ECE 4960: Computational and Software Engineering

Programming Assignment 4: SPICE for ODE Solution

Submitted By: Vishisht M. Tiwari(vmt28) Deepak Agarwal(da475)

Programming Environment

- Language: C++
- IDE: NetBeans 8.2
- OS: MacOS High Sierra

Goal: The goal of the assignment is to implement ODE solvers using transient simulation of circuits and validate the results

Code Structure:
The parent directory contains following files:

- Code Files:

1. Assignment_4_Header.h: It is a global header file included in all code files, containing globally defined functions and other classes declataions.
   Class Global_Functions: Global class defined in the header file containing functions to create dynamically allocated matrix from an array, create dynamic arrays, create full matrix from sparse etc. Exception handling check is added in all functions under a macro EXCEPTION_HANDLING
2. Assignment_4_Main.cpp: Main file performing all the required tasks by calling respective class methods
3. ODE_Solvers.cpp: Code file containing all ODE solvers – Forward Euler, Runge Kutta 34, and Time Adaptive Runge Kutta
4. Circuit_Sim_1.cpp and Circuit_Sim_2.cpp: Code file to implement the electrical circuit equation and solve it by calling solvers in ODE_Solvers.cpp
5. Validation.cpp: Code file to implement validation equation and it calls ODE solvers

- Report documents:

1. Report.pdf and Report.docx: Documentation of code design and testing strategies
2. Readme.txt: File describing code structure, compilation command and testing platforms
3. Output_MacOS.txt: Log output on MacOS
4. Results: A folder containing graph of all solvers and their validation result

Usage of previous assignments:
This assignment required matrix-vector manipulation as part of ODE solver algorithm, along with other error-detection techniques. Various methods were implemented for matrix-vector multiplication in a globally defined class Global_Functions() as part of Programming Assignment

2 and 3. These functions were coded as generically as possible, which eventually helped us in this assignment as we could implemented the same class functions.

Also various row operations like row scaling, row permutation etc were leveraged from programming assignment 2. Apart from leveraging the existing code, many new functions were added to make the library richer like row shifting, row exponential, column shifting, column scaling etc.

Finally keeping in mind the importance of regular error-detection and out-of-bond values, most of the algorithms or functions were added with an exception handler code. The code was put under the macro EXCEPTION_HANDLING which was again leveraged from the previous assignments.

Tasks:

1. **ODE Validation**: Various ODE solvers were implemented: Forward Euler, Runge Kutta 34 and Time Adaptive Runge Kutta were implemented later in the assignment. But before that, validation was performed by checking solution accuracy of this ODE:

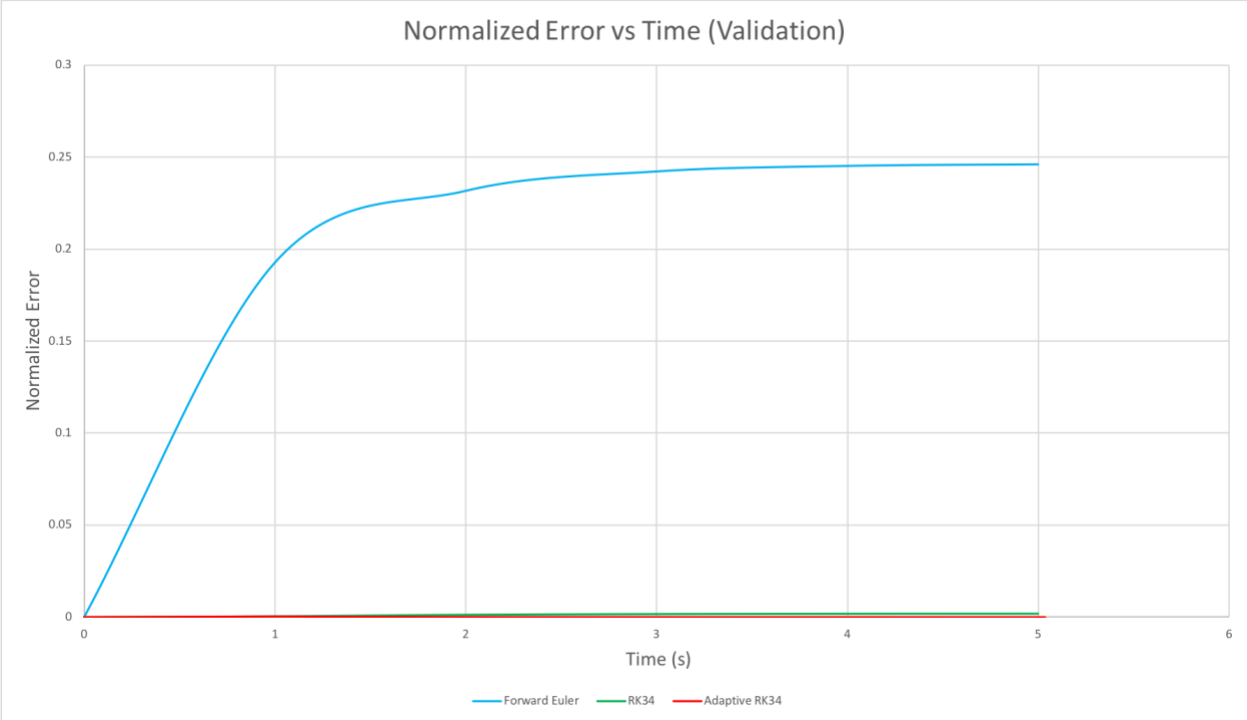
$$\frac{dx}{dt} = 4e^{0.8t} - 0.5x$$

The true solution for this was known already as:

$$x(t) = \frac{4}{1.3}\left(e^{0.8t} - e^{-0.5t}\right) + 2e^{-0.5t}$$

.

where, the initial solution is x(0) = 2



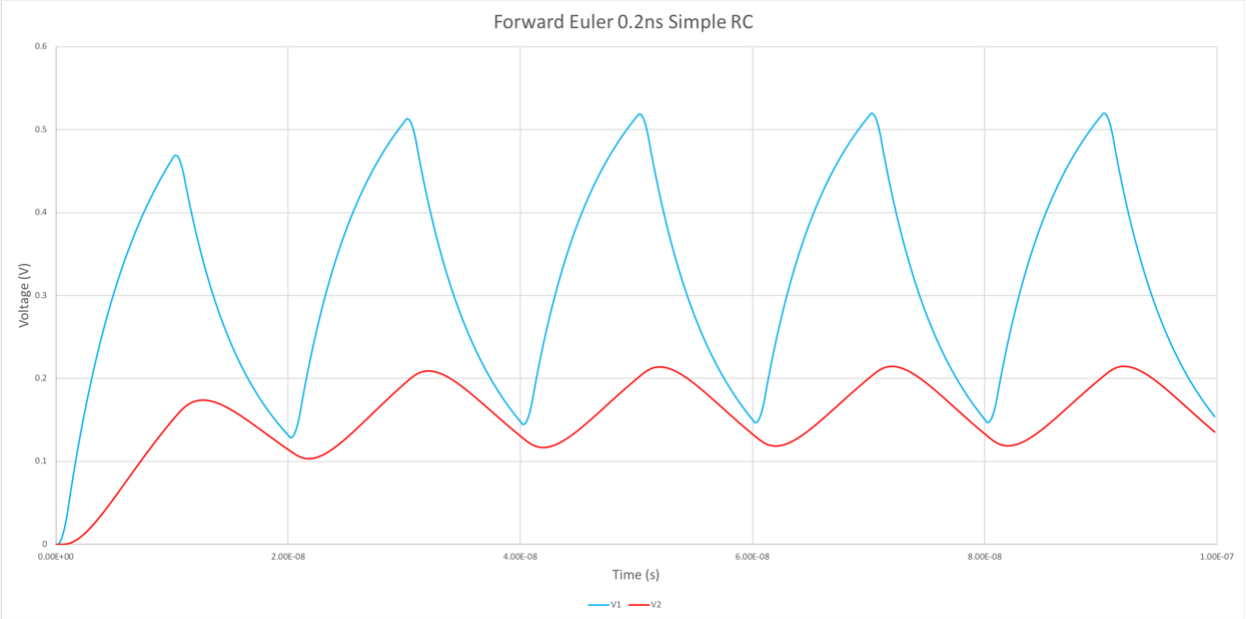
Observation: As shown in the above plot, Forward Euler has maximum normalized error which is improved significantly with RK34 and Adaptive RK34 solver. Also it can be concluded that adaptive RK34 solver performed the best among the three solvers.

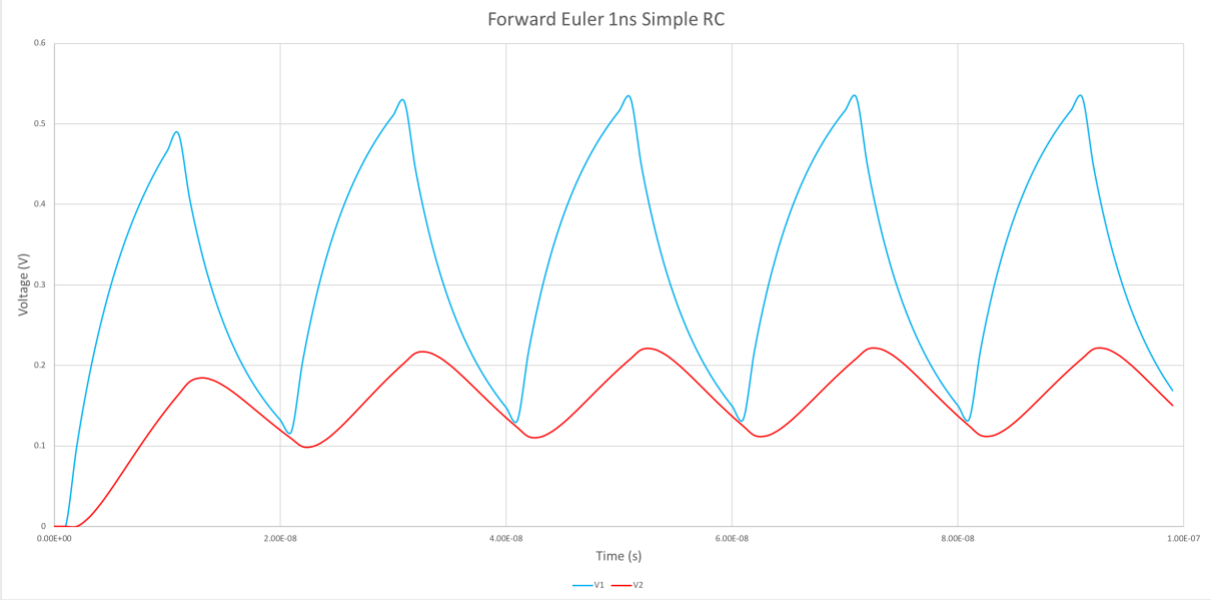
2. Circuit 1: Simple RC Circuit

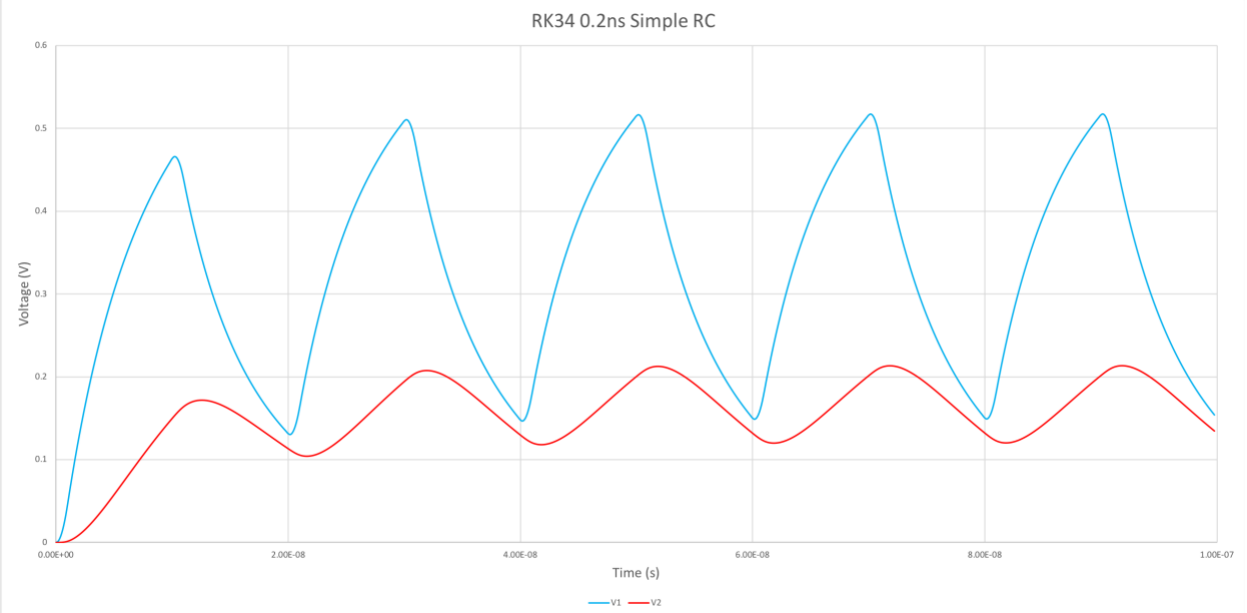
The next task was to solve the circuit shown in Fig.3 of assignment handout for the following circuit equation in the form of $\frac{d\vec{x}}{dt} = \vec{f}(t, \vec{x})$:

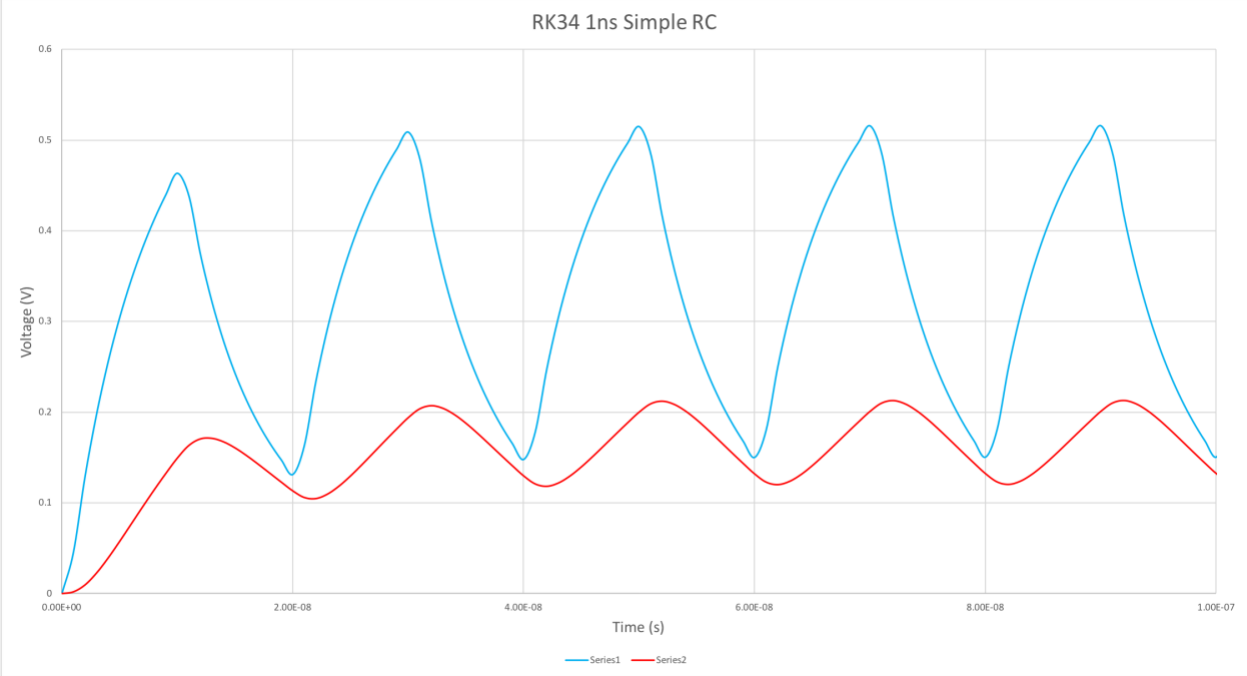
$$\vec{x} = \begin{pmatrix} V_1 \\ V_2 \end{pmatrix}; \qquad \frac{d}{dt}\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} = \begin{pmatrix} -\left(\frac{1}{C_1R_1} + \frac{1}{C_1R_2}\right) & \frac{1}{C_1R_2} \\ \frac{1}{C_2R_2} & -\left(\frac{1}{C_2R_2} + \frac{1}{C_2R_3}\right) \end{pmatrix}\begin{pmatrix} V_1 \\ V_2 \end{pmatrix} + \begin{pmatrix} \frac{i(t)}{C_1} \\ 0 \end{pmatrix}$$

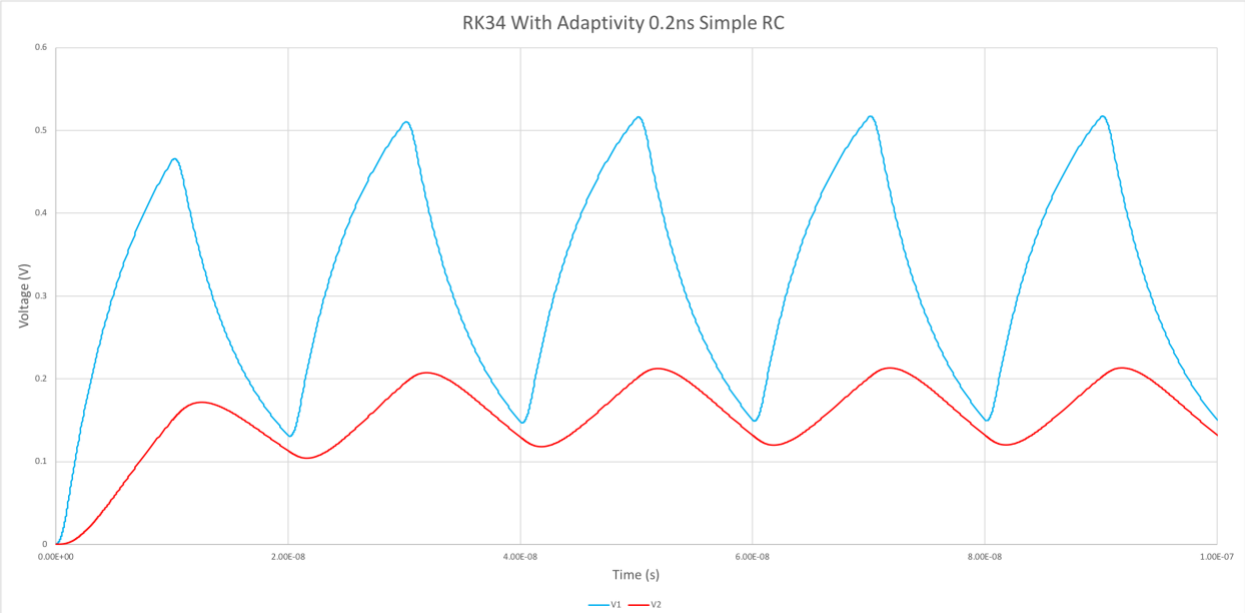
Following up are the results obtained by different ODE solvers:

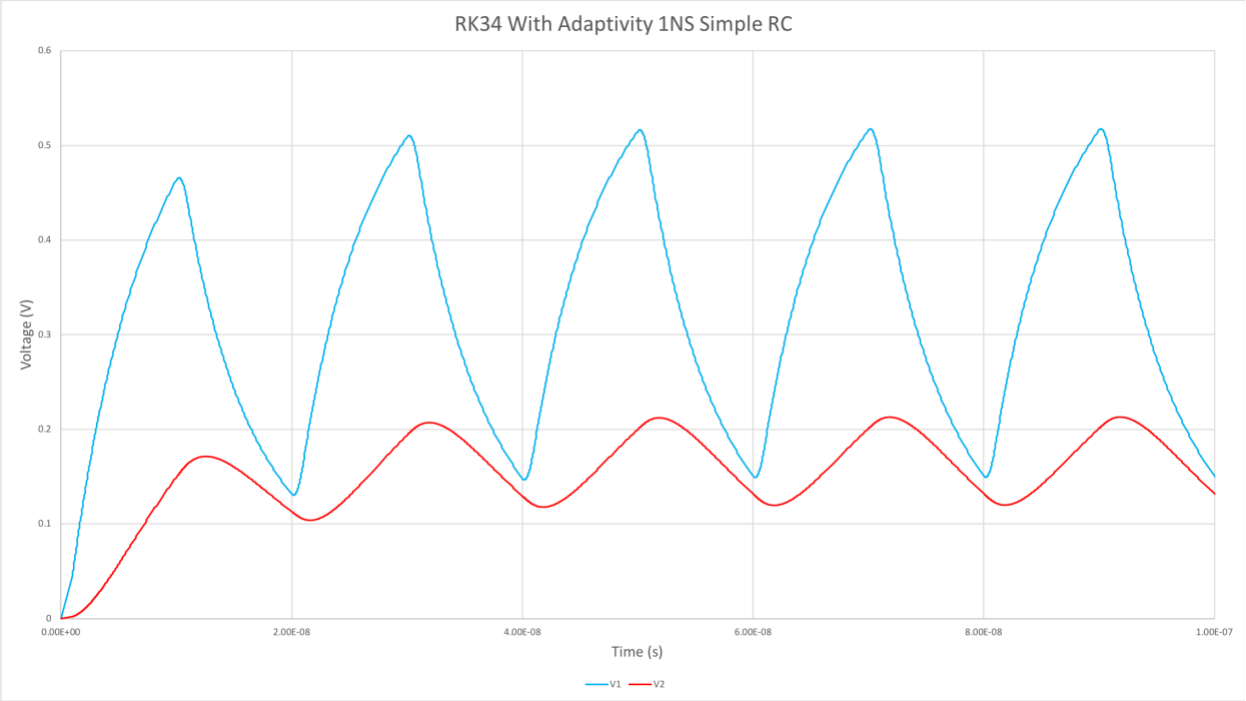

Forward Euler 0.2ns Simple RC
Voltage (V)
0.6
0.5
0.4
0.3
0.2
0.1
0
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
Time (s)
V1
V2


Forward Euler 1ns Simple RC
Voltage (V)
0.6
0.5
0.4
0.3
0.2
0.1
0
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
Time (s)
V1
V2


RK34 0.2ns Simple RC
Voltage (V)
0.6
0.5
0.4
0.3
0.2
0.1
0
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
Time (s)
v1
v2


RK34 1ns Simple RC
Voltage (V)
0.6
0.5
0.4
0.3
0.2
0.1
0
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
Time (s)
Series1
Series2


RK34 With Adaptivity 0.2ns Simple RC
Voltage (V)
0.6
0.5
0.4
0.3
0.2
0.1
0
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
Time (s)
v1
v2

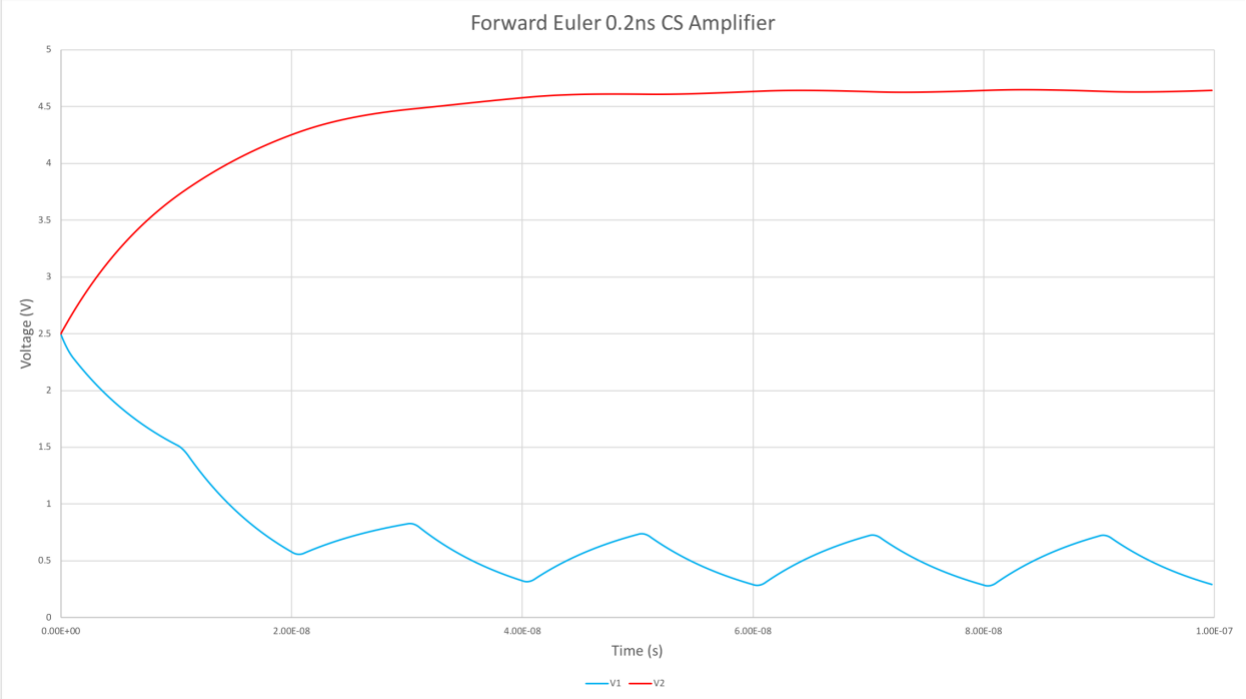

RK34 With Adaptivity 1NS Simple RC
Voltage (V)
0.6
0.5
0.4
0.3
0.2
0.1
0
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
Time (s)
v1
v2

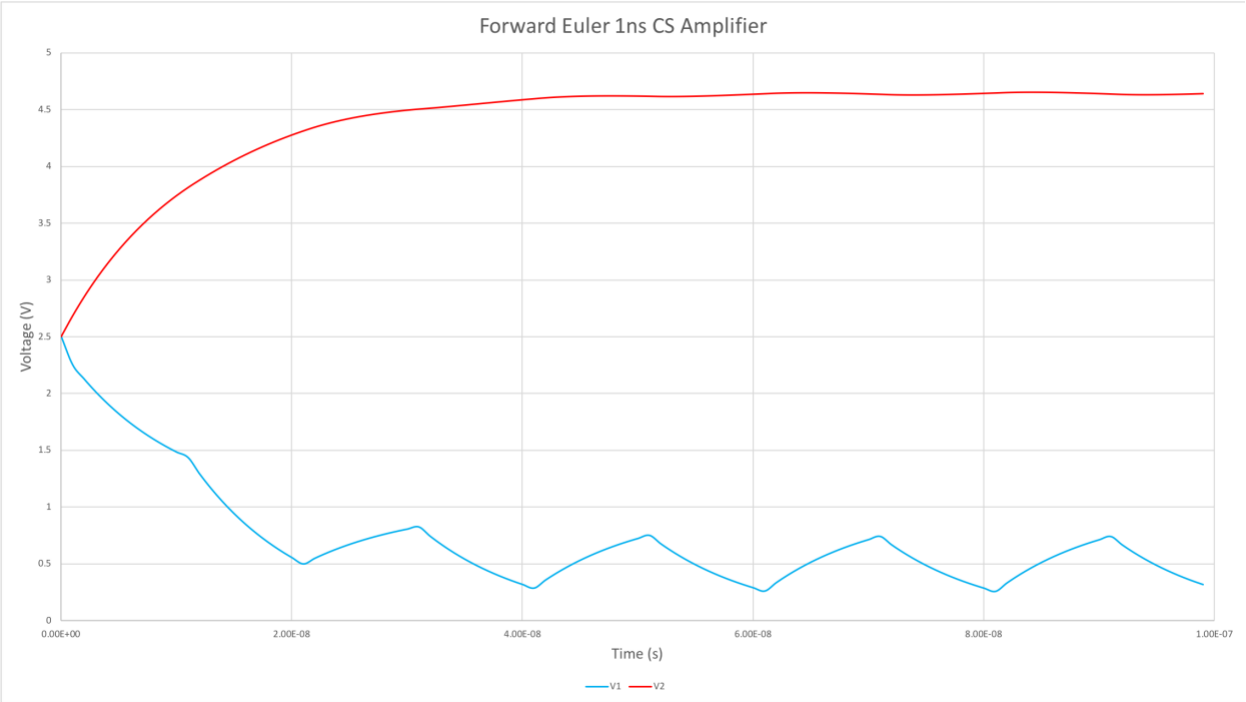
3. **Circuit 3: CS Amplifier:** After solving simple RC circuit, ODE solvers were used to solve circuit equation of common-source CS amplifier. The ODE equation is following:

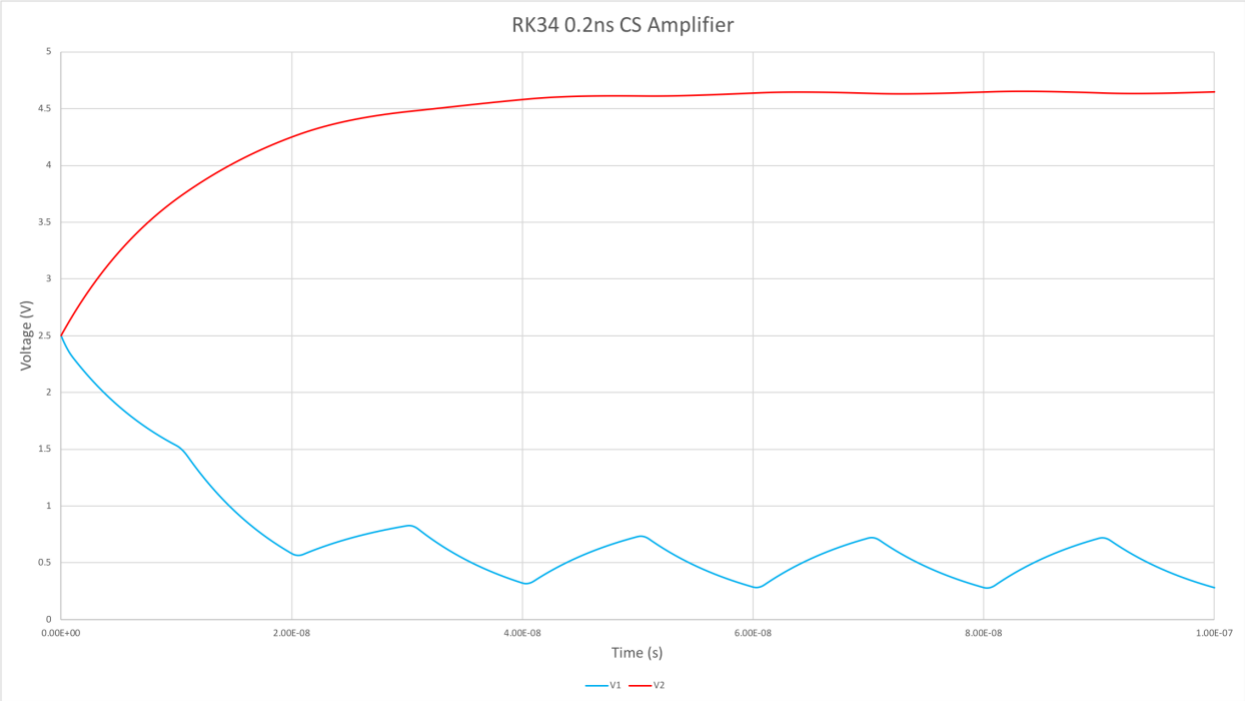
$$\begin{aligned} a_{11}\frac{dV_1}{dt} + a_{12}\frac{dV_2}{dt} &= g_1(V_1, V_2) \\ a_{21}\frac{dV_1}{dt} + a_{22}\frac{dV_2}{dt} &= g_2(V_1, V_2) \end{aligned} \quad or \quad \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} \frac{dV_1}{dt} \\ \frac{dV_2}{dt} \end{bmatrix} = \begin{bmatrix} g_1 \\ g_2 \end{bmatrix},$$

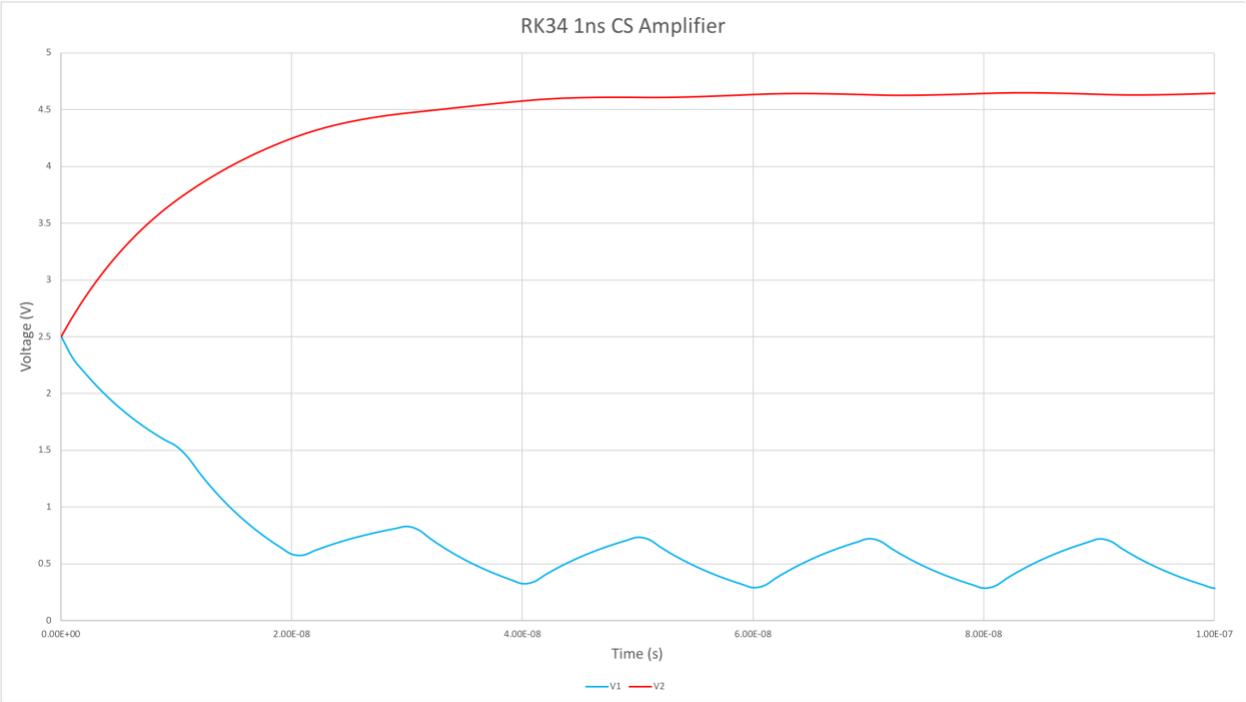
$$\text{where } \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} = \begin{bmatrix} C_1 + C_{GD} & -C_{GD} \\ -C_{GD} & C_2 + C_{GD} \end{bmatrix}$$

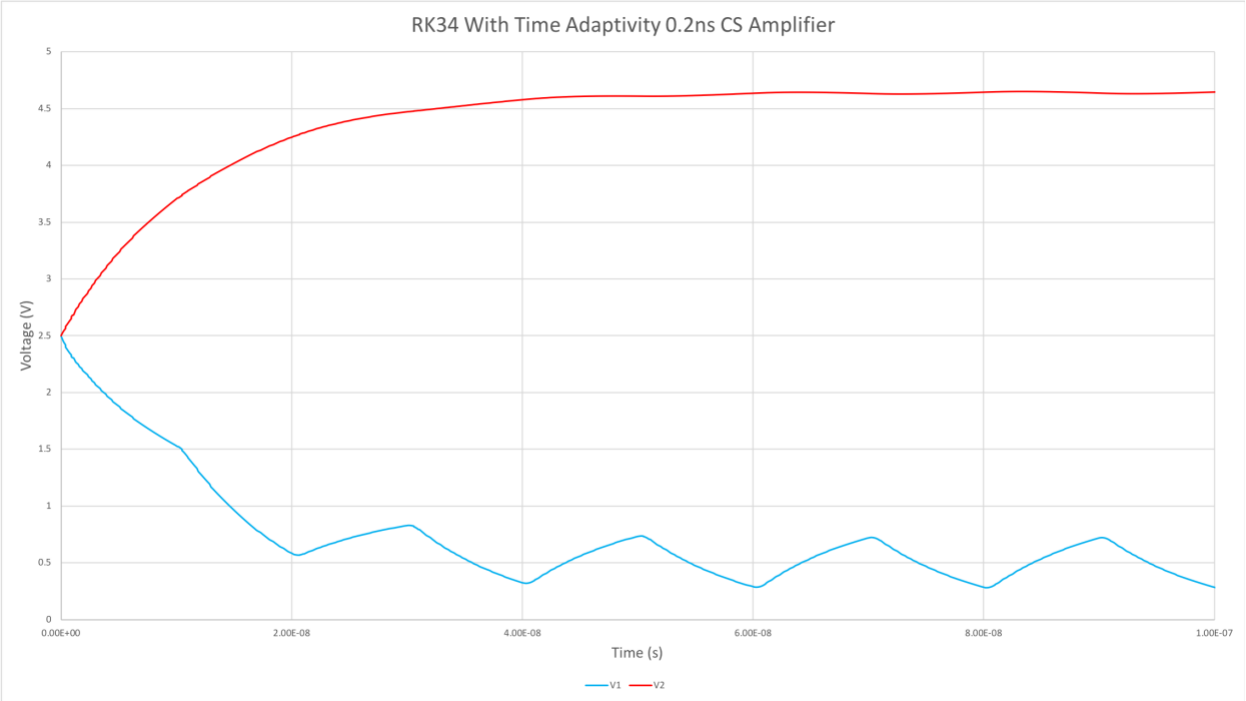
Following are the results observed with various ODE solvers

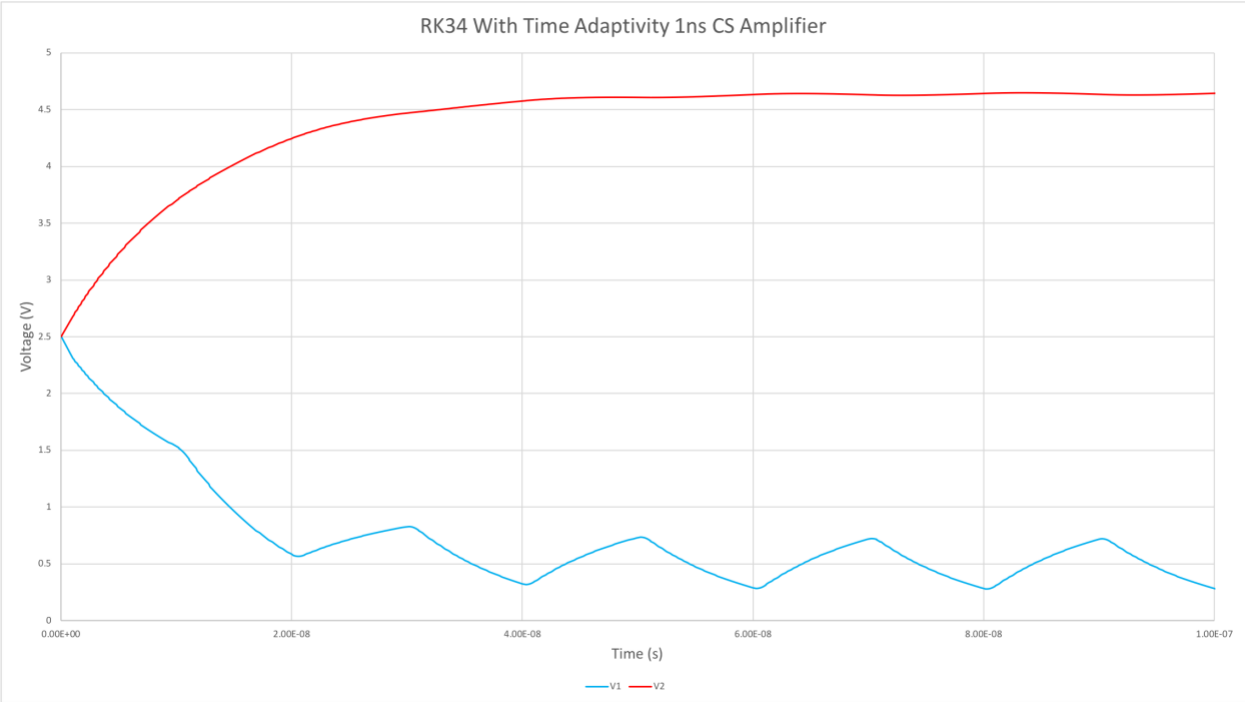

Forward Euler 0.2ns CS Amplifier
Voltage (V)
Time (s)
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
V1
V2


Forward Euler 1ns CS Amplifier
Voltage (V)
Time (s)
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
V1
V2


RK34 0.2ns CS Amplifier
Voltage (V)
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
Time (s)
V1
V2


RK34 1ns CS Amplifier
Voltage (V)
5
4.5
4
3.5
3
2.5
2
1.5
1
0.5
0
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
Time (s)
V1
V2


RK34 With Time Adaptivity 0.2ns CS Amplifier
Voltage (V)
Time (s)
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
V1
V2


RK34 With Time Adaptivity 1ns CS Amplifier
Voltage (V)
Time (s)
0.00E+00
2.00E-08
4.00E-08
6.00E-08
8.00E-08
1.00E-07
V1
V2