PLECS

Tutorial

Solver Accuracy and Settings in PLECS Standalone

Learn about the different numerical solvers found in PLECS and how to configure them

Last updated in tutorials release 1.0

1 Introduction

Variable-step solvers are more accurate and typically faster than fixed-step solvers for simulating power electronic systems. However, configuration of a variable-step solver is more complicated compared to a fixed-step solver since there are four configuration parameters that influence the accuracy and time step of the simulation rather than a single simulation parameter. The four configuration parameters for a variable-step solver are the relative tolerance, absolute tolerance, maximum step size and refine factor.

In this exercise you will learn the following:

- The purpose of the configuration settings for a variable-step solver.
- How to refine the rendered simulation output in an efficient manner.
- How to compare the accuracy of variable- and fixed-step solvers.

Before you begin Ensure the files `solver_settings_1.plecs`, and `step_size.plecs` are located in your working directory. You should also have the reference file `solver_settings_2.plecs` that you can compare with your own model at the end of the exercise.

2 A Resonant LC Circuit

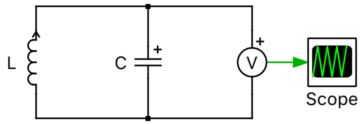

Figure 1: Resonant parallel LC circuit

In this exercise you will use the parallel resonant circuit depicted in Fig. 1 to investigate the effect of changing the different solver settings. The capacitance of the circuit is set to $1\ \mathrm{F}$ and the inductance to $\frac{1}{(2\pi)^2}$ H, yielding a resonant frequency of $1\ \mathrm{Hz}$. When the initial voltage on the capacitor is set to $1\ \mathrm{V}$, the transient response of the voltage across the capacitor can be analytically expressed by:

$$v(t) = cos(2\pi\mathbf{t}) \tag{1}$$

The transient response of the capacitor voltage is depicted in Fig. 2.

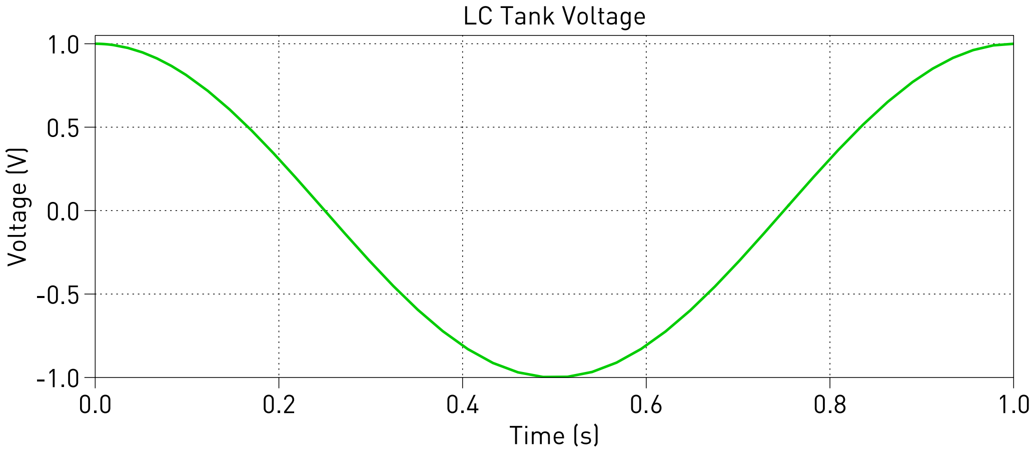

Figure 2: Transient response of resonant LC circuit with the initial capacitor voltage set to 1 V

3 Relative Tolerance

The most important configuration parameter for a variable-step solver is the relative tolerance, which defines the local error limit or acceptable error for the state variables. The relative tolerance is also used by the solver to calculate the step size as follows:

$$h = \left(\frac{tol_{\mathrm{rel}}}{\varepsilon}\right)^{\frac{1}{5}} h_{\mathrm{old}} \tag{2}$$

where tol_{rel} is the relative tolerance, h_{old} is the previous time step, and ε is the relative error. The relative error is the relative difference between the 4th and 5th order solutions and is a measure of the accuracy of the calculated state value.

Table 1: Step sizes for Dopri solver using different relative tolerances

Test No.	Solver Type	Relative Tolerance (%)	Δt (s)
1	Dopri	1e-3	
2	Dopri	1e-4	
3	Dopri	1e-5	

Your Task:

1 Open the model `solver_settings_1.plecs`. In the **Simulation Parameters (Ctrl+E)** select the variable-step **Dopri** solver type and set the **Relative tolerance** to $1\mathrm{e}{-3}$, as shown for Test 1 in Table 1. Set the **Max step size** to a large value such as $1\mathrm{e}3$ so that this setting does not limit the calculated step size. Set the **Absolute tolerance** to `auto` and the **Refine factor** to 1.

2 Run the simulation for 10s. Use the Step Size Measurement block from the file *step_size.plecs* to measure the average step size and display the result on a separate scope plot. You can also use the scope cursors to lock on two consecutive simulation steps and enable the **Delta** function in the **Data** window to measure the step size. Record this value in Table 1. (Answer: 0.135)

3 Repeat the simulation for relative tolerance values of $1\mathrm{e}{-4}$ and $1\mathrm{e}{-5}$, as shown for Tests 2 and 3, and enter the results in Table 1. (Answer: 0.085, 0.054) If you want to keep the solver settings window open during this exercise to easily commit changes without always closing and re-opening it, you must hit the **Apply** button after making a parameter change.

4 Refining the Output

Although the simulated response of a circuit meets the acceptable error constraint that is defined by the relative tolerance, only the major time steps or the points calculated by the solver have this level of accuracy. For larger time steps, the transient response may appear jagged since the scope uses linear interpolation to render the response between major time steps. For such cases, it would be nice to reduce the time step or add additional time steps to improve the rendering.

Your Task:

Table 2: Comparison of different methods of refining the simulation output

Test No.	Solver Type	Relative, Absolute Tolerance (%)	Max Step Size (s)	Refine Factor	$\Delta t(s)$	tsim (s)
1	Dopri	1e-5	1	1		
2	Dopri	1	0.05	1		
3	Dopri	2e-4	1	2		

1 Change the relative tolerance back to $1\text{e}{-3}$ and run a simulation. Uncheck the second signal in the Scope that shows the calculated steps using impulses and verify that the remaining waveform is not smooth and sinusoidal. In fact, each data point has been calculated correctly but the Scope renders a waveform by linearly connecting each step as described above, and these steps are spaced at varied amounts of time with a variable-step solver. Adding more data points in between will improve the smoothness of the waveform.

2 The most obvious method of reducing the time step is to reduce the relative tolerance again. Using the **Dopri** solver, decrease the relative and absolute tolerances both to $1\text{e}{-5}$ and set the maximum step size to 1 s, as shown for Test 1 in Table 2. Run the simulation for $1\text{e}{-4}\text{s}$. The output should be much smoother due the smaller step size. Record the average time step, $\triangle t$, and the simulation time, t$_{sim}$, in the table. To time the simulation you will need write a simple Simulation Script. Add a new Simulation Script from the **Simulation + Simulation scripts**... menu and enter the following code:

```
tic
plecs('simulate')
toc
```

Note: Note that you will need to open up the Octave Console window via the **Window +Show Console** menu to view the output for the simulation time after the simulation has completed.

Also note that the simulation run time of the same model with the same settings may vary between different computers because the processing speed depends on the unique hardware of that computer. Also the simulation time of the same model with the same settings can vary on the same computer depending on the available resources due to other processes running in parallel to PLECS.

3 Another method of reducing the step size is to control the step size directly using the maximum step size parameter. This is possible when the relative tolerance is set to a large value. Set the relative and absolute tolerances to a large value such as 1 to allow the maximum step size parameter to control the step size. Set the maximum step size to 0.05 s, as shown for Test 2. Rerun the simulation and enter the results for $\triangle t$ and t$_{sim}$ in Table 2.

4 A less-obvious but more efficient method of adding additional time steps is to increase the **Refine factor** parameter in the solver settings. The refine factor is the number of points that are calculated along a polynomial fit that is created by the solver. Normally the refine factor is set to 1 and the polynomial fit is used to calculate the output at the next major time step. By increasing the refine factor, additional points or minor time steps are added along the polynomial. Since the polynomial does not require updating for these additional steps, the computational overhead is lower compared to reducing the step size.

Set the relative and absolute tolerances to $2\text{e}{-4}$, the maximum step size to 1 s, and the refine factor to 2 as shown for Test 3 in Table 2. Run the simulation and measure $\triangle t$ using the cursors and

Delta function rather than the Step Size Measurement block. The Step Size Measurement block does not respond to the minor time steps that are induced by using a refine factor greater than 1. Enter the result for Δt and t_{sim} in Table 2.

Is the simulation faster or slower compared to Tests 1 and 2?

The simulation is faster.

Note: With a relative tolerance of $2\mathrm{e}{-4}$, some damping can be seen in the transient voltage waveform. In extreme situations, a variable-step solver can introduce slight numerical damping, depending on the circuit and the step size. The resonant LC circuit is one such case and therefore the relative tolerance or maximum step size should be set appropriately in order to mitigate this effect.

PLECS Blockset users can also take advantage of the output rendering trick described above in Step 4. by accessing a model's settings in Simulink. A **Refine factor** parameter can be found at the bottom of the **Data Import/Export** pane of the **Configuration Parameters** window. Please consult the **Help** button for more information on the different refine factor output options provided by Simulink.

5 Accuracy Properties

For this test, you will investigate the accuracy of the variable-step solvers and the fixed-step solver by comparing the simulated response of each solver with the analytical response given in Eq. (2).

Your Task: To simulate the analytical solution, place a Sine Wave block and set the frequency to $2 * pi$ rad/sec and the phase to $pi/2$ rad. Create an error signal by subtracting the analytical solution from the simulated output voltage and display this signal on a separate scope plot.

At this stage, your model should be the same as the reference model, `solver_settings_2.plecs`.

Your Task: Answer the following questions.

1 Use a **Dopri** solver and set the maximum step size to $0.05\ \mathrm{s}$ and the relative tolerance to the large value of 1 so that the maximum step size setting controls the solver time step. Run the simulation for $10.0\ s$ and measure the peak error. Record this result in Table 3.

2 Repeat Step 1 for the variable-step **Radau** and fixed-step Discrete solver types, recording your results in Table 3. The **Radau** solver is the second variable-step solver type option in PLECS Standalone. To select the Discrete solver choose **Fixed-step** under **Solver Type** and **Tustin** under **Disc. method**.

Which solver is the most accurate for this particular circuit?

The stiff system solver, **Radau.**

3 You will notice that for the Discrete solver, a notable frequency error is present. This can be improved by reducing the time step. Reduce the time step by a factor of 10 to $5\mathrm{e}{-3}$.

By what factor is the error reduced?

A The error is reduced by a factor of 100.

4 Reduce the time step for the **Dopri** and **Radau** solvers to $5\mathrm{e}{-3}$ and measure the error.

? By what factor is the error reduced?

A The error is reduced by a factor of 10^5.

Note: From the time step experiments, it can be seen that reducing the step size by a factor of n with the Discrete solver improves the accuracy by a factor of n^2 because the Discrete solver is a second order solver. On the other hand, reducing the step size n times with the **Dopri** and **Radau** solvers reduces the error by a factor of n^5 since these solvers are fifth order solvers.

Table 3: Variable- and fixed-step solver settings for simulation tests

Test No.	Solver Type	Relative Tolerance (%)	Max Step Size (s)	Refine Factor	Peak Error (%)
1	Dopri	1	0.05	1	
2	Radau	1	0.05	1	
3	Discrete (Tustin)	-	0.05	-	

6 Conclusion

In this exercise you have looked at the different solver settings for variable-step solvers and examined the accuracy of the variable-step solvers and the fixed-step solver. With a variable-step solver, the time step should be normally controlled using the relative tolerance. The time step will be automatically adjusted to meet the acceptable error constraint that is defined by the relative tolerance.

Although a simulation may meet the acceptable error constraint, the response rendered by the scope may appear jagged for larger time steps. To improve the rendering, the time step can be reduced by reducing the relative tolerance or limiting the maximum step size. However, the most efficient means of improving the rendering is to increase the refine factor.

In terms of accuracy, the variable-step solvers are much more accurate than the fixed-step solver for a given step size. Reducing the step size of a solver by a factor of x improves the accuracy by a factor of x^n, where n is the order of the solver.

For further reading on the simulation of continuous systems, the following book is recommended:

References

[1] Francois Cellier and Ernesto Kofman, *Continuous System Simulation*, Springer, New York, 2006.

Revision History:

Tutorials 1.0	First release

How to Contact Plexim:

☎	+41 44 533 51 00	Phone
	+41 44 533 51 01	Fax
✉	Plexim GmbH Technoparkstrasse 1 8005 Zurich Switzerland	Mail
@	`info@plexim.com`	Email
	`http://www.plexim.com`	Web

PLECS Tutorial