Chapter 4
Coupling of Numeric/Symbolic Reduction Methods for Generating Parametrized Models of Nanoelectronic Systems

Oliver Schmidt, Matthias Hauser, and Patrick Lang

Abstract This chapter presents new strategies for the analysis and model order reduction of systems of ever-growing size and complexity by exploiting the hierarchical structure of analog electronical circuits. Thereby, the entire circuit is considered as a system of interconnected subcircuits. Given a prescribed error-bound for the reduction process, a newly developed algorithm tries to achieve a maximal reduction degree for the overall system by choosing the reduction degrees of the subcircuits in a convenient way. The individual subsystem reductions with respect to their prescribed error-bound are then performed using different reduction techniques. Combining the reduced subsystems a reduced model of the overall system results. Finally, the usability of the new techniques is demonstrated on two circuit examples typically used in industrial applications.

4.1 Introduction

In order to avoid immense time and financial effort for the production of deficiently designed prototypes of *integrated circuits* (ICs), industrial circuit design uses mathematical models and simulations for predicting and analysing the physical behavior of electronical systems. Hence, redesigns and modifications of the systems can easily be carried out on a computer screen and tested by subsequent simulation runs. Thereby, analog circuits in general are modelled by systems of *differential-algebraic equations* (DAEs), which are composed of component characteristics and Kirchhoff laws.

The development in fabrication technology of ICs during the last years led to an unprecedented increase of functionality of systems on a single chip. Nowadays, ICs have hundreds of millions of semiconductor devices arranged in several layers

O. Schmidt • M. Hauser • P. Lang (✉)
Fraunhofer Institute for Industrial Mathematics ITWM (Fraunhofer Institut für Techno-und Wirtschaftsmathematik ITWM), Fraunhofer-Platz 1, 67663 Kaiserslautern, Germany
http://www.itwm.fraunhofer.de/abteilungen/systemanalyse-prognose-und-regelung/elektronik-mechanik-mechatronik/analog-insydes.html

P. Benner (ed.), *System Reduction for Nanoscale IC Design*,
Mathematics in Industry 20, DOI 10.1007/978-3-319-07236-4_4

and low-level physical effects such as thermal interactions or electromagnetic radiation cannot be neglected anymore in order to guarantee a non-defective signal propagation. Mathematical models based on DAEs, however, have almost reached their limit and cannot model these effects accurately enough. Consequently, *distributed elements* for critical components such as semiconductor devices and transmission lines are used which yield supplementary model descriptions based on *partial differential equations* (PDEs), where also the spatial dependencies are taken into account. The coupling with DAEs modelling the remaining parts of the circuit then leads to systems of *partial differential-algebraic equations* (PDAEs). A spatial semidiscretization finally results in very high-dimensional systems of DAEs, thus rendering analysis and simulation tasks unacceptably expensive and time consuming.

Since design verification requires a large number of simulation runs with different input excitations, for the reasons mentioned above, *model order reduction* (MOR) becomes inevitable. Dedicated techniques in various areas of research have been developed among which the most popular ones are *numerical methods* taylored for linear systems. Besides these, there also exist *symbolic methods* [8, 10, 15, 19, 20], where *symbolic* means that besides the system's variables also its parameters are given as symbols instead of numerical values (see Sect. 4.1.1). They indeed are costly to compute, but allow deeper analytical insights into functional dependences of the system's linear and nonlinear behavior on its parameters by maintaining the dominant ones in their symbolic form. The basic idea behind these methods is a stepwise reduction of the original system by comparing its *reference solution* to the solution of the so far reduced system by using *error functions* which measure the difference between the two solutions. Since the resulting reduced system contains its parameters and variables in symbolic form, these methods can be seen as a kind of parametric model order reduction (pMOR). Compared to the standard parametric model order reduction techniques [4, 12], the symbolic ones can be additionally applied to nonlinear systems.

In order to avoid infeasibility of analysis and reduction of systems of ever-growing size and complexity, new strategies exploiting their hierarchical structure have been developed in the current research project. They further allow for a coupling of distinct reduction techniques for different parts of the entire circuits.

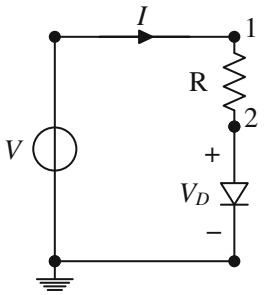
The corresponding algorithms have been implemented in *Analog Insydes* [1], the software tool for symbolic modeling and analysis of analog circuits, that is developed and distributed by the Fraunhofer ITWMin Kaiserslautern, Germany. It is based on the computer algebra system *Mathematica* [21].

The new approach has been successfully applied with significant savings in computation time to both a differential and an operational amplifier typically used in industry. The reduced models also proved to be very robust with regard to different inputs such as highly non-smooth pulse excitations. Thus, the aptitude of the new hierarchical model reduction algorithm to circuits of industrial size has been shown.

4.1.1 Symbolic Modeling of Analog Circuits

In the field of analog electronic circuits, there are different ways of modeling of the devices' behaviors. The approach *Analog Insydes* uses is the combination of Kirchhoff laws with symbolic device models to generate a symbolic system of differential-algebraic equations. As mentioned before, *symbolic* means here that besides the system's variables also its parameters are given as symbols instead of numerical values.

For a better understanding, consider the following circuit consisting of a voltage source V, a resistor R and a diode D.



$$\frac{V_1 - V_D}{R} = I$$

$$AREA \cdot I_S \left(e^{\frac{3.33167 \cdot 10^{-3} V_D \cdot q}{k}} - 1 \right) + G_{MIN} \cdot V_D = I$$

The resulting system of equations contains the following equations modeling the current of the circuit by using the resistor's and diode's model equations. Additional to the system variables, like V_1,V_D and I, the parameters R, $AREA$, I_S, k, q and G_{MIN} are also given as symbols. This allows, besides the simulation after inserting the symbol's values, to analyse this system symbolically. That means in this case, that we could just solve symbolically the system for the voltage in node 1 with respect to the parameters and the voltage at the diode:

$$V_1 = R \cdot \left(AREA \cdot I_S \left(e^{\frac{3.33167 \cdot 10^{-3} V_D \cdot q}{k}} - 1 \right) + G_{MIN} \cdot V_D \right) + V_D$$

The next section follows the notes of [16–18].

4.2 Hierarchical Modelling and Model Reduction

In general, electronic circuits consist of a coupling of blocks such as amplifiers, current mirrors, or polarization circuits. Each block itself might have such a structure or is at least a network of interconnected components like diodes, resistors, transistors, etc. Consequently, the entire circuit is a hierarchical network of interconnected subcircuits, where each of these subcircuits may be modelled differently, e.g. based on netlists, PDEs, or DAEs.

The main idea behind the new *algorithm for hierarchical reduction* developed is the exploitation of the circuit's hierarchical structure in order to perform different reduction techniques on the distinct subcircuits. Besides a suitable choice of the methods according to the modelling of the corresponding subcircuits, this further allows for a faster processing of smaller subproblems if the administrative cost does not get out of hand. Furthermore, particularly in the case of symbolic model order reduction methods, like used in *Analog Insydes*, larger circuits become manageable at all.

Standard graph theoretical methods such as the *modified nodal analysis* (MNA) for transforming a circuit into a system of describing equations, however, lose the structural information available at circuit level. Therefore, we developed a new workflow for separate reductions of single subcircuits in the entire system, which uses information obtained from a previous simulation run. Since, in general, there is no relation between the errors of single nonlinear subsystems and the entire system available, we further introduced a new concept of *subsystem sensitivities*. By keeping track of the error on the output, which is resulting from the simplification of the subsystem, the sensitivities are used to measure the influence of single subsystems on the behavior of the entire circuit. Finally, these sensitivities are used to compute a *ranking of subsystem reductions*. In order to obtain a high degree of reduction for the entire system, it allows to replace the subcircuits by appropriate reduced models in an heuristically reasonable order. The details are explained in the following sections.

4.2.1 Workflow for Subsystem Reductions

Assume an electronic circuit Σ to be already hierarchically segmented into a set of m subcircuits T_i and an interconnecting structure S:

$$\Sigma = (\{T_i \mid i = 1, \ldots, m\}, S)\,. \tag{4.1}$$

As already mentioned, each T_i itself might be recursively segmented into a set of subcircuits and a coupling structure. However, here we only consider a segmentation on the topmost "level 0". If one simply applies methods such as MNA to the circuit Σ in order to set up a set of describing equations, the resulting equations generally involve mixed terms from different subcircuits. In order to maintain the hierarchy information available on circuit level, in a first step the subcircuits are cut out from their connecting structure (cf. Fig. 4.1). Each subcircuit T is then connected to a test bench (a), i.e. a simulation test environment, where the voltage potentials at its terminals are recorded during a simulation run. For example, by simulating the original entire circuit, for each subcircuit T the interconnection of the remaining ones act as a test bench for T.

Note that the reduced model generated by the described method depends strongly on the input signals used. Thus, the input signal of the circuit has to cover the technical requirements of the later usage.

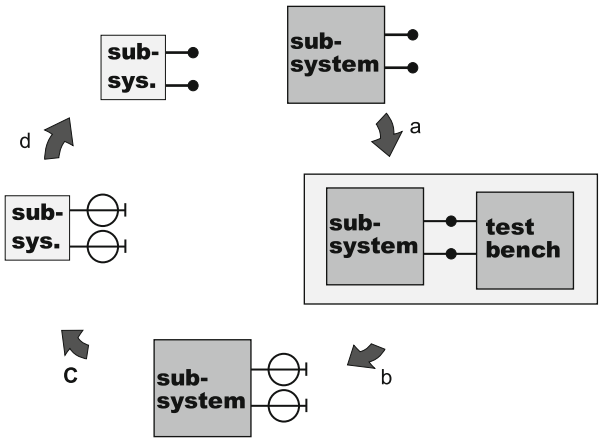


Fig. 4.1 Subsystem reduction via test bench approach

In a second step, the terminals of T are connected to voltage sources that generate exactly the recorded voltage potentials[1] (b). Hence, one has a closed circuit C_T with a defined input-output behavior at the terminals of T. A method such as MNA is used to set up a describing system F_T of equations[2] for C_T. Next, F_T can be reduced using arbitrary appropriate symbolic or numeric reduction techniques (c).

In a last step, the voltage sources at the terminals of the reduced model $\widetilde{F}_T$ are removed (d). Since the terminals of the subsystem are preserved during the reduction process, the original subcircuit T in Σ can easily be replaced by the reduced model $\widetilde{F}_T$ of F_T, thus using the same interconnecting structure S as introduced in (4.1). The entire procedure is repeated several times for each subcircuit T_i in Σ, thus yielding collections of reduced models for each T_i. The whole workflow is summarized in Algorithm 4.1.

It should further be mentioned here that this approach only controls the errors at the terminals of the single subcircuits. A priori, one cannot guarantee a certain *global error*, i.e. the error on the output of the entire circuit Σ, when replacing the original subcircuits T_i by reduced models $\widetilde{F}_{T_i}$. Thus the following algorithms were introduced to control the *global error* during the process.

[1] For doing it best, we first have to determine the voltage and current sources of the circuit that can act as inputs. Thus, the corresponding independent value of each port has to be considered as output. If you connect a voltage source at a port p this would be the current through port p, and vice versa.

For simplicity, we use here voltage sources as inputs and the currents as outputs. Besides of that, it turns out that residual based solvers simulate analog circuits containing transistors faster and more accurate if the voltages are given at the circuit's ports instead of the currents.

[2] Assume we are dealing with systems of DAEs. If PDEs are involved, apply a semidiscretization w.r.t. the spatial coordinates.

Algorithm 4.1 Reduction of subcircuits

Let $T = T_i$ be a subcircuit in an electronic circuit $\Sigma = (\{T_i \mid i = 1, \ldots, m\}, S)$.

a. Connect T to a test bench and record the voltage potentials at its terminals during a simulation run applying a suitable input.

b. Remove the test bench and connect grounded voltage sources to the terminals of T that generate exactly the recorded voltage potentials, thus having T isolated as a closed circuit C_T; further, set up a describing system of equations F_T for C_T.

c. Reduce F_T by using appropriate symbolic or numerical reduction techniques, where the voltages at all terminals of C_T are the inputs and the currents (flowing inwards) are the outputs. Here a family of reduced subsystems with different size and approximation quality is generated.

d. Remove the voltage sources at the terminals after the reduction and finally obtain a family of reduced subsystems, where each reduced subsystem $\widetilde{F}_T$ serves as a behavioral model of T.

4.2.2 Subsystem Sensitivities

In general, there is no relation between the error of the entire system and those of its nonlinear subsystems known. Therefore, in order to use reduced models of appropriate degree for the subsystems, in this section, we investigate the influence of single subcircuits T_i on the behavior of the entire circuit Σ given by (4.1). This offers a high degree of reduction also for Σ.

The goal here is to have an estimate of a subcircuit's *sensitivity*, i.e. the sensitivity of Σ with respect to changes in the corresponding subcircuit's behavior. Our novel approach measures the sensitivity by observing the influence of subcircuit reductions on the output of Σ and finally leads to a *ranking of subcircuit reductions*, i.e. an heuristically optimized order of subcircuit reductions.

Usually, the term *sensitivity analysis* in the background of electronic circuits means the influences of single components or system parameters on certain circuit or network variables. In that case, the *absolute sensitivity* of a variable z w.r.t. changes in a network parameter p is defined by

$$s_a(z, p) = \left.\frac{\partial z}{\partial p}\right|_{p=p_0}, \tag{4.2}$$

whereas

$$s_r(z, p) = p \left.\frac{\partial z}{\partial p}\right|_{p=p_0} = p \cdot s_a(z, p) \tag{4.3}$$

is the *relative sensitivity* of z w.r.t. p. In the two equations above, p_0 is the nominal value of p. Note that

$$s_a(z,p) \approx \left.\frac{\Delta z}{\Delta p}\right|_{p=p_0} = \frac{z-\widetilde{z}}{p_0-\widetilde{p}} \tag{4.4}$$

is an approximation of s_a using perturbed values $\widetilde{z} = z(\widetilde{p})$ and $\widetilde{p}$ of $z = z(p)$ and $p = p_0$. While $z = z(p_0)$ corresponds to a simulation of Σ using the parameter $p = p_0$, $\widetilde{z}$ is obtained by using the perturbed parameter $p = \widetilde{p}$ during the simulation run.

Since we cannot derive the output y of Σ w.r.t. one of its subcircuits, we imitate the meaning of Eq. (4.4) by replacing a single subcircuit T in (4.1) by a perturbed version $\widetilde{T}$, i.e. by a reduced model $\widetilde{F}_T$ of its describing system of equations. Note that any other subsystem in Σ remains original, only T is replaced by one of its reduced models. We then simulate the configuration of Σ at hand and compare the *original* output y, i.e. the reference solution, to the perturbed entire system's output $\widetilde{y}$.

By Definition 4.2.1, the sensitivity of the subcircuit T in Σ is defined as the vector of tuples containing the reduced models and the resulting error on the perturbed entire system. For simplicity, we will not distinguish between sub*circuits* and the corresponding describing sub*systems* based on equations and denote both of them simply by T.

Definition 4.2.1 Let $\Sigma = (\{T_i \mid i = 1,\ldots,m\}, S)$ be an electronic circuit of interconnected subcircuits T_i connected by a structure S. Let further $T = T_i$ be one of the subcircuits in Σ. The ***sensitivity of T in*** Σ is the vector

$$s_T = \left((\widetilde{T}^{(1)}, E(y, y_{\widetilde{T}^{(1)}})), \ldots, (\widetilde{T}^{(m_T)}, E(y, y_{\widetilde{T}^{(m_T)}}))\right) \tag{4.5}$$

that contains tuples of ***reduced models*** $\widetilde{T}^{(j)}$ for T and the resulting error $E(y, y_{\widetilde{T}^{(j)}})$ on the original output y of Σ. In this notation, $y_{\widetilde{T}^{(j)}}$ is the output of the corresponding system

$$\Sigma_{\widetilde{T}^{(j)}} = \left(\{\widetilde{T}^{(j)}\} \cup \{T_i \mid i = 1,\ldots,m\} \setminus \{T\}, S\right), \tag{4.6}$$

where T in comparison to the original circuit Σ is replaced by its jth reduced model $\widetilde{T}^{(j)}$.

In this definition, $\widetilde{T}^{(j)}$ denotes the jth reduced model of T which could be obtained by nonlinear symbolic model order reduction and an accepted error of 10% or by Arnoldi method and k iteration steps for example.

Note that the sensitivity of T involves systems $\Sigma_{\widetilde{T}^{(j)}}$ which are the same as Σ itself except for *exactly one subsystem*, namely T, that is replaced by a reduced version $\widetilde{T}^{(j)}$. Note further that these sensitivities depend again on the chosen input signals, as for the method introduced in Sect. 4.2.1.

Remarks 4.2.2 The sensitivity notion in Definition 4.2.1 can be further augmented by replacing the corresponding error $E(y, y_{\widetilde{T}^{(j)}})$ by a more general ranking expression that takes also additional subsystem criteria, like system size and sparsity, into account [9].

The next section describes how to use these sensitivities in order to obtain an heuristically reasonable order of subsystem reductions for the derivation of a system, that consists of reduced subsystems. Basically, the entries of the sensitivity vector of each subsystem are ordered increasingly with respect to the error on y. Then, following this order, the corresponding reduced models are used to replace the subsystems in Σ.

4.2.3 Subsystem Ranking

In this section, we present a strategy that allows an appropriate replacement of the subsystems of Σ by their reduced models in a reasonable order. The new algorithm presented here uses a ranking for deriving a hierarchically reduced model of the entire system Σ.

The basic idea behind the algorithm is ordering the reduced models of each subsystem increasingly w.r.t. the error[3] on the output y of Σ and subsequently performing the subsystem replacements according to this order. After each replacement, the accumulated error of the current subsystem configuration is checked by a simulation. If the user-given error bound ε for the error of the entire system Σ is exceeded, the current replacement is undone and the tested reduced model is deleted. Otherwise, the next replacement is performed and the procedure is repeated.

Let $\widetilde{T}_i^{(j)}$ denote the jth reduced model of the subsystem T_i. For each T_i in Σ we define a vector L_i which contains the entries of s_{T_i} and is increasingly ordered with respect to the error $E(y, y_{\widetilde{T}_i^{(j)}})$. The *original* subsystems T_i of Σ are then initialized by $\widetilde{T}_i^{(0)}$. In each iteration of the hierarchical reduction algorithm, the subsystem $\widetilde{T}_p^{(q)}$ that corresponds to the minimum entry[4] of the vectors L_i replaces the current (reduced) model $\widetilde{T}_p^{(q_0)}$ that is used for T_p in Σ. If the resulting accumulated error on the output y of Σ exceeds the user-specified error bound ε, the corresponding latest subsystem replacement is undone, i.e. $\widetilde{T}_p^{(q)}$ is reset to $\widetilde{T}_p^{(q_0)}$ in Σ. Furthermore, all reduced subsystems of subsystem T_p are deleted, since we assume that worse rated subsystems would also exceed the error bound. Otherwise only the corresponding sensitivity value $(\widetilde{T}_p{}^{(q)}, E(y, y_{\widetilde{T}_p^{(q)}}))$ of the tested reduced subsystem $\widetilde{T}_p^{(q)}$ is deleted from the vector L_p. This procedure is repeated until all the vectors L_i are empty. For a better overview of this approach see Algorithm 4.2.

[3] See Remarks 4.2.2.

[4] Minimal with respect to the corresponding error $E(y, y_{\widetilde{T}_i^{(j)}})$.

Algorithm 4.2 Heuristically reasonable order of subsystem replacements

Input: segmented electronic circuit $\Sigma = \big(\{T_i \mid i = 1, \ldots, m\}, S\big)$, input u, error bound ε

Output: reduced entire system $\widetilde{\Sigma} = \big(\{\widetilde{T}_i^{(j^*)} \mid i = 1, \ldots, m\}, S\big)$, where $\widetilde{T}_i^{(j^*)}$ are suitably reduced subsystems, $E(y, y_{\widetilde{\Sigma}}) \leq \varepsilon$, and where $y_{\widetilde{\Sigma}}$ is the output of $\widetilde{\Sigma}$

1: **for all** subsystems T_i **do**
2: $\quad L_i := \text{order}(s_{T_i})$ w.r.t. $E(y, y_{\widetilde{T}_i^{(j)}})$
3: $\quad \widetilde{T}_i^{(0)} := T_i$
4: **end for**

5: $L := (L_1, \ldots, L_m)$ ▷ set starting point
6: $\widetilde{\Sigma} := \Sigma$

7: $y := \text{solve}(\Sigma, u)$ ▷ calculate reference

8: **while** $L = \emptyset$ **do**
9: $\quad$ compute $(\widetilde{T}_p^{(q)}, E(y, y_{\widetilde{T}_p^{(q)}})) := \min_{i, L_i \in L}(\min(L_i))$ w.r.t. $E(y, y_{\widetilde{T}_i^{(j)}})$ ▷ choose reduced subsystem
10: $\quad$ replace current $\widetilde{T}_p^{(q_0)}$ by $\widetilde{T}_p^{(q)}$
11: $\quad$ update$(\widetilde{\Sigma})$ ▷ update and solve new reduced overall system
12: $\quad y_{\widetilde{\Sigma}} := \text{solve}(\widetilde{\Sigma}, u)$
13: $\quad \varepsilon_{\text{out}} := E(y, y_{\widetilde{\Sigma}})$
14: $\quad$ delete[5] entry $(\widetilde{T}_p^{(q)}, E(y, y_{\widetilde{T}_p^{(q)}}))$ in L_p

15: $\quad$ **if** $\varepsilon_{\text{out}} \leq \varepsilon$ **then** ▷ check resulting error
16: $\quad\quad$ **if** *dimension*$(L_p) = 0$ **then**
17: $\quad\quad\quad$ delete[5] entry L_p in L
18: $\quad\quad$ **end if**
19: $\quad$ **else**
20: $\quad\quad$ reset $\widetilde{T}_p^{(q)}$ to $\widetilde{T}_p^{(q_0)}$ ▷ undo reduction if error exceeds error bound
21: $\quad\quad$ update$(\widetilde{\Sigma})$
22: $\quad\quad$ delete[5] entry L_p in L
23: $\quad$ **end if**

24: **end while**

Remarks 4.2.3 Note that Algorithm 4.2 can further be improved, e.g. by a clustering of subsystem replacements, where reduced models that cause a similar error on y are bundled in a cluster. Thus, costly multiple simulations for computing the solution $\widetilde{y}$ of the so far reduced entire system $\widetilde{\Sigma}$ are avoided, since they are performed only once after a whole cluster of subsystem replacements is executed. In case the error bound is still not violated, we can continue with the next cluster of subsystem

[5] For a vector $X = (x_1, \ldots, x_n)$, deleting the entry x_i in X means, that a vector $\tilde{X} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)$ of dimension $n - 1$ results.

replacements. Otherwise, however, all replacements in the current cluster have to be rejected and it has to be subdivided for further processing.

Another idea for further improvements is the use of approximate simulations such as *k-step solvers* which quit the Newton iteration for computing the system's solution after k steps. Thus, one obtains an approximate solution $\widehat{y} \approx \widetilde{y}$ for the output of the so far reduced system $\widetilde{\Sigma}$ which can be used for the error check $E(y, \widehat{y}) \leq \varepsilon$ instead of $\widetilde{y}$.

4.2.4 Algorithm for Hierarchical Model Reduction

To combine all the considerations of the preceding sections, the *algorithm for hierarchical model reduction* exploiting the hierarchical structure of electronic circuits is set up. It is schematically shown in Fig. 4.2.

Remarks 4.2.4 Since electronic circuits even nowadays are designed in a modular way using building blocks of network devices and substructures such as current mirrors and amplifying stages, the hierarchical segmentation of an electronic circuit is given in a more or less natural way. Otherwise, the segmentation has to be made manually or by using *pattern matching* approaches[13] in order to detect substructures in the entire circuit.

Note that the presented algorithm (cf. Fig. 4.2) can be applied recursively to the subcircuit levels such that a hierarchically model order reduction results.

4.3 Implementations

The algorithms of the preceding sections have been completely implemented in *Analog Insydes* [1] and the approach for hierarchical model reduction was fully automated. It is divided into three main procedures

- `ReduceSubcircuits`,
- `SensitivityAnalysis`, and
- `HierarchicalReduction`

that have to be executed sequentially. Each of the above procedures takes several arguments among which there are some optional ones.

`ReduceSubcircuits` is called with the specification of an already segmented netlist of the circuit which is to be hierarchically reduced, the specification of the reduction method for each subcircuit, the simulation time interval necessary for recording the voltage potentials at the ports of the subcircuits, and several optional parameters. In accordance with the provided data, the procedure then computes the reduced models for all the specified subcircuits and appends them to the original

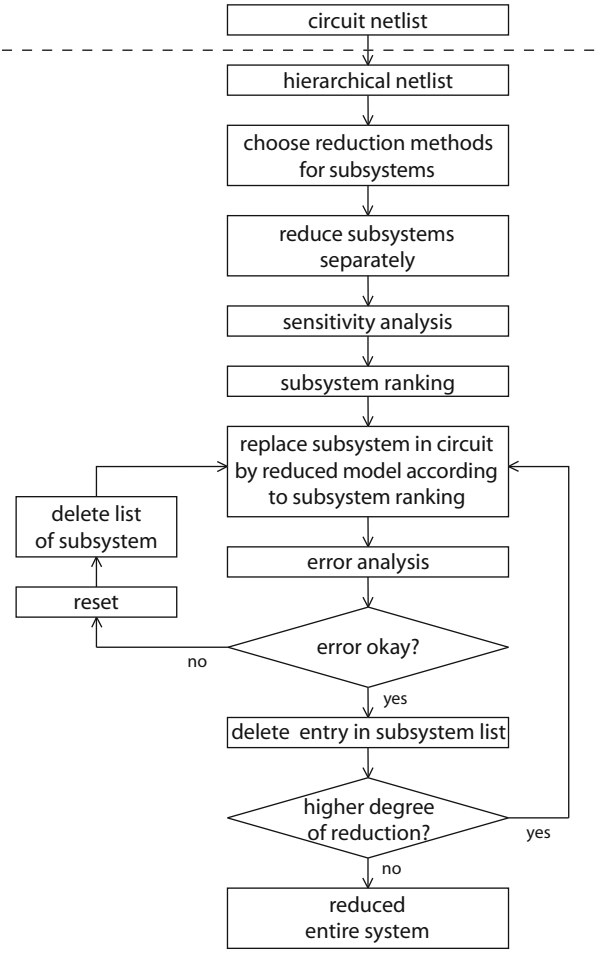


Fig. 4.2 Schematic illustration of the full algorithm for hierarchical model reduction using subsystem sensitivities.

circuit object. This offers an easy switching among the respective models for a single subcircuit.

The return value of `ReduceSubcircuits`, i.e. the hierarchically segmented circuit object together with the reduced models of each subcircuit, is then used as parameter of the function `SensitivityAnalysis`. In addition, the names of the reduced models, a specification of the output variables, the simulation time interval for the error check, and the error function itself to measure the error on the reference solution y are provided. The procedure computes the sensitivity vectors of each subcircuit and returns them ordered increasingly w.r.t. the error on y.

Finally, `HierarchicalReduction` needs a specification of the entire circuit and its reduced subcircuit models, the global error bound, the output variables, the

sensitivities returned by `SensitivityAnalysis`, the simulation time interval necessary for the error check, and several optional arguments. Then the subsystem replacements are performed according to the sensitivities and the accumulated error is checked after each replacement (Algorithm 4.2). The procedure terminates when all sensitivity lists have been processed and deleted.

In addition to the above, there have been implemented several data structures and operators for their manipulation, as well as some well-known reduction algorithms, transmission line models—based on a discretization of a PDE model—and further components based on general state space systems. We further implemented some environments to test the above procedures and functionalities. However, we will not go into detail here, for an overview we refer to [16].

4.4 Applications

In order to demonstrate the large potential of the new hierarchical reduction approach, it is applied in time domain to two analog circuit examples that are typical representants of components used in industrial circuit design. The results of the hierarchical reduction of the two circuits are compared to the direct non-hierarchical approach. Furthermore, some additional input excitations are applied to the circuits in order to show the robustness of the derived reduced models.

Note that we present here the application of the introduced methods on circuits containing strongly nonlinear devices to demonstrate the ability of the approach in the field of nonlinear analog circuits.

4.4.1 Differential Amplifier

The differential-amplifier circuit shown in Fig. 4.3 consists of five subcircuits DUT, DUT 2, L 1, L 8, and L 9, where the latter three ones are transmission lines connecting the supply voltage sources `VCC` and `VEE` and the input voltage source `V1` with the remaining parts of the circuit. For the modelling of the transmission lines, we take a discretized PDE model, namely, the telegrapher's equations (cf., e.g., [5–7, 11]), with 20 line segments each. While `VCC` and `VEE` generate constant voltage potentials of 12 V and −12 V, respectively, the input voltage generated by `V1` is a sine wave excitation with an amplitude of 2 V and a frequency of 100 kHz. Finally, the computations are performed on a time interval $\mathbb{I} = [0.\,\mathrm{s}, 10^{-5}\,\mathrm{s}]$.

Using MNA to set up a system of describing DAEs yields 167 equations containing 645 terms (on "level 0"). A *non-hierarchical* symbolic reduction of the entire system then needs approximately 2 h and 11 min,[6] where most of that time

[6]The computations are performed on a Dual Quad Xeon E5420 with 2.5 MHz and 16 GB RAM.

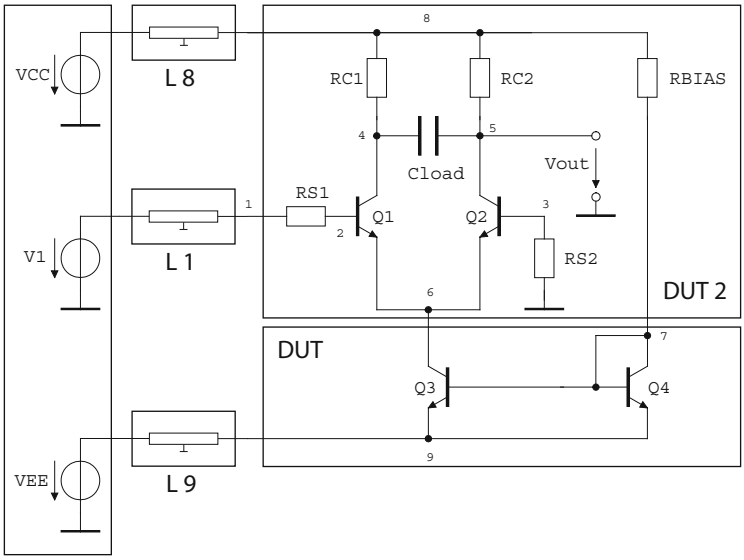


Fig. 4.3 Differential amplifier with its intuitive hierarchical segmentation into five subcircuits DUT, DUT 2, L 1, L 8, and L 9.

(≈95%) is needed for the computation of the transient *term ranking*.[7] Due to this, the computational costs are approximately the same for all choices of the error bound ε. The error function used first discretizes the time interval $\mathbb{I}$ to a uniform grid of 100 points and then takes the maximum absolute difference of the two solutions on this grid as a measure for the error.

With ε equal to 3% the system is reduced to 124 equations and 416 terms, while a permitted error of 10% narrows these numbers down to 44 equations and 284 terms. The results are shown in Fig. 4.4. Note also that the error bound of 10% is fully exploited.

In contrast to the immense time costs of the non-hierarchical approach, the new algorithm for hierarchical reduction reduces the entire system in only 4 min and 50 s. The subcircuits DUT and DUT 2 are reduced symbolically by using a sweep of error bounds

$$sw = \{1\%,\ 10\%,\ 50\%,\ 90\%,\ 100\%\}, \tag{4.7}$$

such that each subsystem yields 5 reduced subsystems. The three transmission lines L 1, L 8, and L 9 are reduced numerically by applying Arnoldi's algorithm [2, 3].

[7] A term ranking is a trade-off between accuracy and efficiency in computation time that estimates the influence of a term in a system of equations on its solution. Here, however, we use full simulations instead of low-accuracy estimates. For more details see [20].

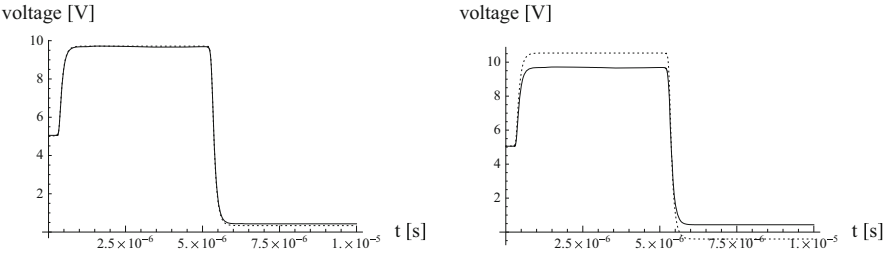


Fig. 4.4 Solution of the original (*solid*) and the non-hierarchically reduced system (*dotted*) allowing 3% (*left*) and 10% (*right*) maximum error, respectively. The input `V1` is $2 \cdot Sin(2\pi 10^5 t)$ Volts

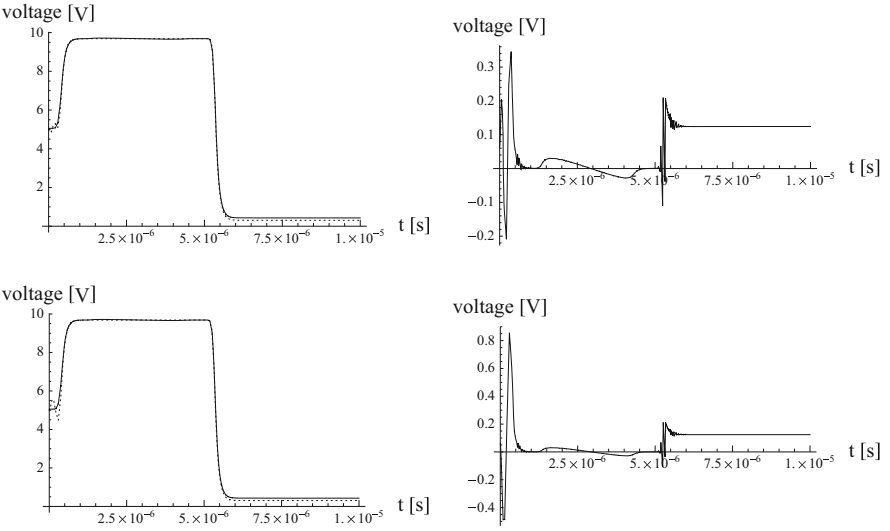


Fig. 4.5 *Left:* Solution of the original (*solid*) and the reduced system (*dotted*) allowing 3% (first row) and 10% (second row) maximum error, respectively. *Right:* The corresponding error plots. The input `V1` is $2 \cdot Sin(2\pi 10^5 t)$ Volts

For L 1 there are five reduced models computed by performing the Arnoldi iteration for up to 5 steps, and for L 8 and L 9 there are made only up to 3 steps, thus yielding three reduced models each for L 8, and L 9.

For $\varepsilon = 3\%$ the resulting reduced overall system contains 62 equations with 315 terms, and $\varepsilon = 10\%$ leads to a reduced overall system with 60 equations and 249 terms. The solutions of the original and the respective reduced systems are shown in Fig. 4.5 together with the corresponding error plots.

In this case we conclude that the hierarchical reduction approach is more than 26 times faster than the non-hierarchical one. Also the number of equations of the reduced model in the 3% error case could be halved. Moreover, by applying further input excitations to both the original and the hierarchically reduced system with

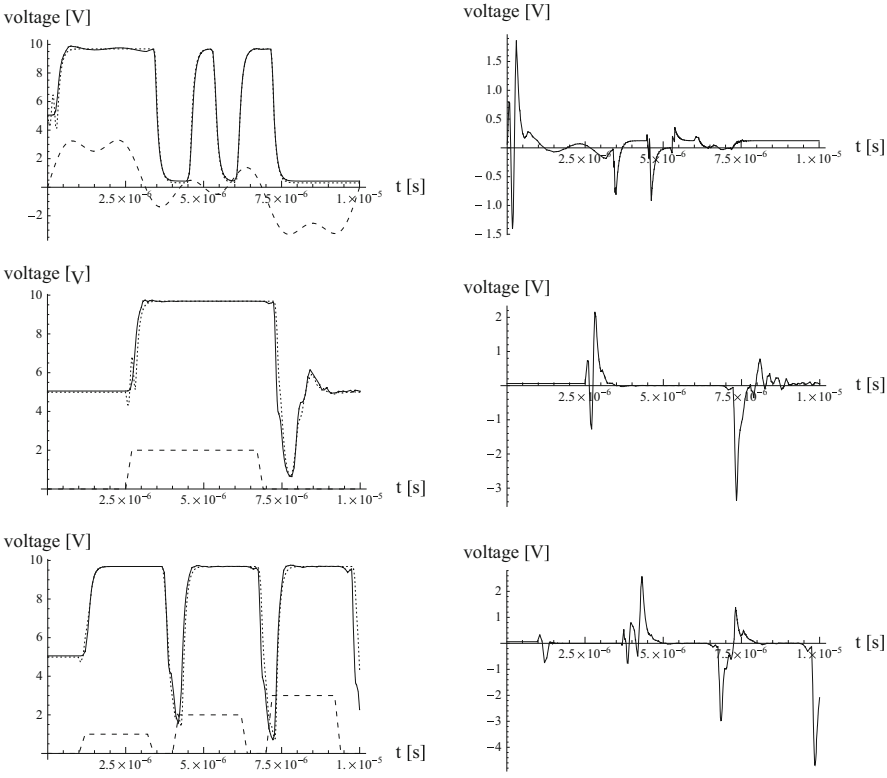


Fig. 4.6 *Left:* Solution of the original (*solid*) and the reduced system (*dotted*, $\varepsilon = 3\%$) together with the input excitation (*dashed*). *Right:* The corresponding error plots

$\varepsilon = 3\%$, it turns out that the derived model is very robust, even w.r.t. highly non-smooth pulse excitations (cf. Fig. 4.6). Note further that the simulation is accelerated approximately by a factor of 5.

4.4.2 *Reduction of the Transmission Line* L1 *by Using an Adapted PABTEC Algorithm*

The tool *PABTEC* [14] uses the Balanced Truncation reduction technique to reduce the linear parts of an analog circuit. Please refer to Chap. 2.6 for further informations about this software.

To demonstrate the coupling of the introduced algorithm with a numeric model order reduction method, we use *PABTEC* to reduce the linear transmission line L1. The remaining subcircuits DUT, DUT2, L8, and L9 have been reduced by the same methods shown in the example before. In doing so, the original entire system

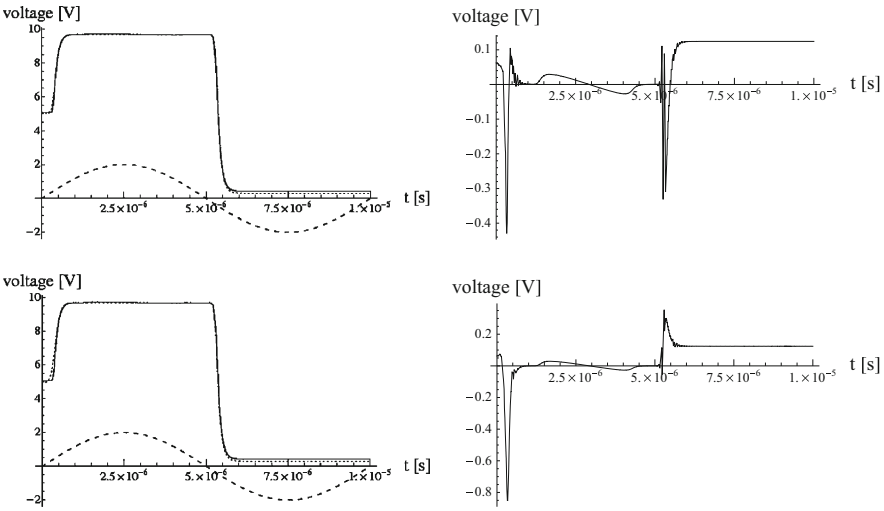


Fig. 4.7 *Left:* Solution of the original (*solid*) and the reduced system (*dotted*) together with the input excitation (*dashed*). *Right:* The corresponding error plots. The first row corresponds to the reduced system obtained by allowing an error of $\varepsilon = 3\%$, while the second row shows the results for $\varepsilon = 10\%$. The input `V1` is $2 \cdot Sin(2\pi 10^5 t)$ Volts

consists of 191 equations containing 695 terms. Applying the hierarchical reduction algorithm with error bounds $\varepsilon = 3\%$ and $\varepsilon = 10\%$ then needs about 8 min and 20 s and yields systems with 96 equations and 2114 terms and 84 equations and 1190 terms, respectively. The results of their simulation (speed-up by a factor of approximately 5) are shown in Fig. 4.7.

4.4.3 Operational Amplifier

The second circuit example to which we apply the new algorithms is the operational amplifier op741 shown in Fig. 4.8. It contains 26 bipolar junction transistors (BJT) besides several linear components and is hierarchically segmented into seven subcircuits CM1–3, DP, DAR, LS, and PP. For a detailed description of their functionality in the interconnecting structure we refer to [16, Appendix C].

The goal is a *symbolic* reduction of the entire circuit in time domain with an overall error bound of $\varepsilon = 10\%$. While the input voltage source `Vid` provides a sine wave excitation of 0.8 V and 1 kHz frequency on a time interval $\mathbb{I} = [0\,\text{s},\ 0.002\,\text{s}]$ to the system, its output is specified by the voltage potential of node 26. The input together with the corresponding output, i.e. the *reference solution*, is shown in Fig. 4.9. Note that the reference solution is pulse-shaped and, thus, the *standard* error function used for the differential amplifier in the preceding sections may lead to large errors for small delays in jumps of the solution. Hence, even with a

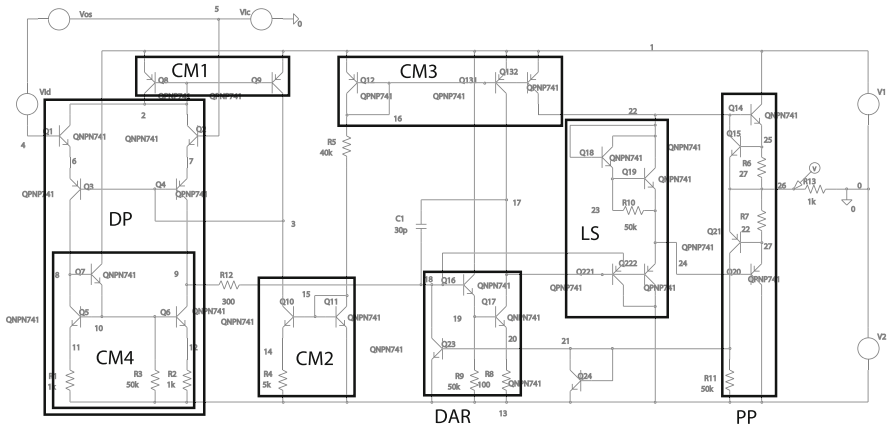


Fig. 4.8 Operational amplifier op741 composed of seven subcircuits CM1–3, DP, DAR, LS, PP

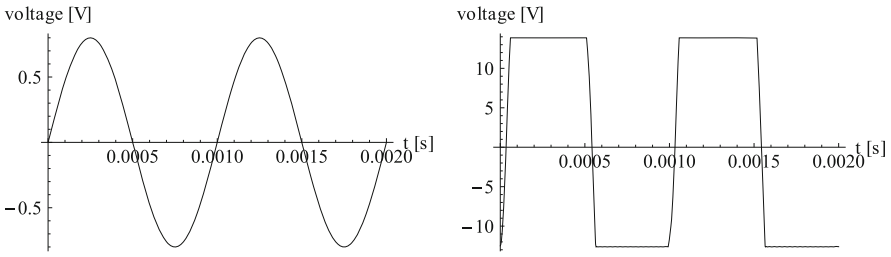


Fig. 4.9 Input voltage excitation (*left*) and the corresponding reference solution (*right*) of the operational amplifier op741

prescribed error bound of 10%, the system might not be reduced at all. In order to cope with these problems, here we use the $\mathscr{L}^2$-norm as error function.

Using MNA to set up a system of describing DAEs for the entire system yields 215 equations and 1050 terms. The direct non-hierarchical symbolic reduction method needs more than 10.5 h and yields a system containing 97 equations and 593 terms. At the same time, providing a sweep of error bounds

$$sw = \{2\%, 10\%, 20\%, 30\%, 50\%, 70\%, 90\%, 100\%\} \tag{4.8}$$

for the separate symbolic reduction of all seven subcircuits and applying the hierarchical reduction algorithm needs only 2 h and 22 min. The resulting system, however, consists of 153 equations and 464 terms, which can be narrowed down to

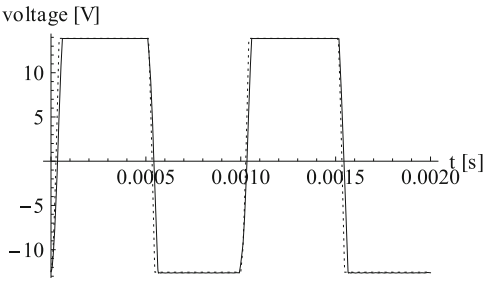


Fig. 4.10 Output of the original (*solid*) and the hybrid reduced entire system (*dotted*)

Table 4.1 Overview of the results of the reduction of the operational amplifier op741

Original system:		215 equations, 1050 terms, 26.0 s simulation time		
Error fct.		Non-hierarchical	Hierarchical	Hybrid
$\mathscr{L}^2$-norm	Time costs	10.5 h	2.5 h	<4 h
	Equations/terms	97/593	139/362	34/92
	Error	2.51%	7.16%	5.68%
	Simulation time	16.0 s	11.4 s	2.2 s
E^*	Time costs	>12 h	2.5 h	<4 h
	Equations/terms	80/405	132/336	34/93
	Error	0.37%	0.08%	5.32%
	Simulation time	9.5 s	13.1 s	2.0 s

The computations were performed on a machine with 8 Quad-core AMD Opteron 8384 "Shanghai" (32 cores in total) with 2.7 GHz and 512 GB RAM on a SuSE Linux 10.1 system

139 equations and 362 terms by slight manual improvements[8] of the hierarchical reduction algorithm.

Considering the obtained systems as *interim solutions* and applying a second non-hierarchical symbolic reduction then reduces the size drastically and leads to a model with only 34 equations and 92 terms. Simultaneously, there are almost no further changes for the non-hierarchically reduced system with 97 equations. Note that the additional time cost is less then 1.5 h, while the simulation time of the "*hybrid*" reduced model is significantly decreased.

Figure 4.10 offers a qualitative impression of the results obtained by the *hybrid* approach. Furthermore, earlier results involved a newly designed alternative error function E^* which is less sensitive with respect to small delays in jumps of the system's solution.

Table 4.1 provides an overview of the best results obtained by the three different approaches. See also Fig. 4.11 which offers some details about the accuracy, time

[8] Due to the structure preserving reduction method, the resulting reduced model contains equations connecting the models of the subcircuits, that can be avoided, like: Voltage of node 24 of subcircuit LS is equal to the voltage of node 24 of subcircuit PP.

Unifying the corresponding variables (i.e. *V*24*LS* and *V*24*PP*) yields a decrease of the number of equations.

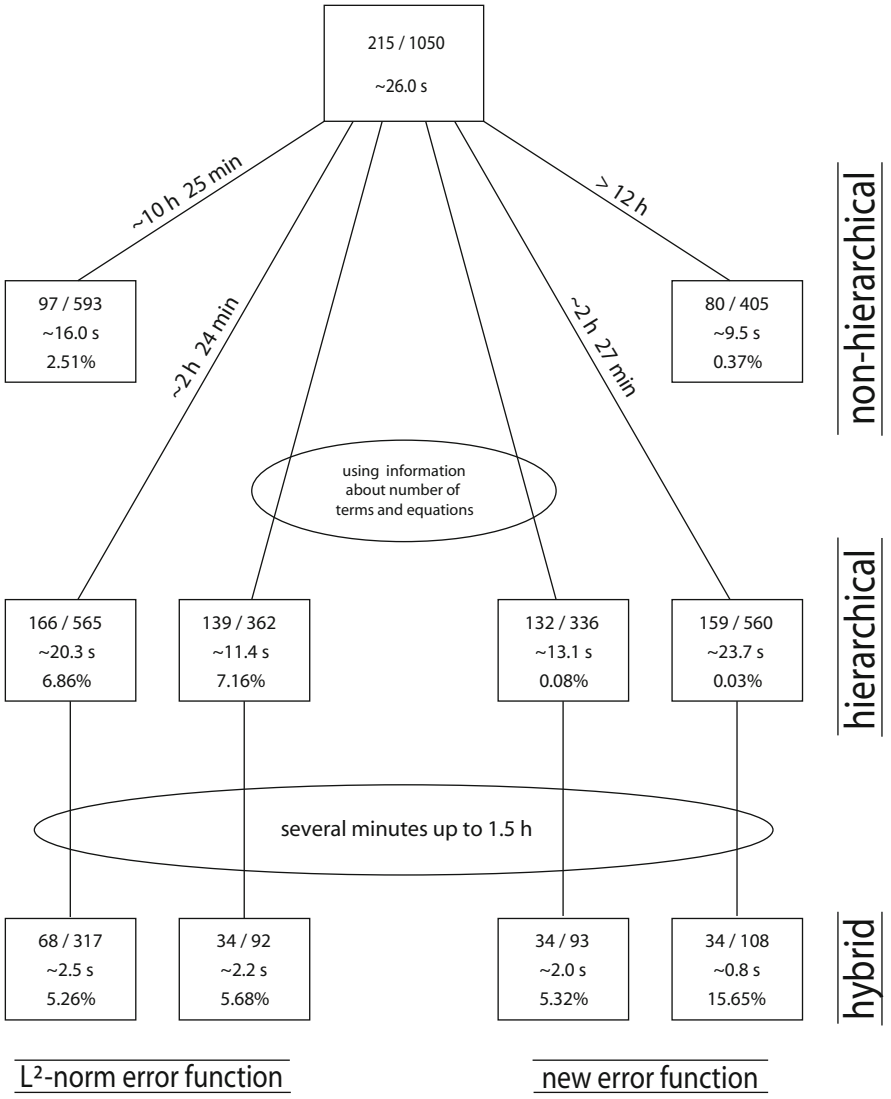


Fig. 4.11 Summary of the reduced models of the op741 amplifier obtained by the three different reduction approaches. The *boxes* contain the number of equations/terms of the reduced models, the time costs of a simulation using the original sine wave excitation, and the error on the output `V$26` of the original amplifier

costs for simulation, and number of equations and terms of the different reduced models. We will not go into detail here, for further information we refer to [16] instead.

With a view towards the robustness of the derived models, we apply some further input excitations, namely, a sine wave with 3 kHz frequency, a sum of sine waves of 250, 500, and 2000 Hz, and a pulse excitation of 250 Hz. In addition to almost perfectly coinciding output curves of the corresponding reduced models (cf. Fig. 4.12), the speed-up in simulation time is up to a factor of 19, see Table 4.2. The presented systems are identified by their number of equations and terms.

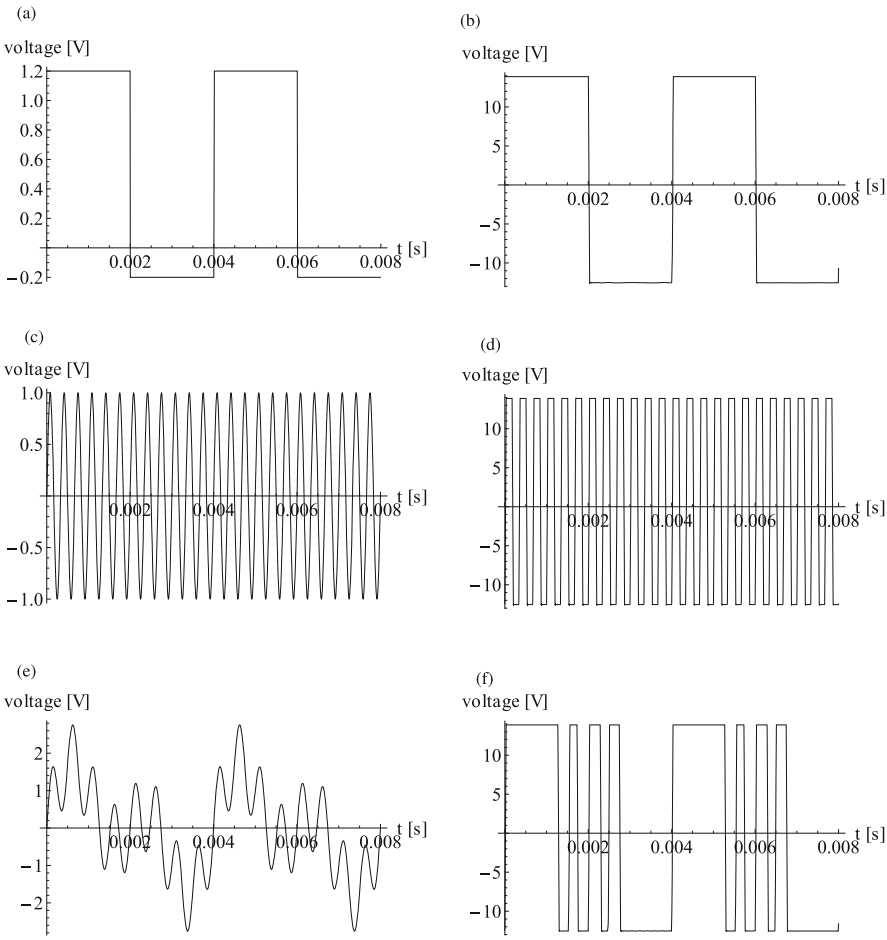


Fig. 4.12 Three different input excitations (*left*) and the resulting outputs of both the original (*solid*) and the *hybrid* reduced system (*dashed*). (**a**) A voltage pulse. (**b**) Output results for the voltage pulse. (**c**) A sine wave with frequency 3000 Hz. (**d**) Outputs applying the input in (c). (**e**) A sum of sine waves. (**f**) The outputs for the sum of sine waves

Table 4.2 Speed-up of simulation of a hybrid reduced entire system w.r.t. the original one

System	Voltage pulse	3 kHz Sine wave	Sum of sine waves
215 = 1050	106 s	273 s	104 s
34 = 92	6 : 6 s	14 : 1 s	10 : 5 s

4.5 Conclusions

To conclude this chapter, we briefly summarize the results: The new hierarchical reduction approach offers enormous savings in computation time, a significant speed-up in system simulations, and yields good reduced models w.r.t. the error, the number of equations and terms of the original system. Moreover, even for highly non-smooth pulse excitations, the reduced models turn out to be very robust. The developed methods were applied to two model classes, circuits consisting of nonlinear subcircuits and circuits containing subcircuits modelled by PDEs, that demonstrated the large potential of the new algorithms.

References

1. Analog Insydes website: http://www.itwm.fraunhofer.de/abteilungen/systemanalyse-prognose-und-regelung/elektronik-mechanik-mechatronik/analog-insydes.html
2. Antoulas, A.C.: Approximation of Large-Scale Dynamical Systems. SIAM, Philadelphia, PA (2005)
3. Bai, Z.: Krylov subspace techniques for reduced-order modeling of large-scale dynamical systems. Appl. Numer. Math. **43**, 9–44 (2002)
4. Feng, L., Benner, P.: A robust algorithm for parametric model order reduction. Proc. Appl. Math. Mech. **7**(1), 1021501–1021502 (2007)
5. Grabinski, H.: Theorie und Simulation von Leitbahnen. Signalverhalten auf Leitungssystemen in der Mikroelektronik. Springer, Berlin (1991)
6. Günther, M.: A Joint DAE/PDE model for interconnected electrical networks. Math. Comput. Model. Dyn. Syst. **6**(2), 114–128 (2000)
7. Günther, M.: Partielle differential-algebraische Systeme in der numerischen Zeitbereichsanalyse elektrischer Schaltungen. Fortschritt-Berichte VDI, Reihe 20, Nr. 343. VDI-Verlag, Düsseldorf (2001)
8. Halfmann, T., Wichmann, T.: Symbolic methods in industrial analog circuit design. In: Anile, A., Alì, G., Mascali, G. (eds.) Scientific Computing in Electrical Engineering. Mathematics in Industry, vol. 9, pp. 87–92. Springer, Heidelberg (2006)
9. Hauser, M., Salzig, C.: Hierarchical model-order reduction for robust design of parameter-varying systems. In: Symbolic and Numerical Methods, Modeling and Applications to Circuit Design (SMACD) (2012)
10. Hennig, E.: Symbolic Approximation and Modeling Techniques for Analysis and Design of Analog Circuits. Ph.D. Thesis, Universität Kaiserslautern, Shaker, Aachen (2000)
11. Miri, A.M.: Ausgleichsvorgänge in Elektroenergiesystemen. Springer, Berlin (2000)
12. Panzer, H., Mohring, J., Eid, R., Lohmann, B.: Parametric model order reduction by matrix interpolation. Automatisierungstechnik **58**, 475–484 (2010)

13. Pelz, G., Roettcher, U.: Pattern matching and refinement hybrid approach to circuit comparison. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **13**, 264–276 (1994)
14. Reis, T., Stykel, T.: PABTEC: passivity-preserving balanced truncation for electrical circuits. IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst. **29**, 1354–1367 (2010)
15. Sommer, R., Hennig, E., Nitsche, G., Schwarz, P., Broz, J.: Automatic Nonlinear Behavioral Model Generation Using Symbolic Circuit Analysis. In: Fakhfakh, M., Tlelo-Cuautle, E., Fernández, F.V. (eds.) Design of Analog Circuits through Symbolic Analysis, pp. 305–341. Bentham Science Publishers, Sharjah (2012). doi:10.2174/978160805095611201010305
16. Schmidt, O.: Structure-Exploiting Coupled Symbolic-Numerical Model Reduction for Electrical Networks. Dissertation, Technische Universität Kaiserslautern, Cuvillier, Göttingen (2010)
17. Schmidt, O.: Structure-exploiting symbolic-numerical model reduction of nonlinear electrical circuits. In: Proceedings der 16th European Conference on Mathematics for Industry ECMI 2010, Wuppertal. Mathematics in Industry, vol. 17, pp. 179–185. Springer, Berlin (2010)
18. Schmidt, O., Halfmann, T., Lang, P.: Coupling of numerical and symbolic techniques for model order reduction in circuit design. In: Benner, P., Hinze, M., ter Maten, E.J.W. (eds.) Model Reduction for Circuit Simulation. Lecture Notes in Electrical Engineering, vol. 74, pp. 261–275. Springer, Dordrecht (2011)
19. Sommer, R., Krauße, D., Schäfer, E., Hennig, E.: Application of symbolic circuit analysis for failure detection and optimization of industrial integrated circuits. In: Fakhfakh, M., Tlelo-Cuautle, E., Fernández, F.V. (eds.) Design of Analog Circuits through Symbolic Analysis, pp. 445–477. Bentham Science Publishers, Sharjah (2012). doi:10.2174/978160805095611201010305
20. Wichmann, T.: Symbolische Reduktionsverfahren für nichtlineare DAE-Systeme. Shaker, Aachen (2004)
21. Wolfram, S.: The Mathematica Book, 5th edn. Wolfram Media, Inc., Champaign (2003)