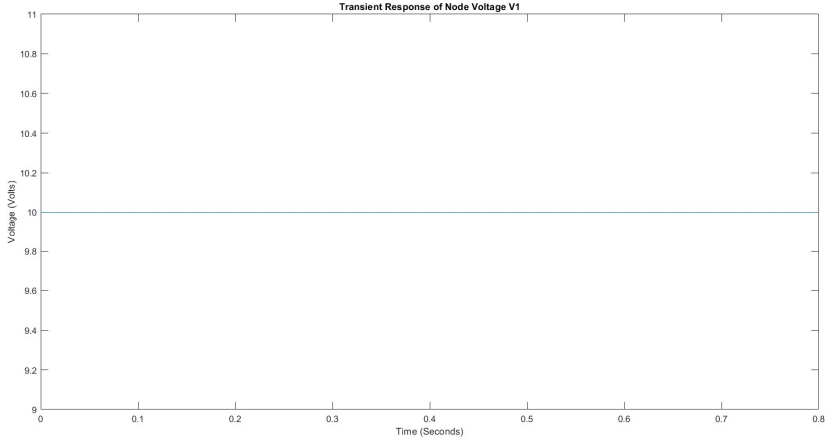
```
1
0.999999500000319
0.999999460000345
0.999999360000409
-9.99999362331395e-11
```

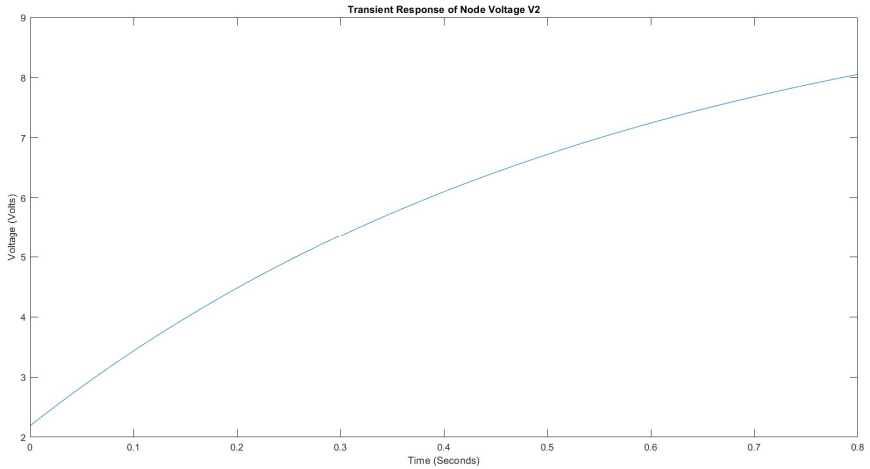
dc_operating_sin.txt - Notepad

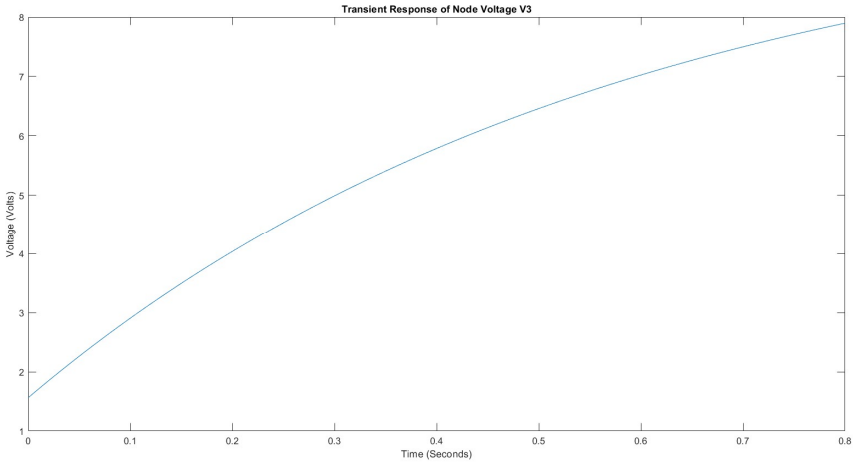
File Edit View

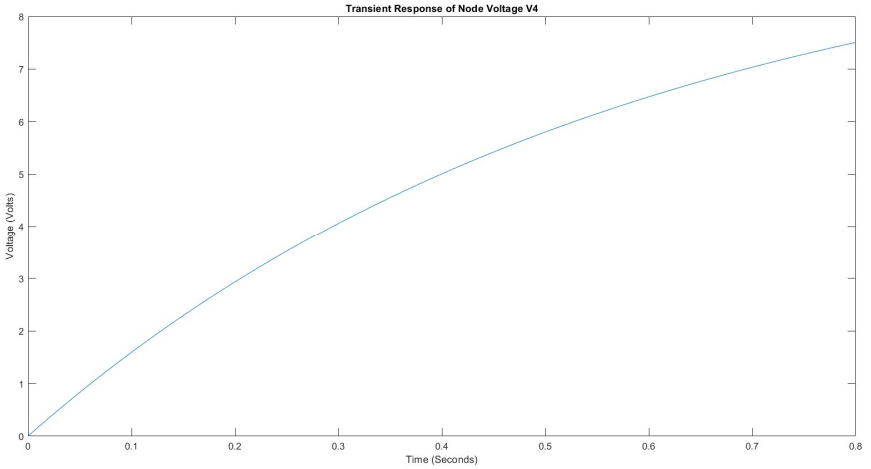
```
2
1.99999900000064
1.99999892000069
1.99999872000082
-1.99999872466279e-10
```

Results : P2test2.sp – DC Voltage

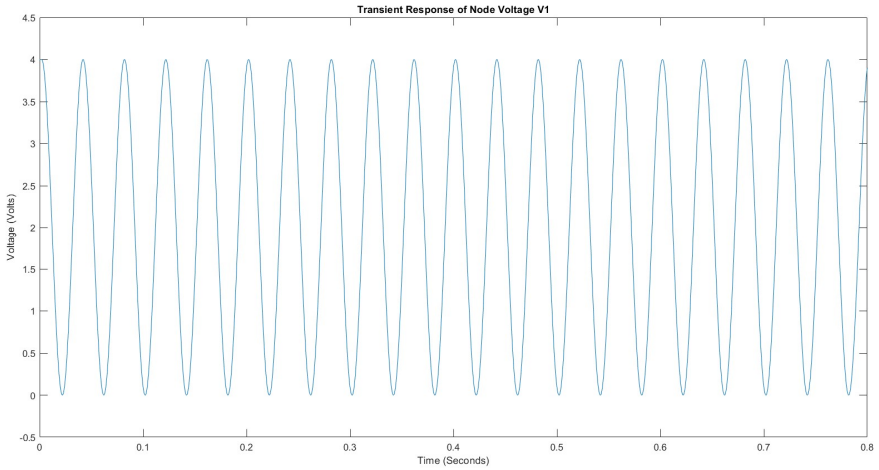


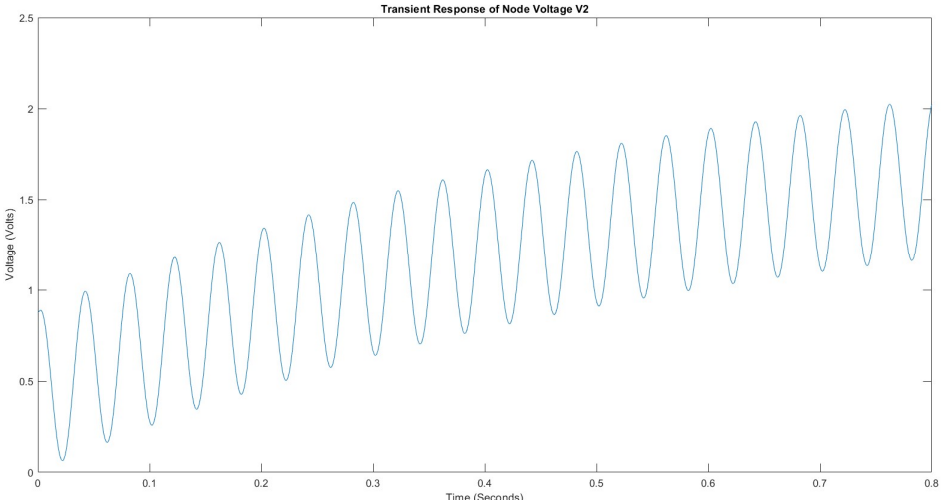


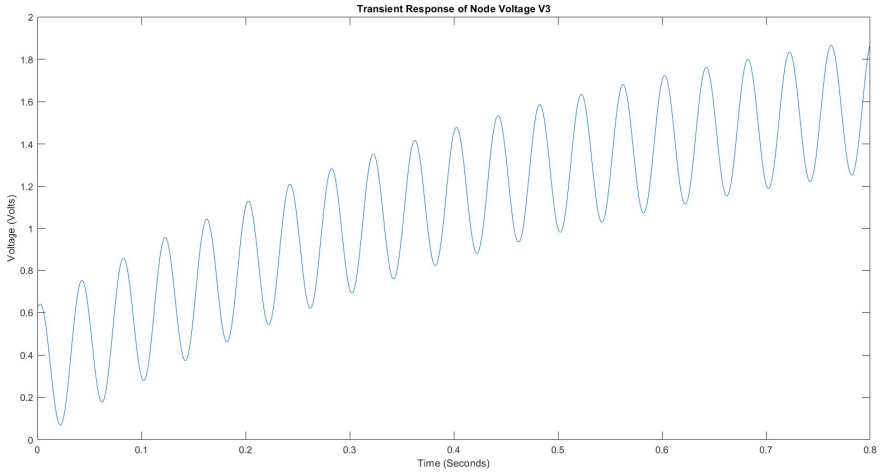


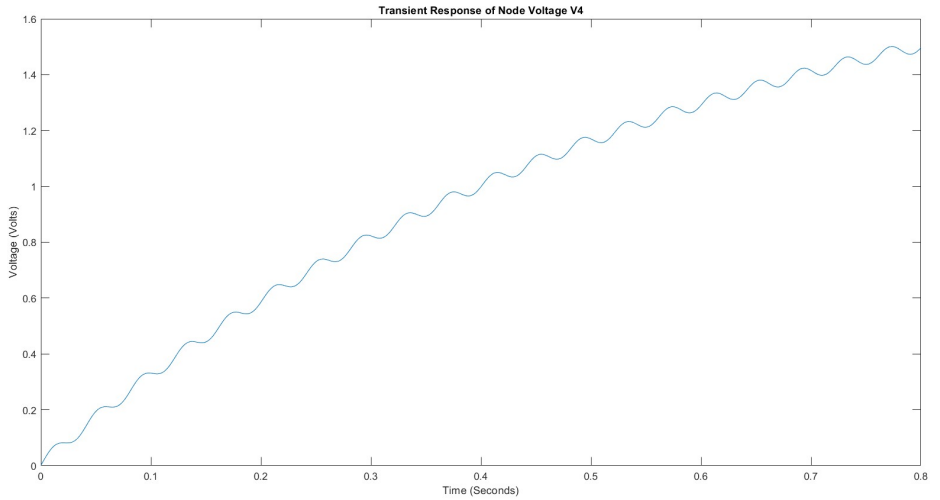


Results : P2test2.sp – Sine Wave

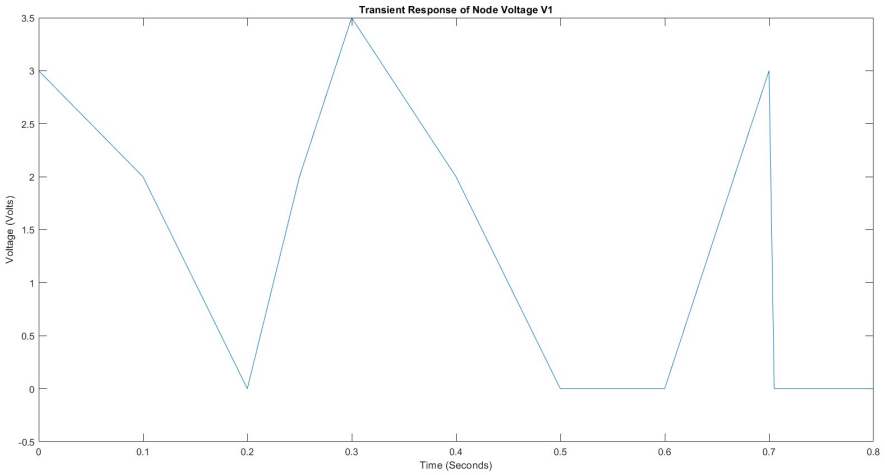


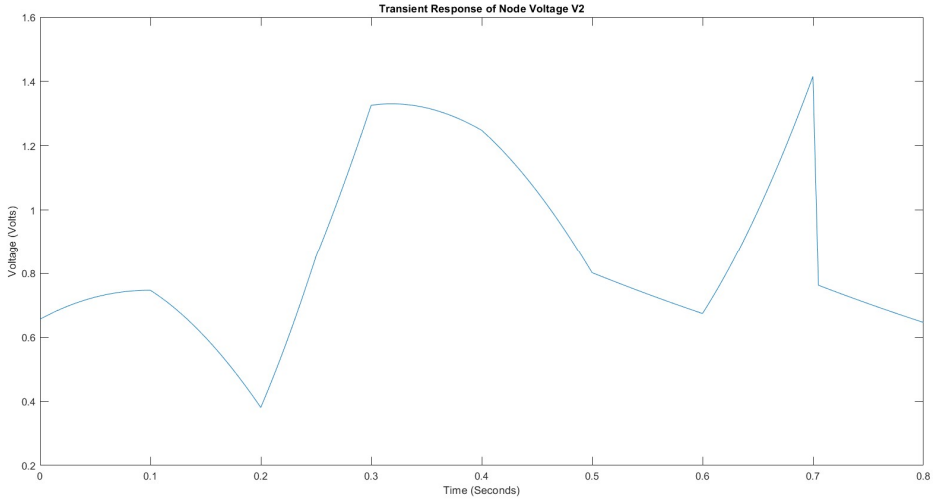


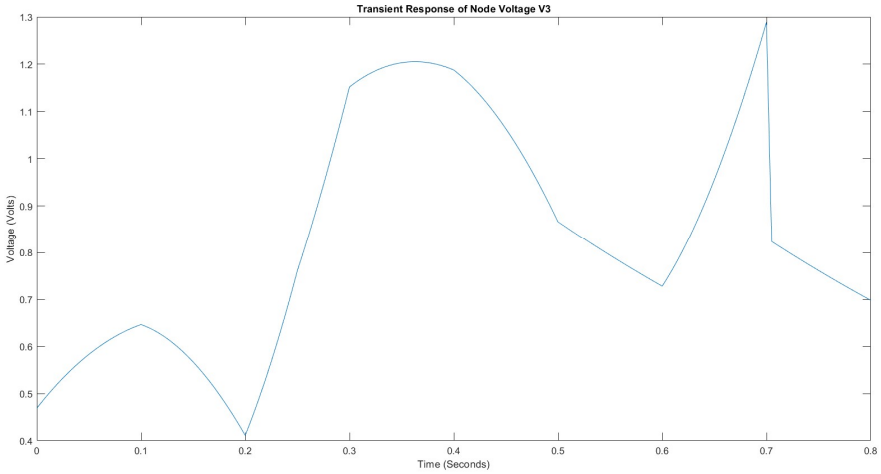


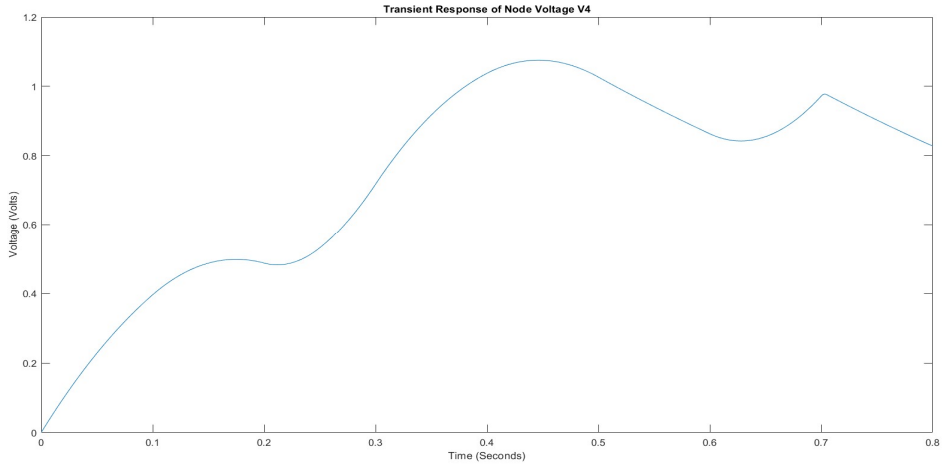


Results : P2test2.sp – Piece Wise Linear Wave

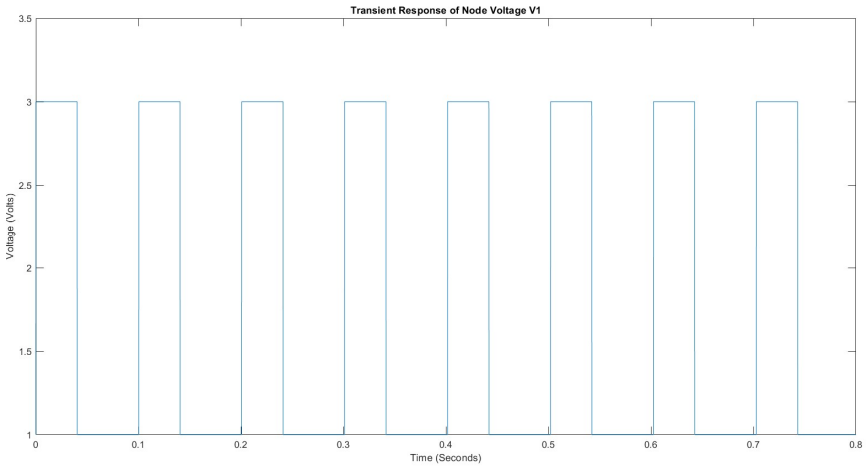


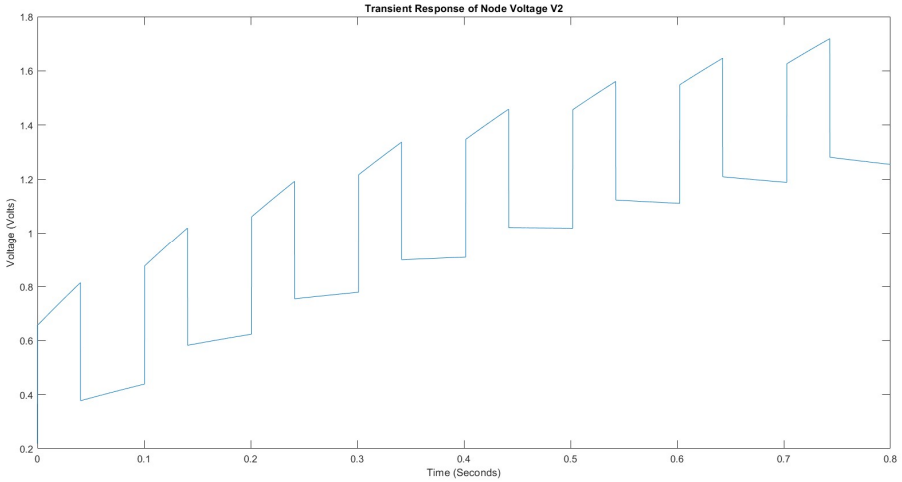


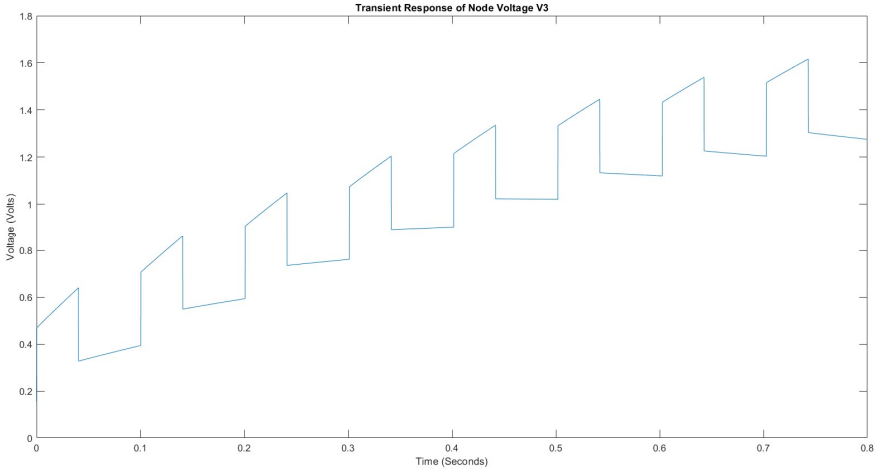


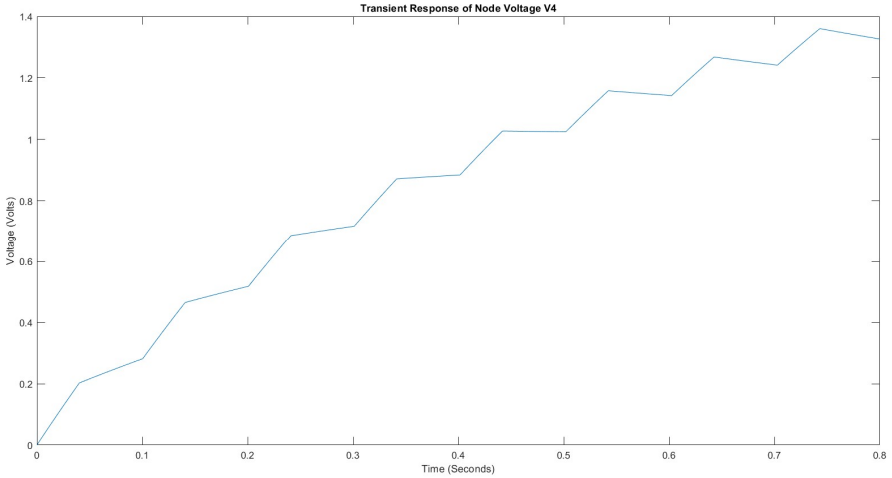


Results : P2test2.sp – Pulse Wave

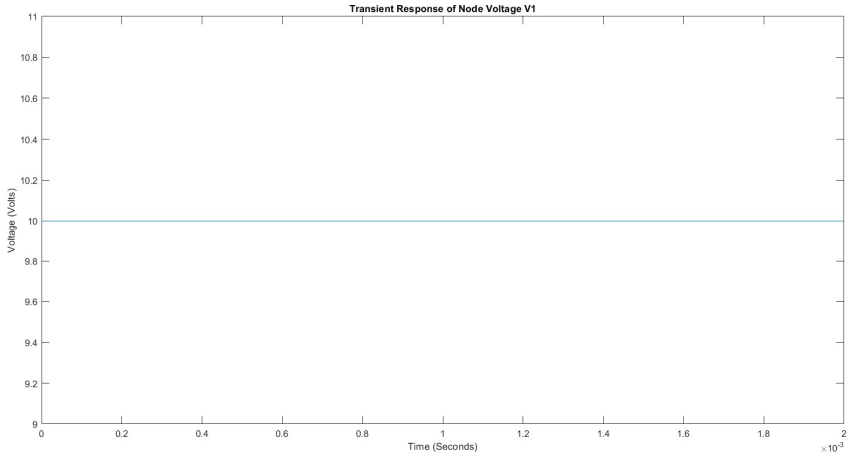


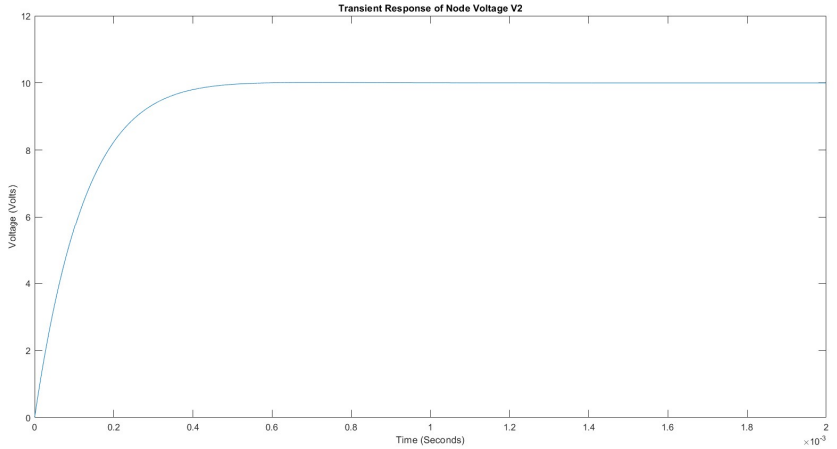


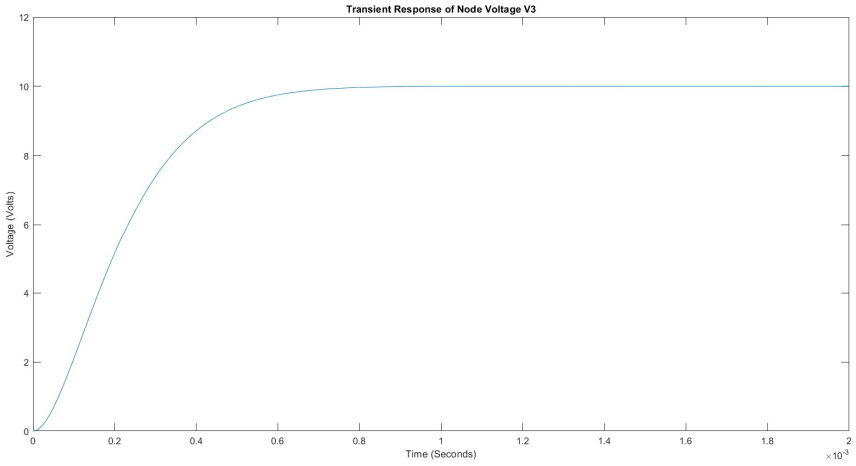


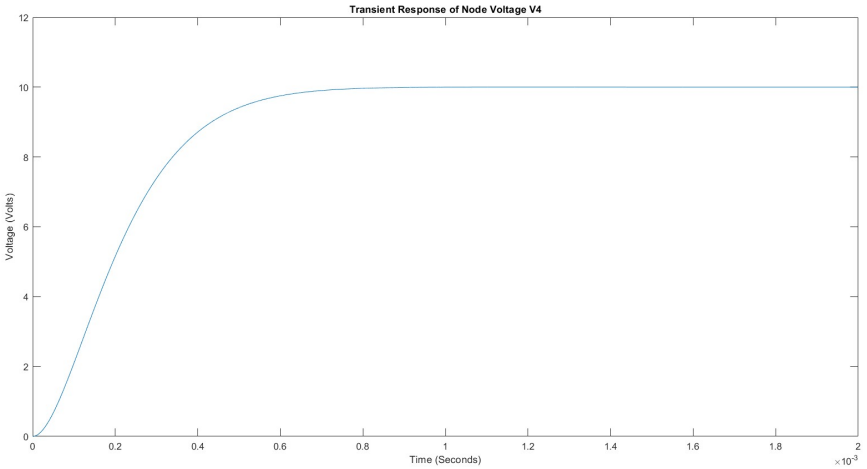


Results : P2test3.sp – DC Voltage









Results : P2test3.sp – DC Operating Point

dc_operating_dc.txt - Notepad

File Edit View

```
10
9.9999990000001
9.9999980000003
9.99999800011141
-9.99999899915233e-10
```

dc_operating_pwl.txt - Notepad

File Edit View

```
0.00022
0.000219999978000002
0.000219999956000007
0.000219999956002451
-2.19999978148691e-14
```

Order of Values

Node Voltage (V1)
Node Voltage (V2)
Node Voltage (V3)
Node Voltage (V4)
Branch Current (Is)

dc_operating_pulse.txt - Notepad

File Edit View

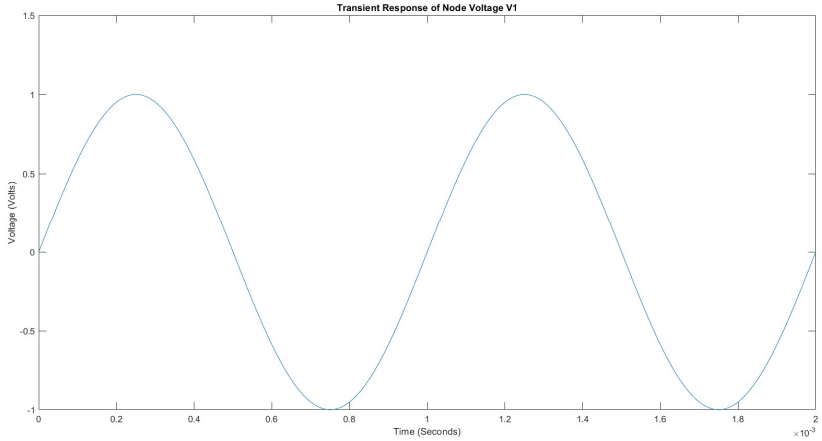
```
0
0
0
-0
0
```

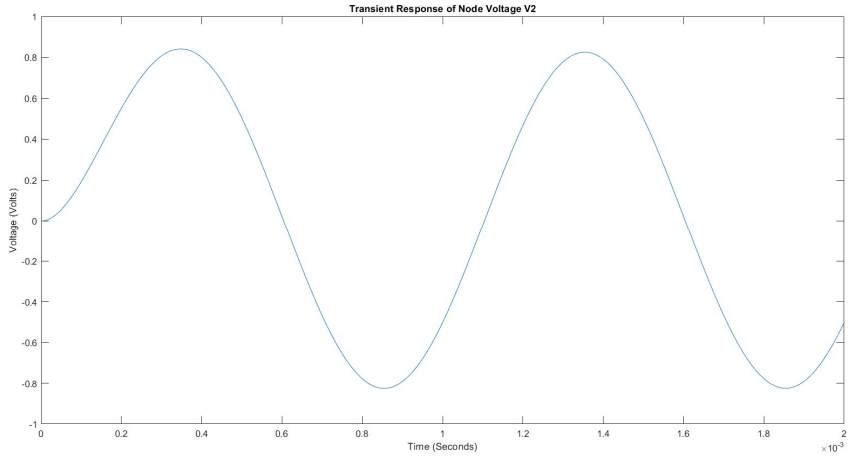
dc_operating_sin.txt - Notepad

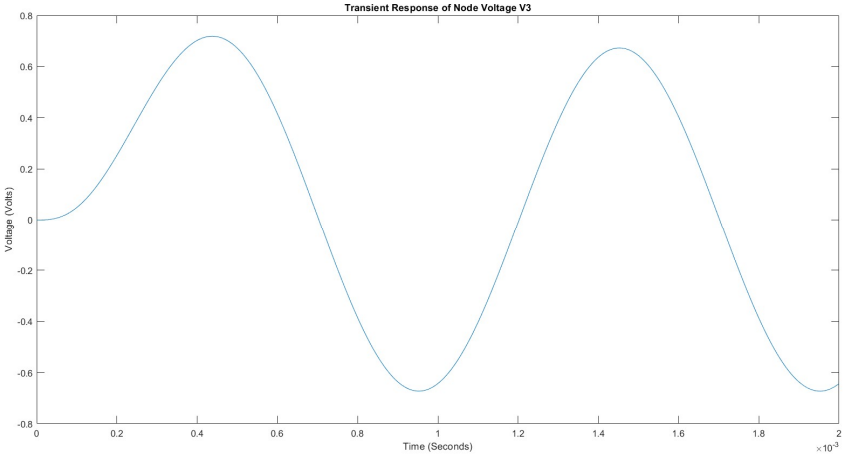
File Edit View

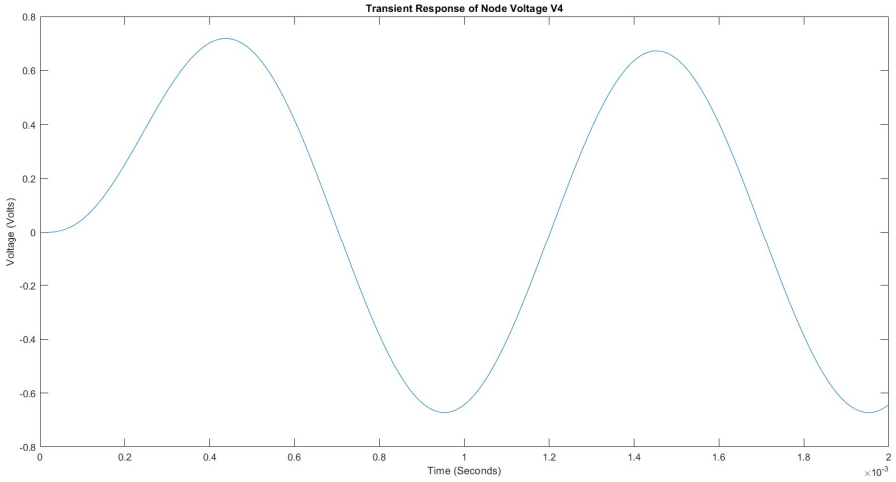
```
0
0
0
-0
0
```

Results : P2test3.sp – Sine Wave

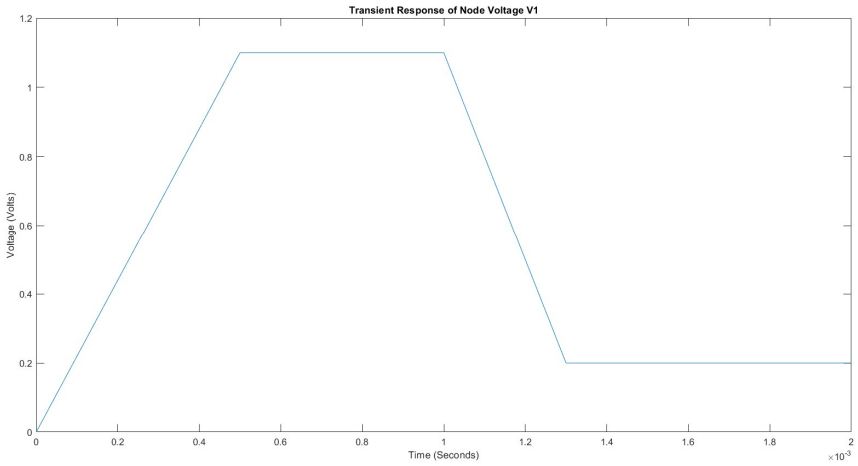


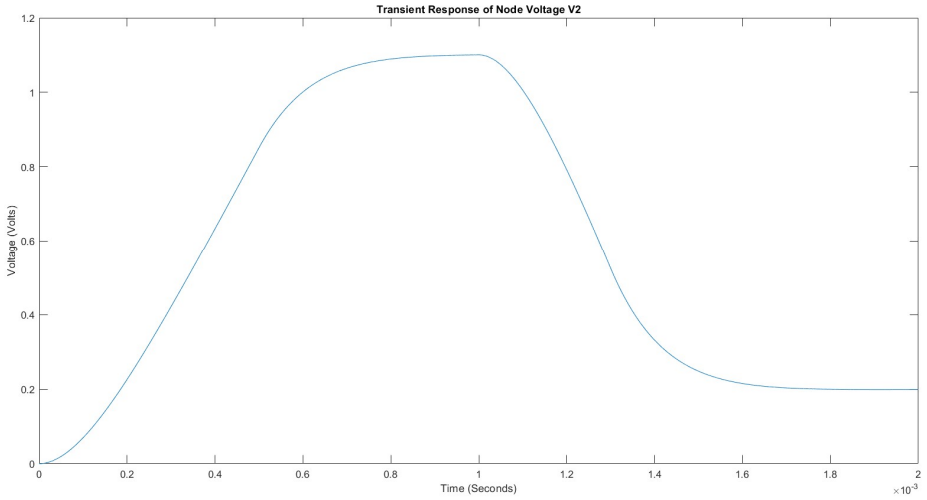


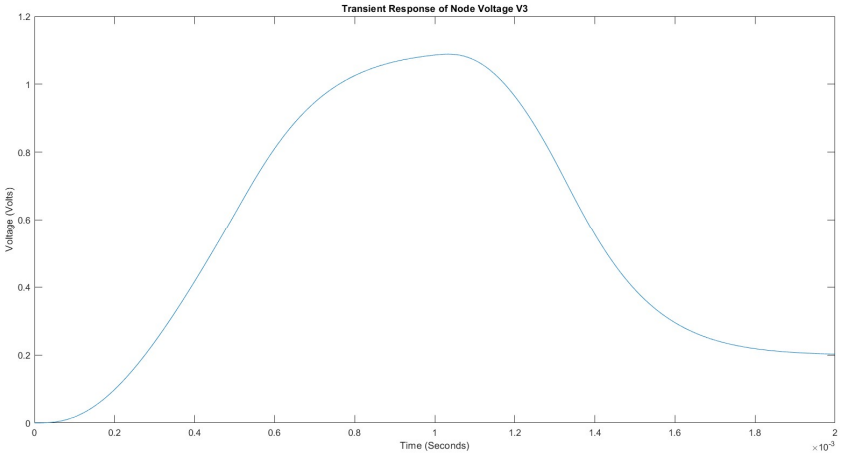


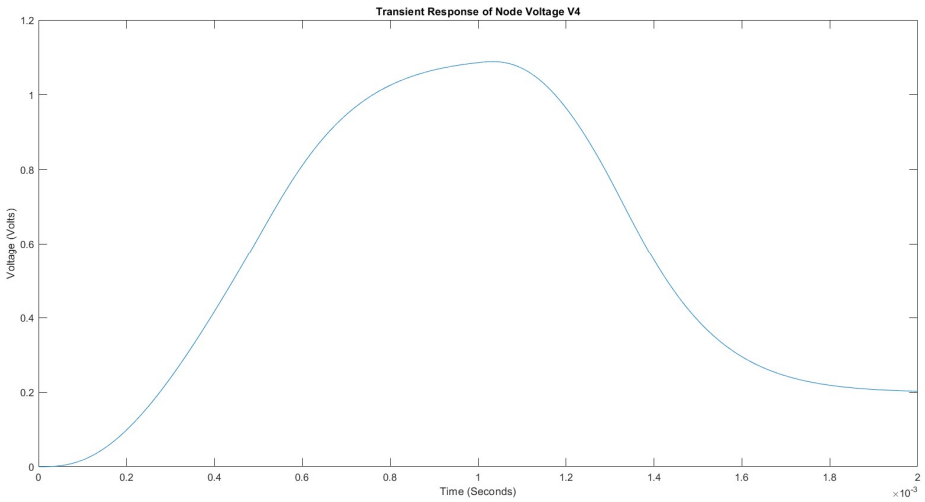


Results : P2test3.sp – Piece Wise Linear Wave

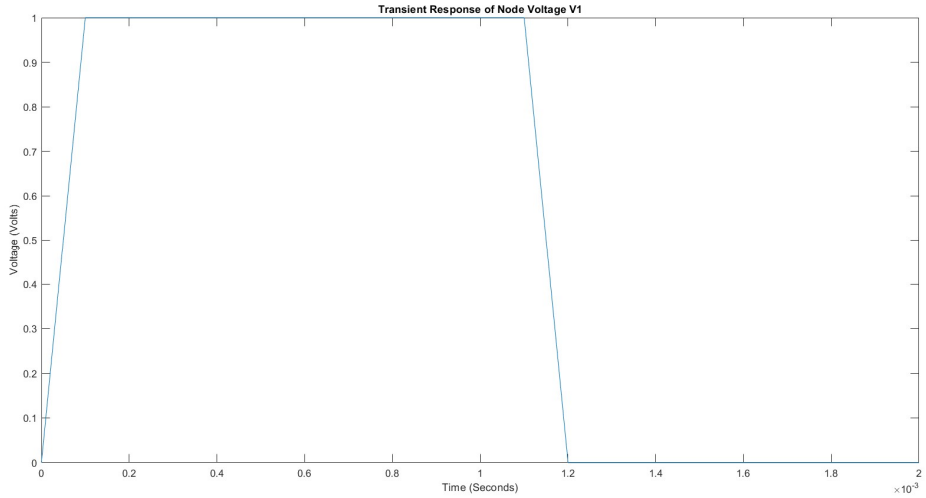


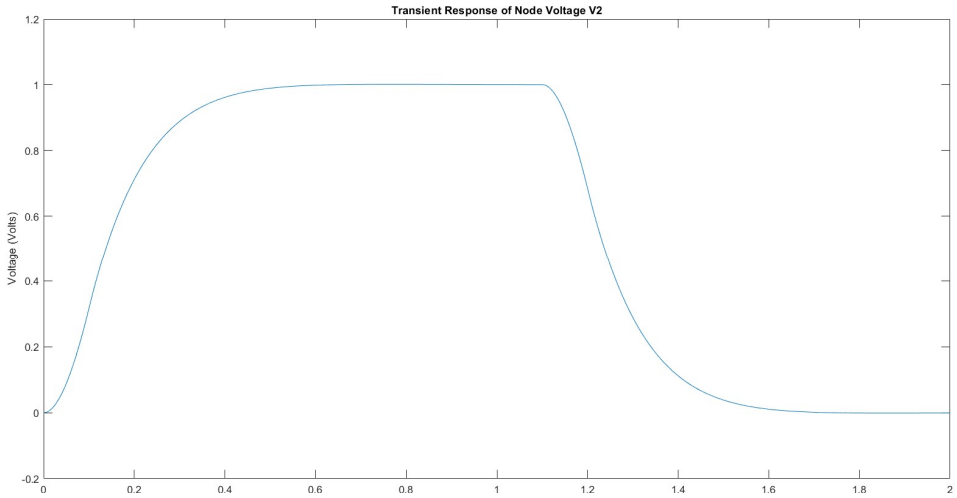


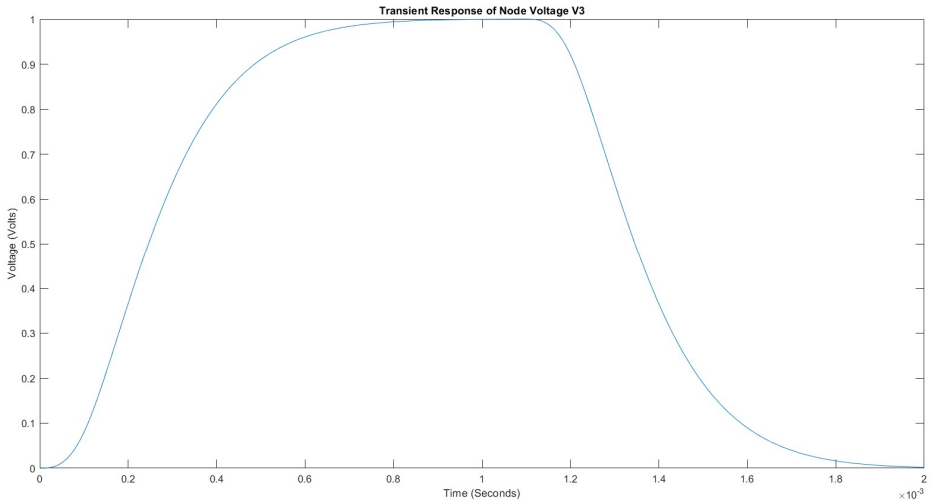


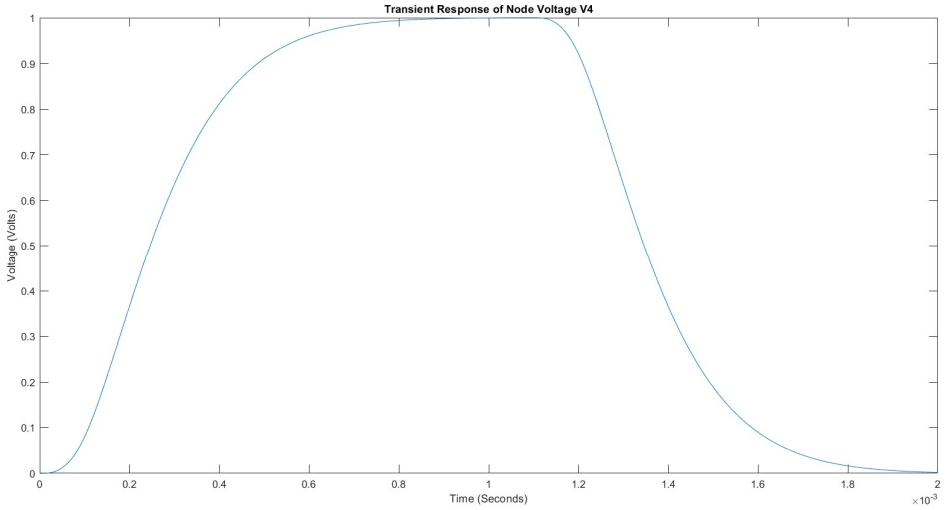


Results : P2test3.sp – Pulse Wave









Results : Comparison to LTSpice

- The results obtained from the written SPICE program are approximately like LTSpice results.
- There is some deviation in the results obtained for P2test2.sp as the step size is very high when compared to other cases.
- The dc operating points obtained are very similar to LTSpice results.
- The sine, pwl, pulse wave generated are also same as the wave generated by the LTSpice.

Conclusions :

- The implemented SPICE program can successfully run the given codes and it can also run the similar types of codes containing R, L, G, C components.
- The code has considered almost all possible cases for the given commands and components. So that any type of code containing these components can be executed by this program.
- As the step time is not adaptive better results can be achieved by take smaller step size.

Improvements that Can be done :

- The submitted program can be improved by adding additional component parameters, taking all possibilities into consideration.
- An adaptive step size can be included for better accuracy of results.