ENGR-050-201-104048 Final Project

Visualizing the Comparison between Numeric and Analytical Calculation

Submitted by

Julian Zhuang (Chun Wei Zhuang)

Thanks to and Instructed by

Dr. Abdie Tabrizi

Objective

This program is intended to acquire the output voltage of an amplifier circuit through numeric calculation with the known differential equation and the initial values below.

$$(d^2V)/dt^2 + (2dV)/dt + 5V = 5Vs = 50$$

$$V0 = 0$$

$$V1 = 0$$

Yet, to examine the accuracy, the output voltages via numeric approach are compared to the ones via the analytical equaiton below.

$$V = 10 - e^{-t}[10\cos(2t) + 5\sin(2t)]$$

Methodology

1. Acquire the voltage in the next second, that is $V(t + dt)$, from the given differential equation.

   $$(d^2V)/dt^2 + (2dV)/dt + 5V = 5Vs = 50$$

   $$(V(t + dt) - 2V(t) + V(t - dt))/dt^2 + (V(t + dt) - V(t))/dt + 5V = 50$$

$$V(t + dt) = ((1/dt)^2 + 2/dt)^{-1}(50 - V(t - dt)/(dt)^2 + V(t)(-5 + 2/(dt)^2 + 2/(dt))$$

2. Subtitue $V0 = 0$ and $V1 = 0$ into $V(t - dt)$ and $V(t)$ respectively, V3 is gathered. Then substitute $V1$ and $V2$ into $V(t - dt)$ and $V(t)$ respectively, V4 is gathered. Repeat this process with a loop, we can collect all output voltages.

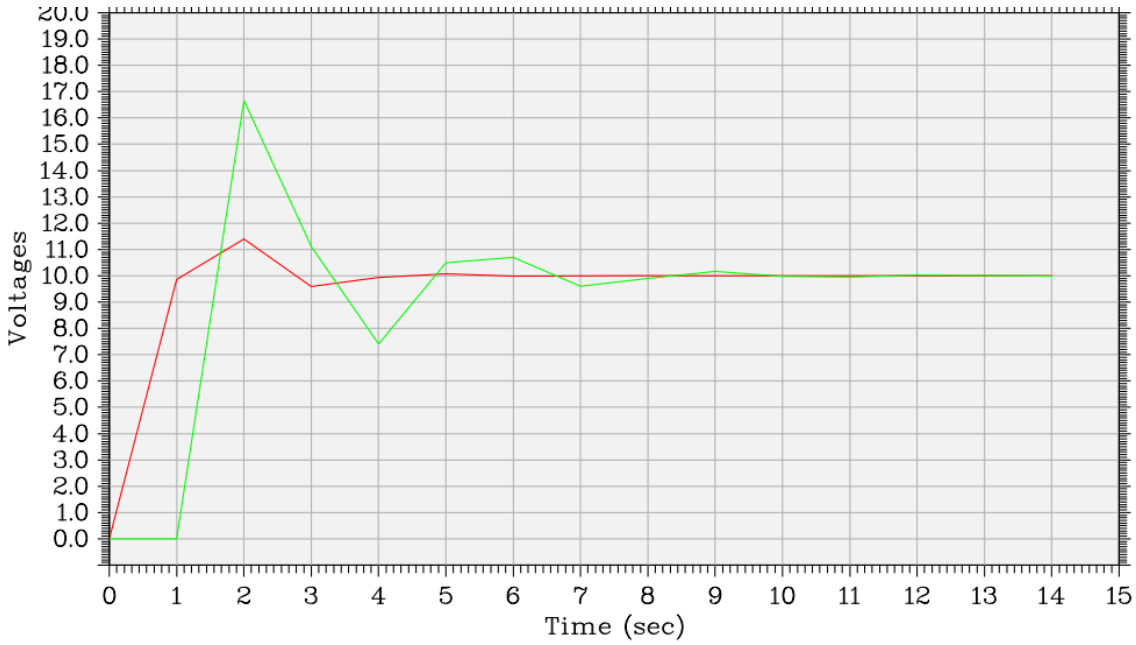
```
for(int i=2; i<n; i++)
{
    v[i] = v_numeric(t[i], v_old, v_new);
    v_old = v_new;
    v_new = v[i];
}
```

Output Result

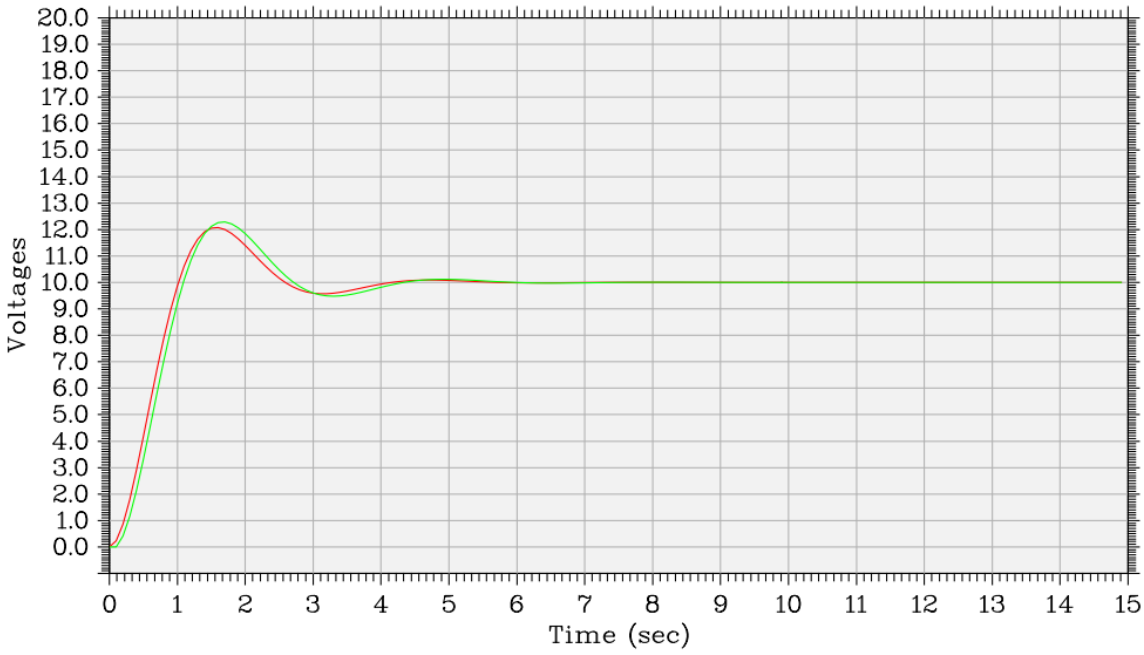
Based on the plots below, as the dt gets smaller, the green curve (numeric) not only looks smoother, but also gets closer to the red one (analytical), which indicates the improvement of the accuracy. The table of output data also reinfrome this trend.

Notice that even though the analytical equation is supposed to provide the exact output voltage, the red curve (analytical) changes over the adjustment of dt as the green one does. The reason is that as dt gets smaller, more points are calculated and connected with the plotting function.

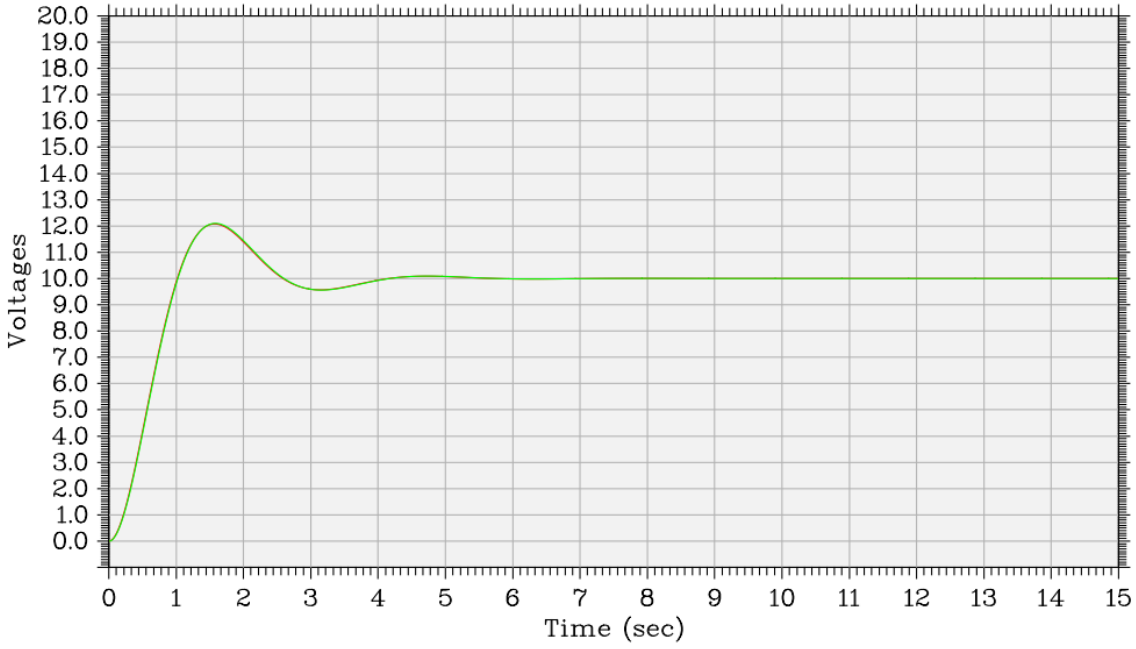
$dt = 1$



$dt = 0.1$



$dt = 0.01$



$dt = 1$

t_max_analy: 11.396721 v_max_analy: 11.396721

t_min_analy: 0.000000 v_min_analy: 0.000000

t_max_num: 1.000000 v_max_num: 16.666666

t_min_num: 0.000000 v_min_num: 0.000000

t	v_analy	v_num
0.000000	0.000000	0.000000
1.000000	9.858359	0.000000
2.000000	11.396721	16.666666
3.000000	9.591516	11.111112
4.000000	9.936046	7.407407
5.000000	10.074864	10.493827
6.000000	9.985733	10.699589
7.000000	9.994237	9.602195
8.000000	10.003695	9.899405
9.000000	9.999648	10.166133

$dt = 0.1$

t_max_analy: 12.074450 v_max_analy: 12.074450

t_min_analy: 0.000000 v_min_analy: 0.000000

t_max_num: 1.600000 v_max_num: 12.285594

t_min_num: 0.000000 v_min_num: 0.000000

t	v_analy	v_num
0.000000	0.000000	0.000000
1.000000	9.858360	9.203454
2.000000	11.396720	11.843132
2.999999	9.591516	9.600876
3.999998	9.936045	9.816604
4.999998	10.074864	10.116971
5.999997	9.985733	9.996117
6.999996	9.994236	9.982208
7.999995	10.003695	10.005710
8.999998	9.999648	10.001227

$dt = 0.01$

t_max_analy: 12.078793 v_max_analy: 12.078793

t_min_analy: 0.000000 v_min_analy: 0.000000

t_max_num: 1.569999 v_max_num: 12.100132

t_min_num: 0.000000 v_min_num: 0.000000

t	v_analy	v_num
0.000000	0.000000	0.000000

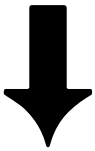
0.100000	0.233174	0.208691
0.200000	0.864847	0.817856
0.300000	1.794275	1.728305
0.400000	2.925543	2.845189
0.500000	4.171009	4.081470
0.600000	5.453765	5.360403
0.700000	6.709165	6.617123
0.800000	7.885510	7.799408
0.899999	8.944045	8.867760
0.999999	9.858355	9.794857
1.099999	10.613320	10.564611
1.199999	11.203755	11.170860

Class Structure

In this program, all the calculation and arrays assignment have been done in class **Amplifier**. Thus, lots of work is pre-done once the class constructor is called in main function. Below is how the operation processes inside the class **Amplifier**.

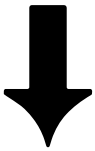
Calculate $V(t + dt)$

v_analytical(float t)
v_numeric(float t, float v_0, float v_1)



Assign the calculated output to the respective member arrays

assign_t_array(float t[])
assign_v_aray_analytical(float t[], float v[])
assign_v_aray_numeric(float t[], float v[])



The wrap-up caller of the three array assignment functions above

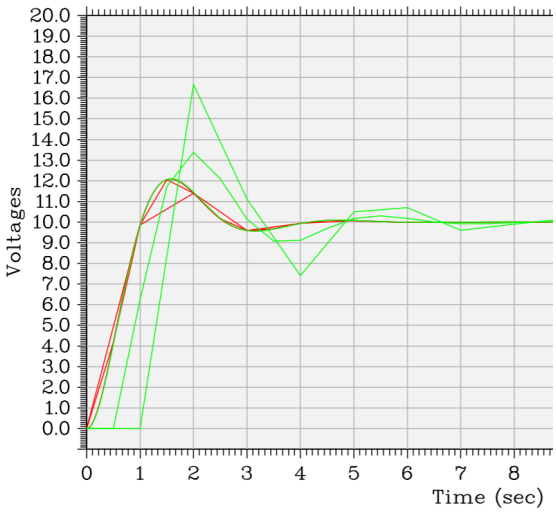
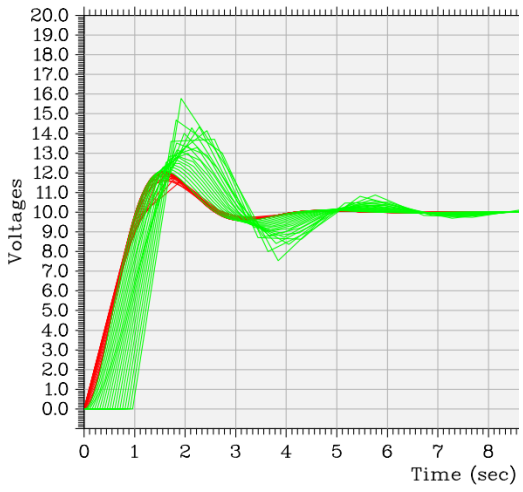
assign_arrays()

After all member arrays have been assigned, examine the max and min values

set_max_analy()
set_min_analy()
set_max_num()
set_min_num()

Constructors: One designed for user and the other for programmer

Amplifier(bool autofill)
Amplifier(float ending_t, float delta_t)

User-Defined Amplifier(bool autofill)	**Programmer-Defined** Amplifier(float ending_t, float delta_t)
	

Reference

Amplifier_circuit_F2020_E50.docx

https://sjcc.instructure.com/courses/22103/files/3021154?module_item_id=1035548