

# NUMERICAL METHODS

**Documentation:** Euler Backward Integration/Problem 5

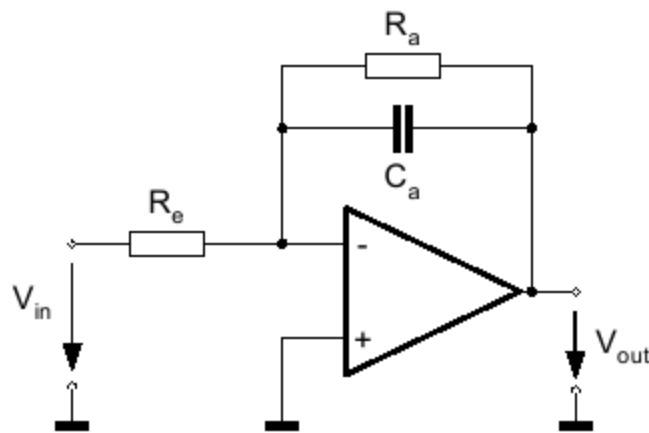
**Author:** Thomas Karanatsios

**Matrikel Number:** 11032154

## CONTENT

1. Tasks.....		1
2. Preparations.....		2
3. Implementation.....		3
4. Experiments .....		4
5. Further Comparison.....		6

## 1. TASKS



*Picture 1: Given Operational amplifier*

The inverting active lowpass filter sketched above is described in time domain by an ODE:

$$\frac{dV_{out}}{dt} + \frac{V_{out}}{R_a \cdot C_a} = -\frac{V_{in}}{R_e \cdot C_a}$$

*Picture 2: given ODE equation*

1. Rewrite the ODE to apply the solution strategy for the backward Euler formula.

2. Complete the code.

1. Check the results for proper function of your integration ( $t_{\text{step}} = 5 \cdot 10^{-6}$ ).  
 Hint: the test frequency is chosen to  $\omega = 10^5$ . This leads to 45 degrees phase shift.  
 Copy the results to a new file: cp out out\_5ms.dat

2. Do a calibration run with half step size ( $t_{\text{step}} = 2 \cdot 10^{-6}$ )  
 Load the waveforms of the previous run into the plot (Data -> Import -> ASCII, select out\_5ms.dat and set 'bad as' to NXY, do not change other settings and leave window by clicking OK)  
 Analyze and record the differences. Discuss the sources of this deviation. Save the recent results to a new file: cp out out\_25ns . dat

3. Integrate the ODE with slightly modified time step (e.g.  $2.49967 \cdot 10^{-6}$ ) and compare the last two runs. You will find only one remarkable difference. Explain this deviation.

## 2. PREPARATIONS

---

### Euler Backward Integration of linear ODEs

ODE is defined by  $\dot{x}(t) = f(x(t), y(t))$

first time step ( $n=1$ ): computation of value  $x_1$

prediction of  $x_1$  from the initial values of  $x = x_0$  and  $\dot{x} = \dot{x}_0$

$$x_1^P = x_0 + h \cdot \dot{x}_0$$

iteration 1

$$\dot{x}_1^{(0)} = f(x_1^P, y_1)$$

$$x_1^{(1)} = x_0 + h \cdot \dot{x}_1^{(0)}$$

iteration 2

$$\dot{x}_1^{(1)} = f(x_1^{(1)}, y_1)$$

$$x_1^{(2)} = x_0 + h \cdot \dot{x}_1^{(1)}$$

•  
•  
•

iteration sequence stops as far as the absolute change of  $x_1$  is less than a user defined value

$$|x_1^{(v+1)} - x_1^{(v)}| < \epsilon$$

second time step (n=2) : prediction of  $x_2$  from  $x = x_2$  and  $\dot{x} = \dot{x}_2$

$$x_2^P = x_1 + h \cdot \dot{x}_1$$

iteration 1

$$\dot{x}_2^{(0)} = f(x_2^P, y_2)$$

$$x_2^{(2)} = x_1 + h \cdot \dot{x}_2^{(0)}$$

•  
•

iteration sequence stops as far as the absolute change of  $x_2$  is less than a user defined value

$$|x_2^{(v+1)} - x_2^{(v)}| < \epsilon$$

The first Task is to rewrite the ODE. After that you get the following Backward Euler:

\_\_\_\_\_

### 3. IMPLEMETATION

Completing the Code with the rewritten Equation:

$\begin{aligned} d\_dt\_Vout\_n &= -(Vout\_n\_last\_iter/(Ra*Ca)) - (Vin\_n/(Re*Ca)); \\ Vout\_n &= Vout\_n1+t\_step*d\_dt\_Vout\_n; \end{aligned}$
---

What we can see here is one iteration of the corrector. As we can assume there will be more steps with this equation above. After this implementation the problem is fixed. But it is also possible to make the code working more along the lines with the instructions we got in the lecture of Mr. Kampmann. This could be possible instead of using the initial value for each step with:

$Vout\_n\_last\_iter = Vout\_n1;$
-----------------------------------

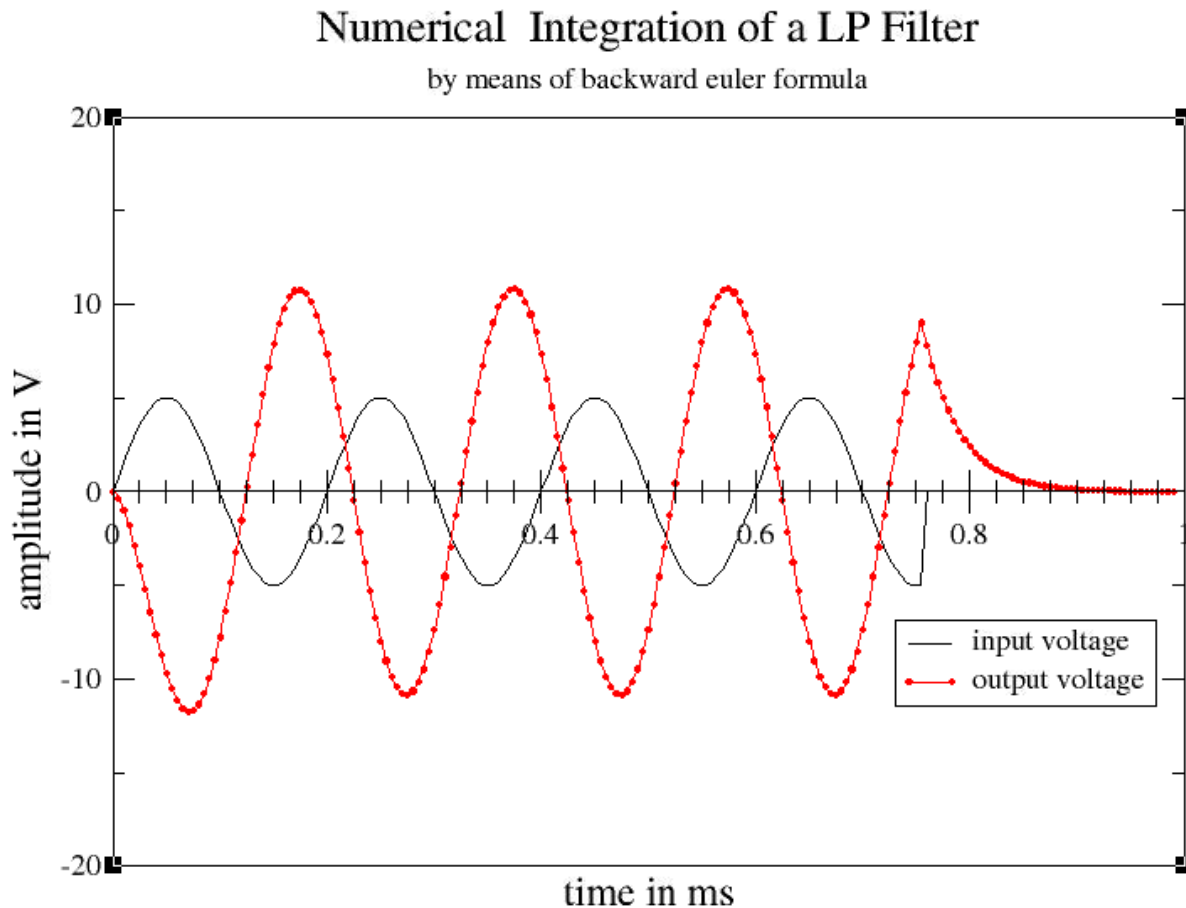
Now it's possible to use the following line of code:

$Vout\_n\_last\_iter = Vout\_n1+t\_step*(-(Vout\_n1/(Ra*Ca)) - (Vin\_n/(Re*Ca)));$
--

So this reduces the number of iterations, which means that one iteration is done outside the loop. In addition to that it's useful to know, that that does not really has any effect of the performance. So you will have an advantage, if the last derivative  $d\_dt\_Vout\_n$  would be saved. This is shown with  $Vout\_n1$ . That means you can reduce the computational effort, as a result of this and you will also increase the demand of memory.

## 4. EXPERIMENTS

In the first experiment and Task a functionality check was necessary. For this the code was checked with a step size of  $5 \cdot 10^{-6}$ . That offers the results on Picture 1. The Picture shows that the output signal is inverted and then it shifts at nearly  $45^\circ$ . But this is not really a surprise because this is the result which was expected from an Active-Low-Pass-Filter (first order) with an input signal of  $f_{3dB}$ .



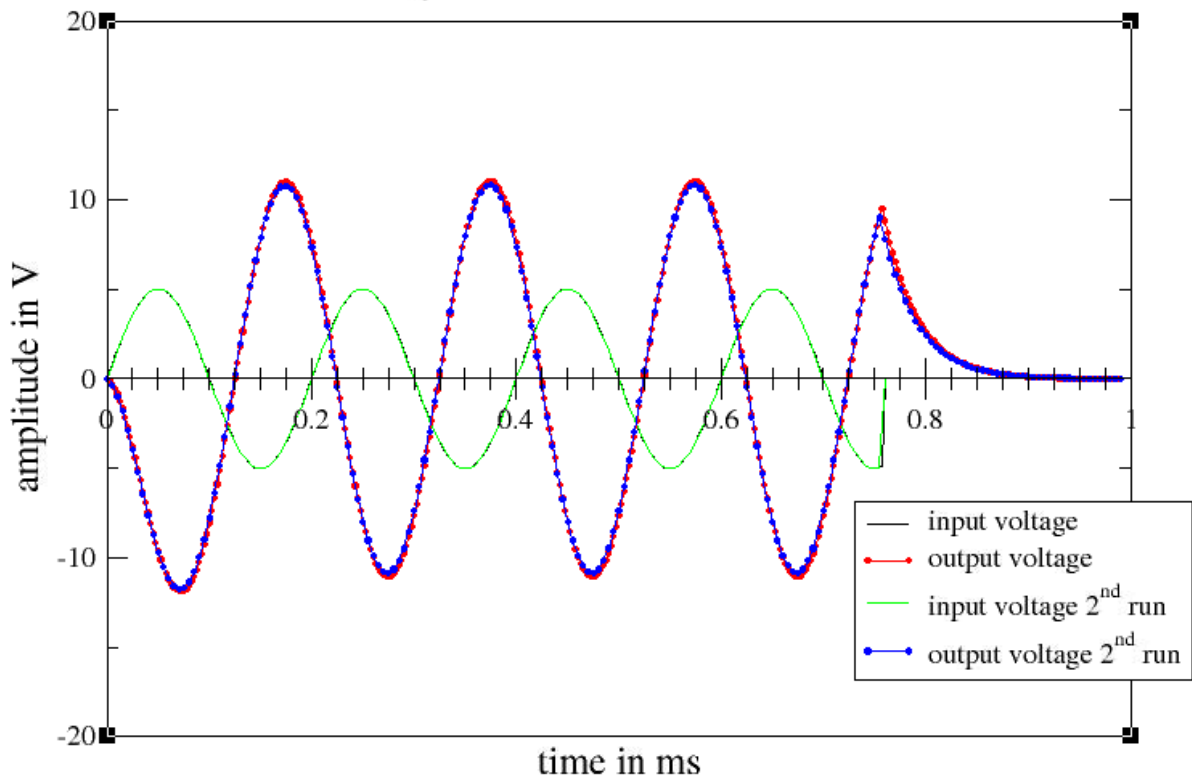
*Picture 1: Results of the functionality check*

The second experiment is the calibration run. In this step it was necessary to do a calibration run with a half step size and compare it with the results from Experiment 1. The result of this you can see in Picture 2.

There might be something confusing in Picture 2 because the Legend decrypted the Graph from the first Experiment with "2<sup>nd</sup> run". This is caused by the automatically created description when adding the data from the first Experiment into the plot. When you compare the Plots on Picture 2 you can see that there is a small difference between the Graphs with different step size. And of course exactly this is the reason of the different from the plots. It affects the accuracy of the solution even the experiment with smaller step size gets small different results, which normally should be more accurate. But it is not possible to see which plot is more accurate from Picture 2. This can be better seen when you zoom into the graph. As you can see in Picture 3 there could be a hint about the accuracy of the Graphs.

## Numerical Integration of a LP Filter

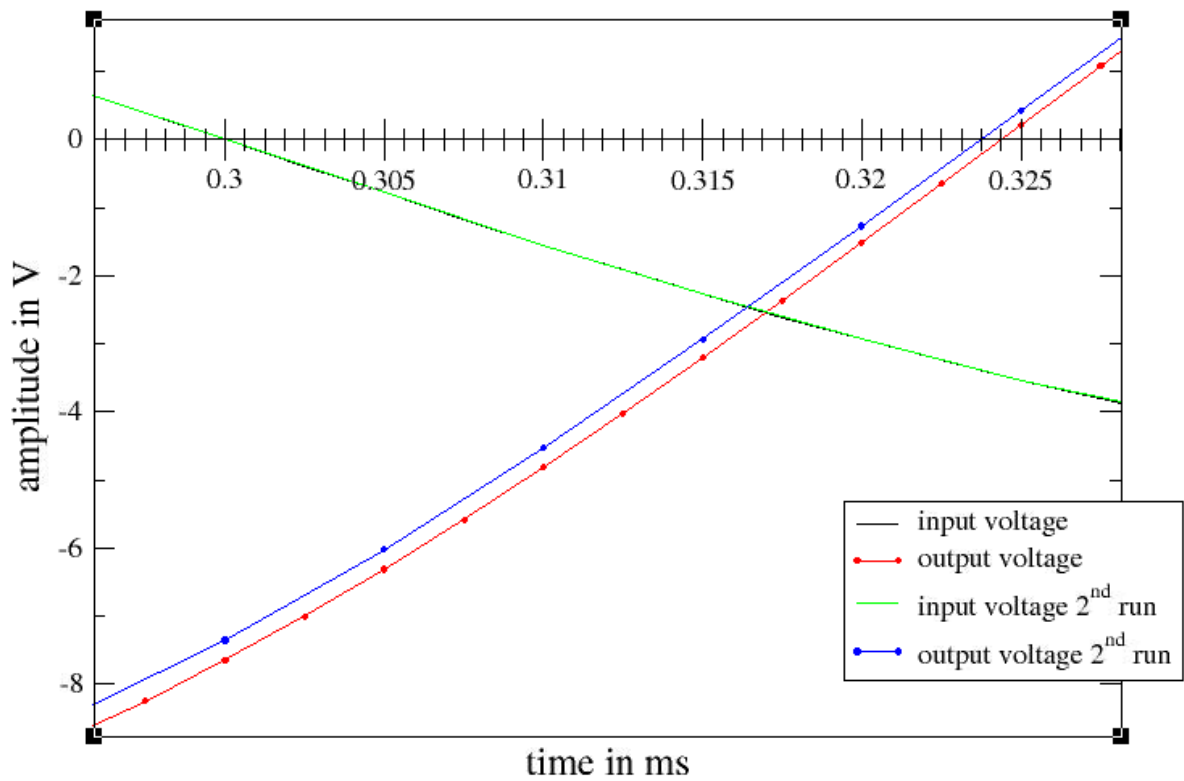
by means of backward euler formula



Picture 2: Results of Calibration Run

## Numerical Integration of a LP Filter

by means of backward euler formula

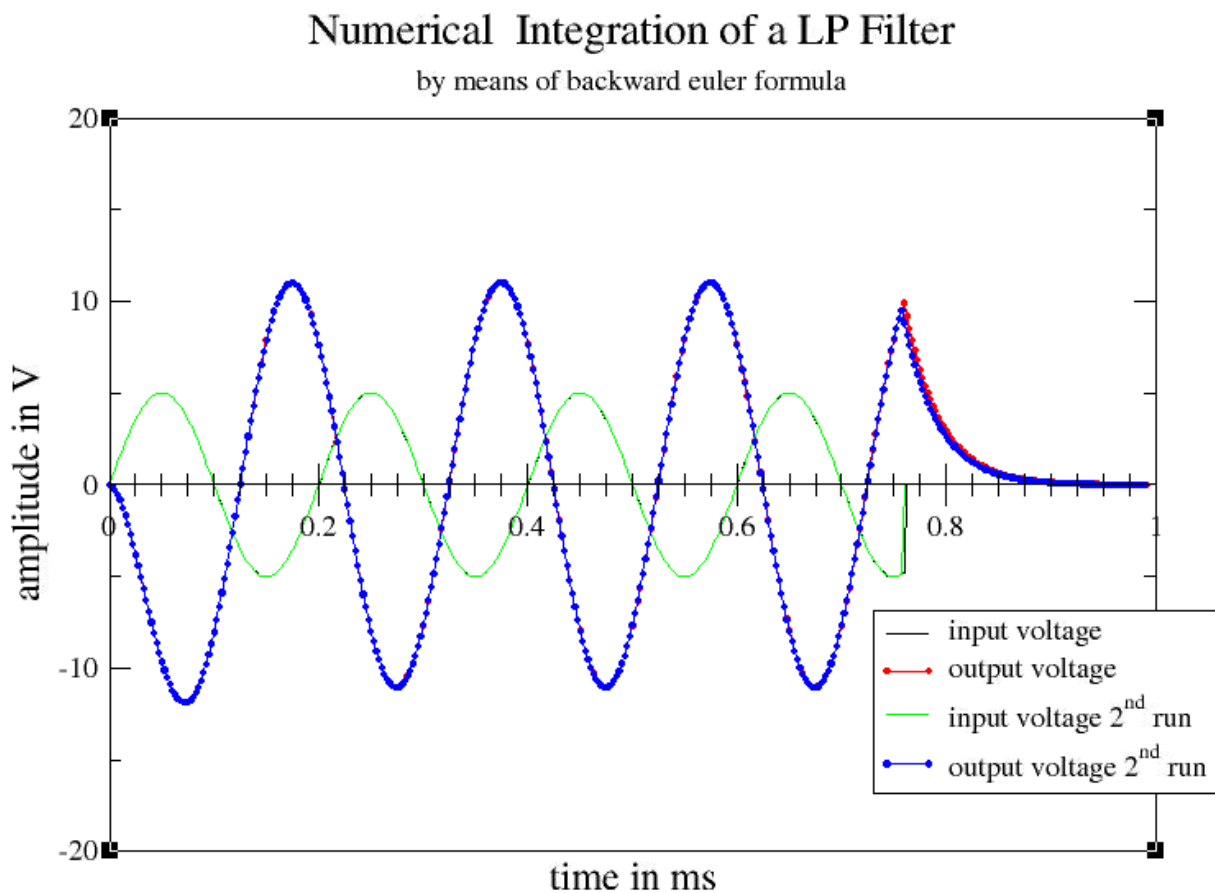


Picture 3: Zoom to see accuracy

You can see a better solution of the curves when you keep in mind that in a real solution the output signal would pass directly through 0.325 and the reason of that is the  $45^\circ$  shift. In the shown solution none of the two graphs reacts like that, but you can see that the red line is close to the 0.325 mark. This allows the conclusion that the shown red one must be more accurate than the Blue one and it can be also seen that the red line is the one with the lower step size. A possibility to verify this was to decrease the step size, which really shows then that the curve is moving closer and closer to the 0.325 mark.

## 5. FURTHER COMPARISON

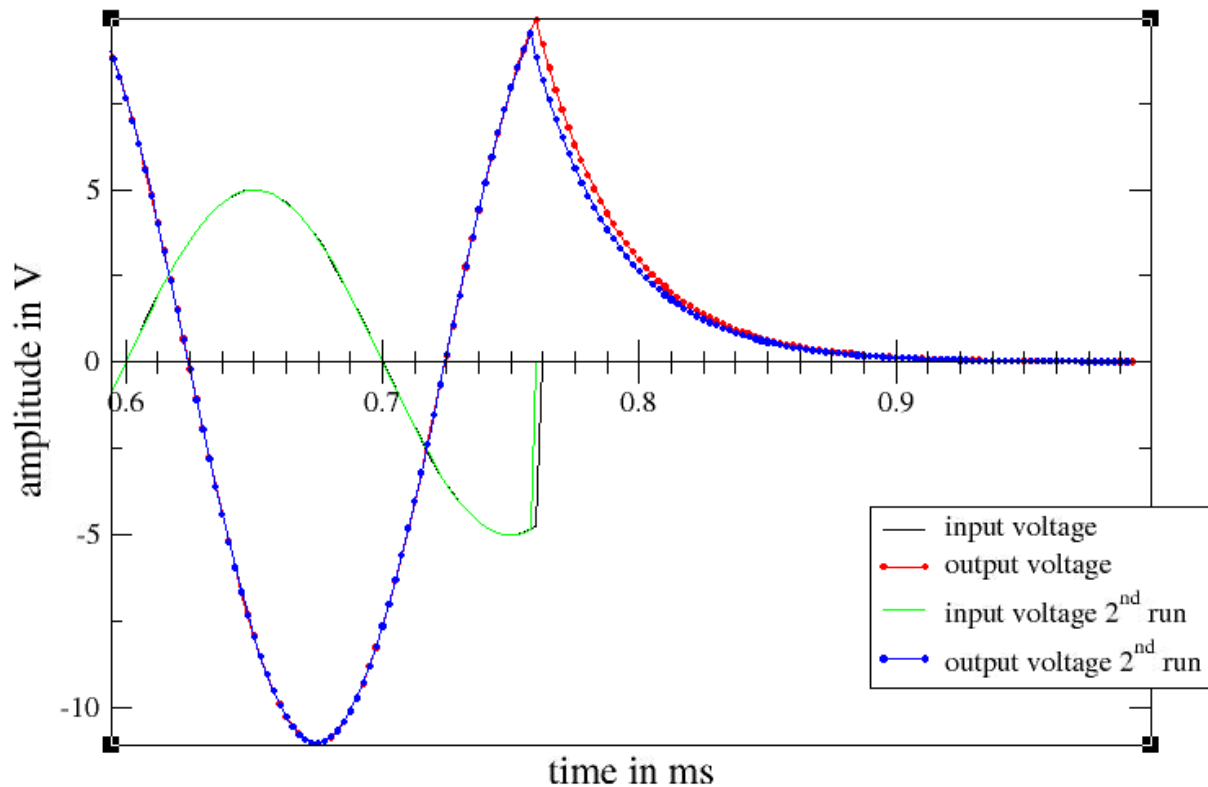
The further Comparison was done with slightly lowered step size. Again the last results are now compared with a slightly lowered step size of  $2.49967e-6$ . For the first one we can say that there should be a little influence on the result. As Picture 4 shows there is one noticeable difference at the end of the time line where the input signal ends. When we again make a zoom you can see the reason for that in Picture 5.



Picture 4: Noticeable difference at the end of the time line

## Numerical Integration of a LP Filter

by means of backward euler formula



Picture 5: Zoom to the function

After the zoom it is easy to see that the green input signal with a step size of  $2.5e-6$  stops one step earlier than the black one with a step size of  $2.49967e-6$ . The other effect is that the red curve rises one step more than the blue one because the red one has done one more step as it's shown in Picture 5. That happens because there is a slight difference in step sizes and the stopping criterion ( $T_a < \text{end\_sine}$ ) is been reached one step earlier for the green graph. This can be verified by dividing the `end_sine` value with the according step sizes. That means for the green graph  $0.76e-3/2.5e-6 = 304$  steps, and for the black graph  $0.76e-3/2.49967e-6 = 304.0401$  steps (305 rounded) which explains the difference of one step between the before shown graphs.