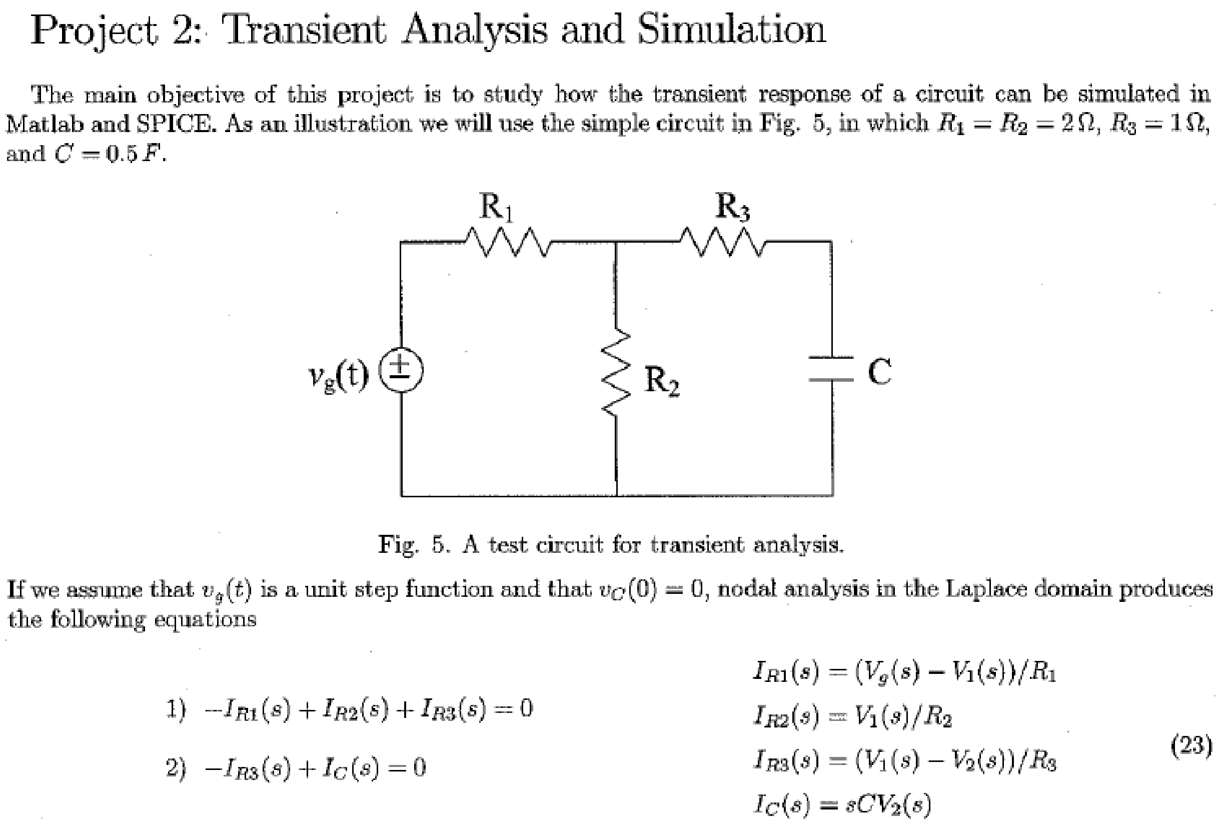
ELEN 100L (Electric Circuits II): Project 2 Tutorial



Project 2: Transient Analysis and Simulation

The main objective of this project is to study how the transient response of a circuit can be simulated in Matlab and SPICE. As an illustration we will use the simple circuit in Fig. 5, in which $R_1 = R_2 = 2\,\Omega$, $R_3 = 1\,\Omega$, and $C = 0.5\,F$.

R1
R3
vg(t)
R2
C

Fig. 5. A test circuit for transient analysis.

If we assume that $v_g(t)$ is a unit step function and that $v_C(0) = 0$, nodal analysis in the Laplace domain produces the following equations

$$\begin{array}{ll} 1)\ \ -I_{R1}(s) + I_{R2}(s) + I_{R3}(s) = 0 & \begin{aligned} I_{R1}(s) &= (V_g(s) - V_1(s))/R_1 \\ I_{R2}(s) &= V_1(s)/R_2 \end{aligned} \\ 2)\ \ -I_{R3}(s) + I_C(s) = 0 & \begin{aligned} I_{R3}(s) &= (V_1(s) - V_2(s))/R_3 \\ I_C(s) &= sCV_2(s) \end{aligned} \end{array} \qquad (23)$$

Hard Copy Deliverables:

1. Hard copy for hand calculations.
2. A MATLAB script and publish the solution using MATLAB's **publish** feature.
3. Turn in MATLAB scripts and a document of the run-time results.
4. Turn in oscilloscope images of the measured results.

Soft Copy Deliverables:

1. Turn in all MATLAB files.
2. Turn in all LTSpice files.
3. Turn in all oscilloscope images of the measured results.

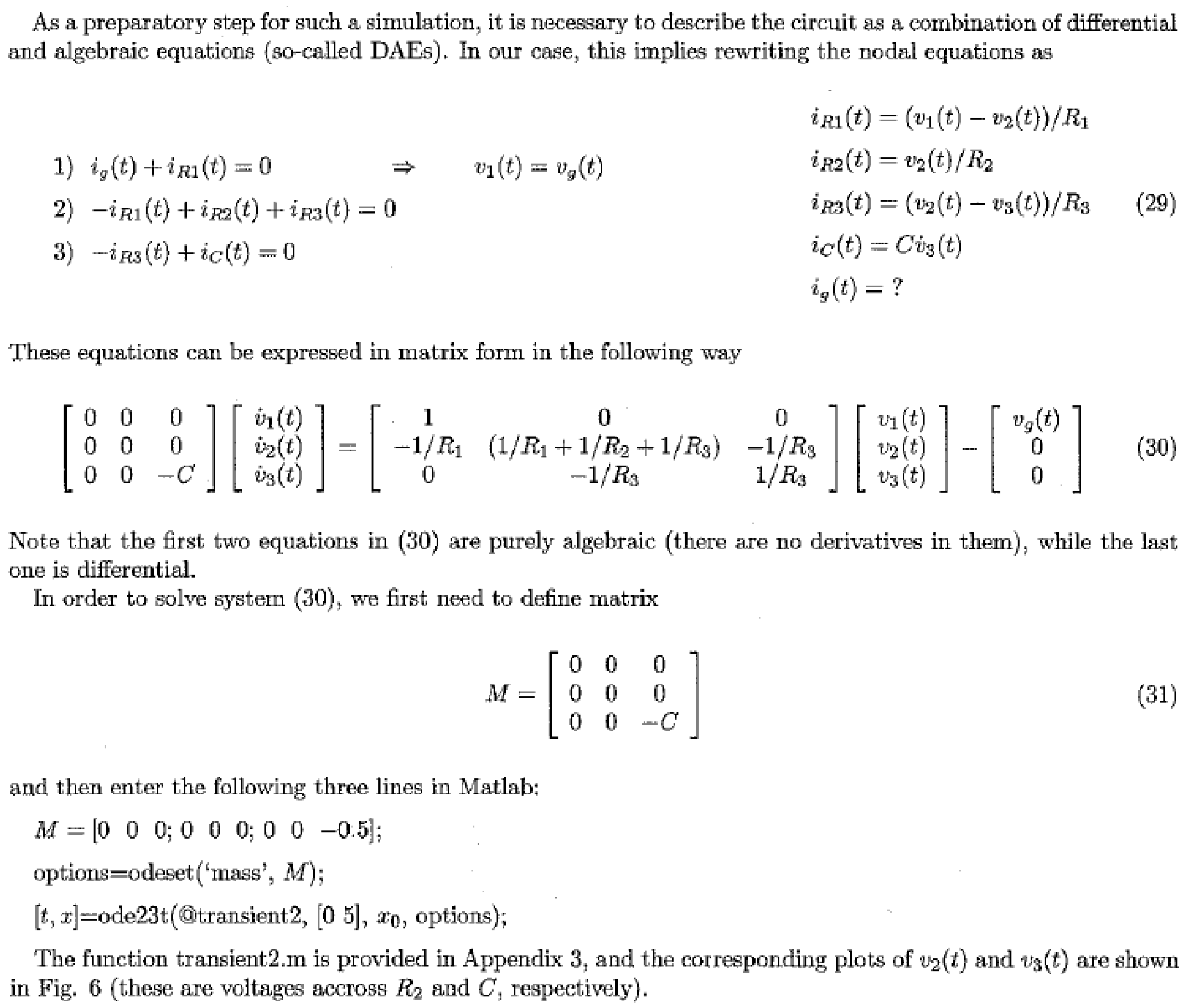
Initialize MATLAB Environment

```
clear; clc; clf; cla; close all;
format long; format compact;
```

Setup global variables

```
% These Ideal Design element values are fixed in the circuit.
VG  = 1;                   % Generator voltage

R1_ideal = 2.0;            % Ohms
R2_ideal = 2.0;            % Ohms
R3_ideal = 1.0;            % Ohms
C1_ideal = 0.5;            % Farads

% Build an array for the R elements.
R_ideal = [R1_ideal, R2_ideal, R3_ideal];

% Build an array for the C elements.
C_ideal = [(0), (0),  (0)        ; ...
           (0), (0),  (0)        ; ...
           (0), (0), -(C1_ideal)];

% Build an array for the source elements.
B = [VG; 0; 0];

% Build an array for the the time vector.
time = [0, 5];

% Build an array for the the initial conditions.
x0 = [0; 0; 0];     % Assume everything is zero to start

% These values are used for plotting purposes.
fignum = 1;
plot_left   = 0;       plot_right = time(2);    % x-axis range (seconds)
plot_bottom = 0;       plot_top   = VG+0.1;     % y-axis range (volts)
```

Problem 2



As a preparatory step for such a simulation, it is necessary to describe the circuit as a combination of differential and algebraic equations (so-called DAEs). In our case, this implies rewriting the nodal equations as

$$\begin{array}{llll} & & & i_{R1}(t) = (v_1(t) - v_2(t))/R_1 \\ 1)\ i_g(t) + i_{R1}(t) = 0 & \Rightarrow & v_1(t) = v_g(t) & i_{R2}(t) = v_2(t)/R_2 \\ 2)\ -i_{R1}(t) + i_{R2}(t) + i_{R3}(t) = 0 & & & i_{R3}(t) = (v_2(t) - v_3(t))/R_3 \\ 3)\ -i_{R3}(t) + i_C(t) = 0 & & & i_C(t) = C\dot{v}_3(t) \\ & & & i_g(t) = ? \end{array} \tag{29}$$

These equations can be expressed in matrix form in the following way

$$\begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -C \end{bmatrix} \begin{bmatrix} \dot{v}_1(t) \\ \dot{v}_2(t) \\ \dot{v}_3(t) \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ -1/R_1 & (1/R_1 + 1/R_2 + 1/R_3) & -1/R_3 \\ 0 & -1/R_3 & 1/R_3 \end{bmatrix} \begin{bmatrix} v_1(t) \\ v_2(t) \\ v_3(t) \end{bmatrix} - \begin{bmatrix} v_g(t) \\ 0 \\ 0 \end{bmatrix} \tag{30}$$

Note that the first two equations in (30) are purely algebraic (there are no derivatives in them), while the last one is differential.

In order to solve system (30), we first need to define matrix

$$M = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -C \end{bmatrix} \tag{31}$$

and then enter the following three lines in Matlab:

$M = [0\ \ 0\ \ 0;\ 0\ \ 0\ \ 0;\ 0\ \ 0\ \ -0.5]$;

options=odeset('mass', M);

$[t, x]$=ode23t(@transient2, [0 5], x_0, options);

The function transient2.m is provided in Appendix 3, and the corresponding plots of $v_2(t)$ and $v_3(t)$ are shown in Fig. 6 (these are voltages accross R_2 and C, respectively).

```
fignum = fignum+1;
```

Display the component values for the Ideal design.

```
display(' ');
display('The Ideal Design component values are:');
fprintf('    R1 = %+11.4f Ohms.\n',   R1_ideal);
fprintf('    R2 = %+11.4f Ohms.\n',   R2_ideal);
fprintf('    R3 = %+11.4f Ohms.\n',   R3_ideal);
fprintf('    C1 = %+11.4e Farads.\n', C1_ideal);
```

```
The Ideal Design component values are:
    R1 =     +2.0000 Ohms.
    R2 =     +2.0000 Ohms.
    R3 =      +1.0000 Ohms.
    C1 = +5.0000e-01 Farads.
```

Calculate the transient response for the Ideal design.

```
% Update the resistor variables used in the proj2E100_transient function
% before calling the ode23t solver.
R1_circuit = R1_ideal;
R2_circuit = R2_ideal;
R3_circuit = R3_ideal;

options = odeset('mass', C_ideal, 'RelTol', 0.1e-9);
[t, x] = ode23t(@proj2E100_transient, time, x0, options);

% Capture a voltage point with index.
v3_point_of_interest = 400e-3; % This particular value is chosen only for
                               % plot annotation illustration purposes.

v3_shifted_mag = abs( x(:,3) - v3_point_of_interest );
v3_point_of_interest_index = ...
    find( min( v3_shifted_mag ) == v3_shifted_mag );
v3_point_of_interest = x(v3_point_of_interest_index, 3);

% Capture the time stamp at voltage point of interest.
t_point_of_interest  = t(v3_point_of_interest_index);
```

Generate the plot for the transient response.

```
fignum = 5;  % Bump the figure number to match tutorial documentation.

fignum = fignum+1; figObj = figure(fignum);  % Establish a figure number
set(fignum, 'Name', ...
    ['Transient Response Ideal Design']);    % Name the figure

Tr_ideal_Plot = plot(t, x);                  % Generate plot
grid on;                                     % Turn grid on
xlabel('Time (seconds)');                    % Label the x-axis
ylabel('Amplitude (volts)');                 % Label the y-axis
axis([plot_left, plot_right, ...
      plot_bottom, plot_top]);               % Bound plot
title(['Figure ',num2str(fignum,'%-2.u'),...
       ': Transient Response']);
legend('v_1(t)', 'v_2(t)', 'v_3(t)', 'Location', 'NorthEast');

% Add annotation to the plot.
str_curs = ['\leftarrow ', num2str(t_point_of_interest), 's, ', ...
            num2str(v3_point_of_interest),'V'];
text(t_point_of_interest, ...
     v3_point_of_interest, ...
     str_curs, 'HorizontalAlignment', 'left', 'FontWeight', 'bold');
```

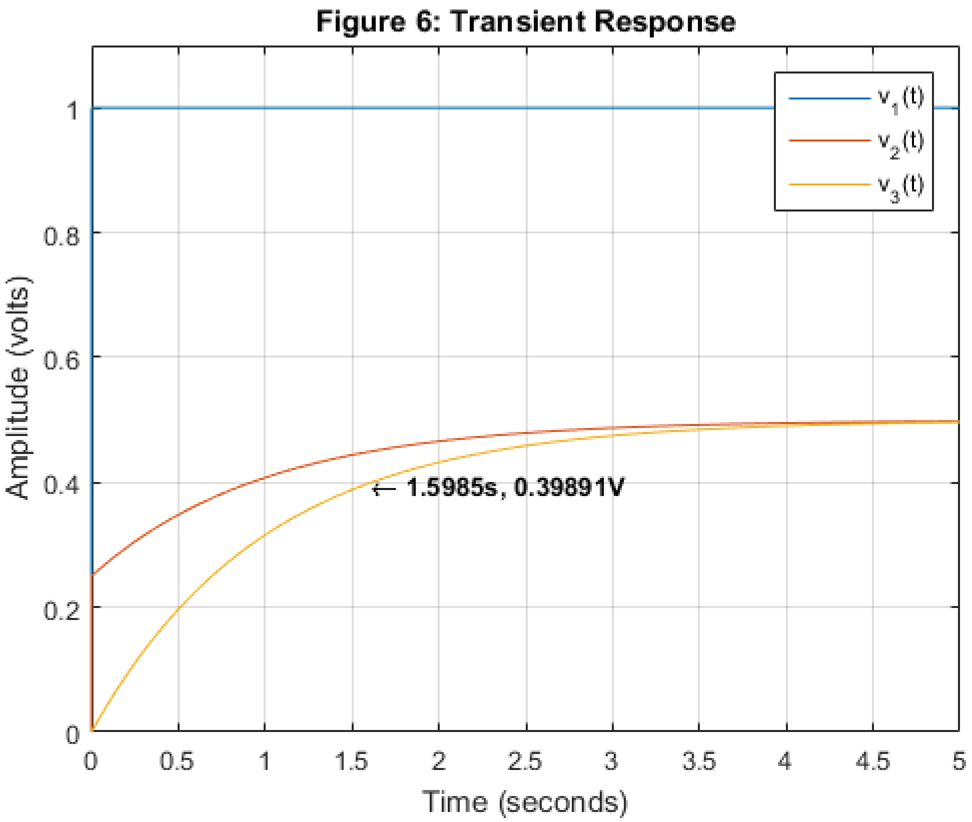


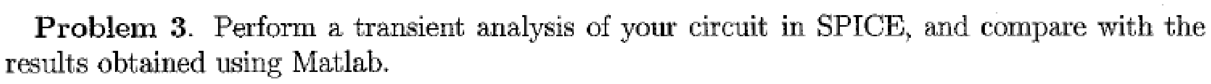
Figure 6: Transient Response

Display the MATLAB point of interest values for the Ideal design.

```
display(' ');
display('The MATLAB point of interest values are:');
fprintf('    V3 p.o.i. = %+11.4f Volts.\n',   v3_point_of_interest);
fprintf('     t p.o.i. = %+11.4f seconds.\n', t_point_of_interest);
```

```
The MATLAB point of interest values are:
    V3 p.o.i. =     +0.3989 Volts.
     t p.o.i. =     +1.5985 seconds.
```

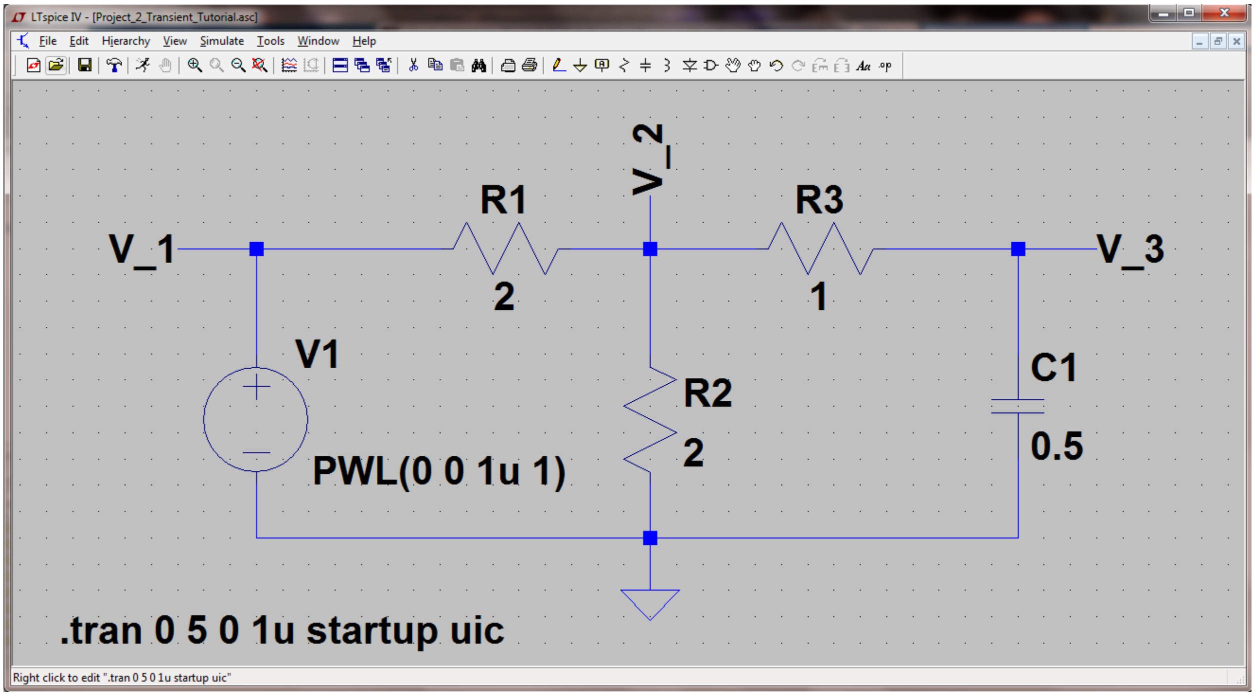
Problem 3



Problem 3. Perform a transient analysis of your circuit in SPICE, and compare with the results obtained using Matlab.

```
fignum = fignum+1;
```

The LTSpice model for the circuit is shown below.



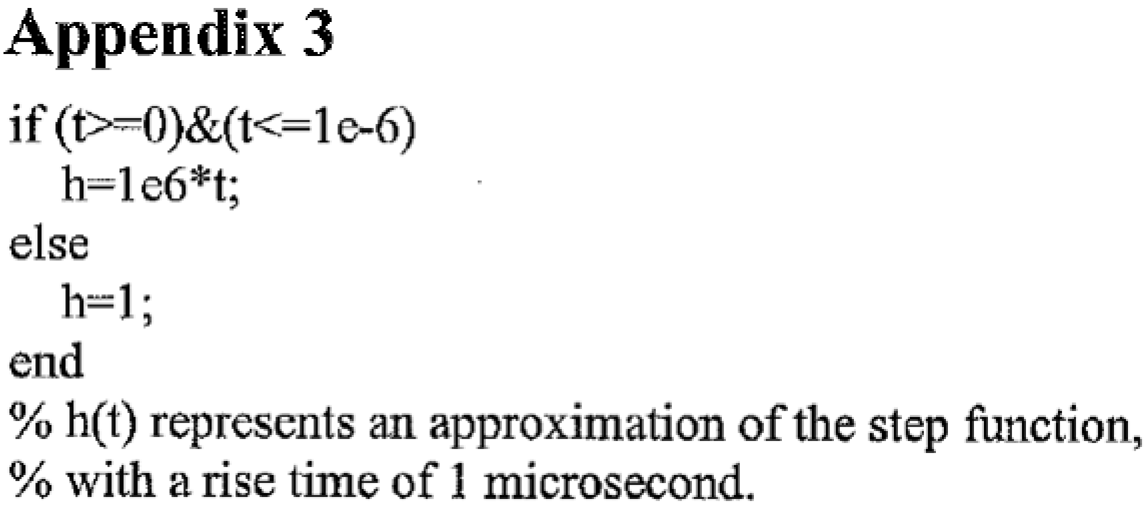
V1 PWL(0 0 1u 1)

.tran 0 5 0 1u startup uic

Right click to edit ".tran 0 5 0 1u startup uic"

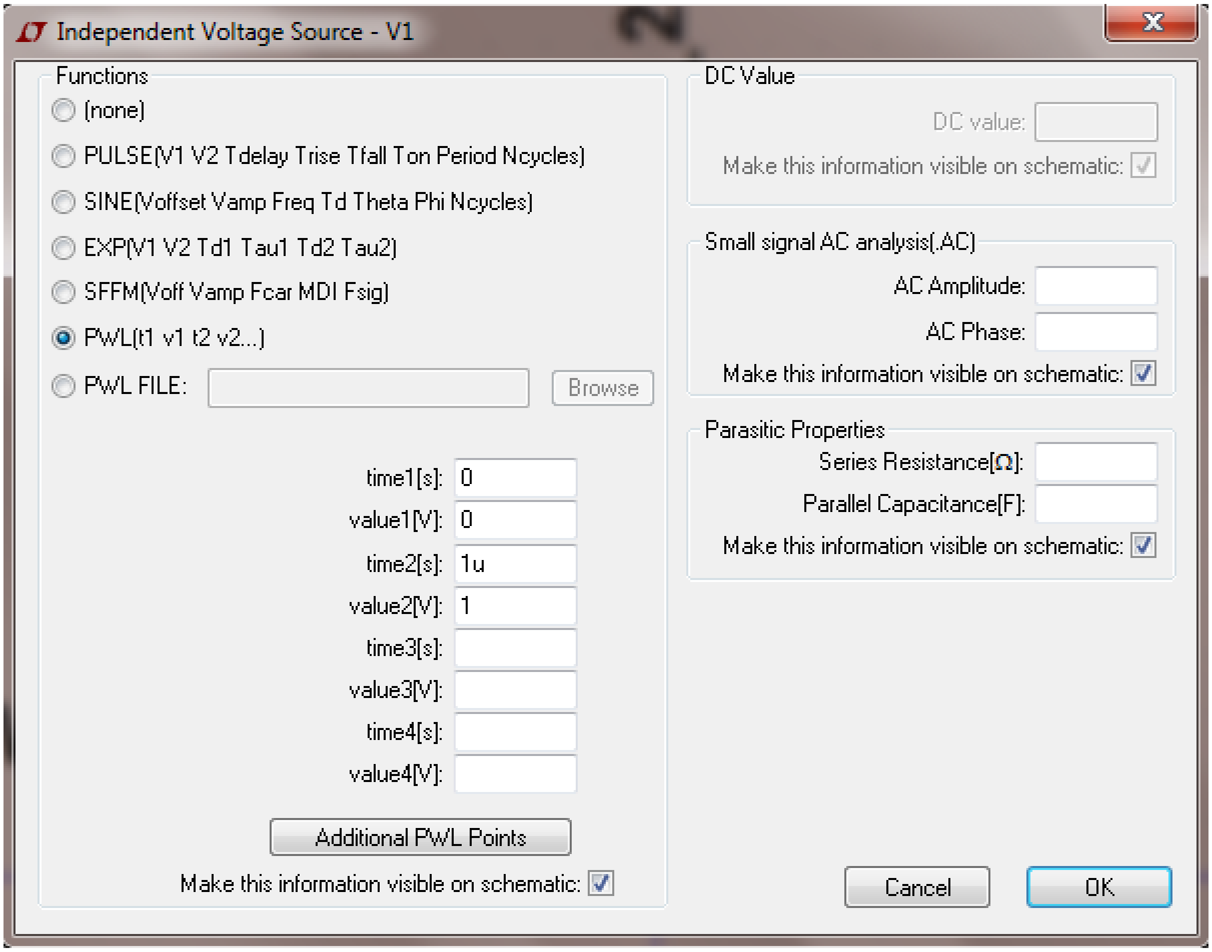
```
fignum = fignum+1;
```

The LTSpice model voltage source setup, to correlate with the MATLAB unit step function definition in Appendix #3, is shown below.



Appendix 3

```
if (t>=0)&(t<=1e-6)
    h=1e6*t;
else
    h=1;
end
% h(t) represents an approximation of the step function,
% with a rise time of 1 microsecond.
```



Independent Voltage Source - V1

Functions

○ (none)

○ PULSE(V1 V2 Tdelay Trise Tfall Ton Period Ncycles)

○ SINE(Voffset Vamp Freq Td Theta Phi Ncycles)

○ EXP(V1 V2 Td1 Tau1 Td2 Tau2)

○ SFFM(Voff Vamp Fcar MDI Fsig)

● PWL(t1 v1 t2 v2...)

○ PWL FILE:

time1[s]: 0

value1[V]: 0

time2[s]: 1u

value2[V]: 1

time3[s]:

value3[V]:

time4[s]:

value4[V]:

Make this information visible on schematic: ☑

DC Value

DC value:

Make this information visible on schematic: ☑

Small signal AC analysis(.AC)

AC Amplitude:

AC Phase:

Make this information visible on schematic: ☑

Parasitic Properties

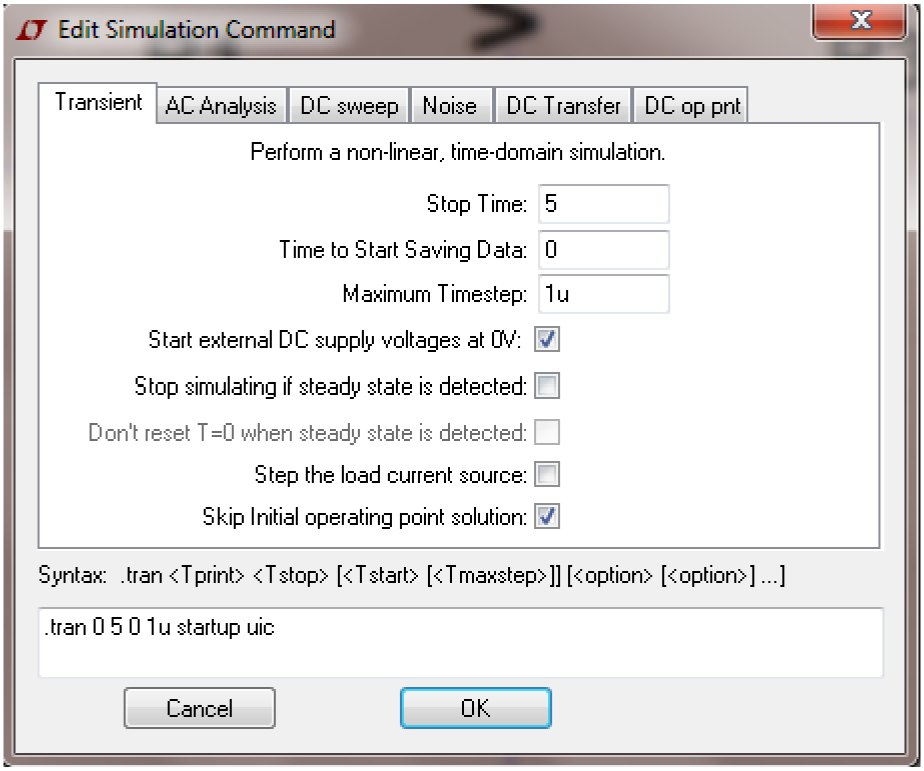
Series Resistance[Ω]:

Parallel Capacitance[F]:

Make this information visible on schematic: ☑

```
fignum = fignum+2;
```

The LTSpice model transient analysis simulation setup is shown below.



Edit Simulation Command

Transient | AC Analysis | DC sweep | Noise | DC Transfer | DC op pnt

Perform a non-linear, time-domain simulation.

Stop Time: 5

Time to Start Saving Data: 0

Maximum Timestep: 1u

Start external DC supply voltages at 0V: ☑

Stop simulating if steady state is detected: □

Don't reset T=0 when steady state is detected: □

Step the load current source: □

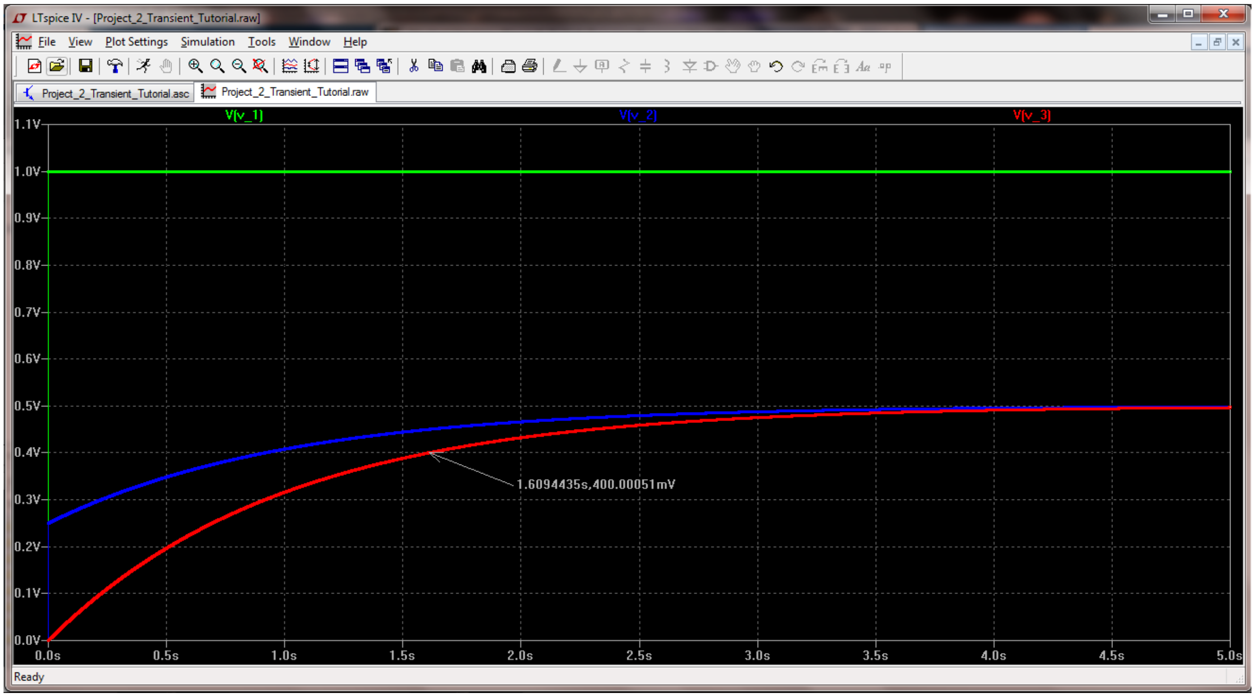
Skip Initial operating point solution: ☑

Syntax: .tran <Tprint> <Tstop> [<Tstart> [<Tmaxstep>]] [<option> [<option>] ...]

```
.tran 0 5 0 1u startup uic
```

```
fignum = fignum+1;
```

The LTSpice model for the simulation result is shown below.



```
fignum = fignum+1;

% Capture point of interest voltage from the plot.
ltspice_v3_point_of_interest = 400.00051e-3;

% Capture point of interest time stamp from the plot.
ltspice_t_point_of_interest  = 1.6094435;
```

Display the LTSpice voltage point of interest values for the Ideal design.

```
display(' ');
display('The LTSpice point of interest values are:');
fprintf('    V3 p.o.i. = %+11.4f Volts.\n', ...
    ltspice_v3_point_of_interest);
fprintf('     t p.o.i. = %+11.4f seconds.\n', ...
    ltspice_t_point_of_interest);
```

```
The LTSpice point of interest values are:
    V3 p.o.i. =     +0.4000 Volts.
     t p.o.i. =     +1.6094 seconds.
```

Calculate the percent difference at the point of interest values between MATLAB and LTSpice Ideal Designs.

```
diff_ideal_v3_point_of_interest  = ...
    (ltspice_v3_point_of_interest - v3_point_of_interest) ...
    /abs(v3_point_of_interest)*100;

diff_ideal_t_point_of_interest  = ...
    (ltspice_t_point_of_interest - t_point_of_interest) ...
    /abs(t_point_of_interest)*100;

display(' ');
display('The % difference between MATLAB and LTSpice at');
display('the point of interest:');
fprintf('    MATLAB  V3 p.o.i. = %+11.4f Volts.\n', ...
         v3_point_of_interest);
fprintf('    LTSpice V3 p.o.i. = %+11.4f Volts.\n', ...
         ltspice_v3_point_of_interest);
fprintf('        %% diff = %+8.4f (%%).\n', ...
         diff_ideal_v3_point_of_interest);

fprintf('    MATLAB   t p.o.i. = %+11.4f seconds.\n', ...
         t_point_of_interest);
fprintf('    LTSpice  t p.o.i. = %+11.4f seconds.\n', ...
         ltspice_t_point_of_interest);
fprintf('        %% diff = %+8.4f (%%).\n', ...
         diff_ideal_t_point_of_interest);
```

```
The % difference between MATLAB and LTSpice at
the point of interest:
    MATLAB  V3 p.o.i. =     +0.3989 Volts.
    LTSpice V3 p.o.i. =     +0.4000 Volts.
        % diff =  +0.2746 (%).
    MATLAB   t p.o.i. =     +1.5985 seconds.
    LTSpice  t p.o.i. =     +1.6094 seconds.
        % diff =  +0.6864 (%).
```

Program execution complete

```
display(' ');
disp('Program execution complete....');
```

```
Program execution complete....
```

MATLAB code listing

Published with MATLAB® R2015b