Speeding up simulation with Simscape Electrical

TN182 | Posted on June 18, 2026 | Updated on June 18, 2026

Shu WANG
Development Engineer

Table of Contents

Starting from MATLAB R2026a, the Specialized Power Systems library (SPS) has been removed from Simscape Electrical (read our blog article on this topic). For many power electronics engineers, this represents a significant loss, as native Simscape Electrical elements behave fundamentally differently, and do not provide equivalent performance unless their operation is well understood, and the model configuration precisely optimized.

This article provides strategies for imperix users to speed up simulation with Simscape Electrical, and benchmarks different configurations against other tools such as SPS and PLECS. It focuses on the plant model side, as the optimization of the controller side is covered in PN131.

Right-sizing the model fidelity

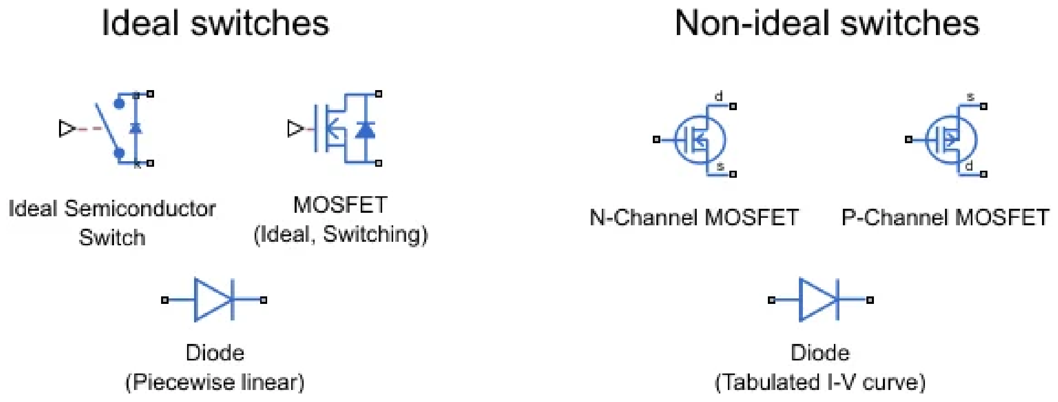
Simscape Electrical is aimed at implementing high-fidelity models. However, this is not always better as excessive details may unnecessarily increase solver complexity and slow down the simulation.

1. For semiconductors: Use ideal switches

Simscape Electrical often provides multiple block variations for the same component. Semiconductor switches are typically available in two types:

- **Ideal:** models switching devices with piecewise-linear I-V curves, similar to SPS and PLECS.
- **Non-ideal:** models switching devices using tabulated I-V curves or SPICE-like nonlinear models, enabling simulation of gate charges and thermal losses.

Imperix recommendation: for control engineers, use ideal switches and ignore gate-driving circuits and thermal losses.



Simscape Electrical switch blocks

2. For passive components: Avoid zero ESRs

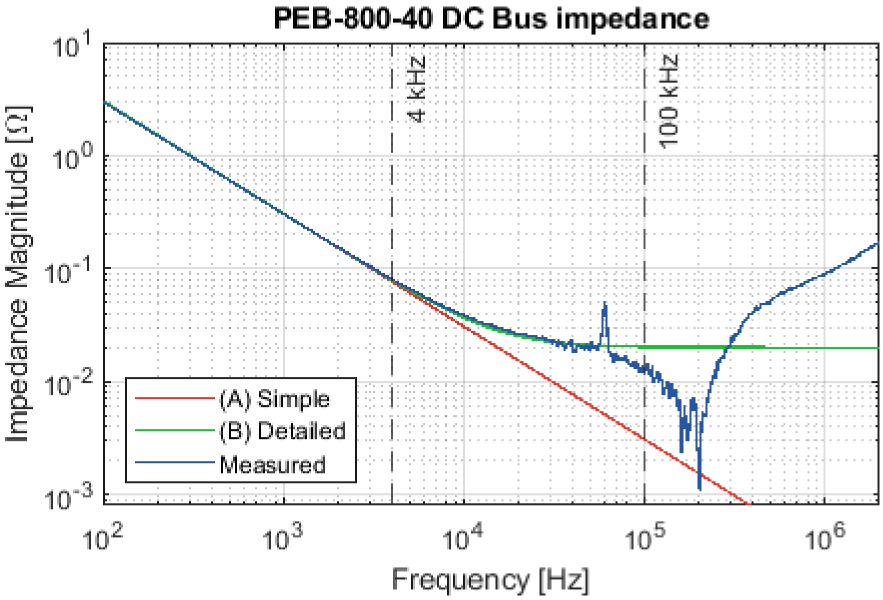
This behavior is highly counterintuitive for engineers coming from SPS or PLECS. With these softwares, setting a component's Equivalent Series Resistance (ESR) to 0Ω is recommended to speed up simulation, as it lets the solver reduce the number of nodes and shrink the state-space matrices.

With Simscape Electrical, if an ideal switch connects to a capacitor or inductor with zero ESR, the physics engine demands an infinite instantaneous current surge, creating a stiff algebraic loop. To resolve this singularity, the variable-step solver shrinks its time step, significantly reducing speed. A realistic ESR limits the instantaneous current and provides mathematical damping, allowing larger, faster time steps.

MathWorks documents this bottleneck in the Simscape User's Guide. Readers can refer to the section titled *"Avoiding Numerical Simulation Issues"* for more detailed information and practical examples.

Imperix recommendation: always use the Imperix Power library. It provides optimized models for use with ideal switches and offers two modeling levels, each approximating the impedance of the passive elements around them within a given frequency validity range:

- **(A) Simple:** models only the main components, sacrificing the accuracy of high-frequency characteristics above 4 kHz.
- **(B) Detailed:** models the parasitics for high modeling accuracy up to 100 kHz.



Model of PEB-800-40 DC bus impedance versus measurement

Imperix recommendation: both the *Simple* and *Detailed* options are maintained in the Simscape Electrical blockset contained within the *Imperix Power Library* (for cross-compatibility reasons). However:

- Prefer the *Simple* model with SPS and PLECS, because it is significantly faster (see below).
- Always use the *Detailed* level in Simscape Electrical, because it contains ESRs to avoid numerical simulation issues.

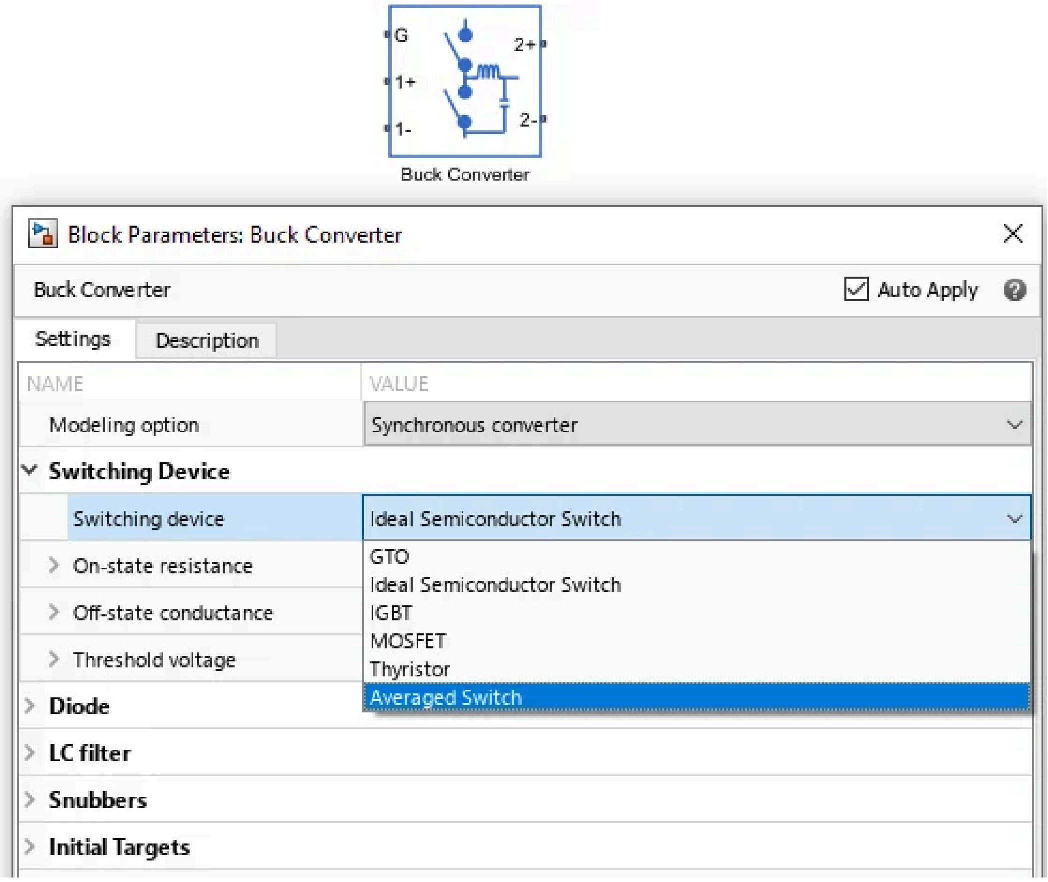
More information on the Imperix Power library is provided in PN150.

3. Using averaged switching models

When a converter is built with individual switches, semiconductors are treated as discrete on/off elements. The variable-step solver must therefore reduce its time step to capture each switching event (zero-crossing) and recompute the equations for the new circuit. This produces excessively small steps and slows down the simulation when signals change rapidly.

Averaged switch models avoid this by replacing discrete gate signals with a continuous input representing the average duty cycle over a PWM period. Because the model stays continuous, the solver avoids heavy zero-crossing detection and takes large time steps.

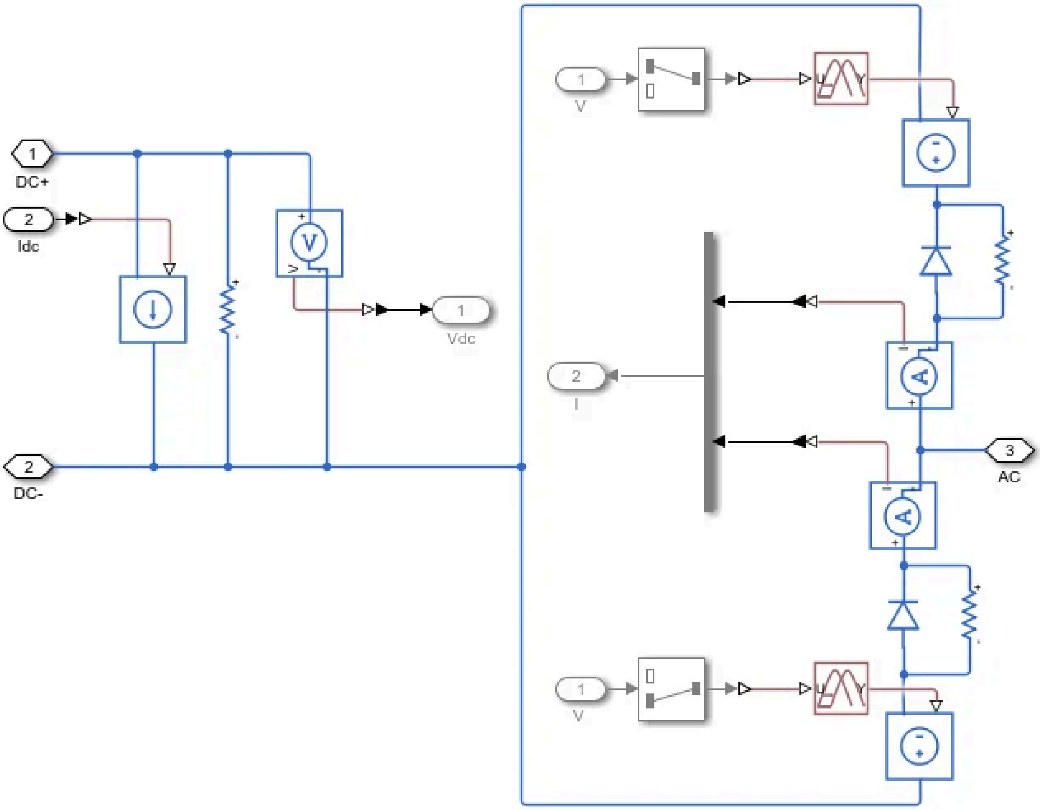
Simulink Simscape provides native average models for some standard converter topologies, such as the Buck Converter and Average-Value Inverter blocks. However, these models have limited versatility. For instance, adapting a buck converter block into an inverter phase-leg is not allowed due to the non-removable capacitance between the AC output and the DC-. This conflicts with the flexible design philosophy of imperix power modules.



Modeling option: Synchronous converter

Switching device: Ideal Semiconductor Switch

Buck converter block from Simscape Electrical



Averaged model of the imperix PEB module

The *Imperix Power Library* circumvents this by offering flexible, topology-agnostic averaged models using controlled current and voltage sources, as SPS and PLECS do. With the *Imperix Control Library*, the pulse-width modulation (PWM) and duty-cycle averaging can be bypassed simultaneously by feeding the controller's continuous duty cycle directly to the converter inputs, greatly reducing simulation time.

Imperix recommendation: use the averaged model for system-level simulations where switching ripple is irrelevant, such as power flow control, primary/secondary control, and energy management. Use the switched model when the modulator's behavior is not negligible, such as when optimizing voltage or current control loops, simulating dead-time effects, or using phase-shift modulation.

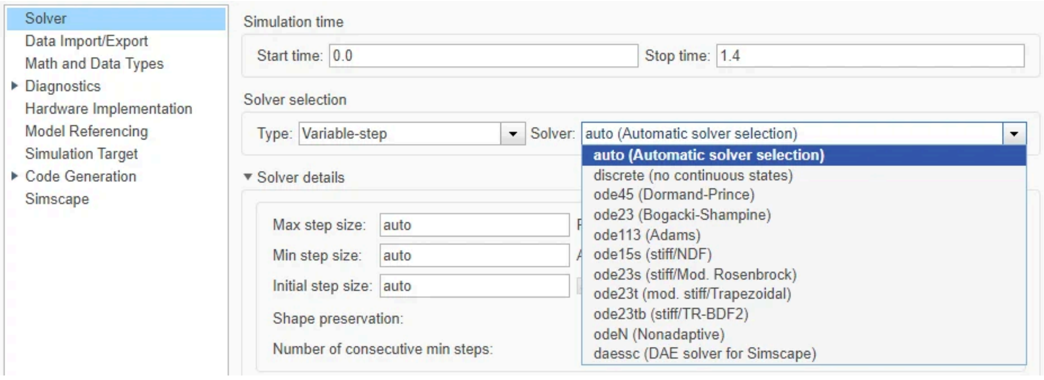
Choosing the appropriate solver

This section assumes familiarity with numerical simulation concepts, particularly the difference between fixed-step and variable-step solvers. New users should review the MathWorks documentation on Important Concepts and Choices in Physical Simulation before proceeding.

The global Simulink solver in *Model Settings* applies to the entire model, including both the controller and plant sides. However, Simscape Electrical can decouple the plant model from this global setting by assigning a dedicated local fixed-step solver specifically for the plant side.

1. Global solver configuration

The *Imperix Control Library* strictly requires a variable-step solver because it contains continuous-time models that cannot be discretized. This is necessary to simulate the time delays of a real-time controller, such as the ADC sampling phase shift and the CPU computation cycle delay. These delays are usually fractional to the control task frequency and can be precisely represented in continuous time, whereas a discretized model limits time resolution to the fixed step size.



Start time: 0.0

Stop time: 1.4

Type: Variable-step

Solver: auto (Automatic solver selection)

Max step size: auto

Min step size: auto

Initial step size: auto

Shape preservation:

Number of consecutive min steps:

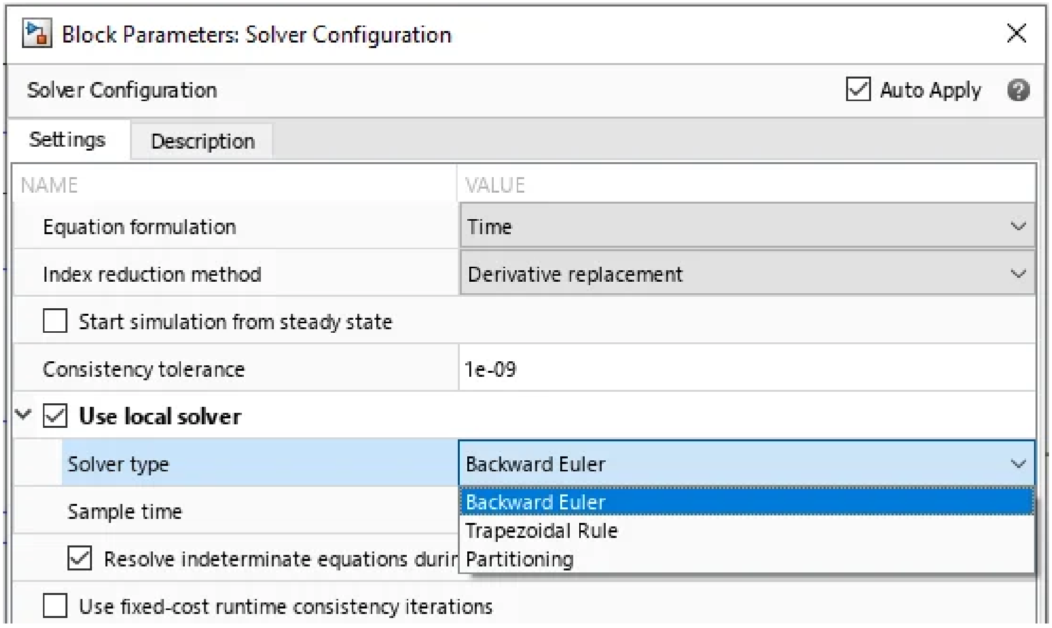
Global Simulink solver options

As documented by MathWorks, Simscape Electrical inherently generates stiff equations because of the nonlinear power electronics models it contains. The following variable-step solvers are therefore most common:

- `ode23t` / `ode23s`: the Simulink default stiff solvers, which perform well when the physical network *can* be reduced to standard Ordinary Differential Equations (ODEs). This occurs with ideal or averaged switches or loosely coupled networks. Thanks to its optimizations, models built with the *Imperix Power Library* generally fall into this category.
- `daessc`: designed to process the Differential-Algebraic Equations (DAEs) generated by Simscape. Although MathWorks promotes it for complex DAEs in large multi-domain networks, it is often overkill for an electrical model that reduces to ODEs. In power electronics, it is only useful together with non-ideal components that prevent ODE reduction, such as temperature-dependent semiconductor models or magnetic saturation in electrical machines.

Imperix recommendation: leave the solver on “auto.” The Simulink compiler profiles the model during initialization, evaluates the stiffness of the algebraic constraints, determines whether DAE-to-ODE reduction is possible, and selects the most efficient global solver.

2. Using the local solver for the plant side



NAME	VALUE
Equation formulation	Time
Index reduction method	Derivative replacement
☐ Start simulation from steady state	
Consistency tolerance	1e-09
☑ **Use local solver**	
Solver type	Backward Euler
Sample time	
☑ Resolve indeterminate equations durin	
☐ Use fixed-cost runtime consistency iterations	

Local Simscape solver options

As mentioned, Simscape supports a local fixed-step solver for the plant side even when the global solver is variable-step. For that, two types are available: Backward Euler and Trapezoidal Rule.

Backward Euler is the recommended default for its high numerical stability. It introduces slight artificial damping, making it robust against the high-frequency ringing and discontinuous transients typical of hard-switched converters. Alternatively, the Trapezoidal rule offers higher accuracy and preserves system energy without artificial damping, but may be prone to numerical oscillation with excessive time steps.

Imperix recommendation: there is no universal rule for when to use a local solver, as fixed-step performance depends heavily on step size.

The impact of step size will be analyzed in the next section. In the presented grid-tied PV inverter example, the maximum error tolerance for the DC bus voltage is set to 1%, which is often comparable to the measurement errors. Results show that a fixed step of 5% of the PWM switching period (5% PWM resolution) is a good starting point for balancing speed and accuracy.

Simulation speed is nearly identical for both solver types; the difference is stability. Start with Backward Euler and switch to trapezoidal rule only when highly accurate tracking of continuous AC oscillations is required, such as for resonant converters and heavily filtered continuous sinusoidal AC networks.

3. When to use the partitioning solver?

Besides the standard local types, Simscape supports a partitioning solver, which uses matrix-tearing techniques to decompose the global system matrix into independent partitions, thereby solving several small equations sequentially instead of a single large one.

Partitioning yields large speedups provided the topology supports such division. It works well in decoupled systems with forced commutation (e.g., standard PWM-driven inverters), as well as in loosely coupled networks (where subsystems are separated by sufficient impedance).

Conversely, it frequently fails or introduces numerical ringing in tightly interconnected circuits, particularly those relying on natural commutation (e.g., diode rectifiers and similar). It also struggles with shared energy nodes, such as when multiple phase-legs are tied directly to a shared DC bus capacitor without sufficient series resistance.

Imperix recommendation: disable partitioning for topologies relying on natural commutation, diode pre-charge sequences, and use the default Backward Euler solver instead. This evaluates the entire system holistically, guaranteeing numerical stability and physical accuracy during complex switching events at the cost of higher computation time.

Comparative results

The results below illustrate the improvements in simulation speed for the example of the Three-phase PV inverter for grid-tied applications.

Comparison 1: Simscape Electrical model with different configurations

Simulation environment

- **MATLAB version**: R2025b
- **Simulation stop time**: 1.4 s
- **Imperix Power library modeling level**: Detailed, unless otherwise stated
- **CLK0 frequency**: 20kHz (PWM period = 5e-5 s)

Simulation storyboard

- At t = 0, the DC bus is already precharged at 500V. The PWMs are not yet activated. The grid connection relays are closed, and the DC bus is charged by the grid via MOSFET body-diodes.
- At t = 0.1 s, the PWMs are activated. The DC bus voltage is regulated to 750V, and the MPPT algorithm gives the boost current reference.
- At t = 0.5 s, the DC voltage reference is reduced to 700V.
- At t = 0.8 s, the power extracted from the PV panel is reduced (simulation of passing clouds).

The following figures show the waveforms of the DC bus voltage and the error with reference to the original model during the simulation.

Results and conclusions

Accuracy is quantified through the percentage difference in results between a given configuration and the original baseline model.

Model configuration	**Simulation run time [s]**	**Normalized run time**
Original model (Detailed)	**112.9**	**1.00**
Simple	137.3	1.22

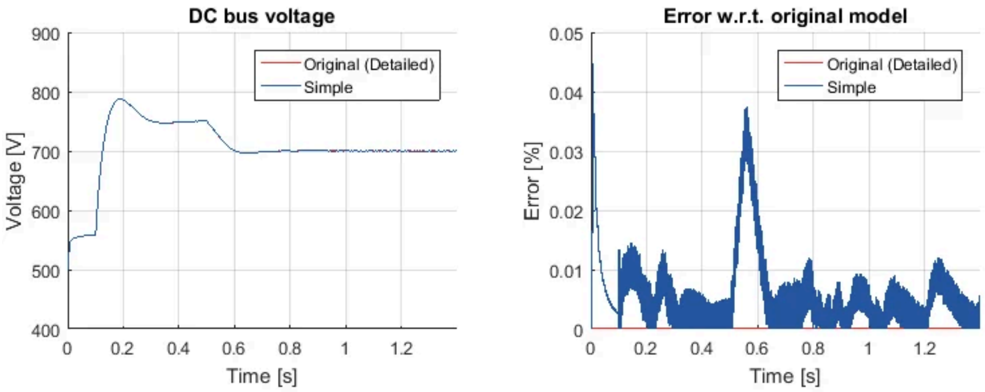


Figure 1: *Simple* model v.s. *Detailed* model

Figure 1 compares the simulation results between the *Simple* and *Detailed* modeling levels. The deviations are negligible (< 0.05%) at a standard 20 kHz switching frequency, but the Simple model is 22% slower because of the zero-ESR numerical issues. The *Detailed* level is therefore always recommended with the *Imperix Power Library* in Simscape Electrical.

Model configuration	**Simulation run time [s]**	**Normalized run time**
Original model (Switched)	**112.9**	**1.00**
Averaged	58.1	0.51

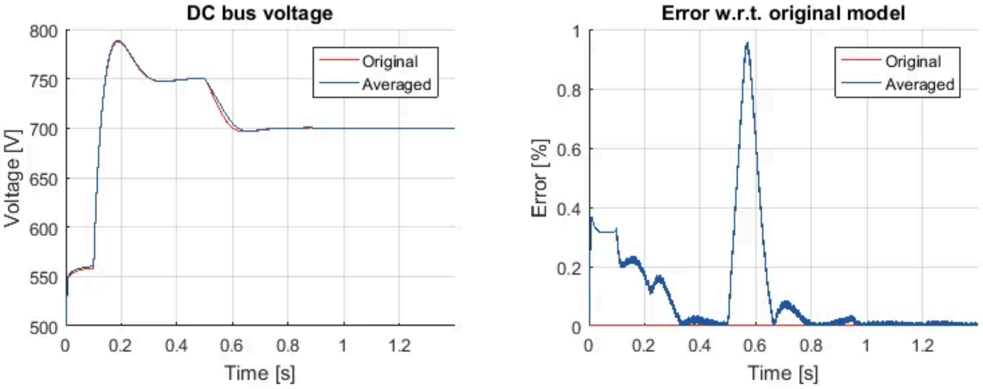


Figure 2: *Switched* model v.s. *Averaged* model

Figure 2 compares the simulation results between the *Switched* and *Averaged* models. As shown, the averaged model nearly halves simulation time while keeping the error below 1%. This is acceptable in most system-level applications as it is often comparable to the measurement accuracy. Using averaged models is therefore recommended whenever modeling switching ripple is not required.

Model configuration	**Simulation run time [s]**	**Normalized run time**
Original model	**112.9**	**1.00**
Local solver (Backward Euler) Sample time = 2.5e-5 s	12.5	0.11*
Local solver (Backward Euler) Sample time = 5e-6 s	23.5	0.21
Local solver (Backward Euler) Sample time = 2.5e-6 s	31.7	0.28
Local solver (Backward Euler) Sample time = 5e-7 s	101.2	0.90

* Result is invalid due to excessive errors

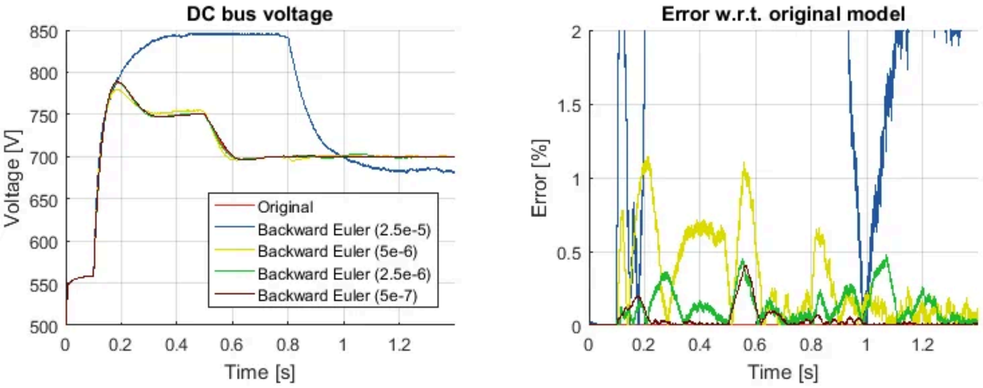


Figure 3: Backward Euler local solver with different step sizes

Figure 3 compares the simulation results using a local fixed-step solver with different step sizes, using Backward Euler as the baseline. Assuming a tolerable peak error of 1%, the optimal step falls between 5e-6 s and 2.5e-6 s, representing 5–10% of the PWM period. Outside this range:

- An excessively large step (2.5e-5 s, 50% of the PWM period) lacks the resolution to capture duty-cycle variations, giving invalid results.
- An excessively small step (5e-7 s, 1% of the PWM period) increases run time without a tangible precision gain.

A step of 5% of the PWM period is hence recommended as the optimal baseline with a fixed-step solver.

Model configuration	**Simulation run time [s]**	**Normalized run time**
Original model	**112.9**	**1.00**
Local solver (Backward Euler) Sample time = 2.5e-6 s	31.7	0.28
Local solver (Trapezoidal Rule) Sample time = 2.5e-6 s	31.1	0.28
Local solver (Partitioning) Sample time = 2.5e-6 s	16.0	0.14

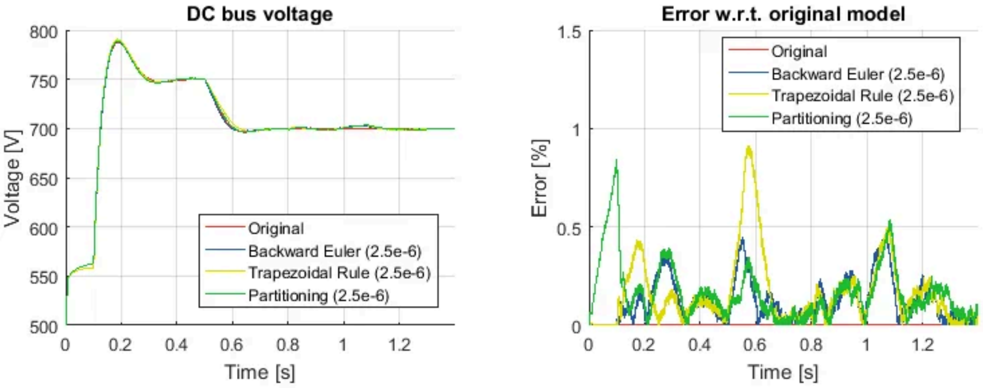


Figure 4: Different local fixed-step solvers

Figure 4 compares the simulation results across different local fixed-step solvers at a constant sample time of 2.5e-6 s. The partitioning solver gives the most aggressive speedup (86% reduction). Its peak error occurs during the AC pre-charge phase (0–0.1 s) due to cross-coupling of the uncontrolled body diodes. Additionally, thanks to the fact that each AC phase leg is sufficiently decoupled by the ESRs of DC bus capacitors and AC inductors, the error stays under 1%. The difference between Backward Euler and Trapezoidal Rule is negligible here: the evaluated signal is the slow DC bus voltage, and the system lacks undamped high-frequency LC resonant tanks, so Backward Euler’s artificial damping has little effect.

Comparison 2: Simscape Electrical versus other simulation tools

To provide a performance comparison with other simulation tools, the same PV inverter model was constructed using alternative simulation software.

As previously discussed, the modeling level has little impact on the accuracy for this specific application at 20 kHz. Therefore, the *Detailed* level is chosen for Simscape Electrical, while the *Simple* level is chosen for the other tools to optimize simulation speed.

Simulation environment

- **MATLAB version**: R2025b
- **PLECS version**: 5.0.1
- **Simulation stop time**: 1.4 s
- **CLK0 frequency**: 20kHz (PWM period = 50 us)

Results and conclusions

Model configuration	**Simulation run time [s]**	**Normalized run time**
Original Simscape Electrical model	**112.9**	**1.00**
Averaged Simscape Electrical model	58.1	0.51
Local solver (Backward Euler) Sample time = 2.5e-6 s	31.7	0.28
Local solver (Partitioning) Sample time = 2.5e-6 s	16.0	0.14
SPS model	25.9	0.23
PLECS model	9.7	0.09

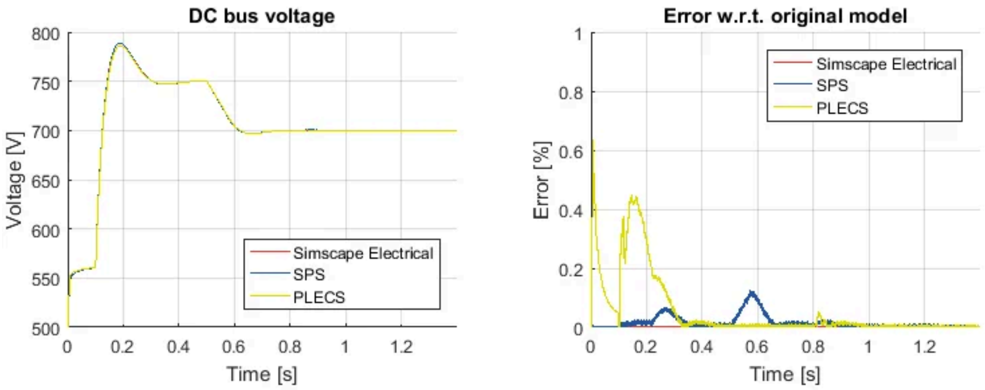


Figure 5: Different simulation tools

These results suggest the following remarks:

- First of all, all tools provide nearly identical results, within 1% error.
- Although MathWorks is transitioning users away from the deprecated SPS blockset toward native Simscape Electrical, the legacy SPS yields significantly better baseline performance than an unoptimized Simscape Electrical model using the same variable-step solver.
- Simscape Electrical can nonetheless be optimized to match or beat SPS, though this requires engineering compromises: first, using averaged switch models sacrifices the analysis of discrete modulator behavior and switching ripple. Second, using the discrete local partitioning solver gives large speedups at the risk of numerical inaccuracies or solver failures on tightly coupled topologies.
- PLECS Standalone shows unbeatable computational efficiency. As a dedicated power electronics engine, it operates without the overhead of multi-domain

physics solvers, running the identical model significantly faster than the Simulink-based alternatives.