Weekly Report

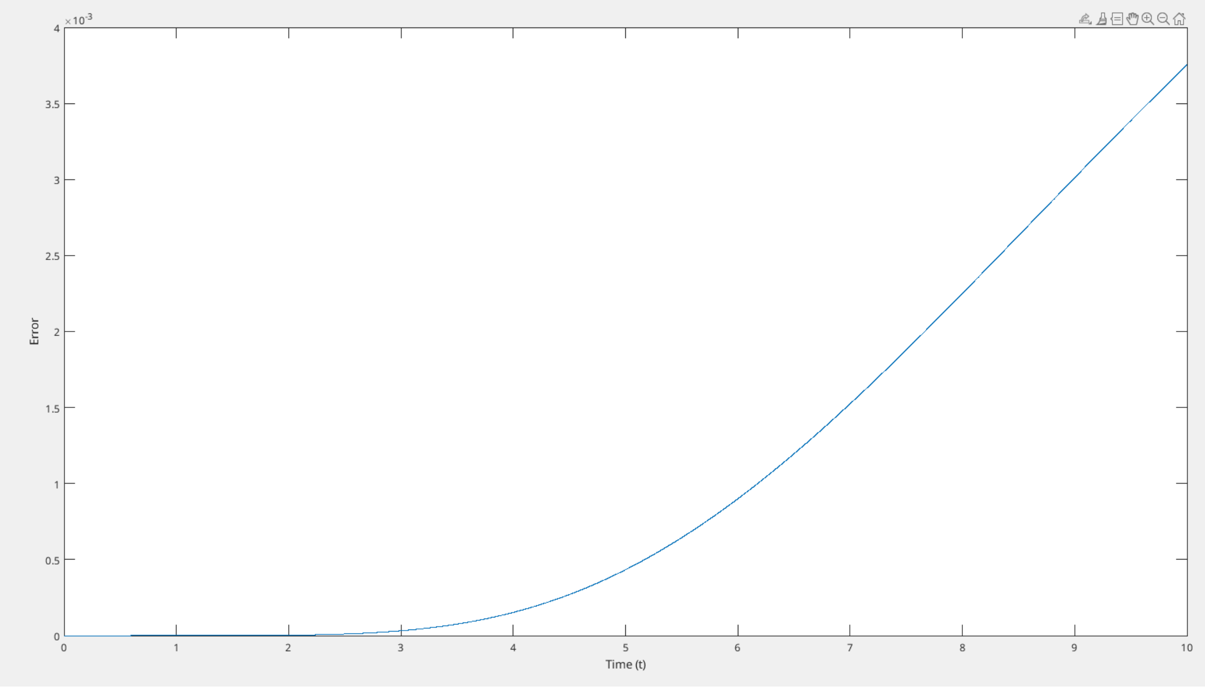
Name: Mohammed Al shuaili

Achieved goals through last week,

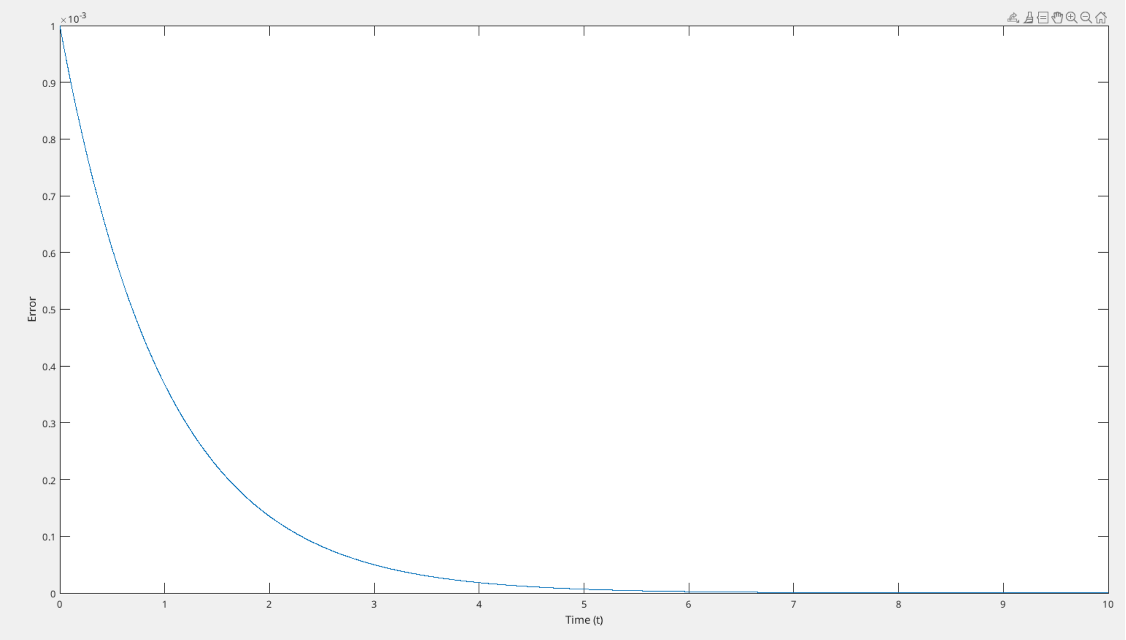
- Consider the previous step to approximate the next step in NILT0 and test it.

I’ve been trying to add this but I'm seeing a strange behaviour, and not sure if this should be expected.

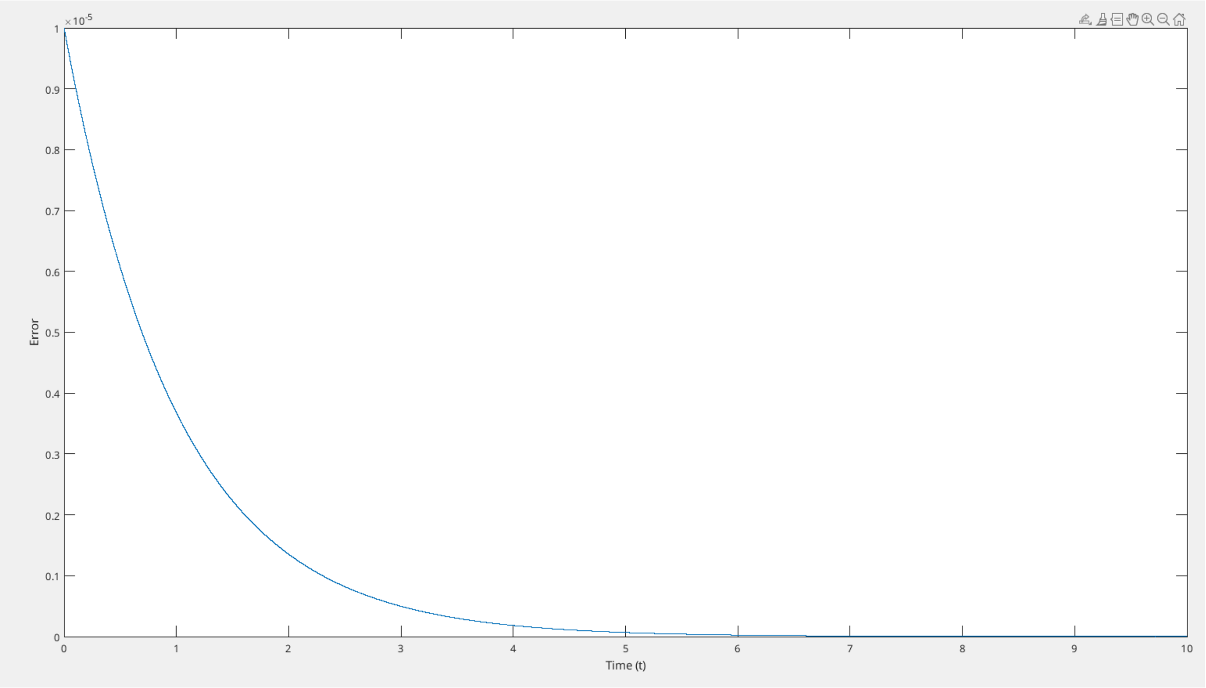
For example for 1/s+1 , if we don’t consider the previous step, the error at M = 5 is very low but it gets higher as time increases this with the unit step response and doesn't change if we change the value of h (step size).



But considering the previous step and using the same input as the top one, I got this ,



The error was kind of high at low t but it went down as time increased. However ,

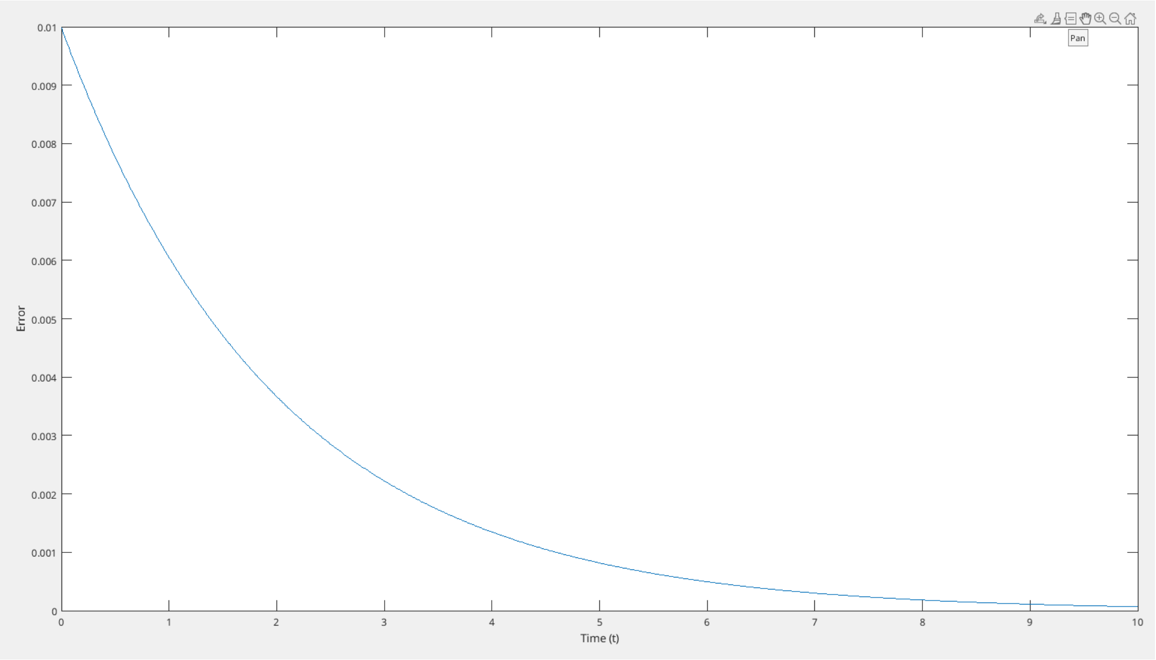


With the same M, but make more sections (i.e lower value for h).

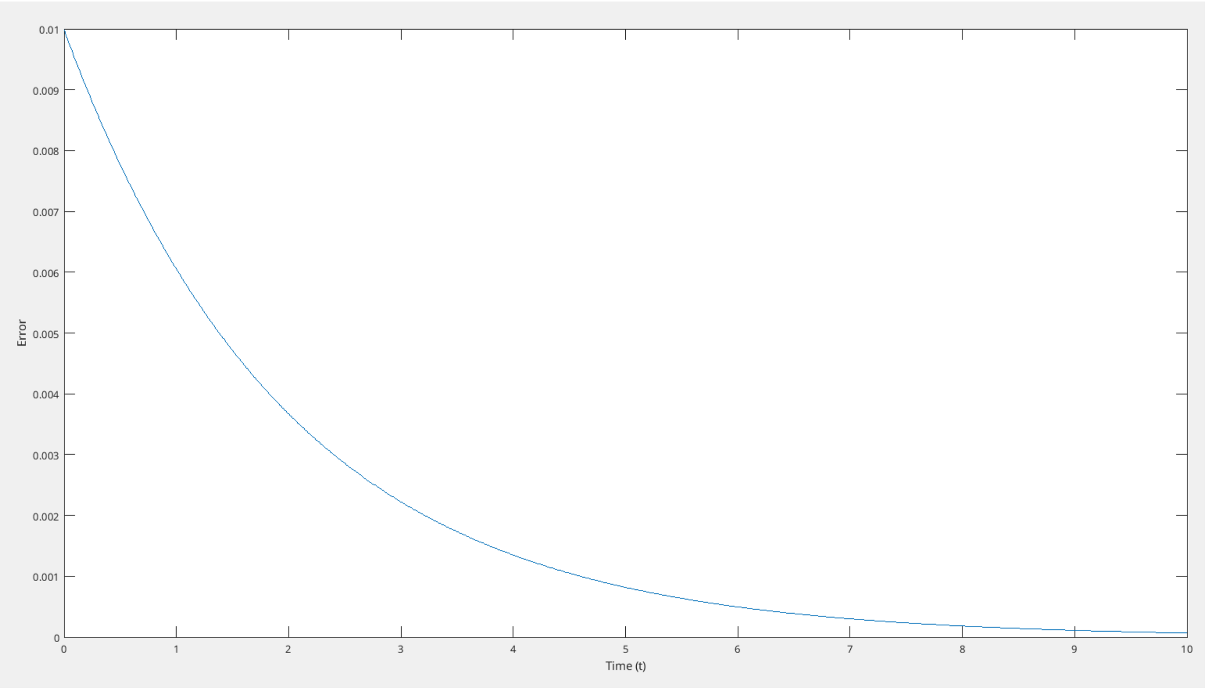
***

However, when I consider dy/dt = -0.5 y +1 , changing the value of M if we consider the previous step doesn't make any difference unless M goes above 13. additionally , only changing h (step size can have an effect).

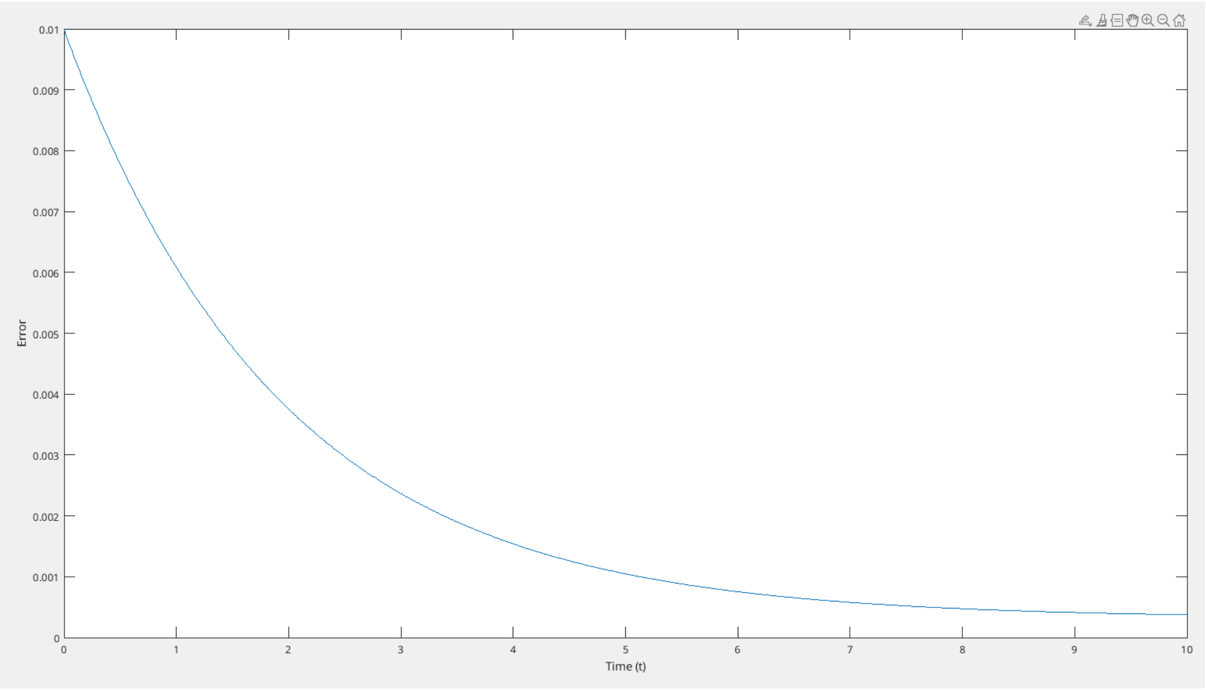
At M = 2 . (h=0.01)



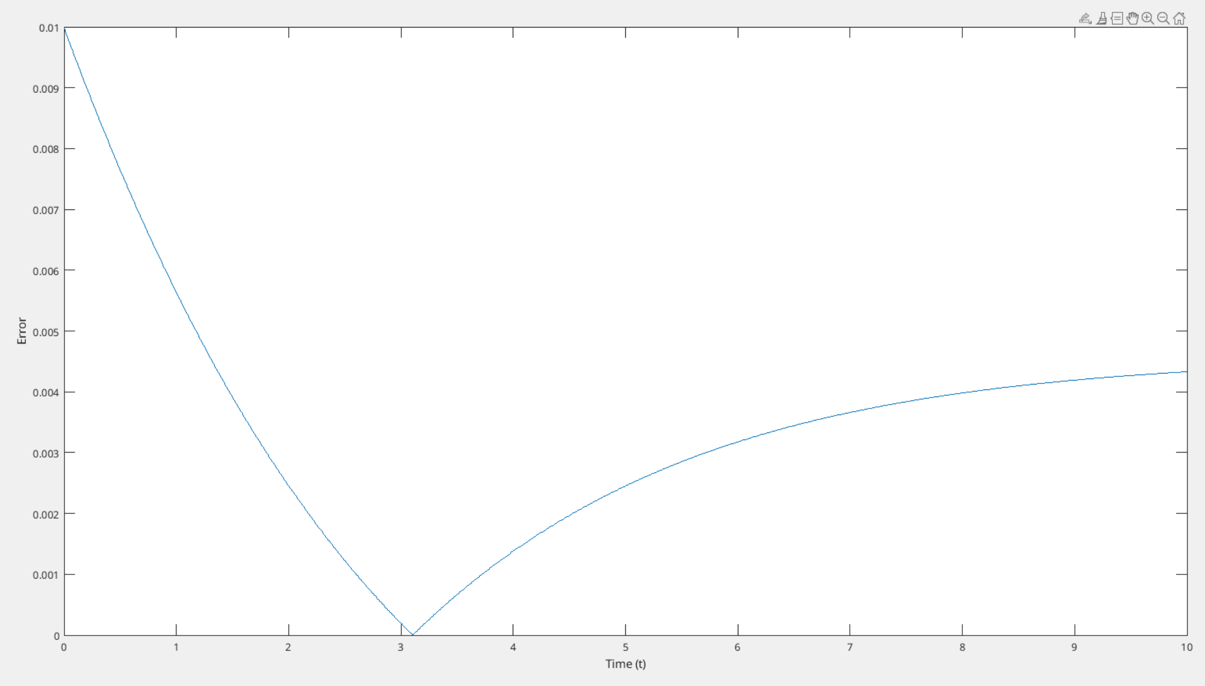
At M = 5,


Error
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
0
1
2
3
4
5
6
7
8
9
10
Time (t)

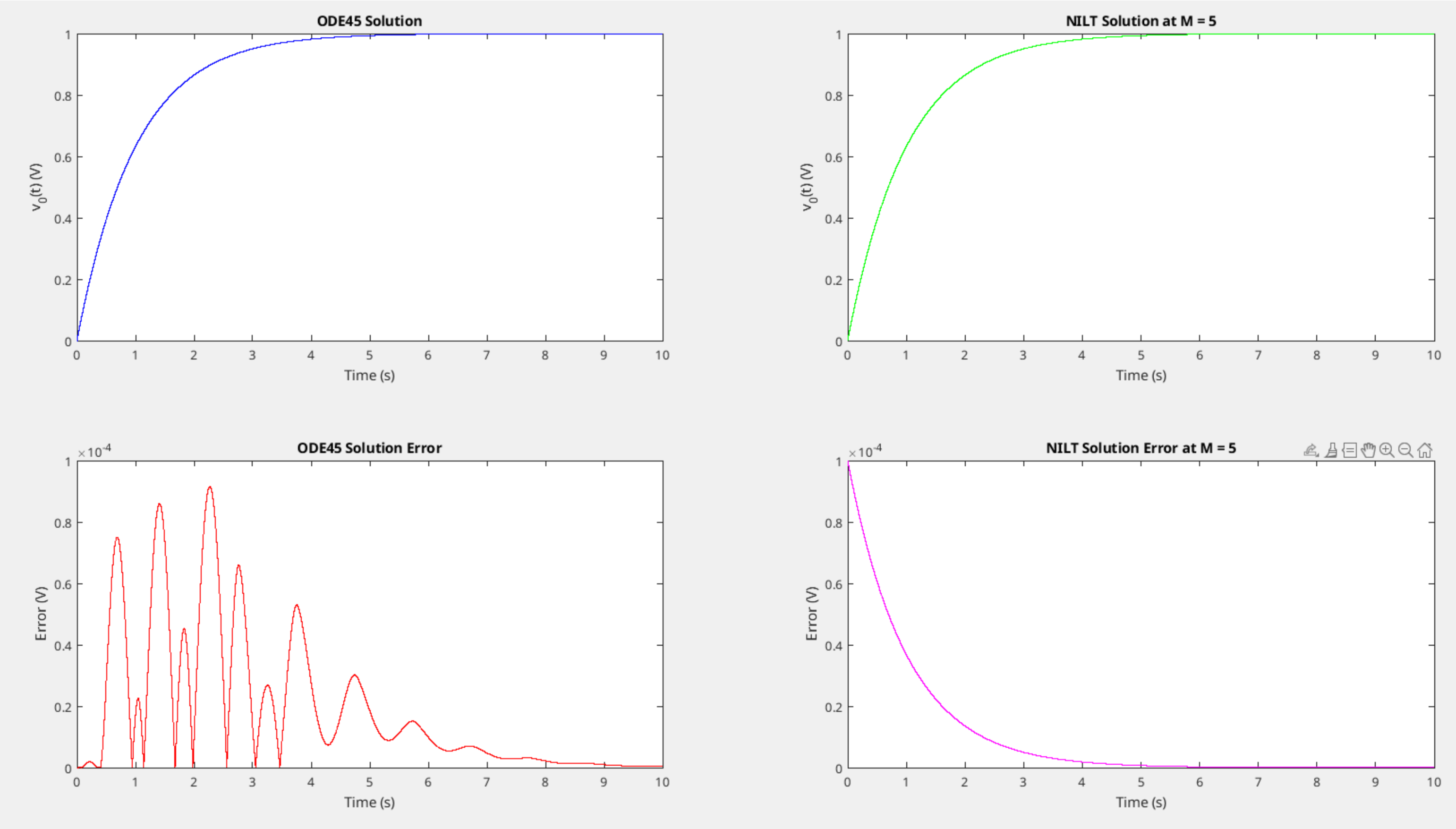
At M = 12


Error
0.01
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
0
1
2
3
4
5
6
7
8
9
10
Time (t)

At M = 13,

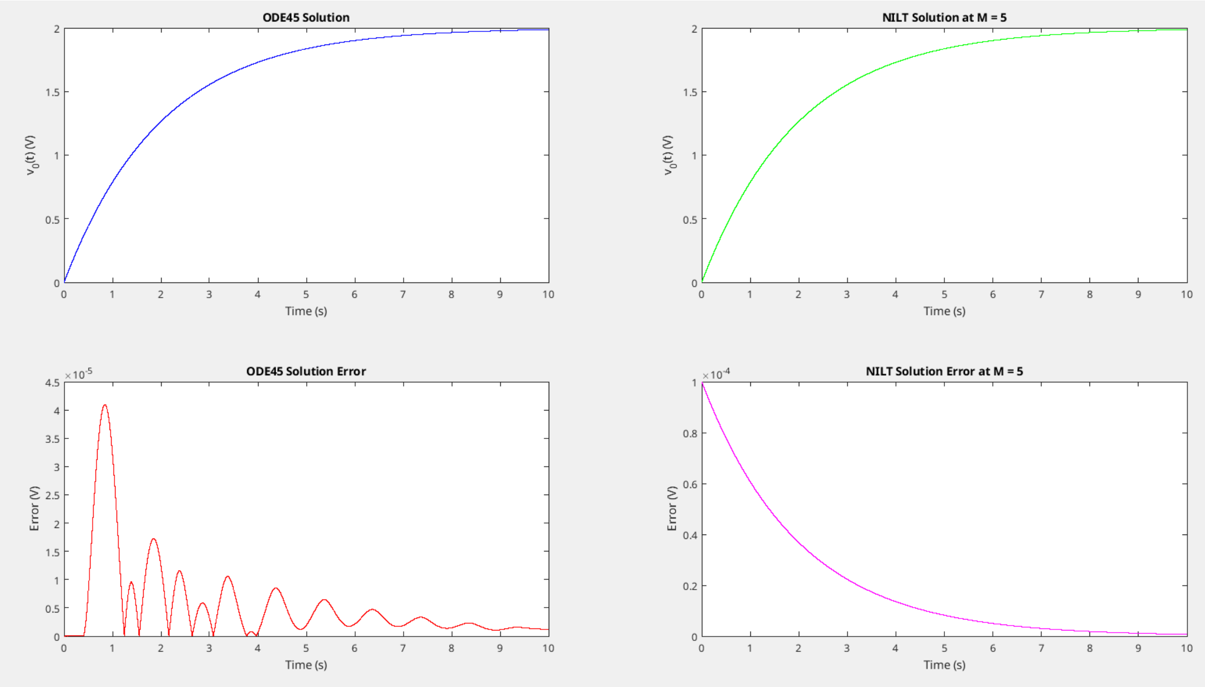


- Ode45 vs NILT0 with (h = 0.0001) with step response of (1/s+1)

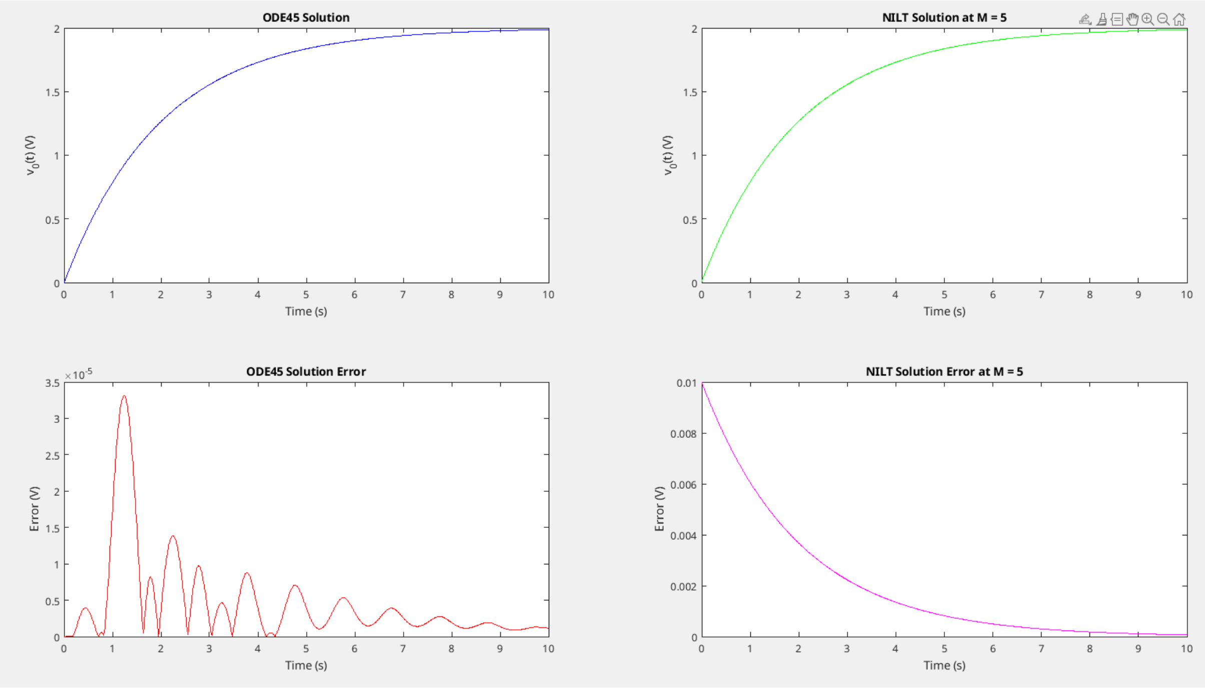


Overall , NILT performed much better than the Ode45 solver as time increased.

Testing with (1/s+0.5).

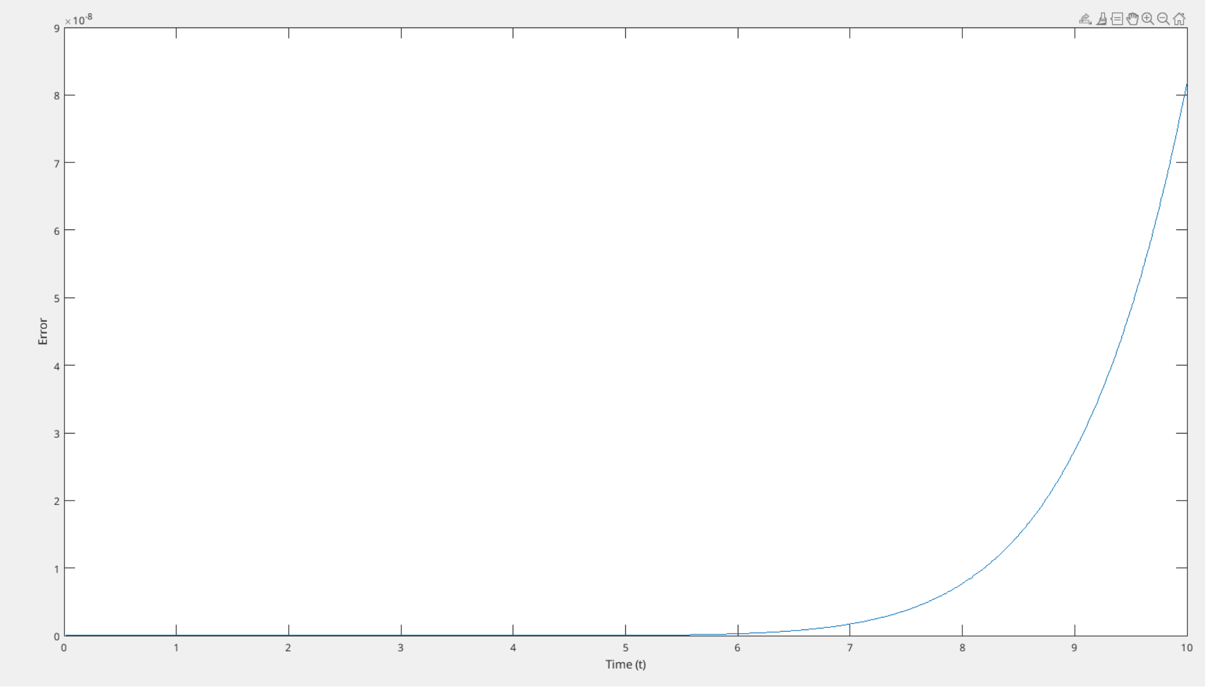


but , with h = 0.01 ,



The error in NILT is very high compared to Ode45.

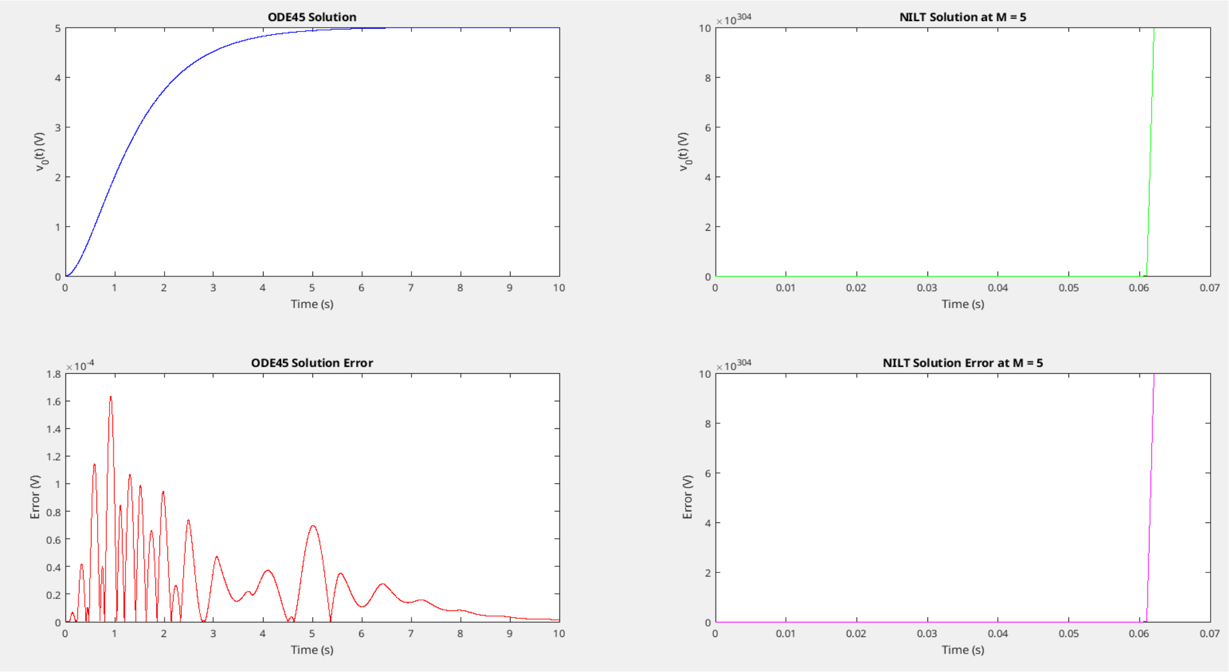
Without considering the time marching, at M=8, for 1/s+0.5, the error was much lower than when considering the previous step.



- Problems:

1. The issue I'm encountering is that, in my implementation, some functions perform better without considering the previous step. Additionally, changing the value of M has minimal impact when I do account for previous steps to approximate the next one. This suggests there may be a problem in the implementation, but I haven't been able to identify it yet.

2. I've struggled to implement the marching time method for second-order differential equations, as on pages 124 and 125. When I try incorporating the previous step into the next step's approximation, I encounter extremely high errors. This means something might be off in my implementation, though I haven't been able to find the issue yet. For reference, I'm looking at an example of an RLC circuit.



ODE45 Solution

NILT Solution at M = 5

ODE45 Solution Error

NILT Solution Error at M = 5