

Co-Simulation and Uncertainty Quantification for Field/Circuit Coupled Problems

Dissertation

zur Erlangung
des akademischen Grades
eines

Doktors der Naturwissenschaften (Dr. rer. nat.)

der
Fakultät für Mathematik und Naturwissenschaften
der
Bergischen Universität Wuppertal (BUW)
vorgelegt von

Kai Gausling

geboren am 02.07.1988 in Gelsenkirchen

betreut von: Prof. Dr. rer. nat Michael Günther (BUW)
PD Dr. rer. nat Andreas Bartel (BUW)

Wuppertal, 18. Januar 2019

The PhD thesis can be quoted as follows:

urn:nbn:de:hbz:468-20190829-145150-8

[<http://nbn-resolving.de/urn/resolver.pl?urn=urn%3Anbn%3Ade%3Ahbz%3A468-20190829-145150-8>]

DOI: 10.25926/qmhq-4a49

[<https://doi.org/10.25926/qmhq-4a49>]

Contents

Contents	I
1 Introduction	1
2 Modeling and Simulation of Electric Circuits and Magnetoquasistatic Devices	5
2.1 Electric Networks	5
2.2 Magnetoquasistatic Approximation for Field Devices	8
2.3 Coupled Field/Circuit Problems	10
2.3.1 Time-Integration for Field/Circuit Coupled Problems	12
2.4 Conclusions	13
3 Co-Simulation	15
3.1 Dynamic Iteration for Coupled Problems	16
3.1.1 Gauss-Seidel-Type Iteration Scheme	19
3.1.2 DAE-DAE Coupling	21
3.1.3 Field/Circuit Co-Simulation	24
3.1.4 Conclusions	27
3.2 Coupling Interfaces	27
3.2.1 R-splitting	28
3.2.2 LR-coupling	34
3.2.3 Exact Recursion Analysis	37
4 Uncertainty Quantification	41
4.1 The Sobol Decomposition	42
4.1.1 Sobol Sensitivity Analysis	43
4.2 The Polynomial Chaos Expansion	45
4.2.1 Orthogonal Polynomials	46
4.2.2 The generalized Polynomial Chaos Expansion	49
4.2.3 Calculation of the gPC Coefficient Functions	51
4.2.4 Polynomial Chaos based Sensitivity Indices	53
5 Multivariate Quadrature	55
5.1 Tensor-Product Grids	56
5.1.1 Gaussian Quadrature	59
5.2 Sparse Grid Quadrature	61
6 Density Estimation in Co-Simulation	63
6.1 Lower Bound Estimate for Purely Algebraic Coupling	64
6.2 The Kernel Density Estimation Approach	66
6.3 The Spectral Approach	67

7	Numerical Examples	71
7.1	Fast Contraction and Higher Order Co-Simulation	72
7.2	Coupling Interfaces in Application	73
7.2.1	R-Splitting for Field/Circuit Coupled Problems	73
7.2.2	LR-Coupling for Field/Circuit Coupled Problems	77
7.3	Accuracy of the Lower Bound Estimator for Purely Algebraic-to-Algebraic Coupling	82
7.4	PDF estimation by using the KDE and Spectral Method	84
7.5	Uncertainty Quantification in Co-Simulation for Coupled Electric Circuits	87
8	Summary	95

1

Chapter 1

Introduction

Nowadays, motivated by the energy transition, the efficient and robust optimization of electrical energy converters, for example, electrical machines or transformers, steadily grow in importance. Due to financial reasons and the lack of space of embedded systems and integrated circuits, the Computer Aided Design (CAD) modeling and system engineering of such devices is frequently done at the limit of what is technically feasible. However, the industrial series manufacturing introduces uncertainties due to production-related tolerances in the devices. For example, this may be the rotor and/or stator diameter, the position of the permanent magnets, material impurities in the iron or tolerance affected electrical components (resistances, capacitances, inductances). Consequently, commercial devices slightly differ from the reference model which may cause a reduced performance or malfunctions.

A physical description of such devices yields a complex multiphysics model that defines, for example, the magnetic field, the movement of the rotor, the heating of iron as well as the supply circuit which drives the engine. The mathematical modeling of such quantities yields a coupled system of partial differential equations (PDEs) for the magnetic field and heating and differential algebraic equations (DAEs) for the electric circuit, [21]. The efficient and robust design of electrical energy converters requires precise information about the field distribution and thus the space discretization of 2D and 3D field devices yields a coupled system of DAEs with frequently millions of unknowns. Numerical simulations of such high-resolution coupled systems are computationally expensive and the simulation time takes several hours until a solution is computed, see e.g. the 2D transformer model using spatial discretization from FEMM, [6], discussed in Chapter 7. Furthermore, the analysis of electric devices with tolerances, often referred to as Uncertainty Quantification (UQ), [45–49], requires to solve the coupled system with its millions of unknowns multiple times (for various parameter values within the tolerance). Therefore, the standard time-integration methods, [28], are not well-suited to perform this task.

To set up the system of equations for coupled problems, single-rate time-integration requires access to the modeling layer for each software-package which are involved in a mixed mode simulation, [23]. Several well known approaches can be applied for the efficient transient simulation of coupled systems, whereas co-simulation often referred to as waveform relaxation or dynamic iteration scheme is frequently used, [2,3,8,20,24]. Especially, when the monolithic description of a complex multiphysics model is not realizable and/or suitable software tools for the subsystems are available, then it is a relevant choice. Co-Simulation tries to exploit that different parts of the multiphysics model acts on various time scales (usually slow changes in the magnetic field and faster changes in the circuit), [3, 8]. For example, this may be a pulse width modulated (PWM) signal which drives a field device, [24]. The idea is to split the multiphysics model into submodels. This allows to define subsystems which can be handled separately with its own time-integrator. Here suitable time stepping methods can be used to capture the structural properties and the dynamics of each subsystem. Then, co-simulation works on certain time periods (windows), where the information between the subsystems are only exchanged at communication points. Here, convergence can be achieved by solving multiple times the subsystems on a small time interval (dynamic iteration). Co-simulation applied to coupled ordinary differential equations (ODEs) is

well understood and always convergent, [4]. Our focus is on time-integration of field/circuit coupled problems. A space discretization of the field, [34, 35], yields a coupled system of DAEs (DAE-DAE coupling). DAEs are structurally different from ODEs. Here, convergence and the number of required repeated model simulations within each window depends on the design of the coupling interface, i.e., how the subsystems communicate with each other, and the computational order amongst others, [24]. However, co-simulation for DAEs might fail if a contraction condition is not fulfilled, [2, 3, 20, 24].

Convergence and contraction of co-simulation with application for field/circuit coupled problems was investigated, e.g. in [8, 24, 32, 52]. The standard way of decoupling for field devices linked with circuits is to separate between the field and circuit part, i.e., cutting at EM device boundaries. Here, Co-simulation has proven to be unconditionally stable and convergent as long as the computation starts with the field subproblem. However, our numerical investigations show that the coupling between field and circuit is commonly weak for the standard type of decoupling such that each subsystem has to be solved many times to ensure a certain accuracy in the solution which ends in a high computational effort.

The primary goal of this thesis is to develop new coupling interfaces for field/circuit coupled problems such that the coupling between both becomes stronger. This would also improve the computational time for UQ, where the general Polynomial Chaos (gPC) expansion is applied for the first time to co-simulation. Furthermore, when co-simulation is used to solve (random) coupled systems, it can suffer from the uncertainties such that it affects the convergence of the dynamic iteration process. Therefore, we aim to assess the divergence probability of co-simulation during the simulation. This requires the computation of the probability density function of the contraction factor, where we focus on two different methods (the Kernel Density Estimation and the gPC based spectral method) to perform this task. The results can be used for an effective time window size control. It also offers the opportunity for varying the time window size to further improve the simulation speed.

Overview

This thesis is basically split into two main parts. The first part is about new coupling strategies for an efficient transient simulation of multiscale problems. This may be, for example, a transformer or an electrical machine coupled to a circuit with fast changing signals, e.g. Pulse-width modulated converter. The convergence and the speed of contraction, i.e., the number of repeated model simulations, of co-simulation depend on the computational sequence in which the subsystems are solved iteratively as well as on the interface modeling amongst others. By numerical simulation tests, the previous (standard) interfaces have revealed to be inefficient with respect to the computational effort, since the number of required repeated simulations to provide a certain accuracy in the solution becomes large. To tackle this problem, we propose new specific interfaces for a more efficient way of coupling the subsystems such that the co-simulation process becomes more contractive. The second main part is about UQ and probability estimation for multiscale models solved within a co-simulation framework. Generally, contraction of a dynamic iteration can only be ensured if certain contraction properties are fulfilled. However, introducing uncertainties may have impact on these contraction properties and can destroy the convergence. Consequently, one is interested to analyze stochastics within the co-simulation to state the probability of divergence. Furthermore, we aim to calculate sensitivities for multiscale models.

This thesis is structured as follows: Chapter 2 introduces into circuit modeling by the technique of modified nodal analysis (Chapter 2.1) and into the modeling of field devices and its magneto-quasistatic formulation using Maxwell's equations in macroscopic form as well as into the space discretization of PDEs (Chapter 2.2). The problem formulation for the field/circuit coupled system is stated in Chapter 2.3, followed by a DAE-index analysis of the resulting equation.

Chapter 3 is the first major part of this thesis and starts with an introduction to co-simulation. Chapter 3.1 treats the idea of the dynamic iteration for coupled problems. It is split into parts on partitioning of timelines into windows, on partitioning of the coupled problem into subsystems as well as on the dynamic iteration process consisting of an extrapolation step followed by several dynamic iteration steps (Chapter 3.1.1). In Chapter 3.1.2, the rate of convergence of co-simulation is analyzed for different splittings of the subsystems using the well known (standard) recursion estimate. Afterwards, co-simulation and its convergence is closely studied for field/circuit coupled problems (Chapter 3.1.3), where it is assumed that the circuit and the field is treated separately within the dynamic iteration process (standard way of splitting). Here, by numerical simulations, we demonstrate that cutting at the EM device boundaries does not work best with respect to the computational effort. Therefore, Chapter 3.2.1 and Chapter 3.2.2 introduce new coupling strategies for a better performance of co-simulation. These are the R-splitting and the LR-coupling interfaces. Both approaches were presented at COMPUMAG 2015 and SCEE 2016, [36, 37]. It also includes the convergence theory based on an extended version (fine structure analysis) of the (standard) error recursion estimate. Chapter 3.2.3 covers the fine structure analysis for coupled DAE systems and proves the inaccuracy of the standard recursion estimation even for a simple coupled circuit. The results were presented at ECMI 2014, [39].

The second major part of this thesis is about UQ applied to co-simulation. Therefore, Chapter 4 introduces into two basic concepts. These are the Sobol decomposition (often referred to as full decomposition) discussed in Chapter 4.1 and the more effective Polynomial Chaos (PC) expansion method (referred to as truncated decomposition) closely studied in Chapter 4.2. However, the Sobol decomposition method becomes inefficient for higher dimensional problems, since the computational cost increases exponentially with the dimension (curse of dimensionality). Here, the PC expansion method tries to break the exponential growth by a suitable approximation technique. It is based on a truncated and weighted sum of orthogonal polynomials (Chapter 4.2.1), where the weighting is done by coefficient functions. Chapter 4.2.2 states the generalized form referred to as gPC expansion followed by the calculation methods to find the unknown coefficient functions (Chapter 4.2.3). Here, we shortly introduce into the Galerkin approach, whereas the Collocation approach is studied more detailed, since it will be used for all our calculations. Finally, Chapter 4.2.4 discusses how sensitivity indices can be derived from a gPC expansion. The gPC based UQ algorithm combined with co-simulation was presented at SCEE 2014, [38]. The Collocation approach requires to solve multiple probabilistic integrals. Therefore, Chapter 5 gives insight into multivariate quadrature formulas to solve such integrals even for high dimensional problems. These are the tensor-product grid and the sparse grid techniques discussed in Chapter 5.1 and Chapter 5.2. Furthermore, we aim to reconstruct the probability density function of the contraction factor as precise as possible to assess the divergence probability of co-simulation when uncertainties are introduced. Commonly, density estimators try to reconstruct the probability density from a given set of samples. Thus, Chapter 6.1 introduces into estimate the contraction factor online, i.e., during the co-simulation. Then, based on these set of samples we apply the Kernel Density Estimation as well as the spectral approach, which is closely discussed in their corresponding chapters (Chapter 6.2 and Chapter 6.3). Both methods with its application in co-simulation were presented at SCEE 2016, [36].

Chapter 7 presents the simulation results and shows the importance and applicability of the proposed methods. Furthermore, it proves the theoretical results numerically. These are: the fast contraction and higher order co-simulation for an RL-circuit (Chapter 7.1), the new R-splitting and LR-coupling approaches with applications for field/circuit coupled problems (Chapter 7.2.1 and Chapter 7.2.2), the accuracy of the lower bound estimation of the contraction factor for algebraic-to-algebraic coupling structures, the probability density function estimation of the contraction factor for uncertain co-simulation models (Chapter 7.4) as well as the calculation of sensitivity indices within a co-simulation framework. Finally, the conclusions and outlook are given in Chapter 8.

2

Chapter 2

Modeling and Simulation of Electric Circuits and Magnetoquasistatic Devices

Many techniques are known for modeling the physical behavior of electric components. These may be, for example, the electric current flow through conductors and wires or the propagation of electric fields in electrical machines and transformers. In any case, the modeling requires a mathematical description, where always a compromise between accuracy and complexity of the model has to be found. Using commercial simulation tools, such as PSpice (Simulation Program with Integrated Circuit Emphasis) by OrCAD, the modeling of electric circuits is commonly done by the technique of Modified Nodal Analysis (MNA), [21], where electric components (resistances, inductances, capacitances, diodes, etc.) are considered as idealized lumped elements by disregarding the spatial distribution. Some of them with a linear behavior, e.g. Ohm's law for the voltage-current relation of a resistor, or a non-linear behavior, e.g. the voltage-current relation for diodes or transistors.

MNA uses the voltage-current relation as well as Kirchhoff's circuit laws to set up the corresponding system of equations, where the equations are set up piecewise for each element. In the end, the electric circuit is described by a set of differential equations and additional algebraic constraints, thus that the problem can be stated as a system of differential algebraic equations (DAEs), [22]. However, for many electric devices, such as transformers or electrical machines, the description by a few lumped elements may be too inaccurate. Here, the behavior and its physical quantities, e.g. fluxes in field devices, are mathematically described by partial differential equations (PDEs), which are mostly computational expensive to solve.

Typically, field devices are powered by electric circuits. The simulation of field devices coupled to circuits is referred to as mixed mode simulation, [23]. Setting up the mixed mode simulation yields a system of partial differential algebraic equations (PDAEs) with expensive PDE part only where necessary, [3, 8]. This chapter will introduce into different models and their modeling as well as the coupling of field devices with circuits. Modified nodal analysis is roughly introduced in Chapter 2.1. Chapter 2.2 focus on modeling of magnetoquasistatic (MQS) devices and its discretized formulation, [24]. Finally, the discretization enables for coupling field devices with circuits by extend the classic MNA, [2, 3, 8, 20, 24], Chapter 2.3. This chapter concludes with a structural analysis of the spatial discretized PDAE model.

2.1 Electric Networks

Electric circuits are commonly modeled via the technique of MNA, [21]. Its idealized lumped basic elements are resistors, capacitors, inductors, diodes as well as time-dependent sources (current and voltage sources). The classical MNA yields a system with unknown node voltages and

branch currents (currents through voltage sources and inductors). It can be extended by adding additional equations for the fluxes and charges (flux/charge oriented MNA), [21, 22].

MNA uses the voltage-current relation of the lumped elements stated as branch equations as well as the Kirchhoff's circuit laws to state the topological relations:

Kirchhoff's current law (KCL): The algebraic sum of all currents within a node is zero:

$$\sum_{k=1}^n i_k = 0,$$

where n is the total number of branches connected to the node, see Fig. 2.1 (left).

Kirchhoff's voltage law (KVL): The algebraic sum of all potential differences (voltages) around any closed loop is zero:

$$\sum_{k=1}^n v_k = 0,$$

where n is the total number of voltages within a closed loop, see Fig. 2.1 (right).

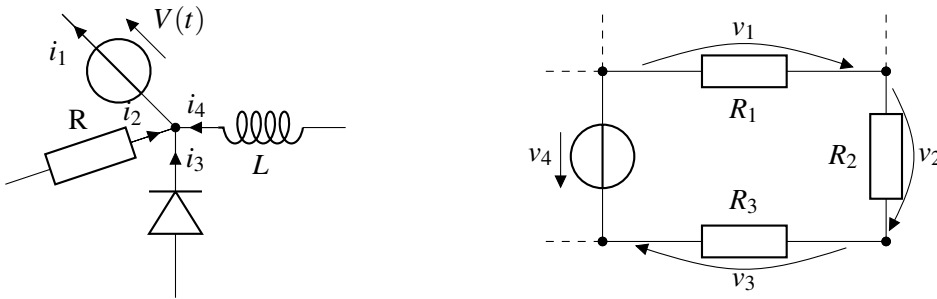


Figure 2.1: (left) KCL for a node with four branches. (right) KVL for a closed loop with four voltages.

Generally, MNA consists of three steps. The first step is to state the Kirchhoff's circuit laws of the circuit. Here, for each node of the circuit one states the currents flowing in and out of the node, where the sum of the currents meeting at a point is zero (KCL). Then, one uses the branch equations in terms of the voltage nodes to replace as many branch currents as possible, e.g. the current through a resistor can be expressed as the potential difference among its two contacts (two nodes) multiplied by the conductance. The simplest kind of circuit modeling via MNA only deals with linear lumped elements, such as resistances, conductances, capacitances, voltage and current sources. However, the technique of MNA also allows for non-linear elements. Therefore, we consider the general case including diodes, transistors, etc..

To state the conventional form of MNA, one has to decompose the equations according to their element types. Assuming that the branches are enumerated in the following order: first all capacitive branches, then all resistive branches, then all inductive branches, then all voltage source branches and finally all current source branches. The incidence matrices $\mathbf{A}_C, \mathbf{A}_R, \mathbf{A}_L, \mathbf{A}_V$ and \mathbf{A}_I state the node-branch relation for each element type, i.e., capacitances, resistances, inductances, independent voltage- and current sources, where each row refers to a network node. Then, all incidence matrices can be summarized block-wise in an overall incidence matrix

$$\mathbf{A} = (\mathbf{A}_C, \mathbf{A}_R, \mathbf{A}_L, \mathbf{A}_V, \mathbf{A}_I) \quad (2.1)$$

of size $\mathbf{A} \in \{-1, 1, 0\}^{n \times b}$ for a circuit with n nodes (excluding the ground potential) and b branches, where each subindex refers to the respective branch type. Therefore, the circuit topology is represented by its incidence matrix \mathbf{A} , where every column of \mathbf{A} corresponds to one branch, i.e., the matrix \mathbf{A} state the node-branch relation for the overall digraph. For large circuits, the incidence matrix \mathbf{A} is typically sparse, while -1 specifies a current which leaves a node, 1 specifies a current which enters a node and 0 specifies the independency between nodes and branches, since they are not in contact with each other.

Using the block-wise notation (2.1), the MNA in flux/charge oriented form can be stated as DAE, [21, 22]

$$\mathbf{A}_C \frac{d}{dt} \mathbf{q} + \mathbf{A}_R \mathbf{g}_R(\mathbf{A}_R^\top \mathbf{u}, t) + \mathbf{A}_L \dot{\mathbf{i}}_L + \mathbf{A}_V \dot{\mathbf{i}}_V + \mathbf{A}_I \dot{\mathbf{i}}_S(t) = 0 \quad (2.2a)$$

$$\mathbf{A}_V^\top \mathbf{u} - \mathbf{v}_S(t) = 0 \quad (2.2b)$$

$$\mathbf{q} - \mathbf{q}_C(\mathbf{A}_C^\top \mathbf{u}, t) = 0 \quad (2.2c)$$

$$\frac{d}{dt} \phi - \mathbf{A}_L^\top \mathbf{u} = 0 \quad (2.2d)$$

$$\phi - \phi_L(\dot{\mathbf{i}}_L, t) = 0 \quad (2.2e)$$

with time variable $t \in [t_0, t_e]$, independent sources $\dot{\mathbf{i}}_S(t)$, $\mathbf{v}_S(t)$ and given functions $\mathbf{q}_C(\mathbf{v}, t)$, $\mathbf{g}_R(\mathbf{v}, t)$ and $\phi_L(\dot{\mathbf{i}}, t)$ for characterizing the relations for their respective circuit element. For a network consisting of n nodes and a total number of $b = n_C + n_L + n_G + n_V + n_I$, $n_C, n_L, n_G, n_V, n_I \in \mathbb{N}_0$ branches, the unknowns of the system are the node potentials $\mathbf{u} \in \mathbb{R}^n$, the currents through inductors $\dot{\mathbf{i}}_L \in \mathbb{R}^{n_L}$ and voltage sources $\dot{\mathbf{i}}_V \in \mathbb{R}^{n_V}$ as well as the charges $\mathbf{q} \in \mathbb{R}^{n_C}$ and fluxes $\phi \in \mathbb{R}^{n_L}$ (where n_V, n_C and n_L denote their respective number). Equation (2.2a) states the Kirchhoff's current law for circuits and (2.2b) assigns all voltage sources to their respective branch equations. By deleting all unknowns for the charges and fluxes, (2.2) can be reduced to a smaller system often referred to as the traditional MNA, where energy conservation is no longer guaranteed, [31].

In practice, oftentimes the physical behavior of many electric devices cannot be described sufficiently accurate by a few idealized lumped elements. Here, the modeling requires the treatment of PDEs. Especially for field devices the set of Maxwell's equations named after **J. C. Maxwell**, [25], is the basis to describe electromagnetic phenomena. The coupling of DAEs (2.2) for circuits with PDEs for field devices yields (after space discretization) an enlarged spatially discretized PDAE system. The coupling is based on an extended version of the classic MNA by plug in the spatially discretized PDE system to the circuit using the idea of controllable sources. The idea is simple and straight forward by inserting an additional unknown current $\dot{\mathbf{i}}_M$ to the branch equation using the incidence matrix \mathbf{A}_M . Then Kirchhoff's Current law (2.2) is extended by the contribution of the field device $\mathbf{A}_M \dot{\mathbf{i}}_M$:

$$\mathbf{A}_C \frac{d}{dt} \mathbf{q} + \mathbf{A}_R \mathbf{g}_R(\mathbf{A}_R^\top \mathbf{u}, t) + \mathbf{A}_L \dot{\mathbf{i}}_L + \mathbf{A}_V \dot{\mathbf{i}}_V + \mathbf{A}_I \dot{\mathbf{i}}_S(t) + \boxed{\mathbf{A}_M \dot{\mathbf{i}}_M} = 0. \quad (2.3)$$

In order to achieve a uniquely solvable system, the equation for the device which describe the unknown currents in terms of the applied voltage drops to the field device are given by $\mathbf{v}_M = \mathbf{A}_M^\top \mathbf{u}$, where \mathbf{u} includes all node potentials of the circuit.

Now, Chapter 2.2 introduces into the modeling of field devices using the magnetoquasistatic formulation which can be obtained by some reasonable assumptions (simplifications) of Maxwell's theory.

2.2 Magnetoquasistatic Approximation for Field Devices

In general Maxwell's partial differential equations in macroscopic formulation are the fundamental theory to describe electromagnetic phenomena. The set of Maxwell's equations reads, [25]:

$$\nabla \times \mathbf{E} = -\frac{\partial \mathbf{B}}{\partial t}, \quad \nabla \times \mathbf{H} = \frac{\partial \mathbf{D}}{\partial t} + \mathbf{J}, \quad \nabla \mathbf{D} = \rho, \quad \nabla \mathbf{B} = 0, \quad (2.4)$$

with electric field strength \mathbf{E} , magnetic flux density \mathbf{B} , magnetic field strength \mathbf{H} , electric flux density \mathbf{D} , electric charge density ρ and the electric current density \mathbf{J} . Within this thesis, we limit ourselves to 2D regions ($\Omega \subseteq \mathbb{R}^2$). All quantities in (2.4) depend on time $t \in [t_0, t_e]$ and space $\mathbf{r} \in \Omega$ and thus Maxwell's equations describe an time and space dependent problem which has to be solved. In order to apply Maxwell's macroscopic equations, one has to specify the relations between \mathbf{D} and \mathbf{E} , \mathbf{H} and \mathbf{B} as well as \mathbf{E} and \mathbf{J} . The constitutive material relations are given (by definition)

$$\mathbf{D} = \varepsilon \mathbf{E}, \quad \mathbf{H} = \nu \mathbf{B}, \quad \mathbf{J} = \sigma \mathbf{E} \quad (2.5)$$

where ε is the permittivity, ν is the reluctivity and σ is the conductivity. Using the magnetic vector potential $\mathbf{A}: [t_0, t_e] \times \Omega \rightarrow \mathbb{R}^2$ for the field quantities \mathbf{E} and \mathbf{B} it follows:

$$\mathbf{B} = \nabla \times \mathbf{A} \quad \text{and} \quad \mathbf{E} = -\frac{\partial \mathbf{A}}{\partial t} - \nabla \varphi, \quad (2.6)$$

where $\varphi: [t_0, t_e] \times \Omega \rightarrow \mathbb{R}$ is the electric scalar potential. Using the magnetic vector potential formulation, the problem can be stated in magnetoquasistatic form, [26]:

$$\sigma \frac{\partial \mathbf{A}}{\partial t} + \nabla \times (\nu \nabla \times \mathbf{A}) = -\rho \nabla \varphi. \quad (2.7)$$

Equation (2.7) is of parabolic type and often referred to as curl-curl equation, where it is assumed that the electric flux density \mathbf{D} behaves constant over the time:

$$\frac{\partial \mathbf{D}}{\partial t} = 0. \quad (2.8)$$

The assumption of (2.8) implies that the field does not change quickly. Consequently, eq. (2.7) only holds for a certain range of frequencies, [26]. However, the curl-curl equation describes the applications in this thesis sufficiently accurate, since all engineering test examples operates within a low frequency domain, e.g. a low frequency power inverter transformers.

Boundary and Initial Conditions: Maxwell's Equations (2.4) as well as the material relations (2.5) are defined on an infinite domain Ω . However, in practice one is interested only in the solution of (2.7) for a finite domain with boundary $\partial\Omega$, where the center of the truncated domain is given by the device (unless symmetries of the device are exploited). Introducing spatially borders is possible for parabolic PDEs as long as the boundaries are defined far enough from the MQS device, where the surrounding is typically modeled by air. Depending on the problem, one has to apply boundary conditions of different type for modeling the behavior in the outside of the truncated problem, [33]. These are the Dirichlet- and Neumann boundary conditions. Furthermore, to exploit symmetric geometries of the device and its resulting electromagnetic symmetries requires to use anti-periodic boundary conditions, where we refer to [24] for details.

The Dirichlet condition directly specifies the values of the tangential part of the solution of the

magnetic vector potential on the boundary:

$$\mathbf{A}(\mathbf{r}) \times \mathbf{n} = \mathbf{A}_d \quad \forall \quad \mathbf{r} \in \partial\Omega, \quad (2.9)$$

where \mathbf{A}_d are predefined values and \mathbf{n} is the outward-pointing normal unit vector. In many cases, the Dirichlet condition is given as a constant value such as $\mathbf{A}_d = 0$ (the field goes to zero at the borders).

The Neumann condition specifies the values that the first derivative of the solution, i.e., the tangential parts of the magnetic field strength, is to take on the boundary of the domain, [24]:

$$\mathbf{H}(\mathbf{r}) \times \mathbf{n} = \mathbf{H}_n \quad \forall \quad \mathbf{r} \in \partial\Omega, \quad (2.10)$$

where \mathbf{H}_n are predefined values and \mathbf{n} is the outward-pointing normal unit vector.

Furthermore, the given parabolic PDE in (2.7) is time dependent with solution on the finite time interval $[t_0, t_e]$. Generally, we set initial conditions to define the solution at $t_0 = 0$ and we are interested in the fields and fluxes at $t \geq 0$. Therefore, beside boundary values one has to define appropriate initial values (IVs) for the magnetic vector potential for the entire domain Ω at time-point t_0 :

$$\mathbf{A}(t_0, \mathbf{r}) = \mathbf{A}_0(\mathbf{r}) \quad \forall \quad \mathbf{r} \in \Omega, \quad (2.11)$$

where in many applications $\mathbf{A}_0(\mathbf{r}) = 0$ is a reasonable choice.

Magnetoquasistatic Solid Conductor Model - A Discretized Formulation

A mixed mode simulation requires the coupling of MQS devices with circuits. In practice, different conductor models such as solid conductors and stranded conductors are known, [24]. Within this work we focus on solid conductors, which is a massive bar with contact surfaces on both sides, see Fig. 2.2. For those models the coupling with a circuit is done by defining parts of the device surface region as circuit branches thus that the circuit communicates with the MQS device only by lumped elements. For coupling, the surface regions which are connected to the circuit have to be linked to the boundary. Therefore, only appropriate boundary conditions have to be defined, where commonly conditions of Dirichlet type (2.9) are applied.

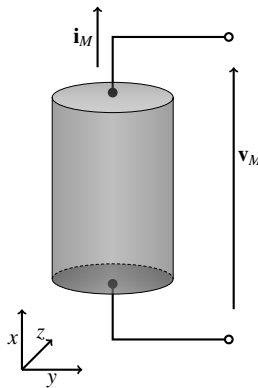


Figure 2.2: MQS solid conductor model connected to a circuit by two wires. v_M is the voltage drop applied to the coils and i_M is the resulting current through the solid conductor.

We focus on the spatial discretized formulation of Maxwell's equation's (2.4) and material relations (2.5). A prominent approach for the discretization of electromagnetic devices in space is the finite element method (FEM), [34, 35], and finite integration technique (FIT), see e.g. [24], where for 2D problems typically triangular elements are used. In the following we state the spatially form of the curl-curl equation, whereas we refer to [24] for details.

The magnetic field of MQS devices given in magnetoquasistatic formulation (2.7) can be described by the spatially discretized curl-curl equations in terms of the discretized magnetic vector potential $\mathbf{a} \in \mathbb{R}^{n_M}$, where n_M depends on the discretization of the MQS device in space. The discretized curl-curl equation reads:

$$\mathbf{M}_\sigma \frac{d}{dt} \mathbf{a} + \mathbf{k}_v \mathbf{a} - \mathbf{M}_\sigma \mathbf{X}_M \mathbf{v}_M = 0, \quad (2.12)$$

where all branch currents through the MQS device are gathered in a (total) current vector \mathbf{i}_M given by

$$\mathbf{i}_M = \mathbf{X}_M^\top \mathbf{k}_v \mathbf{a}. \quad (2.13)$$

The quantities are the conductivity \mathbf{M}_σ , the nonlinear curl-curl matrix \mathbf{k}_v , the voltage drop applied to the conductive region of the MQS device \mathbf{v}_M and \mathbf{X}_M is the distribution matrix for the corresponding current \mathbf{i}_M .

The discretized curl-curl equation (2.12) corresponds to a differential algebraic equation with corresponding initial values $\mathbf{a}(t_0) = \mathbf{a}_0$. Therefore, the next chapter states the coupled system consisting of MQS devices and circuit elements.

2.3 Coupled Field/Circuit Problems

In the previous Chapter 2.1 and Chapter 2.2 the modeling of electric circuits and MQS devices has been introduced. For circuits, the behavior can be modeled using idealized lumped elements by disregarding the spatial distribution. The related system can be stated via MNA to a set of differential algebraic equations, [21, 22]. The situation is different for MQS devices, where a sufficiently accurate description is often not reasonable by a few lumped and idealized elements. The mixed mode simulation (as one system) of circuits with MQS devices requires a coupling between (2.2) and (2.12), [20, 24].

The voltage drop \mathbf{v}_M applied to the coils of the MQS devices includes all nodes which are connect to the conductive surfaces. Therefore, as last step, the extended Kirchhoff's Current law (2.3) is merged with the spatially discretized curl-curl equation for MQS devices (2.12) by the total current \mathbf{i}_M of the MQS device using its corresponding expression (2.13). Finally, the coupled

system can be described as differential algebraic equation, [24]:

$$\mathbf{A}_C \frac{d}{dt} \mathbf{q} + \mathbf{A}_R \mathbf{g}_R(\mathbf{A}_R^\top \mathbf{u}, t) + \mathbf{A}_L \mathbf{i}_L + \mathbf{A}_V \mathbf{i}_V + \mathbf{A}_I \mathbf{i}_S(t) + \boxed{\mathbf{A}_M \mathbf{i}_M} = 0 \quad (2.14a)$$

$$\mathbf{q} - \mathbf{q}_C(\mathbf{A}_C^\top \mathbf{u}, t) = 0 \quad (2.14b)$$

$$\frac{d}{dt} \phi - \mathbf{A}_L^\top \mathbf{u} = 0 \quad (2.14c)$$

$$\phi - \phi_L(\mathbf{i}_L, t) = 0 \quad (2.14d)$$

$$\mathbf{A}_V^\top \mathbf{u} - \mathbf{v}_S(t) = 0 \quad (2.14e)$$

$$\mathbf{M}_\sigma \frac{d}{dt} \mathbf{a} + \mathbf{k}_v \mathbf{a} - \mathbf{M}_\sigma \mathbf{X}_M \boxed{\mathbf{A}_M^\top \mathbf{u}} = 0 \quad (2.14f)$$

$$\mathbf{i}_M - \mathbf{X}_M^\top \mathbf{k}_v \mathbf{a} = 0 \quad (2.14g)$$

with unknown node potentials $\mathbf{u} \in \mathbb{R}^n$, currents through MQS devices, inductors and voltage sources $\mathbf{i}_M \in \mathbb{R}^{n_M}$, $\mathbf{i}_L \in \mathbb{R}^{n_L}$, $\mathbf{i}_V \in \mathbb{R}^{n_V}$, charges and fluxes $\mathbf{q} \in \mathbb{R}^{n_C}$, $\phi \in \mathbb{R}^{n_L}$ and the additional magnetic vector potential $\mathbf{a} \in \mathbb{R}^{n_M}$ of the MQS device.

Differential algebraic equations are generally more difficult to solve as ordinary differential equations. The difficulty of DAEs is stated by its DAE-Index, [27]. Therefore, the next chapter introduces into the differential index analysis with respect to field/circuit coupled problems.

DAE-Index Analysis

From the numerical point of view DAEs are mostly harder to solve by time-integration methods as ODEs, [28], and require a special numerical treatment. The minimum number of differential steps required to transform a DAE into an ODE is known as the differential index of the DAE and referred to as DAE-index, [27, 28]. One can say that the DAE-index number indicates the difficulty to solve the DAE by numerical methods. That is the higher the index the more difficult the DAE to solve.

The structural requirements for a circuit to achieve a certain index is well known and posed in [21]. The DAE-index for field/circuit coupled problems was studied by several authors for linear and non-linear, 2D and 3D MQS devices, e.g. [24, 29, 30, 32]. One can show that under certain conditions regarding the local passivity of all lumped network elements and under some structural requirements of the network, the DAE-index for (2.14) is at most two. This is considered in the following for the traditional MNA, where fluxes and charges are eliminated.

Sorting (2.14) into terms of a differential matrix (dynamic part), non-differential vector (static part) and a vector consisting of all current and voltage sources (source part) yields the form, [24]:

$$\begin{pmatrix} \mathbf{A}_C & 0 & 0 \\ 0 & \mathbf{I} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & \mathbf{M}_\sigma \end{pmatrix} \frac{d}{dt} \begin{pmatrix} \mathbf{A}_C \mathbf{q}_C \\ \phi_L \\ \mathbf{M}_\sigma \mathbf{a} \end{pmatrix} + \begin{pmatrix} \mathbf{A}_R \mathbf{g}_R(\mathbf{A}_R^\top \mathbf{u}, t) + \mathbf{A}_L \mathbf{i}_L + \mathbf{A}_V \mathbf{i}_V + \mathbf{A}_M \mathbf{X}_M^\top \mathbf{k}_v \mathbf{a} \\ -\mathbf{A}_L^\top \mathbf{u} \\ \mathbf{A}_V^\top \mathbf{u} \\ \mathbf{k}_v(\mathbf{a}) \mathbf{a} - \mathbf{M}_\sigma \mathbf{X}_M \mathbf{A}_M^\top \mathbf{u} \end{pmatrix} + \begin{pmatrix} \mathbf{A}_I \mathbf{i}_S(t) \\ 0 \\ -\mathbf{v}_S(t) \\ 0 \end{pmatrix} = 0, \quad (2.15)$$

or more general:

$$\mathbf{A} \frac{d}{dt} \mathbf{d}(\mathbf{x}, t) + \mathbf{b}(\mathbf{x}, t) - \mathbf{f}(t) = 0, \quad (2.16)$$

with unknown solution $\mathbf{x}^\top = (\mathbf{u}^\top, \mathbf{i}_L^\top, \mathbf{i}_V^\top, \mathbf{a}^\top)$.

Index-0:

The DAE (2.15) has index-0 if and only if no voltage sources are included, a tree containing capacitors only and

- i) no MQS devices are included in the network, or
- ii) the network consists of MQS devices with conductive regions only, i.e., \mathbf{M}_σ has full rank.

Index-1:

Assuming that at least one MQS device, one voltage source are included or there is no tree containing capacitors only. Then the DAE (2.15) has index-1 if and only if there is neither

- i) a cutset consisting of inductances, current sources and MQS devices only (LIM-cutset), nor
- ii) a loop consisting of capacitances and at least a voltage source only (CV-loop).

Index-2:

Assuming that at least one MQS device or one voltage source are included or there is no tree containing capacitors only. Then the DAE (2.15) has index-2 if and only if there is either

- i) a cutset consisting of inductances, current sources and MQS devices only (LIM-cutset), or
- ii) a loop consisting of capacitances and at least a voltage source only (CV-loop).

Remark 1. *Within this work we will focus on DAEs (2.15) with DAE-index at most index-1. Thus we assume that neither LIM-cutset nor CV-loop are included.*

Now, having ensured that the DAE (2.15) is of DAE-index at most one, the following section discusses its time-integration.

2.3.1 Time-Integration for Field/Circuit Coupled Problems

The DAE (2.15) is solved in time by using numerical time-integration methods such as Backward Differentiation Formulas (BDF) or implicit Runge-Kutta (RK) methods, [28]. Here, the solution is computed on an discrete time-grid by dividing the simulation interval $[t_0, t_e]$ into $N + 1$ time-points:

$$t_0 < t_1 < t_2 < \dots < t_n \dots < t_N = t_e, \quad (2.17)$$

with fixed step-size $h = (t_e - t_0)/N$ or non-equidistant by using various step-sizes.

For non-equidistant time-points, the size of h essentially depends on how the unknowns of the system behave in time. Here, the step-size has to be chosen smaller in time-regions of high dynamics, i.e., where the system variables change faster. Dealing with field/circuit coupled problems

one usually has different time scales. This comes from the physics, where the field only changes slowly compared to possible changes in the circuit part.

For single rate time-integration, the solution of all unknowns of \mathbf{x} in (2.15) is computed at each time point, where the step size h (defined equidistant by the user or non-equidistant by the time-integration method) must be adapted with respect to the fastest changing signal. However, most of the unknowns of (2.15) are field quantities, i.e., the line integrated magnetic vector potentials, where the number of degrees of freedom (DOF) depends strongly on the size of triangular elements. In practice usually thousands of DOF for the field part. This makes the standard single rate time-integration for (2.15) inefficient.

The way to avoiding this problem is referred to as co-simulation scheme or dynamic iteration, [2, 3, 24], where each subsystem (circuit and field) is treated separately on their own time scales using their own time-integrators with appropriate step sizes. The method of co-simulation will be discussed next.

2.4 Conclusions

Within this chapter it has been shown that the type of equation can be different when physical phenomena are modeled in mathematics.

First we stated the network equations for electric circuits by the technique of modified nodal analysis. This results in a system consisting of differential equations and algebraic constraints. Introducing Maxwell's equations in differential form on macroscopic level leads (under reasonable assumptions) to a magnetoquasistatic formulation for electric field devices (MQS devices) using partial differential equations (PDEs). Merging circuits and MQS devices as one system of equation requires a mixed mode simulation to solve the partial differential algebraic equation model. Using spatial discretization for the MQS device, the PDAE model can be transformed into a system of DAEs, where the underlying DAE-index depends on the network structure. Due to different time scales the computation of those models can be very time-consuming. Therefore, Chapter 3 will discuss the co-simulation strategy for solving such problems in time domain but much more efficient.

3 Chapter 3

Co-Simulation

In Chapter 2, the modeling of field/circuit coupled problems was discussed. It has been shown that a spatial discretization of MQS devices yields a coupled system of differential algebraic equations. For field/circuit coupled problems, single-rate time-integration is inefficient due to the different time scales of the subsystems (usually slow changes in field and faster changes in the circuit). Furthermore, the structural properties of each subsystem can be different due to various modeling techniques. Here suitable time-integration methods must be chosen such that each integrator can capture the structural properties. Moreover, to set up the system of equations for the coupled problem, single-rate time-integration requires access to the modeling layer for each software-package which is involved in the mixed mode simulation. Within this chapter, we focus on a coupling approach of the involved software-packages which requires only the coupling variables to be available.

The coupling of software-packages is a standard procedure for simulating multiphysics problems. Each software-package is used to compute the solution of its own subproblem, i.e., each subsystem is treated separately. This can be, for example, the flux in a transformer coupled to the heat transport in the iron-core supplied by a circuit. The prominent approach for this task is known as co-simulation, waveform relaxation method or dynamic iteration, where the information between the subproblems are only exchanged at communication points. Thus, co-simulation works on certain time periods often referred to as time windows, where convergence can be achieved by solving multiple times the subsystems (iteration), Chapter 3.1. The Co-simulation applied to coupled ODEs is always convergent, see e.g. [4]. Our main focus is on time-integration of field/circuit coupled problems, see Chapter 3.1.3, which yields a coupled systems of DAEs, see Chapter 2.3. In contrast to the case of ODE coupling, the co-simulation of DAEs might fail if a certain contraction condition is not fulfilled [2,3,20], Chapter 3.1.2. Here, the convergence as well as the speed of contraction, i.e., the number of repeated model simulations, depends on the design of the coupling interface [36], i.e., how the subsystems communicates with each other, and the computational order amongst others. One can show that the design of the coupling interface is the most crucial factor in co-simulation. The development of coupling interfaces with its application in electric networks is a major topic of this thesis.

For field/circuit coupled problems the standard approach of decoupling is to separate between the field and circuit part, i.e., cutting at EM device boundaries [36]. Here, the dedicated coupling variables in Chapter 2.3 are replaced by controlled sources (source coupling), [8, 20, 24]. This allows to define field and circuit subsystems which can be handled separately. Previous investigations have shown that co-simulation is always convergent as long as the iteration starts with the computation of the field subproblem, i.e., solving the curl-curl equation (2.12) first, e.g. [3, 8, 24]. However, our numerical investigations show that the coupling between field and circuit is commonly weak for the standard type of decoupling. Thus, oftentimes co-simulation requires a large number of iteration steps to ensure an accurate solution, i.e., each subsystem have to be solved many times. To reduce the number of iteration steps we propose new decoupling approaches, where convergence follows directly from the coupling interface [36, 37], Chapter 3.2. Thus, the convergence of co-simulation is independent from embedded electric circuits or EM devices. Fur-

thermore, numerical investigations show that using dedicated coupling interfaces can enhance the contraction, i.e., they reduce the computational effort in order to achieve a predefined level of accuracy, see Chapter 7.

Afterwards, we will introduce uncertainties into co-simulation (subsystems with uncertain components), [38]. This can be for example: uncertainties in resistors, conductors or inductors in electric circuits or production-related tolerances in electrical machines, e.g. the number of turns, the rotor/stator diameter or material impurities. Introducing uncertainties in this mode might affect the dynamic iteration and can destroy convergence. Hence, one is interested to analyze the stochastic behavior of co-simulation. Details are discussed in Chapter 6, [37].

This chapter is outlined as follows: Chapter 3.1 introduces into the different steps of co-simulation. These are the partitioning into time windows, the partitioning into subsystems as well as the extrapolation step followed by (one or more) dynamic iteration steps. Then, Chapter 3.1.1 and Chapter 3.1.2 introduce into the Gauss-Seidel-type iteration scheme as the main concept of dynamic iteration that we consider within this thesis. Then, Chapter 3.1.3 focus on co-simulation for field/circuit coupled problems and shows how to avoid divergence in practice. Applications to field/circuit coupled problems are given in Chapter 7.

3.1 Dynamic Iteration for Coupled Problems

Within this chapter it is assumed that the differential index of the resulting DAE (2.16) is at most one. This restriction can be assured for electric networks by avoiding LIM-cutsets and CV-loops [21], see Chapter 2.3. This allows to refer to the more general system (2.16) without losing its validity for the field/circuit coupled problem (2.15). Then by decomposing (reordering) the system into differential equations and algebraic constraints the coupled problem can be addressed as semi-explicit differential algebraic initial-value problem (DAE-IVP)

$$\dot{\mathbf{y}} = \mathbf{f}(t, \mathbf{y}, \mathbf{z}) \quad \text{with initial values} \quad \mathbf{y}(t_0) = \mathbf{y}_0, \quad (3.1a)$$

$$\mathbf{0} = \mathbf{g}(t, \mathbf{y}, \mathbf{z}) \quad \text{with initial values} \quad \mathbf{z}(t_0) = \mathbf{z}_0, \quad (3.1b)$$

where \mathbf{f} and \mathbf{g} are vector functions consisting of differential and algebraic unknowns

$$\mathbf{y} \in [t_0, t_e] \rightarrow \mathbb{R}^{n_y} \quad \text{and} \quad \mathbf{z} \in [t_0, t_e] \rightarrow \mathbb{R}^{n_z}. \quad (3.2)$$

The index-1 assumption implies that the DAE-IVP (3.1) has a non-singular Jacobian $\partial \mathbf{g} / \partial \mathbf{z}$ and the initial values \mathbf{y}_0 , \mathbf{z}_0 are consistent, i.e., they solve the algebraic equation (3.1b). Finally, the unknown solution vector can be summarized as:

$$\mathbf{x}^\top = [\mathbf{y}^\top, \mathbf{z}^\top] \quad \text{with} \quad \mathbf{x} \in [t_0, t_e] \rightarrow \mathbb{R}^{n_y + n_z}, \quad (3.3)$$

where the vector \mathbf{y} is composed of variables that are defined by the first derivatives with respect to time and \mathbf{z} is composed of variables that are not described by any derivatives. The DAE-IVP (3.1) solves the problem exact and is referred to as strongly coupled problem or monolithic problem.

Remark 2 (Partitioning for field/circuit coupled problems). *For system (2.15) and index-1 the differential and algebraic part consist of $\mathbf{y} = (\mathbf{q}, \phi, \mathbf{a})^\top$ and $\mathbf{z} = (\mathbf{u}, \mathbf{i}_L, \mathbf{i}_V)^\top$, with algebraic unknown node potentials \mathbf{u} and currents \mathbf{i}_L , \mathbf{i}_V and differential unknown charges \mathbf{q} , fluxes ϕ and magnetic vector potentials \mathbf{a} .*

The following part introduces into the different steps of co-simulation: the partitioning of the timeline into time windows, the partitioning into subsystems as well as the extrapolation step and iteration steps.

Step 1: Partitioning into Time Windows

To enhance convergence, the process of co-simulation works on certain time periods, where convergence can be achieved by solving multiple times the subsystems [3, 8, 20, 24]. To this end, the simulation time $[t_0, t_e]$ is split into time periods of length $[T_n, T_{n+1}]$ with

$$t_0 = T_0 < T_1 < \dots < T_n < T_{n+1} < \dots < T_N = t_e, \quad (3.4)$$

where the time-points T_n are termed as synchronization points or communication points of the co-simulation procedure. The information transport between the subsystems is only managed at the points T_n , ($n = 1, \dots, N$). Then, co-simulation operates on these time periods with communication step size $H_n := T_{n+1} - T_n$, where each subsystem may employ a dedicated time-integrator with step sizes h , see (2.17), on H_n to compute the solution of the coupled problem. The communication step sizes H_n are commonly referred to as time windows or macro steps in contrast to the micro steps h of the numerical time-integration method.

Step 2: Partitioning into Subsystems

In the general case, the strongly coupled problem (3.1) consists of r subsystems. Investigations about the co-simulation of r subsystems have been published in [3]. However, later we will restrict ourselves to $r = 2$ subsystems. This can be, for example, a field model (extracted from the underlying PDE after space discretization) coupled to a circuit model (extracted by MNA) [8, 24].

By reordering the system (3.1) into terms of differential and algebraic equations, the vector functions \mathbf{f} and \mathbf{g} can be decomposed as $\mathbf{f}^\top = [\mathbf{f}_1^\top, \dots, \mathbf{f}_r^\top]$ and $\mathbf{g}^\top = [\mathbf{g}_1^\top, \dots, \mathbf{g}_r^\top]$, where the i -th subproblem is defined as ($i = 1, \dots, r$) [20, 24]:

$$\dot{\mathbf{y}}_i = \mathbf{f}_i(t, \mathbf{y}_1^\top, \dots, \mathbf{y}_r^\top, \mathbf{z}_1^\top, \dots, \mathbf{z}_r^\top) \quad \text{with initial values} \quad \mathbf{y}_i(t_0) = \mathbf{y}_{i,0}, \quad (3.5a)$$

$$\mathbf{0} = \mathbf{g}_i(t, \mathbf{y}_1^\top, \dots, \mathbf{y}_r^\top, \mathbf{z}_1^\top, \dots, \mathbf{z}_r^\top) \quad \text{with initial values} \quad \mathbf{z}_i(t_0) = \mathbf{z}_{i,0}. \quad (3.5b)$$

Again, the index-1 assumption for each subproblem (3.5) implies the non-singularity of the Jacobian $\partial \mathbf{g}_i / \partial \mathbf{z}_i$ for $i = 1, \dots, r$. However, it is important to note that having ensured that each subproblem is at most of index-1 does not warrant index-1 for the coupled problem (3.1). This is demonstrated in [24].

The overall (consistent) initial value can be decomposed as well:

$$\mathbf{y}_0^\top = [\mathbf{y}_{1,0}^\top, \dots, \mathbf{y}_{r,0}^\top], \quad \mathbf{z}_0^\top = [\mathbf{z}_{1,0}^\top, \dots, \mathbf{z}_{r,0}^\top].$$

Now that we have split the strongly coupled system into r subsystems, we are prepared to start co-simulation on these time windows consisting of an extrapolation step followed by (one or more) iteration steps.

Step 3: Constant Extrapolation

Assuming that a solution (approximation) $\tilde{\mathbf{x}}$ of (3.3) has already been computed on $[T_{n-1}, T_n]$. To proceed with the next time window $[T_n, T_{n+1}]$, we employ constant extrapolation, see [2]:

$$\begin{pmatrix} \tilde{\mathbf{y}}|_{[T_n, T_{n+1}]}^{(0)}(t) \\ \tilde{\mathbf{z}}|_{[T_n, T_{n+1}]}^{(0)}(t) \end{pmatrix} = \begin{pmatrix} \tilde{\mathbf{y}}|_{[T_{n-1}, T_n]}(T_n) \\ \tilde{\mathbf{z}}|_{[T_{n-1}, T_n]}(T_n) \end{pmatrix} \quad \text{for all } t \in [T_n, T_{n+1}], \quad (3.6)$$

where the solution of the last time point from the previous time window serves as initial guess for constant extrapolation.

The splitting error for constant extrapolation (3.6) is of order $\mathcal{O}(H)$. It is possible to improve the order of co-simulation by using polynomials of higher degree for the extrapolation step, e.g. linear, quadratic, see [52]. However, constant extrapolation is the most common guess to start up co-simulation.

Step 4: Dynamic Iteration

The extrapolation step is followed by (one or more) iteration steps, where the waveform relaxation on time window $[T_n, T_{n+1}]$ is defined by iteration operators for the differential and algebraic unknown, see [2]:

$$\begin{aligned} \Psi|_{\mathbf{y}, [T_n, T_{n+1}]} : C^1([T_n, T_{n+1}], \mathbb{R}^{n_y}) &\rightarrow C^1([T_n, T_{n+1}], \mathbb{R}^{n_y}), \\ \Psi|_{\mathbf{z}, [T_n, T_{n+1}]} : C^0([T_n, T_{n+1}], \mathbb{R}^{n_z}) &\rightarrow C^0([T_n, T_{n+1}], \mathbb{R}^{n_z}). \end{aligned}$$

The updated solution for the k -th iteration on $[T_n, T_{n+1}]$ can be stated as

$$\begin{pmatrix} \tilde{\mathbf{y}}|_{[T_n, T_{n+1}]}^{(k-1)}(t) \\ \tilde{\mathbf{z}}|_{[T_n, T_{n+1}]}^{(k-1)}(t) \end{pmatrix} \rightarrow \begin{pmatrix} \tilde{\mathbf{y}}|_{[T_n, T_{n+1}]}^{(k)}(t) \\ \tilde{\mathbf{z}}|_{[T_n, T_{n+1}]}^{(k)}(t) \end{pmatrix} = \begin{pmatrix} \Psi|_{\mathbf{y}, [T_n, T_{n+1}]} \\ \Psi|_{\mathbf{z}, [T_n, T_{n+1}]} \end{pmatrix} \begin{pmatrix} \tilde{\mathbf{y}}|_{[T_n, T_{n+1}]}^{(k-1)}(t) \\ \tilde{\mathbf{z}}|_{[T_n, T_{n+1}]}^{(k-1)}(t) \end{pmatrix}, \quad (3.7)$$

where $(k-1)$ addresses the old iteration step.

Now, using constant extrapolation (3.6) and iteration (3.7), each subproblem can be handled as separate subsystem on time window H_n , where the partitioning can be managed by introducing controlled sources for the coupling variables (source coupling), see [24]. The related source coupling is discussed in Chapter 3.1.3.

The decoupling by controlled sources ensures to have each subsystem (3.5) defined by its own DAE-IVP ($i = 1, \dots, r$) with consistent initial values [20, 24]:

$$\begin{aligned} \dot{\tilde{\mathbf{y}}}_{i,n} &= \mathbf{F}_i(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}), & \text{with initial values } \tilde{\mathbf{y}}_{i,n}^{(k)}(T_n) &= \tilde{\mathbf{y}}_{i,n}^{(k-1)}(T_n), \\ 0 &= \mathbf{G}_i(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}), & \text{with initial values } \tilde{\mathbf{z}}_{i,n}^{(k)}(T_n) &= \tilde{\mathbf{z}}_{i,n}^{(k-1)}(T_n), \end{aligned} \quad (3.8)$$

where \mathbf{F}_i , \mathbf{G}_i are referred to as splitting functions. Consequently, for system (3.1), any co-

simulation procedure can be encoded by splitting functions \mathbf{F} , \mathbf{G} :

$$\begin{aligned} \dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{z}) &\leftrightarrow \dot{\tilde{\mathbf{y}}} = \mathbf{F}\left(\tilde{\mathbf{y}}^{(k)}, \tilde{\mathbf{y}}^{(k-1)}, \tilde{\mathbf{z}}^{(k)}, \tilde{\mathbf{z}}^{(k-1)}\right) \\ \mathbf{0} = \mathbf{g}(\mathbf{y}, \mathbf{z}) &\quad \mathbf{0} = \mathbf{G}\left(\tilde{\mathbf{y}}^{(k)}, \tilde{\mathbf{y}}^{(k-1)}, \tilde{\mathbf{z}}^{(k)}, \tilde{\mathbf{z}}^{(k-1)}\right), \end{aligned} \quad (3.9)$$

with $\mathbf{F}^\top = (\mathbf{F}_1^\top, \dots, \mathbf{F}_r^\top)$ and $\mathbf{G}^\top = (\mathbf{G}_1^\top, \dots, \mathbf{G}_r^\top)$. System (3.9) is referred to as weakly coupled, where the splitting functions \mathbf{F} , \mathbf{G} must fulfill the compatibility condition, see [2]:

$$\mathbf{F}(\mathbf{y}, \mathbf{y}, \mathbf{z}, \mathbf{z}) = \mathbf{f}(\mathbf{y}, \mathbf{z}) \quad \text{and} \quad \mathbf{G}(\mathbf{y}, \mathbf{y}, \mathbf{z}, \mathbf{z}) = \mathbf{g}(\mathbf{y}, \mathbf{z}). \quad (3.10)$$

Now, applying both steps, i.e., constant extrapolation (3.6) and dynamic iteration (3.7), for each time window H_n , ($i = 1, \dots, N$) on $[t_0, t_e]$. The dynamic iteration approach computes the solution of each subsystem (3.5) sequentially and iteratively on H_n . Therefore, each subsystem is solved with coupling terms fixed to the (best) so far known approximation. Then, co-simulation is stopped after a finite number of k iteration steps and yields the approximation

$$\tilde{\mathbf{x}}^{(k)} = \begin{pmatrix} \tilde{\mathbf{y}}^{(k)} \\ \tilde{\mathbf{z}}^{(k)} \end{pmatrix} \quad \text{with} \quad \tilde{\mathbf{y}}^{(k)} \in [t_0, t_e] \rightarrow \mathbb{R}^{n_y}, \quad \tilde{\mathbf{z}}^{(k)} \in [t_0, t_e] \rightarrow \mathbb{R}^{n_z}, \quad (3.11)$$

where the (overall) piecewise composed solution satisfies, [20],

$$\tilde{\mathbf{x}} \in C^1([t_0, t_e], \mathbb{R}^{n_y}) \times C^0([t_0, t_e], \mathbb{R}^{n_z}), \quad (3.12)$$

i.e., the differential unknowns \mathbf{y} are at least one times differentiable with respect to time and the algebraic unknowns \mathbf{z} are continuous.

Now, Chapter 3.1.1 discusses the Gauß-Seidel-type iteration scheme, where we set up the standard recursion estimate to derive conditions for a convergent iteration scheme.

3.1.1 Gauss-Seidel-Type Iteration Scheme

Using the Gauß-Seidel-type iteration scheme [20, 24], each time-integration for one part of the unknowns assumes values from another part of the unknowns to be available. For the other part of the unknowns it is similarly. Then, after integration over the time window, we have new time profiles for the unknowns of all parts of the partition. With these new time profiles we can restart the Gauss-Seidel-type iteration process over the same time window to further update the profiles. Thus the solution of the i -th subsystem on the n -th time window $\tilde{\mathbf{x}}|_{i, [T_n, T_{n+1}]}$ includes recent solutions (time profiles) and old time profiles supplied from the other subsystems. This yields the following definition of the Gauß-Seidel scheme, see [24].

Definition 1 (Gauss-Seidel-type Iteration, [24]). *Let $\tilde{\mathbf{y}}_{i,n}^{(k)}$, $\tilde{\mathbf{z}}_{i,n}^{(k)}$ and $\tilde{\mathbf{y}}_{i,n}^{(k-1)}$, $\tilde{\mathbf{z}}_{i,n}^{(k-1)}$ be the solution of the i -th subsystem on the n -th time window after a finite number of k and $k-1$ iteration steps. Consequently, recent solutions and old solutions supplied from other subsystems occur. The IVP for the i -th subsystem ($i = 1, \dots, r$) of the partition (3.8) define a Gauß-Seidel-type iteration scheme*

with encoded splitting functions

$$\mathbf{F}_i(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}) := \mathbf{f}_i(\tilde{\mathbf{y}}_{1,n}^{(k)}, \dots, \tilde{\mathbf{y}}_{i,n}^{(k)}, \tilde{\mathbf{y}}_{i+1,n}^{(k-1)}, \dots, \tilde{\mathbf{y}}_{r,n}^{(k-1)}, \tilde{\mathbf{z}}_{1,n}^{(k)}, \dots, \tilde{\mathbf{z}}_{i,n}^{(k)}, \tilde{\mathbf{z}}_{i+1,n}^{(k-1)}, \dots, \tilde{\mathbf{z}}_{r,n}^{(k-1)}), \quad (3.13a)$$

$$\mathbf{G}_i(\tilde{\mathbf{y}}_n^{(k)}, \tilde{\mathbf{y}}_n^{(k-1)}, \tilde{\mathbf{z}}_n^{(k)}, \tilde{\mathbf{z}}_n^{(k-1)}) := \mathbf{g}_i(\tilde{\mathbf{y}}_{1,n}^{(k)}, \dots, \tilde{\mathbf{y}}_{i,n}^{(k)}, \tilde{\mathbf{y}}_{i+1,n}^{(k-1)}, \dots, \tilde{\mathbf{y}}_{r,n}^{(k-1)}, \tilde{\mathbf{z}}_{1,n}^{(k)}, \dots, \tilde{\mathbf{z}}_{i,n}^{(k)}, \tilde{\mathbf{z}}_{i+1,n}^{(k-1)}, \dots, \tilde{\mathbf{z}}_{r,n}^{(k-1)}). \quad (3.13b)$$

Using the Gauß-Seidel-type iteration scheme, one has to decide, which of the r subsystems is computed first, where the computational order basically affects the iteration operators in (3.7).

For $r = 2$ subsystems, the following example shows the resulting splitting functions.

Example 1. We consider (3.1) to be split into $r = 2$ subsystems. Furthermore, it is assumed that the data exchange between the subsystems is solely organized by differential or algebraic unknowns (not mixed). This claim is ensured for all our engineering test examples in Chapter 7, since the subsystems are split in such a way that only a single variable is used for the information transport. Assuming that subsystem 1 is defined by algebraic equations (AE) only, e.g. a circuit with voltage, current sources and resistors, coupled with subsystem 2 given by an index-1 DAE, e.g. system (2.12). Starting Gauß-Seidel-type dynamic iteration for the AE-DAE coupled problem one has to define which subsystem is computed first.

For subsystem 2 (DAE) first, there are basically two possibilities: the old waveform $\tilde{\mathbf{z}}_1^{(k-1)}$ enters subsystem 2 by differential equation or algebraic constraint depending on the coupling. For simplification we skip the tilde in the following. The splitting functions (3.9) for both cases reads:

(old algebraic waveform enters differential equation)

$$\mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{bmatrix} \mathbf{f}_2(0, \boxed{\mathbf{z}_1^{(k-1)}}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \\ \mathbf{g}_1(0, \mathbf{z}_1^{(k)}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \\ \mathbf{g}_2(0, 0, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \end{bmatrix},$$

or (old algebraic waveform enters algebraic constraint)

$$\mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{bmatrix} \mathbf{f}_2(0, 0, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \\ \mathbf{g}_1(0, \mathbf{z}_1^{(k)}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \\ \mathbf{g}_2(0, \boxed{\mathbf{z}_1^{(k-1)}}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \end{bmatrix}.$$

For subsystem 1 (AE) first, there are again two possibilities: the old differential $\tilde{\mathbf{y}}_2^{(k-1)}$ or old algebraic waveform $\tilde{\mathbf{z}}_2^{(k-1)}$ enters algebraic constraint of subsystem 1. Then, the splitting functions (3.9) for both cases reads:

(old differential waveform enters algebraic constraint)

$$\mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \left[\mathbf{f}_2(0, \mathbf{z}_1^{(k)}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \right],$$

$$\mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{bmatrix} \mathbf{g}_1(0, \mathbf{z}_1^{(k)}, \boxed{\mathbf{y}_2^{(k-1)}}, \mathbf{z}_2^{(k)}) \\ \mathbf{g}_2(0, 0, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \end{bmatrix},$$

or (old algebraic waveform enters algebraic constraint)

$$\mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \left[\mathbf{f}_2(0, 0, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \right],$$

$$\mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{bmatrix} \mathbf{g}_1(0, \mathbf{z}_1^{(k)}, \mathbf{y}_2^{(k)}, \boxed{\mathbf{z}_2^{(k-1)}}) \\ \mathbf{g}_2(0, \mathbf{z}_1^{(k)}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \end{bmatrix}.$$

For contraction, i.e., that $\tilde{\mathbf{x}}|_{[T_n, T_{n+1}]}^{(k)}(t)$ is a better solution as $\tilde{\mathbf{x}}|_{[T_n, T_{n+1}]}^{(k-1)}(t)$, it matters if old differential or algebraic waveforms enters differential or algebraic equation [2, 20, 24]. Here, the wrong computational order can cause divergence. This is discussed in the next chapter, where we analyze the more general case of DAE-DAE coupling. Then, Chapter 3.1.3 treats the Gauss-Seidel iteration scheme with its application to field/circuit coupled problems (2.14). Here, the results for DAE-DAE coupling can be exploit to establish an always convergent co-simulation procedure.

3.1.2 DAE-DAE Coupling

As shown previously, co-simulation starts with an extrapolation step followed by iteration steps. However, the convergence of dynamic iteration requires to be already sufficiently close (in the neighborhood) to the solution of the strongly coupled problem (3.1) (exact solution). Here, using constant extrapolation as initial guess for the extrapolation step is mostly sufficient as long as the time window has been chosen small enough.

Now, convergence of a dynamic iteration is analyzed: Starting from the definition of the neighborhood of a solution, we define how to measure the distance of the approximation $\tilde{\mathbf{x}}$ to the exact solution \mathbf{x} . Following [20], the Gauss-Seidel-type dynamic iteration yields an estimation for the errors in the differential and algebraic variables of (3.11). Finally, we deduce different convergence rates for various splitting functions (3.9).

Definition 2 (Neighborhood). *Let $\mathbf{x}|_{[T_n, T_{n+1}]}$ be the exact solution on the n -th time window. The distance $d > 0$ from the exact solution defines a neighborhood*

$$\mathcal{U}_{n,d} := \{ \mathbf{x}|_{[T_n, T_{n+1}]} \in C^1([T_n, T_{n+1}], \mathbb{R}^{n_y}) \times C^0([T_n, T_{n+1}], \mathbb{R}^{n_z}) : \|\tilde{\mathbf{y}}_n - \mathbf{y}_n\|_{2,\infty}, \|\tilde{\mathbf{z}}_n - \mathbf{z}_n\|_{2,\infty} \leq d \}, \quad (3.14)$$

where $\|\cdot\|_{2,\infty} := \sup_{t \in [T_n, T_{n+1}]} \|\cdot\|_2$ denotes the maximum norm in time and L^2 -norm in space.

Now, starting from two solutions $\mathbf{X}, \tilde{\mathbf{X}}$ on the n -th time window $[T_n, T_{n+1}]$ and performing k

iteration steps. The distance between two waveforms is measured from the differences

$$\delta_{\mathbf{y},n}^{(k)} := \sup_{t \in [T_n, T_{n+1}]} \|\mathbf{Y}_n^{(k)}(t) - \tilde{\mathbf{Y}}_n^{(k)}(t)\|_2, \quad (3.15a)$$

$$\delta_{\mathbf{z},n}^{(k)} := \sup_{t \in [T_n, T_{n+1}]} \|\mathbf{Z}_n^{(k)}(t) - \tilde{\mathbf{Z}}_n^{(k)}(t)\|_2. \quad (3.15b)$$

For convergence, the differences tends to zero as k tends to infinity, i.e., the updated waveforms become closer to the exact solution. Figure 3.1 shows the solution in a node potential of a circuit obtained by Gauss-Seidel iteration on a fixed time window $H_n = [0.4\text{ms}, 0.5\text{ms}]$ for different iteration steps.

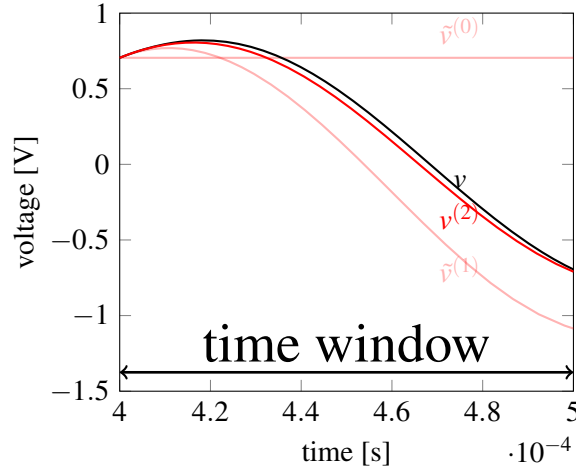


Figure 3.1: Dynamic iteration on time window $H_n = [4\text{ms}, 5\text{ms}]$. (red) solution (waveforms) for $k = 0, 1, 2$ iterations and (black) exact solution obtained by solving the monolithic model.

Now, for DAE-DAE coupling one can derive an recursion estimate for the dynamic iteration of two waveforms in $\mathcal{U}_{n,d}$.

Lemma 3 (Recursion Estimate, [20]). *Let a dynamic iteration (3.9) be given, with consistent splitting functions \mathbf{F} , \mathbf{G} , where \mathbf{F} , \mathbf{G} are Lipschitz-continuous on $\mathcal{U}_{n,d}$, \mathbf{G} is totally differentiable on $\mathcal{U}_{n,d}$ and $\mathbf{G}_{\mathbf{z}}^{(k)}$ is invertible on $\mathcal{U}_{n,d}$. Then there is a constant $C > 1$, such that for two waveforms $\mathbf{x}_n^{(k-1)}, \tilde{\mathbf{x}}_n^{(k-1)} \in \mathcal{U}_{n,d}$ on a time window $H_n < H_{\max}$ the following recursion estimate holds*

$$\begin{pmatrix} \delta_{\mathbf{y},n}^{(k)} \\ \delta_{\mathbf{z},n}^{(k)} \end{pmatrix} \leq \mathbf{K} \begin{pmatrix} \delta_{\mathbf{y},n}^{(k-1)} \\ \delta_{\mathbf{z},n}^{(k-1)} \end{pmatrix} + \text{"initial offset"}, \quad (3.16)$$

with recursion matrix and contraction factor

$$\mathbf{K} := \begin{pmatrix} CH_n & CH_n \\ C & CH_n + \alpha_n \end{pmatrix} \quad \text{and} \quad \alpha_n := (1 + Cd) \|\mathbf{G}_{\mathbf{z}^{(k)}}^{-1} \mathbf{G}_{\mathbf{z}^{(k-1)}}\|_{2,\infty} + Cd, \quad (3.17)$$

with Jacobian $\mathbf{G}_{\mathbf{z}^{(k)}} := \frac{\partial \mathbf{G}}{\partial \mathbf{z}^{(k)}}$ and $\mathbf{G}_{\mathbf{z}^{(k-1)}} := \frac{\partial \mathbf{G}}{\partial \mathbf{z}^{(k-1)}}$.

Proof. We only give a sketch of the proof. For those who are interested in details, see [20, 24]. The proof is basically split into the recursion estimate (3.16) for the differential and algebraic un-

knowns $\delta_{\mathbf{y},n}^{(k)}$ and $\delta_{\mathbf{z},n}^{(k)}$, respectively. The consistency (3.10) of the splitting functions \mathbf{F} , \mathbf{G} implies that the differential function \mathbf{F} is Lipschitz-continuous on the neighborhood (3.14) with Lipschitz constant $L_{\mathbf{F}} > 0$ and \mathbf{G} is totally differentiable on $\mathcal{U}_{n,d}$ with Lipschitz-continuous derivatives.

Inserting two waveforms $\mathbf{X}_n^{(k-1)}$, $\tilde{\mathbf{X}}_n^{(k-1)} \in \mathcal{U}_{n,d}$ yields two differential equations. Subtracting one from the other measures the differences at each time point $t \in H_n$. Now for the ODE-part after the integration for $T_n < \tau \leq T_{n+1}$, Lipschitz continuity yields the estimate

$$|\Delta_{\mathbf{y},n}^{(k)}(\tau)| \leq |\Delta_{\mathbf{y},n}^{(0)}(T_n)| + \int_{T_n}^{\tau} L_{\mathbf{F}}(|\Delta_{\mathbf{y},n}^{(k)}| + |\Delta_{\mathbf{y},n}^{(k-1)}| + |\Delta_{\mathbf{z},n}^{(k)}| + |\Delta_{\mathbf{z},n}^{(k-1)}|) dt. \quad (3.18)$$

For the algebraic unknowns with $\mathbf{Z}^{(k)} = \Phi(\mathbf{Y}^{(k)}, \mathbf{Y}^{(k-1)}, \mathbf{Z}^{(k-1)})$ and $\tilde{\mathbf{Z}}^{(k)} = \Phi(\tilde{\mathbf{Y}}^{(k)}, \tilde{\mathbf{Y}}^{(k-1)}, \tilde{\mathbf{Z}}^{(k-1)})$ the difference yields an estimate for the algebraic unknowns

$$|\Delta_{\mathbf{z},n}^{(k)}| \leq L_{\Phi}(|\Delta_{\mathbf{y},n}^{(k)}| + |\Delta_{\mathbf{y},n}^{(k-1)}| + |\Delta_{\mathbf{z},n}^{(k-1)}|), \quad (3.19)$$

with Lipschitz constant $L_{\Phi} > 0$. Now by inserting (3.19) into (3.18) yields (after reordering) the maximum upper bound for (3.15a) on the respective window H_n . For (3.15b) it is the same strategy. \square

Now, using (3.17) the condition for contraction yields a compact expression.

Lemma 4 (Contraction Condition, [20]). *Let the splitting functions \mathbf{F} , \mathbf{G} fulfill the assumption of Lemma 3. Then for $H_n < H_{\max}$ and d small enough the map (for all k)*

$$\begin{pmatrix} \delta_{\mathbf{y},n}^{(k-1)} \\ \delta_{\mathbf{z},n}^{(k-1)} \end{pmatrix} \rightarrow \begin{pmatrix} \delta_{\mathbf{y},n}^{(k)} \\ \delta_{\mathbf{z},n}^{(k)} \end{pmatrix}$$

is contractive for

$$\|\mathbf{G}_{\mathbf{z}^{(k)}}^{-1} \mathbf{G}_{\mathbf{z}^{(k-1)}}\|_{2,\infty} < 1. \quad (3.20)$$

Proof. For k iteration steps of (3.16) the rougher recursion estimate for the map $\delta^{(0)}(\mathbf{y}, \mathbf{z}) \rightarrow \delta^{(k)}(\mathbf{y}, \mathbf{z})$ implies the multiplication by the k -th power of the recursion matrix \mathbf{K} . Consequently, the map is contractive for a spectral radius $\rho(\mathbf{K}) < 1$. Thus, the claim for convergence is deduced by inspecting the eigenvalues of the recursion matrix \mathbf{K} :

$$\lambda_{1,2} = \frac{1}{2} \left(\alpha_n + 2CH_n \pm \sqrt{\alpha_n^2 + 4C^2H_n} \right). \quad (3.21)$$

See [20] for more details. \square

Remark 5. *Within Chapter 3.2 we will use the spectral radius condition to establish coupling interfaces with always contractive behavior. Thus, the claim is shifted from the structural properties of splitting functions to the network topology.*

The eigenvalues (3.21) determine the convergence rate of the dynamic iteration process. Consequently, the convergence rates are prescribed by the structural properties of the coupling interface. A close consideration of Lemma 3 and Lemma 4 yields the following:

Corollary 6 (Convergence Rates, [24]). *Given a splitting*

i) with

$$4C^2H_n < \alpha_n^2,$$

the dynamic iteration yields the convergence rate $\alpha_n + \mathcal{O}(H_n)$.

ii) with vanishing Jacobian

$$\mathbf{G}_{\mathbf{z}^{(k-1)}} = 0,$$

i.e., no algebraic constraint depends on old algebraic iterates, the contraction factor vanishes, i.e., $\alpha_n = 0$, (simple coupling) and the dynamic iteration process yields the convergence rate $\mathcal{O}(\sqrt{H_n})$.

iii) with vanishing Jacobian

$$\mathbf{G}_{\mathbf{z}^{(k-1)}} = 0 \quad \text{and} \quad \mathbf{G}_{\mathbf{y}^{(k-1)}} = 0,$$

i.e., no algebraic constraint depends on old iterates, the dynamic iteration process yields the convergence rate $\mathcal{O}(H_n)$.

In practice, convergence of co-simulation with at least convergence rate $\mathcal{O}(\sqrt{H_n})$ is intended, which is basically guaranteed for a vanishing contraction factor α_n . That is, the coupling interface must be designed in such a way that algebraic-to-algebraic coupling is avoided, i.e., that no algebraic constraint depends on old algebraic iterates. Then, for $H_n < H_{\max}$ the strength of coupling is basically controlled by the leading constant C in (3.17) having impact to the speed of contraction.

With application to field/circuit coupling the dynamic iteration is always convergent (for $H < H_{\max}$) as long as the co-simulation starts with the computation of the field first, Chapter 3.1.3, where the splitting into subsystems is done by cutting at the EM device boundaries. However, numerical investigations show that for the standard way of splitting the strength of coupling is commonly weak. Thus, within Chapter 3.2 we propose new dedicated coupling interfaces which may enhance the coupling strength. Therefore, the interfaces reduce the computational effort, since co-simulation requires less iteration steps to achieve a prescribed tolerance. Numerical investigations for various interfaces with application in field/circuit coupling are given in Chapter 7.

3.1.3 Field/Circuit Co-Simulation

Within this chapter the dynamic iteration for field/circuit coupled problems (2.14) is closely studied, where it is assumed that the circuit subsystem eqs. (2.14a) to (2.14e) is at most of index-1. Then, the system consisting of a circuit (denoted by subscript C) and field (denoted by subscript M) can be addressed as semi-explicit DAE-IVP:

$$\begin{aligned} \dot{\mathbf{y}}_M &= \mathbf{f}_M(t, \mathbf{z}_M, \mathbf{z}_C), \quad \text{with } \mathbf{y}_M(t_0) = \mathbf{y}_{M,0} & \dot{\mathbf{y}}_C &= \mathbf{f}_C(t, \mathbf{y}_C, \mathbf{z}_C), \quad \text{with } \mathbf{y}_C(t_0) = \mathbf{y}_{C,0} \\ \mathbf{0} &= \mathbf{g}_M(t, \mathbf{y}_M, \mathbf{z}_M), \quad \text{with } \mathbf{z}_M(t_0) = \mathbf{z}_{M,0} & \mathbf{0} &= \mathbf{g}_C(t, \mathbf{y}_C, \mathbf{z}_C, \mathbf{z}_M), \quad \text{with } \mathbf{z}_C(t_0) = \mathbf{z}_{C,0} \end{aligned} \quad (3.22)$$

where the differential and algebraic unknowns of the circuit and the field are summarized in their corresponding vectors \mathbf{y}_M , \mathbf{z}_M , \mathbf{y}_C and \mathbf{z}_C , [24].

Now, two basic concepts can be used for the coupling. This is the source coupling and the parameter coupling. However, within this work we focus only on the source coupling approach and refer for the parameter coupling to [24].

Source Coupling

In order to calculate the circuit and the field subsystem separately on H_n , the field must be supplied by an input signal which is defined by the circuit coupling node, whereas the circuit needs the current of the MQS device to ensure a balanced node \mathbf{v}_M . The idea is the following: having split the strongly coupled problem into a circuit and MQS device subsystem, one has to locate the coupling variables which are responsible for the information transport. These are the voltage drops $\mathbf{v}_M = \mathbf{A}_M^\top \mathbf{u}$ applied to the MQS device in (2.14f) and the current \mathbf{i}_M through the MQS device. The incidence matrix \mathbf{A}_M^\top is structured such that \mathbf{v}_M includes only the nodes of the circuit which are connect to the conductive surfaces, whereas \mathbf{A}_M stamps the current \mathbf{i}_M back to the circuit by the extended Kirchhoff's Current law (2.14a).

This can be done by time-dependend sources, such that each subsystem stands alone without open branches. Figure 3.2 shows the source coupling between circuit and field. The input for the field

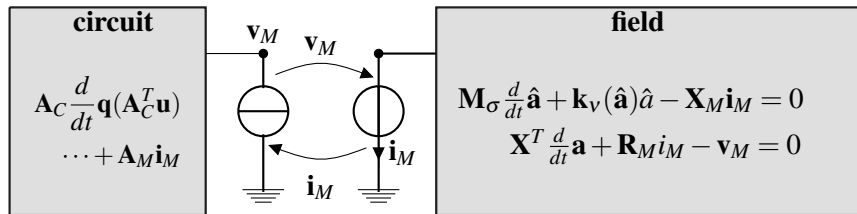


Figure 3.2: Source coupling between field and circuit.

on the n -th time window is given as

$$\mathbf{v}_M(t) = \mathbf{A}_M^\top \mathbf{u}^{(k-1)}, \quad \text{for } t \in [T_n, T_{n+1}], \quad (3.23)$$

where $\mathbf{u}^{(k-1)}$ is the solution of all node potentials of the circuit. Then $\mathbf{v}_M(t)$ is used to update the current through the MQS device \mathbf{i}_M , where the current waveform serves as data for the time-dependent current source of the circuit

$$\mathbf{i}_S(t) = \mathbf{i}_M^{(k)}(t), \quad \text{for } t \in [T_n, T_{n+1}] \quad (3.24)$$

to further update the node potentials.

Remark 7. *The coupling of the circuit with the MQS device as shown in Fig. 3.2 ensures to remain in the index-1 case. However, introducing time-dependent sources can change the differential index of the subsystems. For the case of a current source driven MQS device, the field subsystem becomes index-2, since the interface produces a LIM-cutset. Furthermore, the index-1 condition for the circuit is no longer ensured, since a capacitively grounded node \mathbf{v}_M produce a CV-loop, see Chapter 2.3.*

Splitting Schemes

For the field/circuit coupled problem different splitting functions are obviously possible. Their structure depends on the computational order (field or circuit first) and on the network topology. When the Gauß-Seidel-type dynamic iteration starts with the computation of the field first, the mutual algebraic dependence is avoided, since the old iterate of the circuit, whether differential

or algebraic, enters differential equation of the field (field first):

$$\mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{bmatrix} \mathbf{f}_M(\mathbf{z}_M^{(k)}, \boxed{\mathbf{y}_C^{(k-1)}, \mathbf{z}_C^{(k-1)}}) \\ \mathbf{f}_C(\mathbf{y}_C^{(k)}, \mathbf{z}_C^{(k)}) \end{bmatrix}, \quad (3.25a)$$

$$\mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{bmatrix} \mathbf{g}_M(\mathbf{y}_M^{(k)}, \mathbf{z}_M^{(k)}) \\ \mathbf{g}_C(\mathbf{y}_C^{(k)}, \mathbf{z}_C^{(k)}, \mathbf{z}_M^{(k)}) \end{bmatrix}, \quad (3.25b)$$

Therefore, the contraction factor α_n vanishes and convergence with window-wise convergence rate $\mathcal{O}(H_n)$ can be directly deduced from their related splitting functions, see Cor. 6 (iii).

Now, starting the computation with the circuit, the convergence depends on whether differential or algebraic equation is entered by the algebraic iterate \mathbf{i}_M of the MQS device (circuit first):

$$\mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{bmatrix} \mathbf{f}_M(\mathbf{z}_M^{(k)}, \mathbf{z}_C^{(k)}) \\ \mathbf{f}_C(\mathbf{y}_C^{(k)}, \mathbf{z}_C^{(k)}, \boxed{\mathbf{z}_M^{(k-1)}}) \end{bmatrix}, \quad (3.26a)$$

$$\mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{bmatrix} \mathbf{g}_M(\mathbf{y}_M^{(k)}, \mathbf{z}_M^{(k)}) \\ \mathbf{g}_C(\mathbf{y}_C^{(k)}, \mathbf{z}_C^{(k)}) \end{bmatrix}, \quad (3.26b)$$

or

$$\mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{bmatrix} \mathbf{f}_M(\mathbf{z}_M^{(k)}, \mathbf{z}_C^{(k)}) \\ \mathbf{f}_C(\mathbf{y}_C^{(k)}, \mathbf{z}_C^{(k)}) \end{bmatrix}, \quad (3.27a)$$

$$\mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) := \begin{bmatrix} \mathbf{g}_M(\mathbf{y}_M^{(k)}, \mathbf{z}_M^{(k)}) \\ \mathbf{g}_C(\mathbf{y}_C^{(k)}, \mathbf{z}_C^{(k)}, \boxed{\mathbf{z}_M^{(k-1)}}) \end{bmatrix}. \quad (3.27b)$$

Splitting (3.26) occurs for a capacitively grounded coupling node \mathbf{v}_M , since the node becomes a differential formulation and thus convergence is guaranteed. However, for a non-differential node \mathbf{v}_M the splitting yields mutual algebraic dependent subsystems (algebraic-to-algebraic coupling), see Cor. 6 (i), where convergence is only guaranteed if the contraction condition of Lemma 4 is fulfilled.

To summarize, the following applies to field/circuit coupled problems (3.22), [24]. Let all subsystems be at most of index-1. Using the source coupling with controlled sources (3.23) and (3.23) with voltage driven MQS device, the Gauß-Seidel-type dynamic iteration of Def. 1 for the field/circuit coupled problem (3.22) is unconditionally stable and convergent with convergence rate $\mathcal{O}(H_n)$, if

- i) the iteration starts with the computation of the MQS device.
- ii) the iteration starts with the computation of the circuit with capacitively grounded coupling node \mathbf{v}_M .

3.1.4 Conclusions

Within this chapter the fundamental steps of co-simulation were introduced: the timeline partitioning, the splitting into subsystems, the extrapolation as well as the dynamic iteration. Finally, based on the results of the DAE-DAE coupling and its corresponding recursion estimate, we state the condition for a convergent field/circuit co-simulation.

3.2 Coupling Interfaces

Recall that a convergent co-simulation scheme implies to have the spectral radius of the underlying recursion matrix \mathbf{K} in (3.16) smaller than one. Thus, based on the definition of the contraction factor (3.17), guaranteed convergence can be deduced by inspecting the corresponding splitting functions. Consequently, co-simulation is always convergent by avoiding the mutual algebraic dependence of the subsystems (algebraic-to-algebraic coupling). This can be always ensured for simple coupled or fully differential coupled subsystems, since the Jacobian $\mathbf{G}_{\mathbf{z}^{(k-1)}} = 0$ or $\mathbf{G}_{\mathbf{z}^{(k-1)}} = 0$ and $\mathbf{G}_{\mathbf{y}^{(k-1)}} = 0$ vanish, see Cor. 6. Otherwise, the contraction condition (3.20) is not automatically fulfilled and divergence might occur.

As shown in the previous chapter, the Gauß-Seidel scheme for field/circuit coupled problems is always convergent as long as the dynamic iteration process starts the computation with the MQS device. The reverse order (computation of the circuit first) is also unconditionally stable for a capacitively grounded coupling node. The resulting splitting functions are (3.25) and (3.26). Both splittings yield the fully differential coupling with window-wise convergence rate $\mathcal{O}(H_n)$, see Cor. 6 (iii).

Now, assume to have a splitting scheme that fulfills the contraction conditions of Cor. 6 (ii) or (iii). The eigenvalues of the iteration matrix \mathbf{K} is solely affected by the window size H_n and constant C , where C is an estimate for the leading coefficients and is indicator for the strength of the coupling, having impact to the speed of the contraction. Consequently, the smaller the constant C , the stronger the coupling. One can observe by numerical simulations that the coupling strength is mostly weak for system (3.22) with splitting (3.25) (split at the EM device boundaries and compute the field first), see Chapter 7.2. This is reflected in the number of repeated model simulations (number of iterations) of the field and circuit subsystem which are required to achieve a predefined accuracy.

An extension of the standard recursion estimate of Lemma 3 shows that the coupling strength is basically affected by components which are in the surrounding of the interface, i.e., close to the controlled sources. Using dedicated coupling interfaces may enhance the convergence properties and reduces the computational effort. Therefore, we develop new coupling interfaces with better performances (faster contraction), such that it will beat the standard coupling approach with splitting (3.25). These are the R-splitting and LR-coupling techniques, see Chapter 3.2.1 and Chapter 3.2.2. Here, general stability and contraction follows directly from the network structure. These properties are independent from embedded electric circuits or embedded EM devices, which enables to use the interfaces for various applications. A numerical comparison of both coupling approaches with the standard coupling approach is given in Chapter 7.2.

Furthermore, we show that the standard contraction theory is too coarse even for a simple circuit, Chapter 3.2.3. Therefore, we proof contraction via fine structure analysis, i.e., without estimation.

The result fits to the essential results of R-splitting and LR-coupling.

3.2.1 R-splitting

We expand the standard coupling interface with controlled sources by introducing or identifying an R-link in the respective location. The R-link may model a resistive wire with a capacitive coupling between the isolated potentials. Using decoupling with controlled time-dependent sources, see Chapter 3.1.3, this situation is sketched in Fig. 3.3, where network 1 (NW 1) and network 2 (NW 2) signify each an arbitrary network (circuit or EM device). Two resistances in series is rather academic in practice. In fact, R-splitting can be introduced by split a lumped resistance R into two resistances in series, where the shift is done by a ratio parameter ω :

$$R_{\text{NW } 1} := \omega R \quad \text{and} \quad R_{\text{NW } 2} := (1 - \omega)R \quad \text{with} \quad \omega \in (0, 1). \quad (3.28)$$

Assumption 1. *It is assumed that NW 1 and NW 2 is at most of differential index-1.*

Consequently, by identifying an R-link, both subsystems remain index-1 with non-singular Jacobian $\partial \mathbf{g}_1 / \partial \mathbf{z}_1$ and $\partial \mathbf{g}_2 / \partial \mathbf{z}_2$, since neither LIM-cutset nor CV-loop are caused by controlled sources.

Since the coupling node U_{Co} as well as the coupling current I_{Co} is of algebraic type, an old algebraic constraint depends on an old algebraic iterate. Thus, R-splitting creates a mutual algebraic dependency between the subsystems, such that a contraction factor α_n occurs for both computational sequences, see Cor. 6 (i). Consequently, (3.20) is not trivially fulfilled and convergence is only ensured for $\alpha_n < 1$. Again, the coupled problem can be abstractly addressed by the splitting functions

$$\begin{aligned} \mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) &:= \begin{bmatrix} \mathbf{f}_1(\mathbf{y}_1^{(k)}, \mathbf{z}_1^{(k)}, 0, 0) \\ \mathbf{f}_2(0, 0, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \end{bmatrix}, \\ \mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) &:= \begin{bmatrix} \mathbf{g}_1(\mathbf{y}_1^{(k)}, \mathbf{z}_1^{(k)}, 0, \boxed{\mathbf{z}_2^{(k-1, k)}}) \\ \mathbf{g}_2(0, \boxed{\mathbf{z}_1^{(k, k-1)}}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \end{bmatrix}, \end{aligned} \quad (3.29)$$

where the boxes denote the old algebraic iterate for subs. 1 first or subs. 2 first, respectively.

Now the standard recursion estimate (3.16) with lumped errors for the algebraic and differential variables will be improved. Using the R-splitting strategy, the contraction factor α_n is defined by the ratio of the coupling resistances $R_{\text{NW } 1}$ and $R_{\text{NW } 2}$, which enables to improve the convergence rate.

Extended Splitting Functions

In contrast to address the coupled problem via general splitting functions (3.29) we also take additional differential equations and algebraic constraints into account, which are defined by the components in the surrounding of the controlled sources. These are the equations for the coupling unknowns U_{Co} and I_{Co} and the constraint for the node potential U_1 (of subs.1 and subs. 2):

$$\begin{aligned} 0 &= (U_{B_1, \mathbf{y}_1} - U_{Co, \bar{\mathbf{z}}_1}) \cdot G_{\text{NW } 1} + I_{Co, \bar{\mathbf{z}}_1}(t) \quad \text{and} \quad 0 = (U_{1, \bar{\mathbf{z}}_2} - U_{B_2, \mathbf{y}_2}) \cdot G_{\text{NW } 2} + I_{Co, \bar{\mathbf{z}}_2}, \\ 0 &= U_{1, \bar{\mathbf{z}}_2} - U_{Co, \bar{\mathbf{z}}_2}(t). \end{aligned} \quad (3.30)$$

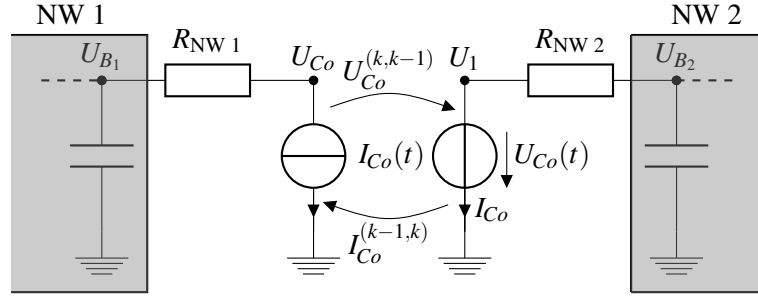


Figure 3.3: R-splitting between two subsystems.

Here, \mathbf{y}_1 , \mathbf{y}_2 and \mathbf{z}_1 , \mathbf{z}_2 refer to the unknowns of NW 1 and NW 2, whereas the subscript bar-notation is used to address explicitly to the additional unknowns which are located in the outside of NW 1 and NW 2. Furthermore, the conductances $G_{NW 1}$ and $G_{NW 2}$ are the inverse of the resistances $R_{NW 1}$ and $R_{NW 2}$. The embedding (coupling) of NW 1 and NW 2 to subs. 1 and subs. 2 is done by the additional current terms

$$\mathbf{A}_{f_1} \cdot G_{NW 1} U_{Co, \bar{\mathbf{z}}_1}, \quad \mathbf{A}_{f_2} \cdot G_{NW 2} U_{1, \bar{\mathbf{z}}_2}, \quad (3.31)$$

where the incidence matrices \mathbf{A}_{f_1} and \mathbf{A}_{f_2} stamp the currents to their respective branch equation of NW 1 and NW 2. Now, the procedure is the same as for the extended Kirchhoff's Current law (2.3) to couple MQS devices with circuits. Note that these are the currents flowing into the gates of NW 1 and NW 2 and that they enter differential equation for capacitively grounded nodes U_{B_1} and U_{B_2} .

Using eq. (3.30) and currents (3.31), the following (extended) splitting scheme defines the partitioned system (for subs. 1 first):

$$\mathbf{F} \left(\mathbf{y}^{(k)}, \mathbf{z}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k-1)} \right) := \begin{bmatrix} \mathbf{f}_1(\mathbf{y}_1^{(k)}, \mathbf{z}_1^{(k)}) + \mathbf{A}_{f_1} \cdot G_{NW 1} \cdot U_{Co, \bar{\mathbf{z}}_1}^{(k)} \\ \mathbf{f}_2(\mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) + \mathbf{A}_{f_2} \cdot G_{NW 2} \cdot U_{1, \bar{\mathbf{z}}_2}^{(k)} \end{bmatrix},$$

$$\mathbf{G} \left(\mathbf{y}^{(k)}, \mathbf{z}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k-1)} \right) := \begin{bmatrix} \mathbf{g}_1(\mathbf{y}_1^{(k)}, \mathbf{z}_1^{(k)}) \\ (U_{B_1, \mathbf{y}_1}^{(k)} - U_{Co, \bar{\mathbf{z}}_1}^{(k)}) \cdot G_{NW 1} + I_{Co, \bar{\mathbf{z}}_1}^{(k-1)}(t) \\ \mathbf{g}_2(\mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \\ (U_{1, \bar{\mathbf{z}}_2}^{(k)} - U_{B_2, \mathbf{y}_2}^{(k)}) \cdot G_{NW 2} + I_{Co, \bar{\mathbf{z}}_2}^{(k)} \\ U_{1, \bar{\mathbf{z}}_2}^{(k)} - U_{Co, \bar{\mathbf{z}}_2}^{(k)} \end{bmatrix}, \quad (3.32)$$

where \mathbf{f}_1 , \mathbf{f}_2 includes the ODE-part and \mathbf{g}_1 , \mathbf{g}_2 includes the algebraic constraints of NW 1 and NW 2. Now, we are prepared to state an extended version of (3.16).

Theorem 8 (Extended Recursion Estimate for R-splitting). *Let a dynamic iteration (3.9) be given, with consistent splitting functions \mathbf{F} and \mathbf{G} , where \mathbf{F} , \mathbf{G} are Lipschitz-continuous on $\mathcal{U}_{n,d}$ and \mathbf{G} is totally differentiable on $\mathcal{U}_{n,d}$. Then, using the R-splitting approach with capacitively grounded nodes U_{B_1} and U_{B_2} , there are constants*

$$C := \left(\frac{G_{NW 2}^2}{G_{NW 1} [1 - L_f(1 + L_\Phi)H_n]} + \frac{G_{NW 2}^2 H_n - G_{NW 2}}{1 - (L_f(1 + L_\Phi) + G_{NW 1})H_n} \right) \quad (3.33)$$

and $C_{\mathbf{y}_1}$, $C_{\mathbf{z}_1}$, $C_{\mathbf{y}_2}$, $C_{\mathbf{z}_2}$, $C_{U_{Co}}$, $C_{U_1} > 0$, where L_Φ , L_f are Lipschitz constants and $G_{NW 1}$, $G_{NW 2}$ are

the conductances of the R-splitting interface, such that for two waveforms $\mathbf{x}_n^{(k-1)}, \tilde{\mathbf{x}}_n^{(k-1)} \in \mathcal{U}_{n,d}$ on a time window $H_n < H_{\max}$ the following extended recursion estimate holds:

$$\begin{pmatrix} \delta_{\mathbf{y}_1}^{(k)} \\ \delta_{\mathbf{z}_1}^{(k)} \\ \delta U_{Co, \bar{\mathbf{z}}_1}^{(k)} \\ \delta_{\mathbf{y}_2}^{(k)} \\ \delta_{\mathbf{z}_2}^{(k)} \\ \delta U_{1, \bar{\mathbf{z}}_2}^{(k)} \\ \delta I_{Co, \bar{\mathbf{z}}_2}^{(k)} \end{pmatrix} \leq \mathbf{K}_e \begin{pmatrix} \delta_{\mathbf{y}_1}^{(k-1)} \\ \delta_{\mathbf{z}_1}^{(k-1)} \\ \delta U_{Co, \bar{\mathbf{z}}_1}^{(k-1)} \\ \delta_{\mathbf{y}_2}^{(k-1)} \\ \delta_{\mathbf{z}_2}^{(k-1)} \\ \delta U_{1, \bar{\mathbf{z}}_2}^{(k-1)} \\ \delta I_{Co, \bar{\mathbf{z}}_2}^{(k-1)} \end{pmatrix} + \text{"initial offset"}, \quad (3.34)$$

with extended recursion matrix and contraction factor

$$\mathbf{K}_e = \begin{pmatrix} 0 & \dots & & \dots & C_{\mathbf{y}_1} H_n \\ \vdots & 0 & & & C_{\mathbf{z}_1} H_n \\ & & \ddots & & C_{U_{Co}} \\ & & & & C_{\mathbf{y}_2} H_n \\ & & & & C_{\mathbf{z}_2} H_n \\ \vdots & & & & C_{U_1} \\ 0 & \dots & & \dots & \alpha_n + CH_n \end{pmatrix} \quad \text{and} \quad \alpha_n = \frac{G_{NW2}}{G_{NW1}} \quad (3.35)$$

Proof. Basically we extend the proof of Lemma 3 by taking additional terms of R-splitting into account. The proof is split into two parts: starting with subs. 1 the estimation technique is applied for all unknowns. Then the estimation is done for subs. 2 while we use the inequalities of the coupling variables to link both subsystems together. Due to the symmetry of the R-splitting interface, the estimation also holds for the computational sequence of subs. 2 first. For simplification we use the notation G_1, G_2 instead of G_{NW1}, G_{NW2} .

Subs. 1: Assume that the co-simulation starts with the computation of subs. 1 first. For the algebraic unknowns of NW1, with $\mathbf{z}_1 = \Phi_1(\mathbf{y}_1)$, the estimation

$$|\Delta_{\mathbf{z}_1}^{(k)}| \leq L_\Phi |\Delta_{\mathbf{y}_1}^{(k)}|, \quad (3.36)$$

holds, where L_Φ is the maximum of the Lipschitz constants of ϕ_i with respect to \mathbf{y}_i and \mathbf{z}_i . For the ODE-part after the integration for $T_n < \tau \leq T_{n+1}$, Lipschitz continuity yields

$$|\Delta_{\mathbf{y}_1}^{(k)}| \leq |\Delta_{\mathbf{y}_1}^{(k-1)}(t_n)| + \int_{T_n}^{\tau} L_f (|\Delta_{\mathbf{y}_1}^{(k)}| + |\Delta_{\mathbf{z}_1}^{(k)}|) + G_1 \cdot |\Delta U_{Co, \bar{\mathbf{z}}_1}^{(k)}| dt, \quad (3.37)$$

with Lipschitz constant L_f (assumed by Lipschitz continuity, consistency of \mathbf{F}). Using the algebraic constraint (3.30) for the coupling node U_{Co} and solving for $\delta_{\mathbf{y}_1}^{(k)}$ yields

$$\begin{aligned} \delta_{\mathbf{y}_1}^{(k)} &\leq \frac{1}{1 - (L_f(1 + L_\Phi) + G_1)H_n} \cdot |\Delta_{\mathbf{y}_1}^{(k-1)}(t_n)| \\ &+ \frac{1}{1 - (L_f(1 + L_\Phi) + G_1)H_n} \cdot H_n \cdot \delta I_{Co, \bar{\mathbf{z}}_1}^{(k-1)}, \end{aligned} \quad (3.38)$$

and for the algebraic part of NW1

$$\begin{aligned} \delta_{z_1}^{(k)} &\leq \frac{L_\phi}{1 - (L_f(1 + L_\Phi) + G_1)H_n} \cdot |\Delta_{y_1}^{(k-1)}(t_n)| \\ &\quad + \frac{L_\phi}{1 - (L_f(1 + L_\Phi) + G_1)H_n} \cdot H_n \cdot \delta I_{Co, \bar{z}_1}^{(k-1)}. \end{aligned} \quad (3.39)$$

Again, starting from the algebraic constraint of U_{Co} and using the estimation $|\Delta U_{B_1, y_1}^{(k)}| \leq |\Delta_{y_1}^{(k)}|$ yields

$$\begin{aligned} \delta U_{Co, \bar{z}_1}^{(k)} &\leq \frac{1}{1 - (L_f(1 + L_\Phi) + G_1)H_n} \cdot |\Delta_{y_1}^{(k-1)}(t_n)| \\ &\quad + \left(\frac{1}{G_1} + \frac{1}{1 - (L_f(1 + L_\Phi) + G_1)H_n} \cdot H_n \right) \cdot \delta I_{Co, \bar{z}_1}^{(k-1)}. \end{aligned} \quad (3.40)$$

Subs. 2: Similarly, by using the same technique for subsystem 2, the corresponding inequalities for (3.36) and (3.37) reads:

$$|\Delta_{z_2}^{(k)}| \leq L_\Phi |\Delta_{y_2}^{(k)}|, \quad (3.41)$$

$$|\Delta_{y_2}^{(k)}| \leq |\Delta_{y_2}^{(k-1)}(t_n)| + \int_{T_n}^{\tau} L_f (|\Delta_{y_2}^{(k)}| + |\Delta_{z_2}^{(k)}|) + G_2 \cdot |\Delta U_{1, \bar{z}_2}^{(k)}| dt. \quad (3.42)$$

With the algebraic equation $U_1 = U_{Co}$ and (3.40) we obtain

$$\begin{aligned} \delta_{y_2}^{(k)} &\leq \frac{1}{1 - L_f(1 + L_\Phi)H_n} \cdot |\Delta_{y_2}^{(k-1)}(t_n)| \\ &\quad + \frac{G_2}{(1 - (L_f(1 + L_\Phi) + G_1)H_n)(1 - L_f(1 + L_\Phi)H_n)} \cdot H_n \cdot |\Delta_{y_1}^{(k-1)}(t_n)| \\ &\quad + \left(\frac{1}{1 - L_f(1 + L_\Phi)H_n} \cdot \frac{G_2}{G_1} + \frac{G_2}{(1 - (L_f(1 + L_\Phi) + G_1)H_n)(1 - L_f(1 + L_\Phi)H_n)} \cdot H_n \right) \cdot H_n \cdot \delta I_{Co, \bar{z}_1}^{(k-1)} \end{aligned} \quad (3.43)$$

And for the algebraic part of NW2

$$\begin{aligned} \delta_{z_2}^{(k)} &\leq \frac{L_\phi}{1 - L_f(1 + L_\Phi)H_n} \cdot |\Delta_{y_2}^{(k-1)}(t_n)| \\ &\quad + \frac{L_\phi G_2}{(1 - (L_f(1 + L_\Phi) + G_1)H_n)(1 - L_f(1 + L_\Phi)H_n)} \cdot H_n \cdot |\Delta_{y_1}^{(k-1)}(t_n)| \\ &\quad + \left(\frac{L_\phi}{1 - L_f(1 + L_\Phi)H_n} \cdot \frac{G_2}{G_1} + \frac{L_\phi G_2}{(1 - (L_f(1 + L_\Phi) + G_1)H_n)(1 - L_f(1 + L_\Phi)H_n)} \cdot H_n \right) \cdot H_n \cdot \delta I_{Co, \bar{z}_1}^{(k-1)} \end{aligned} \quad (3.44)$$

Starting from the algebraic constraint of the coupling current I_{Co} . With $|\Delta U_{B_1, y_2}^{(k)}| \leq |\Delta_{y_1}^{(k)}|$ and the

algebraic equation $U_1 = U_{Co}$, where we use (3.40) as estimation, we obtain for the diagonal link:

$$\begin{aligned}
\delta I_{Co, \bar{z}_2}^{(k)} &\leq \frac{G_2}{1 - L_f(1 + L_\Phi)H_n} |\Delta_{y_2}^{(k-1)}(t_n)| \\
&\quad - \frac{G_2}{1 - (L_f(1 + L_\Phi) + G_1)H_n} \cdot |\Delta_{y_1}^{(k-1)}(t_n)| \\
&\quad + \frac{G_2^2}{(1 - L_f(1 + L_\Phi)H_n)(1 - (L_f(1 + L_\Phi) + G_1)H_n)} \cdot H_n \cdot |\Delta_{y_1}^{(k-1)}(t_n)| \\
&\quad + \left[\underbrace{\frac{G_2}{G_1}}_{=\alpha_n} + \underbrace{\left(\frac{G_2^2}{G_1 [1 - L_f(1 + L_\Phi)H_n]} + \frac{\frac{G_2^2}{1 - L_f(1 + L_\Phi)H_n} H_n - G_2}{1 - (L_f(1 + L_\Phi) + G_1)H_n} \right)}_{=C} \right] \cdot H_n \delta I_{Co, \bar{z}_2}^{(k-1)}.
\end{aligned} \tag{3.45}$$

This concludes the proof. \square

Now, analogously to Lemma 4 one can define a condition such that co-simulation behaves contractive.

Lemma 9 (Contraction Condition). *Let Ass. 1 be fulfilled and let $H_n < H_{\max}$ and d small enough. Then, for a co-simulation with R-splitting technique the map (for all k)*

$$\left(\delta_{y_1}^{(k-1)}, \delta_{z_1}^{(k-1)}, \dots, \delta_{y_2}^{(k-1)}, \delta_{z_2}^{(k-1)} \right)^\top \rightarrow \left(\delta_{y_1}^{(k)}, \delta_{z_1}^{(k)}, \dots, \delta_{y_2}^{(k)}, \delta_{z_2}^{(k)} \right)^\top$$

is contractive for

$$G_{NW2} < G_{NW1} \tag{3.46}$$

independently of the type of NW 1 and NW 2.

Proof. The proof is based on inspecting the eigenvalues of the extended recursion matrix \mathbf{K}_e . The information transport is solely managed by one unknown. Consequently, the recursion matrix \mathbf{K}_e is an upper triangular matrix. Its eigenvalues are the diagonal elements

$$\lambda_{1, \dots, 6} = 0, \quad \lambda_7 = \frac{G_{NW2}}{G_{NW1}} + CH_n, \tag{3.47}$$

which implies for the spectral radius $\rho(\mathbf{K}_e) = \frac{G_{NW2}}{G_{NW1}} + CH_n$. This concludes the proof. \square

Thus, the strength of the coupling is determined by the ratio of the coupling resistances. Here, the constant C as well as α_n can be controlled by resistance splitting. That is, for an increasing resistance R_2 the contraction factor α_n and the constant C decreases. Thus, the spectral radius can be reduced close to zero for a suitable choice of the ratio parameter ω in (3.28).

Typically, the dynamic iteration for field/circuit coupled problems starts with the computation of the field, since the contraction factor vanishes and co-simulation is unconditional stable for splitting 3.25, see Chapter 3.1.3. However, R-splitting provides much faster contraction and accepts much larger time window sizes for convergence.

The question arises how to choose the ratio for optimal convergence. Our investigations suggest that a shift of the full resistance R to subs. 2, i.e., $R_{NW1} = 0$, $R_{NW2} = R$, yields optimal con-

vergence, since the contraction factor α_n vanishes and the constant C is kept as small as possible. However, such a split would violate the underlying splitting functions (3.32), since the old algebraic iterate of subs.2 enters the differential equation of subs. 1 (node U_{B_1}). Therefore, the convergence with window-wise convergence rate $\mathcal{O}(H_n)$ is guaranteed, see Cor. 6 (iii) (simple coupling), and the corresponding splitting functions read (for subs.1 first):

$$\begin{aligned} \mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) &:= \begin{bmatrix} \mathbf{f}_1(\mathbf{y}_1^{(k)}, \mathbf{z}_1^{(k)}, 0, \mathbf{z}_2^{(k-1)}) \\ \mathbf{f}_2(0, 0, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \end{bmatrix}, \\ \mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) &:= \begin{bmatrix} \mathbf{g}_1(\mathbf{y}_1^{(k)}, \mathbf{z}_1^{(k)}, 0, 0) \\ \mathbf{g}_2(0, \mathbf{z}_1^{(k)}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \end{bmatrix}. \end{aligned} \quad (3.48)$$

Again, using eq. (3.30) and currents (3.31), the extended splitting scheme defines the partitioned system (for subs. 1 first). Following the proof of Theorem 8 yields the same structure for the constant C but $\alpha_n = 0$ (as expected). Consequently, the optimal ratio parameter for R-splitting is $\omega = 0$.

Assumption 2 (Avoiding Index-2). *The data exchange between the subsystem is organized via controlled current and voltage sources. Note that for the limits of the ratio parameter ($\omega = 0$, $\omega = 1$), the subsystem which is supplied by the voltage source becomes index-2 if a CV-loop is created. Therefore, it is assumed that the sources are placed in such a way that all subsystems remain an index-1 problem.*

Practical aspects: the current I_{Co} is maximal damped within the first iteration of co-simulation. This becomes important for the co-simulation of MQS devices coupled to a circuit, where the inrush current is typically large (several times their normal full-load current) for a few cycles of the input waveform. Figure 3.4 shows how the R-splitting approach may affects the inrush current when a transformer is energized for the first time. With R-splitting (Fig. 3.4 (left)) the current behaves damped and the co-simulation model is fully updated after one iteration step (no splitting error) such that the waveforms for further iterations steps remains the same. However, using the standard way of field/circuit coupling (Fig. 3.4 (right)) the current reaches two times their normal full-load current. Thus co-simulation requires several iteration steps to be close to the reference solution. Numerical investigations with application in field/circuit coupling are given in Chapter 7.2.1.

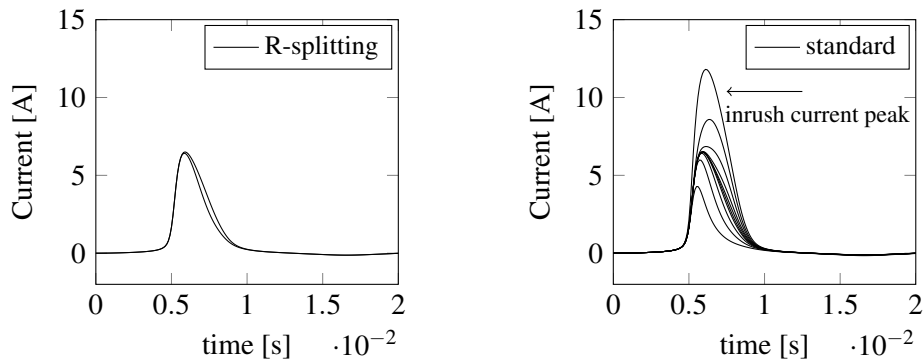


Figure 3.4: Inrush current of a transformer computed by co-simulation with $H_n = 0.01s$ and $k = 1, \dots, 20$ iteration steps. (left) for R-splitting with $G_{NW2} \ll G_{NW1}$ (fast contraction). (right) for the standard splitting (slow contraction).

3.2.2 LR-coupling

Similar to the previous chapter, we expand the coupling interface by introducing or identifying an LR-link consisting of two resistances and one inductance, which may model some physical wire in the respective location with additional inductive parts. Again, using decoupling with controlled sources, this situation is sketched in Fig. 3.5, where NW 1 and NW 2 signify each an arbitrary network (circuit or EM device). Note that for both computational sequences, i.e., subs. 1 or subs. 2 first, an algebraic iterate enters algebraic equation which creates again a mutual algebraic dependency of the subsystems. Thus, the splitting (3.29) is also valid for the LR-coupling approach and the standard recursion estimate yields the general coupling of Cor. 6 (i), i.e., convergence with window-wise convergence rate $\alpha_n + \mathcal{O}(H_n)$. However, the extended recursion estimate shows for the LR-coupling that the contraction factor α_n vanishes and convergence with window-wise convergence rate $\mathcal{O}(H_n)$ is guaranteed. Therefore, it is thus obviously that the standard recursion estimate might be too coarse to detect convergence.

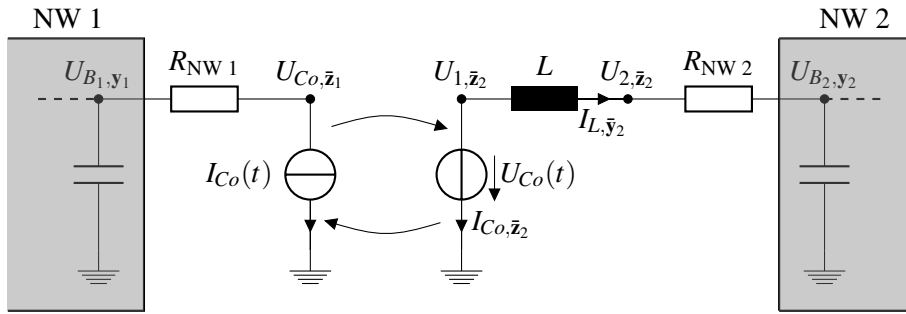


Figure 3.5: LR-coupling between two subsystems. [36]

Extended Splitting Functions

The additional equations for subs.1 and subs. 2 are

$$0 = (U_{B_1, y_1} - U_{Co, \bar{z}_1}) \cdot G_{NW 1} + I_{Co, \bar{z}_1}(t) \quad \text{and} \quad 0 = I_{Co, \bar{z}_2} + I_{L, \bar{y}_2} \quad (3.49a)$$

$$0 = (U_{2, \bar{z}_2} - U_{B_2, y_2}) \cdot G_{NW 2} - I_{L, \bar{y}_2}, \quad (3.49b)$$

$$0 = U_{1, \bar{z}_2} + U_{Co, \bar{z}_2}(t), \quad (3.49c)$$

where the current through the inductance is a differential unknown with respect to time:

$$\frac{d}{dt} I_L = \frac{U_1 - U_2}{L}. \quad (3.50)$$

Similar to the R-splitting approach, the coupling of NW 1 and NW 2 to subs. 1 and subs. 2 is done by the additional currents terms

$$\mathbf{A}_{f_1} \cdot G_{NW 1} U_{Co, \bar{z}_1} \quad \text{and} \quad \mathbf{A}_{f_2} \cdot G_{NW 2} U_{2, \bar{z}_2}, \quad (3.51)$$

with incidence matrices \mathbf{A}_{f_1} and \mathbf{A}_{f_2} . The following (extended) splitting scheme defines the partitioned system for subs. 1 first:

$$\mathbf{F} \left(\tilde{\mathbf{y}}^{(k)}, \tilde{\mathbf{z}}^{(k)}, \tilde{\mathbf{y}}^{(k-1)}, \tilde{\mathbf{z}}^{(k-1)} \right) := \begin{bmatrix} \mathbf{f}_1(\mathbf{y}_1^{(k)}, \mathbf{z}_1^{(k)}) + \mathbf{A}_{f_1} \cdot G_{NW1} \cdot U_{Co, \bar{z}_1}^{(k)} \\ \mathbf{f}_2(\mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) + \mathbf{A}_{f_2} \cdot G_{NW2} \cdot U_{2, \bar{z}_2}^{(k)} \\ (U_{2, \bar{z}_2}^{(k)} - U_{1, \bar{z}_2}^{(k)})/L \end{bmatrix}$$

$$\mathbf{G} \left(\tilde{\mathbf{y}}^{(k)}, \tilde{\mathbf{z}}^{(k)}, \tilde{\mathbf{y}}^{(k-1)}, \tilde{\mathbf{z}}^{(k-1)} \right) := \begin{bmatrix} \mathbf{g}_1(\mathbf{y}_1^{(k)}, \mathbf{z}_1^{(k)}) \\ (U_{B_1, \mathbf{y}_1}^{(k)} - U_{Co, \bar{z}_1}^{(k)}) \cdot G_{NW1} + I_{Co, \bar{z}_1}^{(k-1)}(t) \\ \mathbf{g}_2(\mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \\ I_{Co, \bar{z}_2}^{(k)} + I_{L, \bar{y}_2}^{(k)} \\ (U_{2, \bar{z}_2}^{(k)} - U_{B_2, \mathbf{y}_2}^{(k)}) \cdot G_{NW2} - I_{L, \bar{y}_2}^{(k)} \\ U_{1, \bar{z}_2}^{(k)} + U_{Co, \bar{z}_2}^{(k)}(t) \end{bmatrix} \quad (3.52)$$

with ODE-part \mathbf{f}_1 , \mathbf{f}_2 and algebraic constraints \mathbf{g}_1 , \mathbf{g}_2 of NW 1 and NW 2.

Theorem 10 (Extended Recursion Estimate for LR-coupling). *Let all assumptions of Theorem 8 be fulfilled. Then, using LR-coupling there is a constant*

$$C_{I_{Co}} := c \cdot \left(\frac{L_f}{G_{NW1}} + CH_n \right), \quad (3.53)$$

with $C := c \cdot L_f / (1 - C_0 H_n - G_{NW1} H_n)$ and $c := 1 / \left(1 - C_0 H_n + \frac{L_f + G_{NW2}}{G_{NW2}} H_n + (L_f + G_{NW2}) H_n \right)$, where L_f is a Lipschitz constant and G_{NW1} , G_{NW2} are the conductances of the LR-coupling interface, such that for two waveforms $\mathbf{x}_n^{(k-1)}, \tilde{\mathbf{x}}_n^{(k-1)} \in \mathcal{U}_{n,d}$ on a time window $H_n < H_{\max}$ the following extended recursion estimate holds

$$\begin{pmatrix} \delta_{\mathbf{y}_1}^{(k)} \\ \delta_{\mathbf{z}_1}^{(k)} \\ \delta U_{Co, \bar{z}_1}^{(k)} \\ \delta_{\mathbf{y}_2}^{(k)} \\ \delta_{\mathbf{z}_2}^{(k)} \\ \delta U_{1, \bar{z}_2}^{(k)} \\ \delta U_{2, \bar{z}_2}^{(k)} \\ \delta I_{Co, \bar{z}_2}^{(k)} \end{pmatrix} \leq \mathbf{K}_e \cdot \begin{pmatrix} \delta_{\mathbf{y}_1}^{(k-1)} \\ \delta_{\mathbf{z}_1}^{(k-1)} \\ \delta U_{Co, \bar{z}_1}^{(k-1)} \\ \delta_{\mathbf{y}_2}^{(k-1)} \\ \delta_{\mathbf{z}_2}^{(k-1)} \\ \delta U_{1, \bar{z}_2}^{(k-1)} \\ \delta U_{2, \bar{z}_2}^{(k-1)} \\ \delta I_{Co, \bar{z}_2}^{(k-1)} \end{pmatrix} + \text{"initial offset"}, \quad (3.54)$$

with recursion matrix

$$\begin{pmatrix} 0 & \dots & & \dots & H_n \\ \vdots & 0 & & & H_n \\ & & \ddots & & \frac{1}{G_{NW1}} + H_n \\ & & & & H_n \\ & & & & H_n \\ & & & & \frac{1}{G_{NW1}} + H_n \\ \vdots & & & & H_n \\ 0 & \dots & & \dots & C_{I_{Co}} H_n \end{pmatrix}. \quad (3.55)$$

Proof. Again, we extend the proof of Lemma 3 by taking additional terms of the LR-splitting into account. The proof is split into two parts: starting with subs. 1 the estimation technique is applied for all unknowns. Then the estimation is done for subs. 2 while we use the inequalities of the coupling variables to link both subsystems together. For simplification we use the notation G_1, G_2 instead of $G_{NW 1}, G_{NW 2}$.

Subs. 1: For the algebraic unknowns of NW1, with $z_1 = \Phi_1(y_1)$, the estimation

$$|\Delta_{z_1}^{(k)}| \leq L_\Phi |\Delta_{y_1}^{(k)}|, \quad (3.56)$$

holds, where L_Φ is the maximum of the Lipschitz constants of ϕ_i with respect to y_i and z_i . For the ODE-part after integrating for $T_n < \tau \leq T_{n+1}$, Lipschitz continuity yields

$$\begin{aligned} |\Delta_{y_1}^{(k)}| \leq & |\Delta_{y_1}^{(k-1)}(t_n)| + \int_{T_n}^{T_{n+1}} L_f (|\Delta_{y_1}^{(k)}| + |\Delta_{z_1}^{(k)}|) \\ & + G_1 \cdot |\Delta U_{Co, \bar{z}_1}^{(k)}| dt, \end{aligned} \quad (3.57)$$

with Lipschitz constant L_f . Insert (3.56) into (3.57) and with algebraic constraint for $U_{Co, \bar{z}_1}^{(k)}$ we obtain

$$\delta_{y_1}^{(k)} \leq \frac{1}{1 - C_0 H_n - G_1 H_n} (|\Delta_{y_1}^{(k-1)}(t_n)| + H_n \cdot \delta I_{Co, \bar{z}_1}^{(k-1)}), \quad (3.58)$$

where $C_0 := L_f(1 + L_\Phi)$. Then inserting (3.58) into (3.56) and solving for $\delta_{z_1}^{(k)}$ yields

$$\delta_{z_1}^{(k)} \leq \frac{L_\Phi}{1 - C_0 H_n - G_1 H_n} (|\Delta_{y_1}^{(k-1)}(t_n)| + H_n \cdot \delta I_{Co, \bar{z}_1}^{(k-1)}). \quad (3.59)$$

Subs. 2: Similarly, using the same technique for subsystem 2, the corresponding estimate for $\delta I_{Co, \bar{z}_2}^{(k)}$ reads

$$\begin{aligned} \delta I_{Co, \bar{z}_2}^{(k)} \leq & c \cdot |\Delta_{\bar{y}_2}^{(k-1)}(t_n)| + CH_n \cdot |\Delta_{y_1}^{(k-1)}(t_n)| \\ & + C_{I_{Co}} H_n \cdot \delta I_{Co, \bar{z}_1}^{(k-1)}, \end{aligned} \quad (3.60)$$

with constants $c := 1 / \left(1 - C_0 H_n + \frac{L_f + G_2}{G_2} H_n + (L_f + G_2) H_n\right)$, $C_{I_{Co}} := c \cdot \left(\frac{L_f}{G_1} + CH_n\right)$ and $C := c \cdot L_f / (1 - C_0 H_n - G_1 H_n)$. This concludes the proof. \square

Again, the claim for contraction can be deduced by inspecting the spectral radius of \mathbf{K}_e .

Lemma 11 (Contraction Condition). *Let Ass. 1 be fulfilled. Then, for a co-simulation with LR-splitting technique, there is a window size*

$$H_n < H_{\max},$$

such that the map (for all k)

$$\left(\delta_{y_1}^{(k-1)}, \delta_{z_1}^{(k-1)}, \dots, \delta_{y_2}^{(k-1)}, \delta_{z_2}^{(k-1)} \right)^\top \rightarrow \left(\delta_{y_1}^{(k)}, \delta_{z_1}^{(k)}, \dots, \delta_{y_2}^{(k)}, \delta_{z_2}^{(k)} \right)^\top$$

is contractive independently of the type of NW 1 and NW 2.

Proof. The proof is the same as for Lemma 9. \square

3.2.3 Exact Recursion Analysis

This chapter covers the fine structure analysis for coupled DAEs of Chapter 3.1.2. It shows that using the standard theory no contraction could be inferred for our test circuit. However, by a detailed analysis we can prove the convergence.

Circuit Splitting

We investigate a simple RL-circuit. The modeling via modified nodal analysis yields a DAE (2.2) of index-1. By using the strategy of source coupling the circuit can be modeled as two coupled networks, see Fig. 3.6. Note that the decoupling uses the LR-coupling approach as well as the R-splitting for a reverse assembly of the inductance and resistance. Furthermore, the strongly coupled system corresponds to the same as subsystem 2, since node U_{Co} and U_1 coincide. This makes our model rather academic, but it shows the possibility of deviations between convergence analysis and application in co-simulation, which we want to highlight.

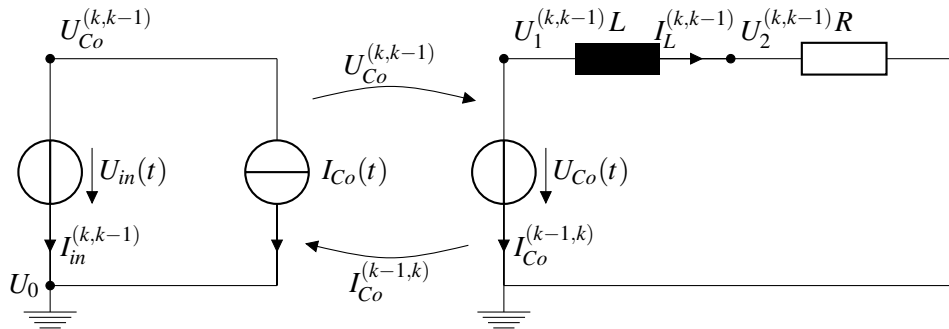


Figure 3.6: Decoupled RL-circuit with source-coupling. The first/second superscript notation $(k)/(k-1)$ denotes the old and new differential/algebraic iterates for subs. 1/subs. 2 first. [39]

The two subsystems for our co-simulation read in the form of (2.16):

$$\text{Subs. 1: } 0 = \begin{pmatrix} 0 & 1 \\ -1 & 0 \end{pmatrix} \begin{pmatrix} U_{Co} \\ I_{in} \end{pmatrix} - \begin{pmatrix} -I_{Co}(t) \\ -U_{in}(t) \end{pmatrix}, \quad (3.61a)$$

$$\text{Subs. 2: } 0 = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & L \end{pmatrix} \begin{pmatrix} \dot{U}_1 \\ \dot{U}_2 \\ \dot{I}_{Co} \\ \dot{I}_L \end{pmatrix} + \begin{pmatrix} 0 & 0 & 1 & 1 \\ 0 & G & 0 & -1 \\ -1 & 0 & 0 & 0 \\ -1 & 1 & 0 & 0 \end{pmatrix} \begin{pmatrix} U_1 \\ U_2 \\ I_{Co} \\ I_L \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ -U_{Co}(t) \\ 0 \end{pmatrix}, \quad (3.61b)$$

with inductance L , conductance $G = 1/R$, supply voltage $U_{in}(t)$, unknown node potentials U_1, U_2, U_{Co} and unknown currents I_{in}, I_L, I_{Co} , where the nodes U_{Co} and I_{Co} are the coupling variables for the information transport between both systems.

Standard Structure Analysis

Next we use the standard theory of Chapter 3.1.2 to analyze the convergence for the coupled system (3.61). To this end, we generalize system (3.61) to the following semi-explicit form:

$$\begin{aligned} \dot{\mathbf{y}}_2 &= \mathbf{f}_2(\mathbf{y}_2, \mathbf{z}_2), \\ 0 &= \mathbf{g}_1(\mathbf{z}_1, \mathbf{z}_2), \quad 0 = \mathbf{g}_2(\mathbf{z}_1, \mathbf{y}_2, \mathbf{z}_2), \end{aligned} \quad (3.62)$$

with non-singular Jacobians $\partial \mathbf{g}_i / \partial \mathbf{z}_i$, $i = 1, 2$, where subsystem 1 is merely a system of algebraic equations and subsystem 2 is a DAE. This corresponds to a AE-DAE coupling of example 1. The variables of the subsystems are

$$\mathbf{z}_1 := [U_{Co}, I_{in}]^\top \quad \text{and} \quad \mathbf{y}_2 := I_L, \quad \mathbf{z}_2 := [U_1, U_2, I_{Co}]^\top. \quad (3.63)$$

The corresponding splitting functions read (for subs. 1 first):

$$\begin{aligned} \mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) &:= [\mathbf{f}_2(0, 0, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)})], \\ \mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) &:= \begin{bmatrix} \mathbf{g}_1(0, \mathbf{z}_1^{(k)}, 0, \mathbf{z}_2^{(k-1)}) \\ \mathbf{g}_2(0, \mathbf{z}_1^{(k)}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \end{bmatrix}. \end{aligned} \quad (3.64)$$

Notice that the old algebraic iterate $\mathbf{z}_2^{(k-1)}$ (the current $I_{Co}^{(k-1)}$) enters algebraic equation of subs. 1. The reverse computational sequence yields the splitting functions (for subs. 2 first):

$$\begin{aligned} \mathbf{F}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) &:= [\mathbf{f}_2(0, 0, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)})], \\ \mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)}) &:= \begin{bmatrix} \mathbf{g}_1(0, \mathbf{z}_1^{(k)}, 0, \mathbf{z}_2^{(k)}) \\ \mathbf{g}_2(0, \mathbf{z}_1^{(k-1)}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)}) \end{bmatrix}. \end{aligned} \quad (3.65)$$

Here, the old algebraic iterate $\mathbf{z}_1^{(k-1)}$ (the node potential $U_{Co}^{(k-1)}$) enters algebraic equation of subs. 2. Therefore, both computational sequences create a mutual algebraic dependency and does not fulfill (3.20) trivially, i.e.,

$$\|\mathbf{G}_{\mathbf{z}^{(k)}}^{-1} \mathbf{G}_{\mathbf{z}^{(k-1)}}\|_{2,\infty} < 1.$$

Consequently, following the standard recursion estimate of Lemma 3 divergence of a co-simulation might occur for $\alpha_n \geq 1$. Thus, the convergence analysis for the coupled problem requires the calculation of the contraction factor, i.e., the Jacobians $\mathbf{G}_{\mathbf{z}^{(k)}}^{-1}$, $\mathbf{G}_{\mathbf{z}^{(k-1)}}$.

Splitting the Jacobian of $\mathbf{G}(\mathbf{y}^{(k)}, \mathbf{y}^{(k-1)}, \mathbf{z}^{(k)}, \mathbf{z}^{(k-1)})$ into parts of $\mathbf{G}_{\mathbf{y}^{(k)}}$, $\mathbf{G}_{\mathbf{y}^{(k-1)}}$, $\mathbf{G}_{\mathbf{z}^{(k)}}$ and $\mathbf{G}_{\mathbf{z}^{(k-1)}}$

yields for

$$\begin{aligned} \text{Subsystem 1 first: } \mathbf{G}_{\mathbf{z}^{(k)}} &= \begin{pmatrix} 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & G & 0 & 0 & 0 \\ -1 & 0 & 1 & 0 & 0 \end{pmatrix} \Rightarrow \mathbf{G}_{\mathbf{z}^{(k)}}^{-1} = \begin{pmatrix} 0 & 1 & 0 & 0 & -1 \\ 0 & 0 & 0 & R & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 & 0 \end{pmatrix}, \mathbf{G}_{\mathbf{z}^{(k-1)}} = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}, \\ \text{Subsystem 2 first: } \mathbf{G}_{\mathbf{z}^{(k)}} &= \begin{pmatrix} 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 \\ 0 & G & 0 & 0 & 0 \\ -1 & 0 & 0 & 0 & 0 \end{pmatrix} \Rightarrow \mathbf{G}_{\mathbf{z}^{(k)}}^{-1} = \begin{pmatrix} 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & R & 0 \\ 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 \\ 1 & 0 & -1 & 0 & 0 \end{pmatrix}, \mathbf{G}_{\mathbf{z}^{(k-1)}} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix}, \end{aligned}$$

and finally for

$$\begin{aligned} \text{Subsystem 1 first: } \|\mathbf{G}_{\mathbf{z}^{(k)}}^{-1} \mathbf{G}_{\mathbf{z}^{(k-1)}}\|_2 &= \|(0 \ 0 \ 0 \ 0 \ 1)^\top\|_2 = 1, \\ \text{Subsystem 2 first: } \|\mathbf{G}_{\mathbf{z}^{(k)}}^{-1} \mathbf{G}_{\mathbf{z}^{(k-1)}}\|_2 &= \|(-1 \ 0 \ 0 \ 0 \ 0)^\top\|_2 = 1. \end{aligned} \quad (3.66)$$

It is obviously that both computational sequences does not fulfill the contraction condition (3.20) and convergence cannot be inferred directly by the standard theory. However, the exact fine structure analysis proves the convergence of co-simulation for system (7.6). Moreover, it shows that the iteration process is contractive with the best possible speed of contraction, i.e., no splitting errors occur after both subsystems are updated once.

Exact Fine Structure Analysis

We aim to calculate the recursion matrix \mathbf{K}_e explicitly for all unknowns (3.63). As usual, $\Delta_{\mathbf{X}}^{(k)} X_i := X_i^{(k)}(t) - \tilde{X}_i^{(k)}(t)$ measures the difference of two waveforms on the n -th time window after k iterations steps. For simplicity of notation the subindex n is skipped.

Assuming that the Gauß-Seidel scheme starts with subsystem 1 first. Following the relations to old and new iterates we find for the algebraic variables by taking differences:

$$\begin{aligned} \Delta_{\mathbf{z}_1}^{(k)} I_{in} &= -\Delta_{\mathbf{z}_1}^{(k-1)} I_{Co}, & \Delta_{\mathbf{z}_1}^{(k)} U_{Co} &= \Delta U_{in} = 0, \\ \Delta_{\mathbf{z}_2}^{(k)} I_{Co} &= -\Delta_{\mathbf{y}_2}^{(k)} I_L, & \Delta_{\mathbf{z}_2}^{(k)} U_1 &= \Delta_{\mathbf{z}_1}^{(k)} U_{Co} = 0, & \Delta_{\mathbf{z}_2}^{(k)} U_2 &= \frac{1}{G} \Delta_{\mathbf{y}_2}^{(k)} I_L. \end{aligned} \quad (3.67)$$

Notice that the coupling node $U_{Co}^{(k)}$ is given by the time-dependent source $U_{in}(t)$ and thus $\Delta_{\mathbf{z}_1}^{(k)} U_{Co} = 0$ holds. From the differential equation for the induction current I_L we obtain

$$\frac{d}{dt} \left(\Delta_{\mathbf{y}_2}^{(k)} I_L \right) = \frac{\Delta_{\mathbf{z}_2}^{(k)} U_1 - \Delta_{\mathbf{z}_2}^{(k)} U_2}{L} = \frac{1}{G \cdot L} \Delta_{\mathbf{y}_2}^{(k)} I_L$$

and thus we find for any $t \in [T_n, T_n + H_n]$

$$|\Delta_{\mathbf{y}_2}^{(k)} I_L(t)| = |\Delta_{\mathbf{y}_2}^{(k)} I_L(t_n)| \cdot e^{(t-t_n)/(G \cdot L)} = |\Delta_{\mathbf{y}_2}^{(k-1)} I_L(t_n)| \cdot e^{(t-t_n)/(G \cdot L)}. \quad (3.68)$$

With (3.68) in (3.67) and by using absolute values, we finally find the exact error propagation (for

subs. 1 first):

$$\begin{bmatrix} |\Delta_{y_2}^{(k)} I_L| \\ |\Delta_{z_1}^{(k)} I_{in}| \\ |\Delta_{z_1}^{(k)} U_{Co}| \\ |\Delta_{z_2}^{(k)} I_{Co}| \\ |\Delta_{z_2}^{(k)} U_1| \\ |\Delta_{z_2}^{(k)} U_2| \end{bmatrix} = \underbrace{\begin{bmatrix} \mathbf{0} & \mathbf{A} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}}_{=:\mathbf{K}_{exact}} \begin{bmatrix} |\Delta_{y_2}^{(k-1)} I_L| \\ |\Delta_{z_1}^{(k-1)} I_{in}| \\ |\Delta_{z_1}^{(k-1)} U_{Co}| \\ |\Delta_{z_2}^{(k-1)} I_{Co}| \\ |\Delta_{z_2}^{(k-1)} U_1| \\ |\Delta_{z_2}^{(k-1)} U_2| \end{bmatrix} + e^{(t-t_n)/(G \cdot L)} \begin{bmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ \frac{1}{G} \end{bmatrix} |\Delta_{y_2}^{(k-1)} I_L(t_n)|, \quad (3.69)$$

where $\mathbf{0}$ is zero matrix and

$$\mathbf{A} = \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}.$$

The spectral radius of the exact recursion matrix yields $\rho(\mathbf{K}_{exact}) = 0$, since all eigenvalues are zero. Consequently, the recursion (3.69) satisfies the spectral radius condition $\rho(\mathbf{K}_{exact}) < 1$ trivially for splitting (3.64). Therefore, co-simulation is contractive and the iteration process is finished after all unknowns are updated only once. It should be noted that the result is in line with Theorem 8 of R-splitting, since no resistance between the current and voltage source is involved, i.e., $R_1 = 0$, the contraction factor α_n vanishes.

Furthermore, the exact recursion estimate matches the numerical results of Chapter 7.1.

Remark 12 (Information loss by lumping). *Clearly, the relation between \mathbf{K}_{exact} and \mathbf{K} is the lumping of all differential and algebraic unknowns in (3.69). Applying the maximum norm, we obtain the estimate*

$$\begin{bmatrix} |\Delta^{(k)} \mathbf{y}| \\ |\Delta^{(k)} \mathbf{z}| \end{bmatrix} \leq \mathbf{K} \begin{bmatrix} |\Delta^{(k-1)} \mathbf{y}| \\ |\Delta^{(k-1)} \mathbf{z}| \end{bmatrix} + \gamma := \begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} |\Delta^{(k-1)} \mathbf{y}| \\ |\Delta^{(k-1)} \mathbf{z}| \end{bmatrix} + \begin{bmatrix} C \\ C \end{bmatrix} |\Delta^{(k-1)} \mathbf{y}(t_n)|,$$

with $C = (1 + \frac{1}{G})e^{(t-t_n)/(G \cdot L)}$ and $\rho(\mathbf{K}) = 1$. The result fits the calculations of (3.66).

Thus without fine structure analysis, the contraction disappears from the estimate even for our simple test circuit.

Conclusions

It has been shown that the standard theory of co-simulation does not always detects convergence. This holds already for a simple electric circuit, which we have investigated. Therefore, we analyzed the coupled system by express the exact error propagation (fine structure analysis) and proved the convergence for our example. Clearly, the information about stability and contraction disappeared during lumping, which we have demonstrated.

4 Chapter 4

Uncertainty Quantification

The Computer Aided Design (CAD) modeling and system engineering of complex devices is becoming increasingly important. Frequently, the modeling process results in a system of equations, for example, an ODE, DAE, PDE or PDAE system, with an appropriate large number of variables (model inputs) and outputs. In practice, it is desirable to have knowledge about the impact (high or low sensitivity) of the system variables to the output behavior, where the measurement of sensitivities requires mathematical algorithms. The information about sensitivities can be used for various purposes: the reduction of complex models, i.e., models with a large number of variables and high computational effort are replaced by "cheaper" models. Here, the calculation of sensitivities can be used to set up a reduced order model by setting variables with low sensitivity as a constant, [41]. The resulting system is of lower dimension and can be computed faster. Furthermore, the industrial CAD of complex electric circuit, e.g., integrated circuits on silicon wafers or circuits with radio-frequency application can benefit from sensitivity analysis. These circuits are often of size of a few micrometer and thus parasitic effects must be taken into account. Here, a sensitivity analysis can support the layouting-process. In addition, the knowledge about the impact of electric components such as resistors, conductors and inductors to the behavior of electric circuits can be used for reducing manufacturing costs, since the information can be exploit to use components with low tolerance only where necessary.

Uncertainty Quantification (UQ) refers to a family of methods to calculate sensitivities. UQ is based on a description of a system whose components (parameters) are considered as random variables with given distributions such as normal or uniform distribution. Therefore, UQ will attempt to answer the following questions:

- which system parameters have the most impact to the system behavior?
- which system parameters have less impact to the system behavior?
- how do the system parameters interact?

The concept of UQ unites a set of different methods, which can be classified into local and global sensitivity approaches. Local approaches evaluate changes in the model outputs with respect to variations in a single input, [42–44]. The input variable are typically changed once a time in small increments and the effect of this perturbation on the model output is calculated by using local sensitivity indices:

$$S_i|_{[p\%]} = \frac{|Y_{[+p\%]} - Y_{[-p\%]}|}{Y_0}, \quad (4.1)$$

where $Y_{[+p\%]}$, $Y_{[-p\%]}$ is the model response for a variation of $\pm p\%$ and Y_0 is the nominal output value. For the global sensitivity analyzes, several approaches are known such as (amongst others) the weighted average of local sensitivities, partial rank correlation coefficients, multiparametric sensitivity analysis, Fourier amplitude sensitivity analysis, Monte-Carlo method as well as the Sobol decomposition method, [41, 46–49].

Within this chapter we focus on variance-based methods, see e.g. [45], often referred to as analysis of variance (ANOVA), which belong to the family of global approaches. The variance-based calculation of sensitivities offers the advantage to measure sensitivities across the whole input space (global). Furthermore, they can handle non-linear responses, i.e., they can also measure the effect of interactions in non-linear systems.

The classic Sobol decomposition belongs to the family of variance-based methods. However, it is useless for high dimensional problems, since the computational effort increases exponentially with the amount of uncertain variables, Chapter 4.1.1. This problem is referred to as curse of dimensionality. In practice, one can observe that only a few variables have a major impact to the model output. This phenomena is often referred to as effective dimension. Now, different methods try to exploit this circumstance to break the curse of dimensionality. Therefore, we focus on the general Polynomial Chaos (gPC) expansion, Chapter 4.1, where stochastic systems are approximated by a truncated series of orthogonal basis functions, [9]. The received sensitivities are only approximations of the exact values computed by the (full) Sobol decomposition method. The most expensive part of the gPC expansion is to determine some coefficient functions. For this purpose, two main concepts can be used: the Stochastic Galerkin approach as intrusive method and the Stochastic Collocation approach which is a non-intrusive methods [9, 45, 50], Chapter 4.2.3. We focus on the Stochastic Collocation approach, which requires to solve multiple probabilistic integrals. The efficient calculation of multiple integrals is a separate problem. Therefore, the numerical treatment of such integrals is discussed in Chapter 5.

This chapter is outlined as follows: Chapter 4.1 introduces into the decomposition of the variance of a stochastic model via Sobol decomposition and into the calculation of Sobol based sensitivities. Afterwards the Polynomial Chaos (PC) expansion is discussed in Chapter 4.2. Then, orthogonal basis functions of a Polynomial Chaos expansion are introduced in Chapter 4.2.1 followed by the concept of the gPC expansion, Chapter 4.2.2, which is an extension of the PC approach. Finally, this chapter concludes with the calculation of the gPC based sensitivity indices, Chapter 4.2.4.

4.1 The Sobol Decomposition

The concept to decompose the variance of the output of a model in fractions which can be attributed to system variables was first developed by **I. M. Sobol** in 1990, [41]. The advantage of the Sobol decomposition is to determine the contribution of each variable and their mutual interactions to the model output variance.

From a perspective of a black-box concept, any model may be considered as a function

$$y = f(\mathbf{x}), \quad (4.2)$$

with n variables stored in a vector $\mathbf{x} = (x_1, \dots, x_n)$ and y is a chosen univariate model output, e.g. a selected node potential or branch current of a circuit or the electric field strength of a field device. We define the n -dimensional unit hypercube for the entire domain as

$$K^n := [0, 1]^n. \quad (4.3)$$

Following [41] it is possible to decompose such a model function f as a series of functions. The

function decomposition reads

$$f(\mathbf{x}) = f_0 + \sum_{i=1}^n f_i(x_i) + \sum_{1 \leq i < j \leq n} f_{ij}(x_i, x_j) + \cdots + f_{12\dots n}(x_1, x_2, \dots, x_n), \quad (4.4)$$

where the unknown functions f_i , f_{ij} up to $f_{12\dots n}$ have to be determined. All functions in (4.4) are constructed such that they are orthogonal to each other with respect to the L^2 -norm:

$$\int_{K^n} f_{\mathbf{u}}(\mathbf{x}_{\mathbf{u}}) \cdot f_{\mathbf{v}}(\mathbf{x}_{\mathbf{v}}) d\mathbf{x} = 0 \quad \forall \mathbf{u} \neq \mathbf{v}. \quad (4.5)$$

The unknown function terms in (4.4) can be constructed recursively by

$$f_{i_1, \dots, i_s}(x_{i_1}, \dots, x_{i_s}) = \int_{K^{n-s}} f(\mathbf{x}) d\mathbf{x}_{\sim [i_1, \dots, i_s]} - \left(f_0 + \sum_{i=1}^n f_i(x_i) + \sum_{1 \leq i < j \leq n} f_{ij}(x_i, x_j) + \cdots \right), \quad (4.6)$$

where the recursion starts with the calculation of the leading (constant) term f_0 given by integration over the entire domain:

$$f_0 = \int_{K^n} f(\mathbf{x}) d\mathbf{x}. \quad (4.7)$$

The subscript $d\mathbf{x}_{\sim [i_1, \dots, i_s]}$ in (4.6) denotes the integration over the entire domain except the variables x_{i_1}, \dots, x_{i_s} .

Remark 13. Equation (4.6) requires to solve multiple integrals of dimension $n - s$. In particular for spatial discretized PDAE systems of Chapter 2.3 such integrals cannot be solved analytically. Consequently, the numerical treatment by quadrature formulas is required to compute approximations, which is discussed in Chapter 5.

4.1.1 Sobol Sensitivity Analysis

The aim is to determine how much of the system output is affected by each of the input parameters, either due to a single parameter or due to interactions between them. Sobol sensitivity analysis is based on a decomposition of the variance of the system output into a sum of partial variances of the input parameters.

From now on, assuming n random input variables X_i , ($i = 1, \dots, n$) in a stochastic model stored in a random variable vector $\mathbf{X} = (X_1, \dots, X_n)$, where the variables are mutually independent and (for now) uniformly distributed. Furthermore, each variable is considered to range over some finite interval which may be assumed, after rescaling, to be in the unit hypercube (4.3).

Under some probabilistic interpretation of the input variables in (4.2), the model output $Y = f(\mathbf{X})$ is a random variable with expectation value

$$\mathbb{E}(Y) = \int_{K^n} f(\mathbf{X}) d\mathbf{X} \quad (4.8)$$

and (total) variance

$$\text{Var}(Y) = \int_{K^n} f^2(\mathbf{X}) d\mathbf{X} - \mathbb{E}(Y)^2. \quad (4.9)$$

Now, the Sobol decomposition (4.4) allows to decompose the variance (4.9) of a stochastic model as a finite sum of partial variances. In other words, we are interested in the decomposition of the output variance of a model into contributions from effects of single variables, combined effects of pairs of variables, and so on.

Analogously to the decomposition (4.4) such a decomposition of variances (4.9) exists and is given by

$$D(Y) = \sum_{i=1}^n D_i(Y) + \sum_{1 \leq i < j \leq n} D_{ij}(Y) + \dots + D_{12\dots n}(Y), \quad (4.10)$$

where we used the abbreviation:

$$D(Y) = \text{Var}(Y), \quad D_i(Y) = \text{Var}[\mathbb{E}(Y|X_i)], \quad D_{ij}(Y) = \text{Var}[\mathbb{E}(Y|X_i, X_j)] - D_i(Y) - D_j(Y),$$

and so on.

The partial variances in (4.10) are given explicitly by

$$D_{i_1, \dots, i_s}(Y) = \int_{K^s} f_{i_1, \dots, i_s}^2(X_{i_1}, \dots, X_{i_s}) d\mathbf{X}_{[i_1, \dots, i_s]}, \quad (4.11)$$

with functions $f_{i_1, \dots, i_s}(X_{i_1}, \dots, X_{i_s})$ already defined in (4.6). The subscript $d\mathbf{X}_{[i_1, \dots, i_s]}$ denotes the integration over all remaining random variables X_{i_1}, \dots, X_{i_s} .

Assuming that the multiple integral in (4.6) can be solved without errors, the decomposition of the variance (4.9) by (4.10) is exact.

Now, Sobol indices specify the sensitivity of the input variables to the system output and identify the variable or the set of variables that have the most influence to the system output, i.e., it provides insight into which system parameter contributes most to the variability of the system output. Once the partial variances are received, the calculation of sensitivities can be done by a simple scaling. The Sobol index for the subset of variables X_{i_1}, \dots, X_{i_s} is defined as

$$S_{i_1, \dots, i_s} := \frac{D_{i_1, \dots, i_s}(Y)}{D(Y)}. \quad (4.12)$$

In general, the higher the sensitivity indice value, the higher the impact to the model. Here, one distinguishes between main effects and high order effects.

- The main effect (first order contribution) describes the contribution from a single variable X_i to the model output variance and is often referred to as first-order contribution. Its sensitivity index reads $S_i = \frac{D_i}{D}$.
- The second-order contribution $S_{ij} = \frac{D_{ij}}{D}$ is the proportion of the variance of the model output that appears due to the interactions of two variables X_i, X_j .
- This can be easily extended to higher-order contributions: $S_{ijk} = \frac{D_{ijk}}{D}, \dots$

The Sobol indices (4.12) sum up to 1:

$$1 = \sum_{i=1}^n S_i + \sum_{1 \leq i < j \leq n} S_{ij} + \cdots + S_{12\dots n}, \quad (4.13)$$

since S_{i_1, \dots, i_s} is defined by the partial variance for each set of variables divided by the total variance.

Total-Effect: Furthermore, in order to measure the main effect and all higher-order effects of a single variable X_i , one can define the total-order sensitivity index

$$S_i^T := S_i + S_{ij} + \cdots + S_{1\dots i\dots n}. \quad (4.14)$$

Computational Effort

The Sobol decomposition (4.4) requires to compute

$$\sum_{k=0}^n \binom{n}{k} = 2^n$$

function terms. In addition, the decomposition into partial variance (4.10) requires the computation of 2^n multiple integrals. Thus, the computational effort to identify all sensitivity indices of a model function (4.2) using the Sobol sensitivity analysis is given by

$$2^n + 2^n = 2^{n+1}.$$

Consequently, the complexity of an UQ algorithm based on Sobol decomposition is $\mathcal{O}(2^n)$.

Obviously, the determination of sensitivities for the full range of input variables and interactions between each set of variables makes the method computational expensive, which is the main drawback. However, in practice one can observe that the contribution of higher-order effects decreases with increasing number of parameters which are considered in interaction with each other. Therefore, usually one is not interested in all Sobol sensitivity indices. To detect sensitive parameters it is mostly sufficient to calculate sensitivities up to a certain order, e.g. third-order indices. This circumstance can be exploited to reduce the computational effort. Chapter 4.2 introduces the generalized Polynomial Chaos approach which attempts to break the curse of dimensionality by truncation.

4.2 The Polynomial Chaos Expansion

The Polynomial Chaos (PC) expansion, also often referred to as PC expansion, allows to analyze stochastic processes of uncertain models with significant reduced effort as the full Sobol decomposition. The main idea of the PC expansion is that a reduced model for the output can be expressed as a series expansion consisting of orthogonal polynomials. The strategy is, in a certain sense, similar to the Sobol decomposition of Chapter 4.1. The Polynomial Chaos approach has its origin in 1938 developed by **N. Wiener** [18, 19]. He has proven that each normal distributed random variable can be represented by an infinite sum of Hermite polynomials. Later, **D. Xiu** extend the concept to the generalized Polynomial Chaos (gPC) expansion, such that it can be used for a broad range of distributions, see [9]. Roughly speaking, the gPC expansion express random

variables with arbitrary distributions, e.g. normal-, uniform distribution, etc., as a polynomial function (sequence), where the distribution is already known.

There are other methods which can be used for a sensitivity analysis. One prominent approach is the Monte-Carlo analysis [49], which is based on a brute force sampling of the random parameter space. Compared to Monte-Carlo, the implementation of an algorithm based on the gPC expansion is more complex, however the gPC expansion can be much more efficient, [45].

4.2.1 Orthogonal Polynomials

This chapter gives insight into the main properties of orthogonal polynomials. The most frequently used orthogonal polynomials in mathematics are the Hermite polynomials, the Laguerre polynomials, the Jacobi polynomials, the Chebyshev polynomials as well as the Legendre polynomials. All this types of polynomials are orthogonal with respect to the inner product for a measure with support in a real interval. A sequence of such polynomials is referred to as orthogonal polynomial sequence.

First we introduce into the univariate case of orthogonal polynomial sequences. Afterwards, the concept can be easily extend to multivariate case. It enables to use the gPC expansion for stochastic models with an arbitrary number of random input variables. In fact, depending on the distribution of the random input variables, different types of orthogonal polynomials have to be used for the gPC expansion, Chapter 4.2.2. Finally, the gPC method requires the scaling of orthogonal polynomials. The result is a sequence of orthonormal polynomials. Thus, we introduce into the construction of orthonormal polynomial sequences and state the main advantage.

The Univariate Case

Let $\tilde{p}: [a, b] \rightarrow \mathbb{R}^+$ be weight function [45]. We define the weighted inner product of two functions.

Definition 3 (Weighted Inner Product). *Let \tilde{p} with $\tilde{p}: [a, b] \rightarrow \mathbb{R}^+$ be weight function of two functions $f, g: [a, b] \rightarrow \mathbb{R}$. Furthermore let $f, g \in L^2$. The inner product of two functions on a finite interval $[a, b]$ with respect to the weight function \tilde{p} is defined as*

$$\langle f, g \rangle_{\tilde{p}} := \int_a^b f(x) \cdot g(x) \cdot \tilde{p}(x) dx. \quad (4.15)$$

Let $Q_n(x)$, $n \in \mathbb{N}_0$ be a polynomial with $\deg Q_n = n$:

$$Q_n(x) = a_n x^n + a_{n-1} x^{n-1} + \dots + a_0, \quad a_n \neq 0. \quad (4.16)$$

Two polynomials are said to be orthogonal if their inner product becomes zero. Now, we aim to define a sequence $(Q_n)_{n \in \mathbb{N}_0}$ of orthogonal polynomials. The orthogonal sequence with respect to

its weight function \tilde{p} is defined as

$$\langle Q_n, Q_m \rangle_{\tilde{p}} = \gamma_n \delta_{nm}, \quad \text{with} \quad \gamma_n := \int_a^b Q_n^2(x) \tilde{p}(x) dx, \quad (4.17)$$

where δ_{nm} is the Kronecker delta function

$$\delta_{nm} = \begin{cases} 1, & \text{if } n = m, \\ 0, & \text{if } n \neq m. \end{cases} \quad (4.18)$$

Usually, the polynomial sequence is required to be orthonormal and its polynomials \tilde{Q}_n are referred to as orthonormal polynomials if (4.17) holds for $\gamma_n = 1$. The orthonormal polynomial sequence $(\tilde{Q}_n)_{n \in \mathbb{N}_0}$ can be constructed from orthogonal polynomials by scaling:

$$\tilde{Q}_n = Q_n / \sqrt{\gamma_n}, \quad (4.19)$$

with γ_n already defined in (4.17). The gPC expansion exploit the property of orthonormal polynomial sequences, which will be discussed in Chapter 4.2.3.

The question arises which family of polynomials should be used to construct the orthogonal sequence. The candidates are the Hermite polynomials, the Laguerre polynomials, the Jacobi polynomials, the Chebyshev polynomials and the Legendre polynomials. In fact, the type of basis polynomials is related to the distribution of the random variables which should be approximated as gPC expansion. The correlation between the distribution of random variables and its corresponding polynomials is topic of Chapter 4.2.2. However, within this work we focus on uniformly distributed variables, where Legendre polynomials are used. Thus, we state the Legendre polynomials, the scaling to construct the orthonormal polynomial sequence as well as the transformation of uniformly distributed random variables to coincide with its domain of definition.

Legendre Polynomials: With respect to the gPC expansion for uniformly distributed random variables, the Legendre polynomials are the best choice of basis functions [9, 40, 45].

Definition 4 (Legendre polynomials). *The n -th Legendre polynomial $P_n: [-1, 1] \rightarrow \mathbb{R}$, $n \in \mathbb{N}_0$ is defined by the Rodrigues formula*

$$P_n(x) := \frac{1}{2^n n!} \cdot \frac{d^n}{dx^n} \left((x^2 - 1)^n \right), \quad (4.20)$$

or by recursion

$$(n+1)P_{n+1}(x) = (2n+1)xP_n(x) - nP_{n-1}(x), \quad (n = 1, 2, \dots) \text{ with } P_0 = 1; P_1 = x. \quad (4.21)$$

The Legendre polynomials fulfill the property of orthogonality with respect to the inner product (4.17) on the finite interval $[-1, 1]$ for $\tilde{p} = \frac{1}{2}$. The scaling to construct the orthonormal sequence is given by

$$\tilde{P}_n = P_n / \gamma_n \quad \text{with} \quad \gamma_n = \sqrt{2n+1}. \quad (4.22)$$

The Multivariate Case

Within this thesis, the multivariate polynomial sequence is used within the gPC expansion to approximate the stochastic process of models whose input is a random input vector. From now on, it is useful to think of each variable as a random variable stored in a random input vector $\mathbf{X} = (X_1, \dots, X_n)$. The following requires the assumption of independency.

Assumption 3. Consider the domain $\mathbf{D} = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n]$. It is assumed that all random variables $X_i \in [a_i, b_i]$, ($i = 1, 2, \dots, n$) are mutual independent.

First, one has to define the corresponding multivariate weight function. With Ass. 3, the multivariate weight function $\tilde{p}_{\mathbf{X}}: \mathbf{D} \rightarrow \mathbb{R}^+$ is defined as the product of the univariate weight functions:

$$\tilde{p}_{\mathbf{X}} := \prod_{i=1}^n \tilde{p}_i(X_i), \quad \text{with } \tilde{p}_i: \mathbb{R} \rightarrow \mathbb{R}^+ \quad (4.23)$$

Now, the construction of multivariate orthogonal polynomial sequences consisting of polynomials up to a certain degree is straightforward by using the multiindex notation.

Definition 5 (Multivariate Polynomial). Let Ass. 3 be fulfilled and let $\beta = (\beta_1, \beta_2, \dots, \beta_n) \in \mathbb{N}_0^n$ be a multiindex with cardinality $|\beta| := \sum_{i=1}^n \beta_i$. Furthermore, let Q_{β_i} be univariate orthogonal polynomial with $\deg Q_{\beta_i} = \beta_i$. Then, the multivariate orthogonal polynomial Φ_{β} with $\deg \Phi_{\beta} = |\beta|$ is defined as

$$\Phi_{\beta}(\mathbf{X}) := \prod_{i=1}^n Q_{\beta_i}(X_i). \quad (4.24)$$

Hence, the corresponding multivariate orthogonal polynomials can be constructed by multiplying univariate orthogonal polynomials with respect to a given multiindex. It is straightforward to proof that (4.17) also holds for multivariate orthogonal polynomial sequences $(\Phi_{\beta})_{\beta \in \mathbb{N}_0^n}$.

Corollary 14 (Inner product of multivariate polynomials). Let Ass. 3 be fulfilled. Furthermore, let $\alpha, \beta \in \mathbb{N}_0^n$ be two multiindices. The corresponding multivariate polynomials $\Phi_{\alpha}, \Phi_{\beta}$ constructed by (4.24) also fulfill the orthogonality (4.17) on \mathbf{D} with respect to the multivariate weight function $\tilde{p}_{\mathbf{X}}$ as constructed by (4.23).

Now, similar to the univariate case, using normalized univariate polynomials (4.19) in (4.24), the sequence $(\tilde{\Phi}_{\beta})_{\beta \in \mathbb{N}_0^n}$ is referred to as multivariate orthonormal sequence if

$$\langle \tilde{\Phi}_{\alpha}, \tilde{\Phi}_{\beta} \rangle_{\tilde{p}_{\mathbf{X}}} = \delta_{\alpha\beta} \quad \forall \tilde{\Phi} \in (\tilde{\Phi}_{\beta})_{\beta \in \mathbb{N}_0^n}, \quad \text{with } \delta_{\alpha\beta} = \begin{cases} 0, & \text{if } \alpha \neq \beta \\ 1, & \text{if } \alpha = \beta, \end{cases} \quad (4.25)$$

is satisfied, where $\delta_{\alpha\beta}$ is the n -dimensional Kronecker delta function with $\delta_{\alpha\beta} = \delta_{\alpha_1\beta_1} \dots \delta_{\alpha_n\beta_n}$.

In practice, the domain of distribution of random variables differs from the domain where Legendre polynomials are defined, i.e. from $[-1, 1]$. Therefore, a linear transformation is required to be able to use Legendre polynomials as basis polynomials.

Linear Transformation: Assuming that the random input vector $\mathbf{X} = (X_1, \dots, X_n)$ is defined on the domain $\mathbf{D} = [a_1, b_1] \times [a_2, b_2] \times \dots \times [a_n, b_n]$, where each variable is independent and uniformly distributed on their respective interval, i.e., $X_i \in \mathcal{U}(a_i, b_i)$, ($i = 1, \dots, n$).

An appropriate linear transformation for each variable X_i , ($i = 1, \dots, n$) enables to define a transformed random input vector $\tilde{\mathbf{X}} := (\tilde{X}_1, \dots, \tilde{X}_n)$ with $\tilde{X}_i \in \mathcal{U}(-1, 1)$. The mapping from $[a_i, b_i]$ to $[-1, 1]$ is done by the "shifting" function (for $i = 1, \dots, n$):

$$\tilde{X}_i = \frac{b_i - a_i}{2} X_i + \frac{b_i + a_i}{2}. \quad (4.26)$$

Usually, the transformation is already considered in the polynomial sequence and its corresponding polynomials are referred to as "shifted polynomials".

Example 2. Table 4.1 shows the polynomial sequence $(\Phi_\beta)_{\beta \in \mathbb{N}_0^n}$ and their corresponding multi-index β for $n = 3$ random variables up to a polynomial degree of two, i.e., cardinality $|\beta| \leq 2$:

$\text{deg } \Phi_\beta$	β	Φ_β
0	(0, 0, 0)	$Q_0(X_1) Q_0(X_2) Q_0(X_3)$
1	(1, 0, 0)	$Q_1(X_1) Q_0(X_2) Q_0(X_3)$
1	(0, 1, 0)	$Q_0(X_1) Q_1(X_2) Q_0(X_3)$
1	(0, 0, 1)	$Q_0(X_1) Q_0(X_2) Q_1(X_3)$
2	(2, 0, 0)	$Q_2(X_1) Q_0(X_2) Q_0(X_3)$
2	(0, 2, 0)	$Q_0(X_1) Q_2(X_2) Q_0(X_3)$
2	(0, 0, 2)	$Q_0(X_1) Q_0(X_2) Q_2(X_3)$
2	(1, 1, 0)	$Q_1(X_1) Q_1(X_2) Q_0(X_3)$
2	(0, 1, 1)	$Q_0(X_1) Q_1(X_2) Q_1(X_3)$
2	(1, 0, 1)	$Q_1(X_1) Q_0(X_2) Q_1(X_3)$

Table 4.1: Shows all polynomials and their related multiindex up to a polynomial degree of two, where Q_0, Q_1, Q_2 are the univariate polynomials.

4.2.2 The generalized Polynomial Chaos Expansion

The mathematician **N. Wiener** was the first one who has introduced the way to approximate stochastic Gaussian processes by using Hermite polynomials. In the literature, his method is often referred to as homogeneous chaos or Wiener chaos, see [18]. Later R. Cameron and W. Martin extend the concept of homogeneous chaos and proved that such an expansion converges for an arbitrary stochastic process, [19]. Based on the results of Cameron-Martin, the mathematician **D. Xiu** generalized the homogeneous chaos to various continuous and discrete distributions, [9]. This applies to the most physical systems.

Homogeneous Chaos

According to **N. Wiener**, a stochastic process of a model whose random input vector \mathbf{X} consists of normal distributed variables can be represented as a series expansion consisting of orthogonal polynomials (homogeneous chaos)

$$f = \sum_{j=0}^{\infty} f_j H_j(\mathbf{X}), \quad (4.27)$$

where H_j are the Hermite polynomial and f_j are the corresponding coefficient functions, which have to be computed from a limited number of model simulations, see Chapter 4.2.3. The homogeneous chaos only provides optimal convergence for stochastic processes which are of Gaussian

type [18, 19]. However, an extension of (4.27) for further stochastic processes with different distributions exists as well. This leads to the generalized form of the homogeneous chaos known as the generalized Polynomial Chaos expansion.

The gPC Expansion

As mentioned the gPC framework is able to handle various stochastic processes. In the probability theory, the fact that a value of a random variable would equal to a desired sample is specified by its density function. One prominent function is the density function of Gaussian type of distribution. The gPC expansion for the multivariate case, i.e., at least two random variables, requires the theory of probability spaces. That is, the density function is generalizable to the concept of multivariate distributions.

Definition 6 (Joint Density Function). *Let $\mathbf{X} = (X_1, \dots, X_n)$ be random input vector defined on $\mathbf{D} \in \mathbb{R}^n$, and let all random variables X_i , ($i = 1, \dots, n$) be independent distributed with given density function $f_{X_i} : D_i \rightarrow \mathbb{R}^+$. Let (x_1, x_2, \dots, x_n) be a sample out of the probability space. The function $f_{\mathbf{X}} : \mathbf{D} \rightarrow \mathbb{R}^+$ with*

$$f_{\mathbf{X}}(x_1, x_2, \dots, x_n) = f_{X_1}(x_1) \cdot f_{X_2}(x_2) \cdots f_{X_n}(x_n) \quad (4.28)$$

is joint density function of the random input vector \mathbf{X} .

For optimal convergence of the gPC expansion, it matters that the joint density function of the random input vector \mathbf{X} is equal or at least similar to the weight function $\tilde{p}_{\mathbf{X}}$ of the multivariate basis polynomials Φ_{β} . Table 4.2 shows the distribution of a random variable and the corresponding polynomial for optimal convergence as well as the domain of definition, [40].

Distribution	Polynomial	Domain of definition	Density- and weight function
Normal	Hermite	$(-\infty, \infty)$	$\frac{1}{\sqrt{2\pi}} \exp(-\frac{1}{2}x^2)$
Gamma	Laguerre	$[0, \infty)$	$\exp(-x)x^{\alpha-1} \frac{1}{\Gamma(\alpha)}$
Beta	Jacobi	$[-1, 1]$	$\frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)\Gamma(\beta)2^{\alpha+\beta-1}}(x+1)^{\alpha-1}(1-x)^{\beta-1}$
Uniform	Legendre	$[-1, 1]$	$\frac{1}{2}$

Table 4.2: Polynomial and corresponding distribution.

In practice, the distribution of random variables of a stochastic process can be different, e.g. some of them are normal distributed while others are uniformly distributed. However, this does not represent a restriction of the scope of the gPC expansion. Here, appropriate basis polynomials have to be used with respect to Table 4.2. Therefore, the homogeneous chaos (4.27) can be extended to a series expansion consisting of suitable (multivariate) orthogonal polynomials:

$$f = \sum_{j=0}^{\infty} f_j \Phi_j(\mathbf{X}). \quad (4.29)$$

In order to use this approach in practice the series is truncated to a finite number of terms. Therefore, for the most of non-academic model functions the gPC expansion only provides an approximation of the exact Sobol decomposition. However, the gPC expansion provides an adequate accuracy with much less computational effort.

The truncated sum consists of P terms, which is prescribed by the problem dimension n and the maximum polynomial degree p , where the dimension is given by the number of random variables. Therefore, the number of terms can be controlled by the polynomial degree p , which is the only degree of freedom.

Corollary 15 (Computational Effort, [40]). *Let $\deg \Phi_i \leq p$ for $j = 0, \dots, P-1$, i.e., the gPC expansion is of order p , and n is the number of random variables of the stochastic process. Then, the gPC expansion (4.29) is truncated after*

$$P+1 = \binom{n+p}{p} = \frac{(n+p)!}{n!p!} \quad (4.30)$$

terms.

Thus, eq. (4.35) can be written as a truncated formulation using a finite number of P terms:

$$f_{\text{gPC}}(\mathbf{X}, t) = \sum_{j=0}^{P-1} f_j(t) \Phi_j(\mathbf{X}). \quad (4.31)$$

Definition 7 (Approximation Error). *The approximation error caused by the gPC expansion of order p is defined as*

$$\delta(\mathbf{X}, p, n) := |f(\mathbf{X}, t) - f_{\text{gPC}}(\mathbf{X}, t)|. \quad (4.32)$$

Figure 4.1 shows the growth of terms as a function of n and p .

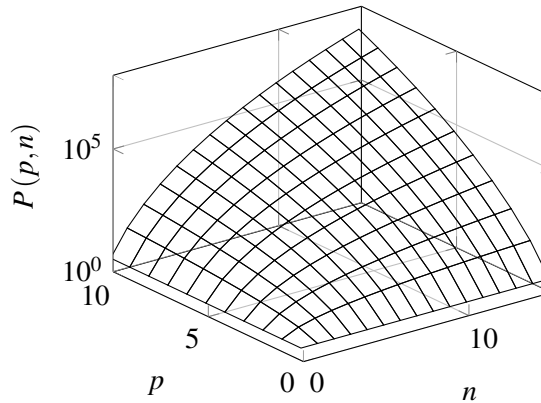


Figure 4.1: Number of terms of a gPC expansion for a given dimension n and maximum polynomial degree p .

In fact, the calculation of the unknown coefficient functions f_j in (4.31) is the most time-consuming part. Here, two different approaches are basically known: the Stochastic Galerkin method and the Stochastic Collocation method. Within Chapter 4.2.3 we focus on the Stochastic Collocation approach as the method of choice, whereas the Stochastic Galerkin approach is introduced briefly.

4.2.3 Calculation of the gPC Coefficient Functions

The major challenge of the gPC expansion is to determine the unknown coefficient functions. For this task, two approaches exist: the Stochastic Galerkin approach (intrusive method) and the

Stochastic Collocation approach (non-intrusive method) [9, 45, 50], where the related method is termed as intrusive or non-intrusive gPC expansion.

Intrusive method: The main idea of the intrusive gPC expansion is to replace all random variables and dependent variables in the related equations with their gPC expansion (for $j = 0, 1, \dots, P-1$):

$$\left\langle f \left(t, \sum_{j=0}^{P-1} f_j(t) \Phi_j(\mathbf{X}), \mathbf{X} \right), \Phi_l(\mathbf{X}) \right\rangle \quad (4.33)$$

The inner product yields a coupled system of deterministic equations which can be solved in the same way as the origin deterministic model. Due to the fact that introducing uncertainties is mostly the second step after setting up the deterministic model, the whole program code has to be reimplemented. This task becomes expensive for multiphysics simulations, since it requires the interplay between different software-packages. Therefore, the Galerkin approach is not being considered further within this thesis.

Non-intrusive method: The non-intrusive gPC expansion allows to compute the unknown coefficient functions without changing the model itself (the program code). This is the main benefit and makes the application of a gPC based UQ to co-simulation straightforward. The non-intrusive gPC expansion considers the stochastic model from a perspective of a black-box and tries to calculate the coefficient functions by using the strategy of random space sampling technique. Thus, it calculates the coefficients from a limited number of model simulations. For multivariate stochastic models, i.e, more than one random variable, one has to solve multiple (probabilistic) integrals. The spatial structure of the sampling grid and the number of points depends on the sampling technique. The most common technique is based on quadrature, where the underlying gPC method is referred to as quadrature-based non-intrusive gPC expansion. Using full grids, the number of grid points growth exponentially with the number of random variables. To bypass this problem, numerical techniques based on sparse grid can be used to reduce the numerical effort. This is topic of Chapter 5.

Stochastic Collocation Approach

The collocation approach predict the unknown coefficient functions by a limited number of model simulations. The calculation is done explicitly by solving probabilistic integrals:

$$f_j(t) := \frac{\langle f(t, \mathbf{X}), \Phi_j(\mathbf{X}) \rangle_{\tilde{p}_{\mathbf{X}}}}{\langle \Phi_j^2(\mathbf{X}) \rangle_{\tilde{p}_{\mathbf{X}}}}, \quad \text{for } j = 0, \dots, P-1, \quad (4.34)$$

where the nominator $\langle f(t, \mathbf{X}), \Phi_j(\mathbf{X}) \rangle_{\tilde{p}_{\mathbf{X}}}$ as well as the denominator $\langle \Phi_j^2(\mathbf{X}) \rangle_{\tilde{p}_{\mathbf{X}}}$ requires the numerical treatment of multiple integrals. Now that we have defined the unknown coefficient functions, we are prepared to state the generalized non-intrusive gPC expansion.

Definition 8 (Generalized Non-Intrusive Polynomial Chaos Expansion). *Let Ass. 3 be fulfilled and let $\beta = (\beta_1, \dots, \beta_n) \in \mathbb{N}_0^n$ be given multiindex. Furthermore, let $\Phi_{\beta} : \mathbf{D} \rightarrow \mathbb{R}$ be a multivariate basis polynomial with polynomial degree $|\beta| = \beta_1 + \dots + \beta_n \leq p$ and let $\tilde{p}_{\mathbf{X}} : \mathbf{D} \rightarrow \mathbb{R}^+$ be the corresponding weight function. The generalized non-intrusive gPC expansion of order p is defined as*

$$f_{gPC}(\mathbf{X}, t) := \sum_{|\beta| \leq p} f_{\beta}(t) \Phi_{\beta}(\mathbf{X}) \quad (4.35)$$

with coefficient function f_{β} given by (4.34) and orthogonal basis polynomials Φ_{β} .

Using orthonormal polynomials $\tilde{\Phi}_j$, see Chapter 4.2.1, as polynomial basis in (4.31) the denominator yields $\langle \tilde{\Phi}_j^2(\mathbf{X}) \rangle_{\tilde{p}_{\mathbf{X}}} = 1$ and (4.34) simplifies to

$$f_j(t) = \int_{\mathbf{D}} f(t, \mathbf{X}) \tilde{\Phi}_j(\mathbf{X}) \tilde{p}_{\mathbf{X}}(\mathbf{X}) d\mathbf{X}, \quad \text{for } j = 0, \dots, P-1, \quad (4.36)$$

with multivariate weight function $\tilde{p}_{\mathbf{X}}: \mathbf{D} \rightarrow \mathbb{R}$ as given in (4.23).

Remark 16. *Let Ass. 3 be fulfilled. For a uniformly distributed random input vector \mathbf{X} , the joint density function $f_{\mathbf{X}}$ is constant on \mathbf{D} , where the class of Legendre polynomials has to be chosen as orthogonal polynomial sequence in (4.31), see Table 4.2. Therefore, (4.36) further simplifies to*

$$f_j = \frac{1}{2^n} \int_{\mathbf{D}} f(t, \mathbf{X}) \tilde{\Phi}_j(\mathbf{X}) d\mathbf{X}, \quad \text{for } j = 0, \dots, P-1. \quad (4.37)$$

4.2.4 Polynomial Chaos based Sensitivity Indices

Chapter 4.1 shows the possibility to determine stochastic quantities by the Sobol decomposition. However, the amount of effort increases exponentially with the number of random variables, which makes the method impracticable for higher dimensional problems. As an alternative, approximations of the mean value and variances can be derived by extract it directly from its approximating gPC expansion.

Once all unknown coefficient functions in (4.35) are determined, the mean value and variance of the model output $Y = f(\mathbf{X})$ can be directly obtained by [45]:

$$\begin{aligned} \mathbb{E}(Y) &= f_0 \Phi_0, \quad \text{and} \quad DgPC(Y) = \text{Var} \left[\sum_{0 < |\beta| \leq p} f_{\beta} \Phi_{\beta} \right] \\ &= \sum_{0 < |\beta| \leq p} f_{\beta}^2 \mathbb{E} \left[\Phi_{\beta}^2 \right]. \end{aligned} \quad (4.38)$$

By using orthonormal polynomials $\tilde{\Phi}_{\beta}$ as basis, see Chapter 4.2.1, the inner product yields $\langle \tilde{\Phi}_{\beta}^2 \rangle = \mathbb{E} \left[\tilde{\Phi}_{\beta}^2 \right] = 1$ and (4.38) corresponds to:

$$\mathbb{E}(Y) = f_0, \quad \text{and} \quad DgPC(Y) = \sum_{0 < |\beta| \leq p} f_{\beta}^2, \quad (4.39)$$

i.e., a simple summation of the square number of all coefficient functions, whereas the mean value is given by the constant f_0 . The error in the variance (4.9) due to truncation is

$$\text{Err}[\text{Var}(Y), DgPC(Y)] = |\text{Var}(Y) - DgPC(Y)|. \quad (4.40)$$

Remark 17.

- *The accuracy of the mean value derived by (4.39) is only affected by the numerical integration error done by solving the probabilistic integral (4.36) ones.*
- *But, the accuracy of the variance derived by (4.39) depends on numerical integration errors which are done by solving the probabilistic integral $P - 1$ times as well as the truncation*

error of the gPC expansion. This is discussed in Chapter 5.

In contrast to identify sensitivity indices by the Sobol decomposition, a gPC based UQ allows to approximate sensitivities directly by simple regrouping and summation.

Definition 9 (gPC based Sensitivity Indices, [45]). *Let f_β be the coefficient functions of an gPC expansion (4.35). Furthermore, for an index set (i_1, \dots, i_s) we define a set of multi-indices θ_{i_1, \dots, i_s} as*

$$\theta_{i_1, \dots, i_s} := \{\beta \in \mathbb{N}_0^n : \beta_k > 0 \text{ for } k \in \{i_1, \dots, i_s\}, \beta_k = 0 \text{ for } k \notin \{i_1, \dots, i_s\}\}.$$

The partial gPC based sensitivity index for a subset of variables X_{i_1}, \dots, X_{i_s} is defined by the sum of the squares of the gPC coefficient functions:

$$SgPC_{i_1, \dots, i_s}(Y) := \sum_{\substack{\beta \in \theta_{i_1, \dots, i_s} \\ |\beta| \leq p}} f_\beta^2 \mathbb{E} \left[\Phi_\beta^2 \right] / DgPC, \quad (4.41)$$

with variance $DgPC$ as defined in (4.38). Using orthonormal polynomials as basis polynomials, (4.41) simplifies to

$$SgPC_{i_1, \dots, i_s}(Y) := \sum_{\substack{\beta \in \theta_{i_1, \dots, i_s} \\ |\beta| \leq p}} f_\beta^2 / \sum_{0 < |\beta| \leq p} f_\beta^2. \quad (4.42)$$

Analogously, one can define the total-order gPC sensitivity indices as an approximation of the (exact) total-order Sobol sensitivity indices in (4.14), which involves the main effect and all cross effects up to a certain order:

$$SgPC_i^T := SgPC_i + SgPC_{ij} + SgPC_{ijk} + \dots \quad (4.43)$$

Remark 18. *We will use the gPC based sensitivity indices in Chapter 7.5, where we focus on the behavior of the stochastic process during the co-simulation procedure. We analyze the rate of convergence of the stochastic quantities by extracting mean value and variance directly from its gPC representation.*

The objective is to approximate Sobol indices as best as possible by gPC sensitivity indices. Here, the limiting factor is the order p in (4.35) that states the maximum number of random variables which can be considered in interaction. Only if the stochastic model is well approximated by the gPC expansion of order p , one can expect that (4.42) are good approximations of (4.12).

Remark 19. *In practice one has made the observation, the more parameters are considered in mutual interaction the smaller are their corresponding sensitivity indices, see e.g. [51]. Thus, it is mostly sufficient to use basis polynomials up to a certain degree, e.g. in particular for high dimensional problems $1 \leq p \leq 3$ is frequently a good compromise between accuracy and computation effort.*

5 Chapter 5

Multivariate Quadrature

Within this chapter, we aim to calculate definite integrals of functions of more than one variable:

$$I^n[f] = \int_{\mathbf{D}} f(\zeta) d\zeta, \quad \zeta \in \mathbf{D}. \quad (5.1)$$

In the majority of cases multiple integrals cannot be analytically solved, since the function f may have a difficult form. This may be the case if f is a model function received from a modeling process of electric networks, see Chapter 2. For example, the spatial discretized PDAE system for field/circuit coupled problems discussed in Chapter 2.3. We aim to analyze sensitivities for such a spatial discretized PDAE system, which requires to solve multiple integrals of (5.1) multiple times. Therefore, suitable techniques are used to solve the integrals by numerical integration which is referred to as quadrature, see e.g. [54]. In numerical analysis, quadrature formulas are approximations of definite integrals, usually stated as a weighted sum of function values at specified points within the domain of integration, see e.g. [57, 58]. Thus, multivariate quadrature formulas are based on a set of points (often referred to as nodes) arranged in a grid and a set of corresponding weights, see e.g. [53]. The full set of nodes is commonly referred to as tensor-product grid or full grid. Numerical integration techniques based on tensor-product grids are computationally expensive for high dimensional problems, since the amount of data increases exponentially with the dimension (curse of dimensionality). In order to avoid the exponential growth of data, we introduce into the so-called sparse grid technique [57]. Sparse grids break the curse of dimensionality, [59], of common tensor-product grids and reduce the computational effort for high dimensional problems. This becomes important particularly for a sensitivity analysis of field/circuit coupled problems tackled by co-simulation, since the computation of the field is expensive and thus the user aims to minimize the repeated model simulations. Consequently, using numerical quadrature for the probabilistic integrals (4.36) of the (sample-based) non-intrusive gPC expansion only provides approximations of the unknown coefficient functions.

Problem Definition

The modeling process of field/circuit coupled problems yields a system of differential algebraic equations. Introducing random processes, the system (2.15) corresponds to the following form:

$$\mathbf{E}(\mathbf{p}) \dot{\mathbf{x}} + \mathbf{A}(\mathbf{p}) \mathbf{x} = \mathbf{f}(t), \quad \mathbf{x} \in [t_0, t_e] \rightarrow \mathbb{R}^n, \quad (5.2)$$

with dynamic part \mathbf{E} , static part \mathbf{A} and time dependent sources \mathbf{f} , where the matrices \mathbf{E} and \mathbf{A} include some physical parameters $\mathbf{p} = (p_1, \dots, p_Q)^T$ which are assumed to be uncertain, i.e., we suppose that a subset of parameters are not known exactly. Again, all uncertain parameters are considered to be independent random variables.

Now, by separation of system (5.2) into subsystems, the Gauss-Seidel type iteration scheme of

Def. 1 defines splitting functions

$$\begin{aligned}\dot{\tilde{\mathbf{y}}} &= \mathbf{F}\left(t, \tilde{\mathbf{y}}^{(k)}(t, \mathbf{p}), \tilde{\mathbf{y}}^{(k-1)}(t, \mathbf{p}), \tilde{\mathbf{z}}^{(k)}(t, \mathbf{p}), \tilde{\mathbf{z}}^{(k-1)}(t, \mathbf{p})\right), \\ \mathbf{0} &= \mathbf{G}\left(t, \tilde{\mathbf{y}}^{(k)}(t, \mathbf{p}), \tilde{\mathbf{y}}^{(k-1)}(t, \mathbf{p}), \tilde{\mathbf{z}}^{(k)}(t, \mathbf{p}), \tilde{\mathbf{z}}^{(k-1)}(t, \mathbf{p})\right).\end{aligned}\quad (5.3)$$

Within this thesis, we aim to compute statistics of the solution $\tilde{\mathbf{x}}(t, \mathbf{p}) = (\tilde{\mathbf{y}}(t, \mathbf{p}), \tilde{\mathbf{z}}(t, \mathbf{p}))^\top$ for $t \in [t_0, t_e]$ for the coupled problem (5.3). Statistics of uncertain systems can be computed by the non-intrusive gPC expansion as introduced in Chapter 4.2.2. The computation of the expectation value (4.8) and variance (4.9) are approximated by (4.38). Thus, non-intrusive gPC expansion requires the sampling of the random parameter space by a grid $\mathcal{G} = \{\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(m)}\}$ with grid points $\mathbf{p}^{(i)}$. The spatial arrangement of the grid points is topic of this chapter.

First, Chapter 5.1 introduces into the concept of tensor-product grids. Then, based on these general concept, sparse grids can be easily constructed, see Chapter 5.2.

5.1 Tensor-Product Grids

First, this chapter introduces into univariate quadrature. Afterwards, the concept can be easily extended to higher dimensions by using the cartesian product of their univariate sets.

Definition 10 (Univariate Quadrature). *Consider the univariate function $f: [a, b] \rightarrow \mathbb{R}$ to be integrated within the interval $[a, b]$. The univariate quadrature formula of order N is defined as*

$$Q_N^1[f] := \sum_{i=1}^N \omega^{(i)} \cdot f(p^{(i)}), \quad (5.4)$$

with N given nodes $p^{(i)} \in [a, b]$ and weights $\omega^{(i)}$. All nodes $p^{(i)}$ for $i = 1, \dots, N$ are stored in a 1-D univariate grid $\Gamma_N^1 = \{p^{(1)}, \dots, p^{(N)}\}$.

Now, multivariate quadrature formulas are constructed based on such univariate quadrature formulas. The construction is straightforward by nesting univariate grids, where the resulting multivariate quadrature is referred to as tensor-product grid or full grid, [55].

Definition 11 (Multivariate Quadrature). *Consider a multivariate function $f: \mathbf{D} \rightarrow \mathbb{R}$ on $\mathbf{D} = D_1 \times \dots \times D_k \times \dots \times D_n \subseteq \mathbb{R}^n$ with $D_k = [a_k, b_k]$. The subscript (k) denotes the k -th direction. The multivariate quadrature formula is constructed as tensor-product (nesting) of their univariate sets:*

$$Q^n[f] := Q_{N_1}^{(1)} \otimes \dots \otimes Q_{N_k}^{(k)} \otimes \dots \otimes Q_{N_n}^{(n)}[f] = \sum_{i_1=1}^{N_1} \omega_1^{(i_1)} \dots \sum_{i_k=1}^{N_k} \omega_k^{(i_k)} \dots \sum_{i_n=1}^{N_n} \omega_n^{(i_n)} \cdot f(\mathbf{p}^{(i)}), \quad (5.5)$$

with given nodes $\mathbf{p}^{(i)} = (p_1^{(i_1)}, \dots, p_n^{(i_n)})$ and weights $(\omega_1^{(i_1)}, \dots, \omega_n^{(i_n)})$ for $i = 1, \dots, m$, where

$$Q_{N_k}^{(k)}[f_k] = \sum_{i_k=1}^{N_k} \omega_k^{(i_k)} \cdot f_k(p_k^{(i_k)}), \quad k = 1, \dots, n,$$

is a set of univariate quadrature formulas with corresponding 1-D grids $\Gamma_{N_k}^{(k)}$.

The related tensor-product grid consisting of $m = N_1 \cdots N_n$ nodes $\mathbf{p}^{(i)}$ in (5.5) is a set of the form

$$\mathcal{G} = \Gamma_{N_1}^{(1)} \otimes \cdots \otimes \Gamma_{N_n}^{(n)} = \left\{ \mathbf{p}^{(i)} : p_k^{(i)} \in \Gamma_{N_k}^{(k)}, i = 1, \dots, m, k = 1, \dots, n \right\}. \quad (5.6)$$

Typically, all weights in (5.5) are concentrated in an overall weight, where the merging is done by multiplication:

$$\omega^{(i)} = \prod_{k=1}^n \omega_k^{(i_k)}. \quad (5.7)$$

Quadrature formulas are given for a broad class of different integrations problems. One of the most important and widespread methods are the Gaussian quadrature formulas [53, 60]. From the structural property point of view, all these formulas only differ in the way how the nodes are located within the domain of integration. Within this thesis, we focus on the Gauss-Legendre quadrature, since they are most suitable for solving probabilistic integrals of the gPC expansion of Chapter 4.2 when uniform distributed variables are involved. Figure 5.1 (left) shows the two-dimensional tensor-product grid \mathcal{G}^2 on $[-1, 1]^2$ and (right) the total number of nodes for an tensor-product grid with N_k nodes in each directions in dependence of the dimension of integration n .

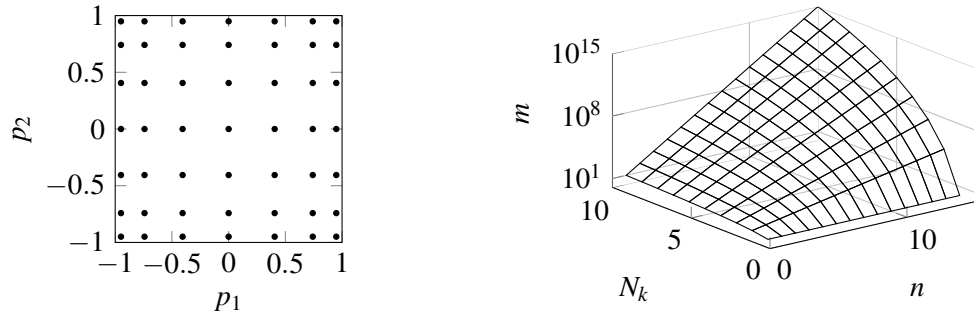


Figure 5.1: (left) Two-dimensional tensor-product grid \mathcal{G}^2 with 49 nodes. (right) Exponential growth of nodes for a tensor-product grid with N_k nodes in each direction.

Quadrature Error

Recall that the non-intrusive gPC expansion requires to determine the unknown coefficient functions $f_j(t)$ by the inner product $\langle f(t, \mathbf{X}), \tilde{\Phi}_j(\mathbf{X}) \rangle_{\tilde{p}_{\mathbf{X}}}$ for $j = 0, \dots, P-1$. The subsequent calculation of the variance of a model output suffers from truncation by the gPC expansion and thus (4.39) is only an approximation of (4.9) with error (4.40). However, (4.40) assumes to solve the probabilistic integrals (4.36) analytically, i.e., it ignores quadrature errors.

Definition 12 (Quadrature Error). *Let $I^n[f]$ be the exact integration value and $Q^n[f]$ be its approximation obtained by quadrature. We define the error caused by (5.5) as*

$$E^n[f] := |I^n[f] - Q^n[f]|. \quad (5.8)$$

Solving the probabilistic integrals (4.36) by quadrature (5.5) with a finite number of nodes usually

provides only an approximation of the exact integration value:

$$\hat{f}_j(t) = \sum_{i=1}^m \omega^{(i)} f(t, \mathbf{p}^{(i)}) \tilde{\Phi}_j(\mathbf{p}^{(i)}). \quad (5.9)$$

Therefore, the quadrature error (5.8) for each coefficient function is $|f_j - \hat{f}_j|$ and finally the error in the variance (4.39) caused by quadrature reads

$$\text{Err} [DgPC(Y), \hat{D}gPC(Y)] = \left| \sum_{j=1}^{P-1} f_j^2 - \hat{f}_j^2 \right|. \quad (5.10)$$

Rounding Errors

Numerical quadrature (5.5) requires the function evaluation of the model function f for all nodes $(\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(m)}) \in \mathcal{G}$. The received function value on a computer is denoted as $\hat{f}(\mathbf{p}^{(i)})$, where typically $\hat{f}(\mathbf{p}^{(i)}) \neq f(\mathbf{p}^{(i)})$ holds due to rounding errors etc.. Consequently, the quadrature to find the coefficient functions of gPC expansion reads:

$$\hat{\hat{f}}_j(t) = \sum_{i=1}^m \omega^{(i)} \hat{f}(t, \mathbf{p}^{(i)}) \tilde{\Phi}_j(\mathbf{p}^{(i)}). \quad (5.11)$$

It follows for the error

$$|\hat{f}_j(t) - \hat{\hat{f}}_j(t)| = \left| \sum_{i=1}^m \omega^{(i)} \left(f(t, \mathbf{p}^{(i)}) - \hat{f}(t, \mathbf{p}^{(i)}) \right) \tilde{\Phi}_j(\mathbf{p}^{(i)}) \right|. \quad (5.12)$$

Under the assumption that $|f(\mathbf{p}^{(i)}) - \hat{f}(\mathbf{p}^{(i)})| \leq \varepsilon$ for all $\mathbf{p}^{(i)} \in \mathcal{G}$ and $|\tilde{\Phi}_j(\mathbf{p}^{(i)})| \leq \max_{\mathbf{p} \in \mathcal{G}} |\phi_j(\mathbf{p})|$ we find an upper bound for the error caused by inaccuracies in function evaluations:

$$|\hat{f}_j(t) - \hat{\hat{f}}_j(t)| \leq \left[\sum_{i=1}^m \omega^{(i)} \right] \max_{\mathbf{p} \in \mathcal{G}} |\phi_j(\mathbf{p})| \cdot \varepsilon. \quad (5.13)$$

It follows for the error in the variance caused by inaccuracies in function evaluations:

$$\text{Err} [\hat{D}gPC(Y), \hat{\hat{D}}gPC(Y)] = \left| \sum_{j=1}^{P-1} \hat{f}_j^2 - \hat{\hat{f}}_j^2 \right|. \quad (5.14)$$

Accumulation of Errors and Numerical Stability

The truncation error (4.40), the quadrature error (5.10) as well as the rounding error (5.14) accumulates within gPC procedure. Here, the upper bound for the error in the variance can be stated as

$$\begin{aligned} |\text{Var}(Y) - \hat{D}gPC(Y)| &\leq \text{Err}[\text{Var}(Y), DgPC(Y)] + \text{Err} [DgPC(Y), \hat{D}gPC(Y)] \\ &\quad + \text{Err} [\hat{D}gPC(Y), \hat{\hat{D}}gPC(Y)], \end{aligned} \quad (5.15)$$

where $\text{Var}(Y)$ is the exact value and $\hat{\hat{D}}gPC(Y)$ is the computed value.

An important aspect of quadrature formulas is their numerical stability. Using negative weights may lead to an amplification of errors caused by function evaluations, [61]. Again, consider the rounding error (5.12) caused by a computation procedure. The upper bound is given by (5.13), where the error ε is not amplified if all weights are positive, since $\sum_{i=1}^m \omega^{(i)} = 1$ holds. Within Chapter 5.1.1 we discuss Gaussian quadrature formulas, which are numerically stable.

5.1.1 Gaussian Quadrature

Within this chapter, we discuss the Gaussian quadrature named after the mathematician **Carl Friedrich Gauss** [60]. The class of Gaussian quadrature includes different methods, where its numerical stability is the major advantage of all formulas, since the quadrature weights are strictly positive. Furthermore, by a suitable choice of all nodes $p^{(i)}$, $i = 1, \dots, N$ within the domain of integration $[a, b]$ and weights $\omega^{(i)}$, the Gaussian quadrature yields exact results for all polynomials with maximum polynomial degree $2N - 1$. Consequently, no quadrature error (5.8) occurs and the formula reads

$$\int_a^b Q_n(x) dx = \sum_{i=1}^N \omega^{(i)} \cdot Q_n(p^{(i)}).$$

Now, consider the integrand g which can be split into two parts $g = \tilde{p} \cdot f$, where f is approximated by a polynomial Q_n in $[a, b]$ at nodes $p^{(i)}$ and \tilde{p} is related weight function, see Table 4.2. The received integral can be computed exactly and the method has the form:

$$\int_a^b g(x) dx = \int_a^b \tilde{p}(x) f(x) dx \approx \int_a^b \tilde{p}(x) Q_n(x) dx = \sum_{i=1}^N \omega^{(i)} f(p^{(i)}). \quad (5.16)$$

It can be extended to the multivariate case by (5.5). Notice that the domain of integration is not limited to finite intervals. The nodes $p^{(i)}$ are simply the roots of a polynomial belonging to the class of orthogonal polynomials. Within this work we only consider uniformly distributed random variables. Therefore, with respect to Table 4.2 we take a close look to the Gauss-Legendre quadrature which is based on Legendre polynomials of Def. 4.

Gauss-Legendre quadrature

Using the roots of the Legendre polynomials as nodes $p^{(i)}$ for $i = 1, \dots, N$ in (5.16) the method is referred to as Gauss-Legendre quadrature of order N and yields exact results for polynomials with degree $2N - 1$ or less, [54]. The corresponding weights of the Gauss-Legendre quadrature can be obtained by

$$\omega^{(i)} = \frac{2}{(1 - x^{(i)2}) \left[\frac{d}{dx} P_n(x) \right]_{x=x^{(i)}}^2}, \quad (5.17)$$

where P_n is Legendre polynomial with $\deg P_n = n$, see [53].

Notice that (5.16) fits the form of the probabilistic integrals (4.36) to find the coefficient functions $f_j(t)$ of the non-intrusive gPC expansion, Chapter 4.2.3, from which sensitivity indices can be

derived. Recall that the multivariate case requires to solve probabilistic integrals of the form:

$$f_j(t) = \int_{\mathbf{D}} f(t, \mathbf{X}) \tilde{\Phi}_j(\mathbf{X}) \tilde{p}_{\mathbf{X}}(\mathbf{X}) d\mathbf{X}, \quad \text{for } j = 0, \dots, P-1, \quad (5.18)$$

where f is model function, $\tilde{\Phi}$ is orthonormal multivariate polynomial constructed by (4.24) and $\tilde{p}_{\mathbf{X}}$ is multivariate weight function obtained by (4.23). Dealing with uniformly distributed random variables within the gPC expansion, Legendre polynomials should be used as basis polynomials. Consequently, for a fixed number of nodes, the Gauss-Legendre quadrature yields the best numerical result for the probabilistic integral (5.18). For uniformly distributed random variables, the multivariate weight function $\tilde{p}_{\mathbf{X}}$ is constant, see Table 4.2, and $\tilde{\Phi}_j$ for $j = 0, \dots, P-1$ is orthonormal multivariate Legendre polynomial. Therefore, the multivariate quadrature to find approximations of the gPC coefficient functions (4.37) reads

$$f_j(t) \approx \frac{1}{2^n} \cdot \sum_{i_1=1}^{N_1} \omega_1^{(i_1)} \sum_{i_2=1}^{N_2} \omega_2^{(i_2)} \cdots \sum_{i_n=1}^{N_n} \omega_n^{(i_n)} f(\mathbf{p}^{(i)}) \tilde{\Phi}_j(\mathbf{p}^{(i)}), \quad \text{for } j = 0, \dots, P-1, \quad (5.19)$$

where $\mathbf{p}^{(i)} = (p_1^{(i_1)}, \dots, p_n^{(i_n)})$ are the roots of the Legendre polynomials and $\omega^{(i)} = (\omega_1^{(i_1)}, \dots, \omega_n^{(i_n)})$ are the corresponding weights computed by (5.17).

Accuracy: As mentioned, the univariate Gauss-Legendre quadrature of order N is exact for all polynomials Q_n with degree at most $2N-1$. The accuracy of the multivariate quadrature can be derived by consider the accuracy of the univariate set. Then, the multivariate Gauss-Legendre quadrature with corresponding tensor-product grid $\mathcal{G} = \Gamma_{N_1}^{(1)} \otimes \cdots \otimes \Gamma_{N_n}^{(n)}$, see (5.6), is exact for all polynomials

$$P \in \text{span} \left\{ \prod_{k=1}^n x_k^{q_k} : q_k \leq 2N_k - 1 \text{ for all } k \right\}. \quad (5.20)$$

In particular, if the number of nodes is the same in each direction, i.e., $N_1 = N_2 = \cdots = N_n$, then Gauss-Legendre quadrature is exact for all polynomials with monomial degree $2N-1$ or less.

5	y^5	xy^5	x^2y^5	x^3y^5	x^4y^5	x^5y^5
4	y^4	xy^4	x^2y^4	x^3y^4	x^4y^4	x^5y^4
3	y^3	xy^3	x^2y^3	x^3y^3	x^4y^3	x^5y^3
2	y^2	xy^2	x^2y^2	x^3y^2	x^4y^2	x^5y^2
1	y	xy	x^2y	x^3y	x^4y	x^5y
0	1	x	x^2	x^3	x^4	x^5
	0	1	2	3	4	5

Table 5.1: Accuracy of the Gauss-Legendre quadrature using three nodes in both directions.

Table 5.1 shows all monomial for which a two-dimensional Gauss-Legendre quadrature with $N_1 = N_2 = 3$ nodes provides exact results.

Suppose that the gPC expansion (4.35) is of order p . Consequently, basis polynomials Φ_j with maximum monomial degree of p appear. For dimension $n = 2$ and order $p = 5$, these are all monomial of Table 5.1 up to the diagonal element. Now, the objective is to approximate the coefficient functions of the gPC expansion using quadrature as good as possible. The common Gauss-Legendre quadrature with N_k nodes for $k = 1, \dots, n$ in the k -th direction is exact for all polynomials with $\deg \Phi \leq 2 \cdot \min\{N_1, N_2, \dots, N_n\} - 1$, see (5.20). However, such a high accuracy

is not required. This is the point where sparse grids become important, see Chapter 5.2. Here, few integration nodes are enough to yield a good approximation of integrals.

5.2 Sparse Grid Quadrature

Sparse grids were originally developed by **S. A. Smolyak** [57]. They are based on a particular sparse construction referred to as Smolyak construction and enables to approximate multiple integrals of (5.1) in an more efficient way [56,57]. Compared to the full set of nodes of tensor-product grids, sparse grids balance computational effort and accuracy in an appropriate manner, since the resulting grid is much more sparse. The number of nodes get independent of the dimension of integration up to a logarithmic factor and thus sparse grids break the curse of dimensionality.

Recall that the Gaussian quadrature formulas are able to integrate the full monomial space (5.20) up to a degree of $2 \cdot \min\{N_1, N_2, \dots, N_n\} - 1$ exactly, see e.g. Table 5.1. However, in the majority of cases such an accuracy is not required. Here, sparse grids provide smaller set of nodes that are not exact for the full monomial space but for all polynomials up to a certain total degree.

Definition 13 (Smolyak Quadrature). *For $k = 1, \dots, n$ we define the difference quadrature formula*

$$\Delta_N^1[f_k] := (Q_N^1 - Q_{N-1}^1)[f_k] \quad \text{with} \quad Q_0^1[f_k] := 0, \quad (5.21)$$

where

$$Q_N^1[f_k] = \sum_{i=1}^N \omega^{(i)} f_k(p^{(i)}),$$

is the univariate quadrature formula with corresponding 1-D grid Γ_N^1 . The multivariate Smolyak quadrature of level l is defined as

$$Q_l^n[f] := \sum_{|\mathbf{k}| \leq l+n-1} \left(\Delta_{N_{k_1}}^1 \otimes \dots \otimes \Delta_{N_{k_n}}^1 \right) [f], \quad (5.22)$$

where \mathbf{k} is multiindex and $|\mathbf{k}| = \sum_{i=1}^n k_i$.

Remark 20. *Using the supremum-norm $\|k\|_\infty = \max\{k_1, \dots, k_n\}$ in (5.22) one obtains quadrature formulas based on tensor construction (5.5).*

The underlying sparse grid is a set of the form

$$\mathcal{G}_l = \{\mathbf{p}_{l,i} : 1 \leq i \leq N_l \subset [-1, 1]^n\}, \quad (5.23)$$

with increasing density of nodes from the middle to the edges, see Fig. 5.2.

Accuracy: For a given dimension n and maximum polynomial degree d we consider the polynomial space

$$\mathbb{P}_d^n := \text{span}\{x_1^{i_1} \cdot x_2^{i_2} \cdot \dots \cdot x_n^{i_n} \mid (i_1, i_2, \dots, i_n) \in \mathbb{N}_0^n, i_1 + i_2 + \dots + i_n \leq d\}.$$

The multivariate Smolyak quadrature Q_l^n with corresponding sparse grid \mathcal{G}_l is exact for all polynomials in \mathbb{P}_l^n , see [58], i.e.,

$$E_l^n[f] = 0 \quad \text{for all } f \in \mathbb{P}_l^n.$$

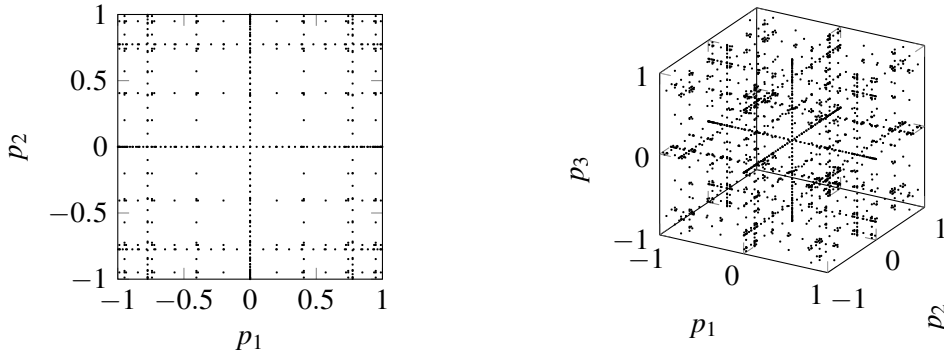


Figure 5.2: (left) Two-dimensional sparse grid of level $l = 5$ with 613 nodes in $[-1, 1]^2$. (right) Three-dimensional sparse grid of level $l = 5$ with 2070 nodes in $[-1, 1]^3$.

Remark 21. Suppose that the univariate quadrature formula Q_1 is exact for all polynomials with degree at most s . Furthermore, assuming that the multivariate quadrature (5.22) uses the same level l for every direction. Then, the Smolyak quadrature provides exact results for all monomials

$$\sum_{|i| \leq s} c_i x_1^{i_1} \dots x_n^{i_n}, \quad c_i \in \mathbb{R}. \quad (5.24)$$

These are all monomials up to the diagonal elements of Table 5.1. The important aspect is that it is exactly the required (optimal) accuracy to solve the probabilistic integrals (4.36) of the non-intrusive gPC expansion of order p , provided that $s = p$.

Conclusions

Within this chapter, the tensor-product grid and sparse grid based quadrature formulas were introduced. It was shown that tensor-product grids provide a high accuracy. However, they are oversized with respect to the number of nodes to find approximations of probabilistic integrals of the gPC expansion. Here sparse grids are well suited, since they are exact for all monomials up to a certain total degree, which fits exactly the structural property of the multivariate orthogonal polynomial basis of the gPC expansion.

6 Chapter 6

Density Estimation in Co-Simulation

In this chapter, we consider time-dependent coupled problems including uncertain parameters. These may be, for example, uncertainties in electric circuits or production-related tolerances in electrical machines. In practice, various circumstances could cause the consideration of uncertainties within the modeling process: tolerances in resistances, inductances and capacitances, variations in the diameter of the rotor and stator or the displacement of permanent magnets of electrical machines. The uncertain parameters are modeled as random variables with their corresponding probability density function (PDF).

The modeling and analysis of uncertainties within the computer-aided development process is well known and several approaches have been developed and optimized in recent time. However, when co-simulation is used to solve uncertain coupled systems (5.3), the uncertainties may affect the convergence of the dynamic iteration process, since the splitting scheme may be defines (one or more) random variables as old iterates. That is, the iterative solution determined by co-simulation becomes a random process and the contraction factor (3.17) becomes stochastic too with its own PDF. Consequently, uncertainties in co-simulation can cause divergence.

Now, the objective is to reconstruct the PDF of the random variable α_n as precise as possible to assess the divergence probability of co-simulation under the effect of uncertainties. Several well known approaches can be found in the literature for measuring the PDF of random variables, see [14]. One prominent technique is the so-called Kernel Density Estimation (KDE). These method is based on a brute-force sampling of the random parameter space, see [16]. A different approach is the spectral method developed by **J. Li** and **Y. Marzouk** from the Massachusetts Institute of Technology. The spectral approach determines the PDF analytically by covering the stochastic process using the gPC expansion of Chapter 4, see [1].

The KDE and spectral approach are entirely different and provide specific benefits. To our knowledge, both methods have never been applied to co-simulation with respect to PDF measuring problems. Thus, a comparison within a co-simulation process is of great interest. In general, dealing with uncertain components in a co-simulation framework was an unknown field of research and was investigated by us for the first time in [38].

To calculate statistics on a timeline $[t_0, t_e]$ an initial value problem of system (5.3) has to be resolved m times for all parameter-sets out of Ω . Consequently, we have to solve the problem

$$\begin{aligned} \dot{\tilde{\mathbf{y}}} &= \mathbf{F}\left(t, \tilde{\mathbf{y}}^{(k)}(\mathbf{p}^{(i)}), \tilde{\mathbf{y}}^{(k-1)}(\mathbf{p}^{(i)}), \tilde{\mathbf{z}}^{(k)}(\mathbf{p}^{(i)}), \tilde{\mathbf{z}}^{(k-1)}(\mathbf{p}^{(i)})\right), \\ \mathbf{0} &= \mathbf{G}\left(t, \tilde{\mathbf{y}}^{(k)}(\mathbf{p}^{(i)}), \tilde{\mathbf{y}}^{(k-1)}(\mathbf{p}^{(i)}), \tilde{\mathbf{z}}^{(k)}(\mathbf{p}^{(i)}), \tilde{\mathbf{z}}^{(k-1)}(\mathbf{p}^{(i)})\right), \end{aligned} \quad (6.1)$$

sequential for each $H_n \subseteq [t_0, t_e]$ and iterative for a fixed number of iteration steps k for all grid points $i = 1, \dots, m$. However, numerical time-integration methods only yield a discrete solution computed on a finite set of time points t_n (microsteps) with $t_0 < t_1 < \dots < t_n < t_{n+1} < \dots < t_e$. Consequently, the output for the different realizations of the parameters is the overall but

discrete solution $\tilde{\mathbf{x}}(t_n, \mathbf{p}^{(i)})$ for all $i = 1, \dots, m$. Finally, for a fixed grid point $\mathbf{p}^{(i)}$, interpolation over the entire timeline yields the continuous solution $\tilde{\mathbf{x}}(\mathbf{p}^{(i)})^{(k)}$. Based on these set of simulations (waveforms) we compute statistics for either the complete time interval or just at the final time.

This chapter is structured as follows: The KDE as well as the spectral method requires a set of samples from which a probability density function can be reconstructed. Therefore, Chapter 6.1 introduce into estimate the contraction factor of a co-simulation procedure. Then, Chapter 6.2 and Chapter 6.3 gives inside into the KDE and spectral method where the samples are used to estimate the corresponding PDF. A comparison of both methods with respect to the attainable accuracy and their computational cost is done numerically in Chapter 7.4.

6.1 Lower Bound Estimate for Purely Algebraic Coupling

In multiphysics the data exchange is frequently managed by algebraic constraints which are entered by coupling variables of algebraic type. In such cases the dynamic iteration yields the window-wise convergence rate $\alpha_n + \mathcal{O}(H_n)$, see Cor. 6 (i). In the following, we investigate coupled DAEs with the assumption of purely algebraic-to-algebraic coupling.

Assumption 4 (Purely Algebraic-to-Algebraic Coupling). *Assuming that all subsystems remains an index-1 problem such that the co-simulation scheme can be encoded by splitting functions \mathbf{F} and \mathbf{G} . Furthermore it is assumed, that the coupling of the subsystems is organized by algebraic constrains and that only old algebraic iterates enter the coupling equations. Then, the differential functions \mathbf{f}_1 , \mathbf{f}_2 , i.e., the ODE parts, are decoupled and the particular splitting scheme reads:*

$$\mathbf{F}(\cdot, \cdot, \cdot, \cdot) = \begin{bmatrix} \mathbf{f}_1 \left(\mathbf{y}_1^{(k)}, \mathbf{z}_1^{(k)}, 0, 0 \right) \\ \mathbf{f}_2 \left(0, 0, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)} \right) \end{bmatrix}, \quad \mathbf{G}(\cdot, \cdot, \cdot, \cdot) = \begin{bmatrix} \mathbf{g}_1 \left(\mathbf{y}_1^{(k)}, \mathbf{z}_1^{(k)}, 0, \boxed{\mathbf{z}_2^{(k-1,k)}} \right) \\ \mathbf{g}_2 \left(0, \boxed{\mathbf{z}_1^{(k,k-1)}}, \mathbf{y}_2^{(k)}, \mathbf{z}_2^{(k)} \right) \end{bmatrix}, \quad (6.2)$$

where the boxed superscript denotes the computational sequence of subs. 1 or subs. 2 first.

When co-simulation is used for the simulation of multiphysics coupled problems with mutual algebraic dependence, one is interested to measure the contraction factor online, i.e., during the co-simulation procedure. However, oftentimes physical principles are described by non-linear models and for these kind of models the computation of the Jacobians $\mathbf{G}_{\mathbf{z}^{(k)}}^{-1}$ and $\mathbf{G}_{\mathbf{z}^{(k-1)}}$ in (3.20) is time-consuming. Especially for real-world problems, the Jacobians can not be calculated within an appropriate timeframe. To bypass this problem, we propose an lower bound estimate for the contraction factor based on the partial solution $\tilde{\mathbf{x}}(\mathbf{p}^{(i)})|_{[T_n, T_{n+1}]}^{(k)}$ for a given grid point $\mathbf{p}^{(i)}$ and various iteration steps k in (6.1). The accuracy of the estimate is analyzed numerically in Chapter 7.3.

Knowledge about the contraction factor during the co-simulation can be used for an effective time window size control, such that non-equidistant communication point $t_0 = T_0 < T_1 < \dots < T_n < T_{n+1} < \dots < T_N = t_e$ can be defined with respect to the dynamic of the system. It offers the opportunity for varying the time window size H_n to ensure the convergence and the fast contraction over the entire simulation time. A second field of application is related to coupled systems with uncertain parameters. Here, the estimation of the contraction factor enables to set-up a set of samples that can be used to measure the corresponding probability density function, see Chapter 6.2 and Chapter 6.3.

Now, the idea is to exploit the iterative behavior of waveforms on the time window H_n , i.e., the waveform relaxation.

Lemma 22 (Lower Bound Estimate). *Let $\mathbf{X}^{(k)}(t)$, $\tilde{\mathbf{X}}^{(k)}(t)$ be two waveforms on the n -th time window H_n after the k -th iteration. The error after k iteration steps is measured by the difference $\delta_n^{(k)} = \|\mathbf{X}_n^{(k)}(t) - \tilde{\mathbf{X}}_n^{(k)}(t)\|_{2,\infty}$. Furthermore, let Ass. 4 be fulfilled. Consequently, the Jacobians in (3.20) exist, i.e., $\mathbf{G}_{\mathbf{z}^{(k)}}^{-1} \neq \mathbf{0}$ and $\mathbf{G}_{\mathbf{z}^{(k-1)}} \neq \mathbf{0}$, and a contraction factor α_n occurs for both computational sequences, see Cor. 6. Then, for the Gauß-Seidel-type dynamic iteration approach of Def. 1 and $H_n \leq H_{\max}$ the contraction factor is bounded from below by*

$$\alpha_n \geq \left(\frac{\delta_{\mathbf{z},n}^{(k)}}{\delta_{\mathbf{z},n}^{(0)}} \right)^{1/k} - CH_n, \quad (6.3)$$

with lumped errors $\delta_{\mathbf{z},n}^{(k)}$, $\delta_{\mathbf{z},n}^{(0)}$ and constant $C > 0$.

Proof. The proof is basically the same as for the contraction condition of Lemma 4. For algebraic-to-algebraic coupling the error propagation for the differential and algebraic part after k iteration steps reads:

$$\begin{pmatrix} \delta_{\mathbf{y},n}^{(k)} \\ \delta_{\mathbf{z},n}^{(k)} \end{pmatrix} \leq \tilde{\mathbf{K}}^k \begin{pmatrix} \delta_{\mathbf{y},n}^{(0)} \\ \delta_{\mathbf{z},n}^{(0)} \end{pmatrix} \quad \text{with} \quad \tilde{\mathbf{K}} = \begin{pmatrix} 0 & CH_n \\ 0 & CH_n + \alpha_n \end{pmatrix}, \quad (6.4)$$

where the k -th power of the iteration matrix $\tilde{\mathbf{K}}$ yields

$$\tilde{\mathbf{K}}^k = (\alpha_n + CH_n)^k \begin{pmatrix} 0 & \frac{CH}{\alpha_n + CH_n} \\ 0 & 1 \end{pmatrix}. \quad (6.5)$$

Inserting (6.5) in (6.4) yields an error bound for the algebraic unknowns:

$$\delta_{\mathbf{z},n}^{(k)} \leq (\alpha_n + CH_n)^k \delta_{\mathbf{z},n}^{(0)}. \quad (6.6)$$

Further suppose that the error of the solution (waveforms) of the algebraic, using an extrapolation step as initial guess for the coupling variables, is greater than zero. Then, we found that the contraction factor is bounded from below by

$$\alpha_n \geq \sqrt[k]{\delta_{\mathbf{z},n}^{(k)} / \delta_{\mathbf{z},n}^{(0)}} - CH_n. \quad (6.7)$$

This concludes the proof. □

Hence (6.3) enables to estimate the contraction factor for H_n small enough. However, when co-simulation is applied to multiphysics problems no analytic solution is available and thus the errors $\delta_n^{(k)}$, $\delta_n^{(0)}$ cannot be calculated directly (exactly). Therefore in practice, the errors can be predicted via Richardson Extrapolation.

Richardson Extrapolation

In order to measure errors in co-simulation, different approaches have already been used in applications of network simulations, see [5, 10–12]. Within this thesis, we focus on estimates based on Richardson Extrapolation.

Suppose that co-simulation has reached the communication point T_{n-1} . An error estimate for all unknowns of the system on the next time window $H_n = [T_{n-1}, T_n]$ can be deduced as follows:

In each time window, the subsystems are integrated twice. At first, one step with the time window size H_n is carried out which yields the numerical solution $\mathbf{X}_n^{(k)}(t)$. Then, the same step is executed using two steps with the half time window size $H_{n/2}$ yielding the more accurate solution $\mathbf{X}_{n/2}^{(k)}(t)$. Finally, the error of the solution $\mathbf{X}_n^{(k)}(t)$ can be estimated by

$$\delta_n^{(k)} = \frac{2^{p+1}}{2^{p+1} - 1} \|\mathbf{X}_n^{(k)}(t) - \mathbf{X}_{n/2}^{(k)}(t)\| + \mathcal{O}(H_n^{p+2}), \quad (6.8)$$

where p denotes the degree of the polynomials that are used to approximate the input data of the sources.

6.2 The Kernel Density Estimation Approach

Kernel density estimates are closely related to histograms. The technique of histogram can deliver exact results for uniformly distributed random variables $x \in \mathcal{U}(a, b)$ with discontinuities at the two boundaries a and b . However, in contrast to determine stochastic distributions by simple histograms, the KDE approach allows to get PDFs with properties such as smoothness and continuity. This allows to calculate PDFs with arbitrary function profiles.

Definition 14 (Kernel Density Estimator). *Let $\mathbf{X} = (x^{(1)}, \dots, x^{(m)})$ be an independent sample drawn from some distribution with unknown density p_X . Via KDE, the PDF can be estimated by*

$$\hat{p}_X = \frac{1}{mh} \sum_{i=1}^m K\left(\frac{x - x^{(i)}}{h}\right), \quad (6.9)$$

where $K : \mathbb{R} \rightarrow \mathbb{R}_+$ is a non-negative function that integrates to one and has mean zero and $h > 0$ is smoothing parameter.

Smoothness or continuity of the resulting distribution \hat{p}_X can be controlled by using a suitable kernel $K(x)$. Here, well known kernel-functions are the gaussian-, cauchy-, picard- and epanechnikov-kernel [14]. In practice, the gaussian-kernel is often used due to its convenient mathematical properties:

$$K(x) = \frac{1}{\sqrt{2\pi}} e^{-x^2/2} \quad (6.10)$$

Note that the usage of uniform distributed kernel-functions in (6.9) is also allowed and coincides with the standard histogram technique. To avoid the effect of oversmoothing or undersmoothing, the KDE method requires judicious choice of the bandwidth h . For example, smaller h makes the KDE more accurate but more noisy, whereas large values for h obscures much of the underlying structure. The effect of the smoothing parameter is shown in Fig. 6.1. The example shows that one should be careful with the choice of the smoothing parameter. Starting from the same set of samples \mathbf{X} and using a gaussian-kernel. The reconstructed PDF is completely different depending on the choice of the smoothing parameter h . For $h = 1$ and $h = 3$ the estimated PDF is undersmoothed. The choice of $h = 4$ yields a PDF which is very similar to the exact PDF. However, a further increase of h cause the effect of oversmoothing. By Scott's rule-of-thumb the choice of h can be chosen optimal in some sense, see [13]. However, by using MATLAB-routines

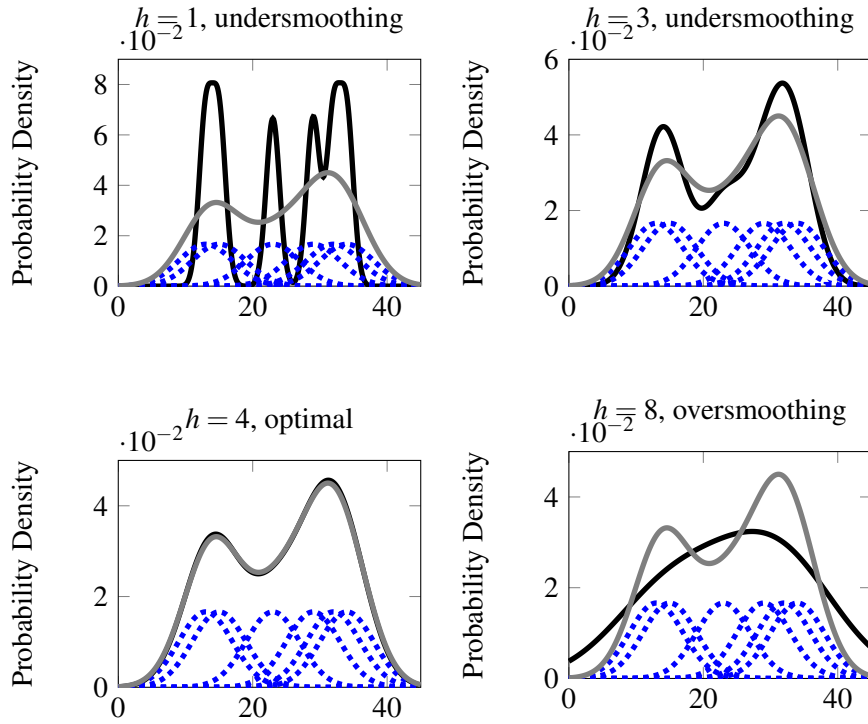


Figure 6.1: Probability density function using the same set of samples for different smoothing parameter h . (gray) is exact PDF. (black) is reconstructed PDF. (blue) set of samples with underlying gaussian-kernel.

the KDE approach combined with an smoothing parameter control is already implemented such that the setting of the smoothing parameter is not further treated in this thesis.

For uncertain coupled problems we aim to estimate the PDF of the contraction factor α_n by using the KDE approach (6.9) with gaussian-kernel (6.10). The sampling of the random parameter space of system (6.1) can be done arbitrary via the Monte-Carlo sampling methods or structural by using tensor-product grids or sparse grids as discussed in Chapter 5.1 and Chapter 5.2. For the structural sampling, the parameter space will be sampled with respect to their distributions, e.g. uniform, normal. Then, a repeated co-simulation of system (6.1) with different parameter realizations out of Ω for all grid points $\mathbf{p}^{(i)} \in \mathcal{G}$ results in a set of samples of scattered values for the contraction factor $\alpha_n = (\alpha_n^{(1)}, \dots, \alpha_n^{(m)})$ calculated by (6.3). Now, the KDE technique tries to estimate the underlying PDF based on these finite amount of input data.

6.3 The Spectral Approach

The spectral approach to determine PDFs is based on the concept of the gPC expansion discussed in Chapter 4.2. The method has been introduced for the first time by **J. LI** and **Y. Marzouk**, see [1]. The idea is to approximate the stochastic process via the gPC expansion and reconstruct the corresponding PDF of the observed output analytically.

Suppose that the system (6.1) is sampled by a grid $\mathcal{G} = \{\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(m)}\}$ of tensor-product or sparse type with grid points $\mathbf{p}^{(i)}$. The approximation of the output $\tilde{\mathbf{x}}(t, \mathbf{p})$ by the gPC expansion in a fixed time point t_n (microstep) requires the calculation of the truncated sum

$$\tilde{\mathbf{x}}_{\text{gPC}}(t_n, \mathbf{p}) = \sum_{j=0}^{P-1} \mathbf{f}_j(t_n) \tilde{\Phi}_j(\mathbf{p}), \quad (6.11)$$

with orthonormal basis polynomials $\tilde{\Phi}_j$ satisfying the condition $\mathbf{E}(\tilde{\Phi}_j, \tilde{\Phi}_i) = \delta_{ji}$ and unknown coefficients $\mathbf{f}_j(t_n)$. Then, after co-simulation has reached some final time t_e for all $\mathbf{p} \in \Omega$, statistical data can be reconstruct by quadrature formulas using the same grid points $\mathbf{p}^{(i)} \in \mathcal{G}$:

$$\mathbf{f}_j(t_n) \approx \sum_{i=1}^m \omega^{(i)} \tilde{\mathbf{x}}(t_n, \mathbf{p}^{(i)}) \tilde{\Phi}_j(\mathbf{p}^{(i)}) \quad \text{for } j = 0, \dots, P-1. \quad (6.12)$$

Consequently, to calculate the gPC expansion (6.11) for all outputs $\tilde{\mathbf{x}} \in \mathbb{R}^n$ and each microstep t_n ($n = 0, \dots, N$) requires the computation of $(N+1) \cdot m \cdot n$ function evaluations of (6.1).

However, usually only one up to a few outputs are considered within the UQ analysis such that the number of global approximations by (6.11) is reducible. Furthermore, the computational effort can be reduced if the global approximations $\tilde{\mathbf{x}}_{\text{gPC}}(t, \mathbf{p})$ for $t \in [t_0, t_e]$ are only computed at the communication points T_n with $t_0 = T_0 < T_1 < \dots < T_n < T_{n+1} < \dots < T_N = t_e$. According to the different time scales of the subsystems, the procedure has the main advantage that each output is given by (6.11) on the same discretized time-grid, i.e., $\tilde{\mathbf{x}}_{\text{gPC}}(T_n, \mathbf{p})$ for $n = 0, \dots, N$.

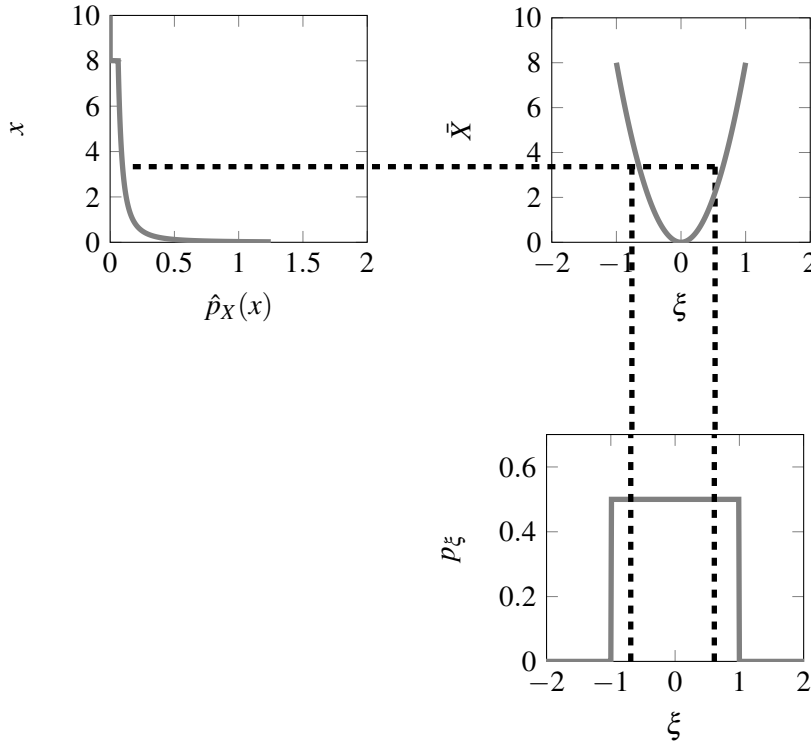


Figure 6.2: Illustrates the mapping of distribution via the spectral method. (bottom, right) distribution of the variable. (top, right) mapping function of the stochastic process. (top, left) resulting probability density function.

We now aim to construct the PDF for the contraction factor α_n of the coupled problem (6.1) within a fixed time window H_n . Therefore, the lower bound estimate (6.3) applied to the solutions $\tilde{\mathbf{x}}(t, \mathbf{p}^{(i)}) \in H_n$ for $i = 1, \dots, m$ (waveforms for all sample points $\mathbf{p}^{(i)}$ on H_n) generates a distributed set of scattered values $\alpha_n = (\alpha_n^{(1)}, \dots, \alpha_n^{(m)})$. This step is the same as for the KDE method. Then, the approximation of the stochastic process of α_n by the gPC expansion based on these finite set of data is given by

$$\alpha_{n_{\text{gPC}}}(\mathbf{p}) = \sum_{j=0}^{P-1} \alpha_{n_j} \Phi_j(\mathbf{p}). \quad (6.13)$$

For simplicity, we abbreviate (6.13) as

$$\bar{X}(\mathbf{p}) = \alpha_{n_{\text{gPC}}}(\mathbf{p}). \quad (6.14)$$

Now we want to deduce a PDF for \bar{X} , say $\hat{p}_{\bar{X}}$. Suppose we want to evaluate this PDF at some $x \in \mathbb{R}$. Then

$$\hat{p}_{\bar{X}}(x) = \sum_{\hat{\xi} \in R_x} \frac{p_{\mathbf{X}}(\hat{\xi})}{|D\bar{X}(\hat{\xi})|} \quad \text{with } R_x = \{\hat{\xi}_1, \dots, \hat{\xi}_N\}, \quad (6.15)$$

where $\hat{\xi}_1, \dots, \hat{\xi}_N$ are the N roots of the polynomial $\bar{X}(\mathbf{p}) - x = 0$, see [1]. In other words, many possible $\hat{\xi}$ may give use this particular x and all of them contribute to the probability density at position x . For one dimension $\bar{X}: \mathbb{R} \rightarrow \mathbb{R}$ the denominator becomes the absolute value of the first derivative: $|D\bar{X}(\hat{\xi})| = |d\bar{X}/d\xi|$ evaluated at $\hat{\xi}$.

Remark 23. To generalize (6.15) to multiple dimensions, i.e., more than one random variable, the quantity $|D\bar{X}(\mathbf{p})|$ becomes the absolute value of the Jacobian-determinant. Here, some further steps are required for $\dim(\bar{X}) \neq \dim(\mathbf{p})$, see [1]. Furthermore, the root-finding becomes much more difficult.

Figure 6.2 illustrates the non-linear mapping of a uniformly distributed random variable. The stochastic process is approximated by a second order gPC expansion ($p = 2$), i.e., polynomials with degree at most two are involved.

Notice that the drawn PDF is only exact if the stochastic process is exactly represented by the gPC expansion. Due to truncation, this requirement shall not apply for the most cases. However, an increase in the order of the gPC expansion provides a better approximation and thus a better estimate of the PDF by (6.15).

7 Chapter 7

Numerical Examples

Within this chapter the theoretical results and new methods of Chapter 3 - 6 are numerically verified. For this purpose, we take different examples from the electrical engineering.

Modeling Aspects and Simulator Packages

The circuit part is modeled via modified nodal analysis (MNA), see Chapter 2, by the Octave Circuit Simulator (OCS) package. The field device is a single phase isolated transformer modeled in 2D. The modeling is done with the Finite Element Method Magnetics (FEMM) software package. The spatial discretization of the 2D surface is done by triangulation, see [24], and thus the problem can be traced back to a common DAE-DAE coupling, see Chapter 3.1.2. The OCS package communicates with the Field Device Simulator (FIDES), see [7], for magnetoquasistatic field simulations, see Chapter 2.2. For the time-integration of all subsystems the microsteps are chosen small enough such that the total time points are sufficient to render the dynamics of the systems (field and circuit). Consequently, time-integration inaccuracies can be neglected, since they are comparatively small compared to the co-simulation splitting errors. The coupling of the circuit with the field device is done by add additional unknowns (coupling variables) to the network equations (source coupling), see eq. (2.3). The circuit is excited by an additional controlled current or voltage source driven by the previously computed waveform (branch current or node voltage on H_n) of the field device provided that they share two nodes (including the ground-potential), see Chapter 2.3. For the inverse computational sequence the way of update the field is the same. The strongly coupled network (monolithic model) is simulated by using the FIDES and OCS packages as well. Here, the magnetoquasistatic equations (2.12) are solved along the timeline together with the circuit equations (2.2) by using the same time stepping scheme.

All simulations ran on a common PC with Macintosh Operating System (OS X 10.10.5): processor type Intel Core i5, 2.9 GHz, L2-Cache (per core): 256 KB, RAM: 8 GB.

The following topics are discussed in their corresponding chapters:

Chapter 7.1: the convergence and the contraction of the RL-circuit studied in Chapter 3.2.3 are numerically verified. The results were first presented at ECMI 2014 and published in [39].

Chapter 7.2: the R-splitting and LR-coupling with their application in field/circuit coupled problems are considered. Basically, the theoretical results of Chapter 3.2 are numerically verified. Both splittings are compared to the standard way of coupling (cutting at the EM device boundaries) with respect to the convergence rate and the speed of contraction. The results were presented at COMPUMAG 2015 and SCEE 2016 and published in [36,37].

Chapter 7.3: the accuracy of the lower bound estimate of the contraction factor given in Chapter 6.1 is numerically verified with application in a field/circuit co-simulation, see [37].

Chapter 7.4: the KDE and the spectral approach of Chapter 6 are used to estimate the corresponding probability density function of the contraction factor for an uncertain field/circuit coupled problem. The results were first published in [37].

Chapter 7.5: the non-intrusive gPC expansion discussed in Chapter 4.2.2 is applied to co-simulation. Basically, the propagation of stochastic quantities (expectation and standard deviation) are considered within the dynamic iteration process. The results were presented at SCEE 2014 and published in [39].

7.1 Fast Contraction and Higher Order Co-Simulation

The RL-circuit as discussed in Chapter 3.2.3 (see Fig. 3.6) is numerically verified with respect to the convergence rate and the number of iteration steps that are required to attain a predefined accuracy. The following set of parameter-values are employed: resistance $R = 10\text{k}\Omega$ and inductance $L = 1\text{mH}$. The circuit is supplied by a voltage source $U_{\text{in}}(t) = 1\text{V} \cdot \sin(\omega t)$ with angular frequency $\omega = 2\pi \cdot 5 \cdot 10^3\text{s}^{-1}$ ($f = 5\text{kHz}$).

To investigate contraction and convergence, the co-simulation is studied within a fixed time interval $[t_0, t_0 + H_n]$ with initial time $t_0 = 0.4\text{ms}$ and time window size $H_n = 10^{-4}\text{s}$. This is sufficient to cover a half of the period of the input voltage source U_{in} . The accuracy of the solutions on the n -th time window after k iterations $\tilde{\mathbf{X}}^{(k)}(t)$ is measured by comparing with a reference solution $\mathbf{X}_m(t)$ computed by a monolithic simulation (strongly coupled): $\Delta_n^{(k)}(t) = \mathbf{X}_m(t) - \tilde{\mathbf{X}}^{(k)}(t)$ and lumped error $\delta_n^{(k)} := \sup_{t \in H_n} \|\Delta_n^{(k)}\|_2$. For both splitting functions (3.64) and (3.65) (subs. 1 and subs. 2 first) a constant extrapolation of the initial value is used as initial guess $\tilde{\mathbf{X}}^{(0)}(t)$ on H_n .

Contraction and Convergence: Fig. 7.1 (left) shows convergence and (right) contraction for both computational sequences. Thus, we have convergence even so the estimate (3.66) does not indicate this behavior, see eq. (3.66). Moreover, both subsystems are precisely solved, i.e., no more splitting errors occur, after each subsystem is updated only once, where the solution of each subsystem (waveforms on H_n) is bounded by the time-integrator accuracy of about $\text{AbsTOL} = \text{RelTOL} = 10^{-6}$. The tolerances are used to limit the local discretization error. If the difference between a high-accuracy and low-accuracy time-integration on H_n is higher than one of the tolerances, then the microsteps size h_n will be reduced. However, the time-integration can meet much better accuracies for very small time window sizes. The simulation results confirm exactly the theoretical result predicted by the fine structure analysis, see eq. (3.69), since the spectral radius of the corresponding (exact) recursion matrix is zero and thus no splitting errors occur after one iteration. Additionally, we observe two different convergence rates for the different computational sequences. For subs. 1 first the window-wise convergence $\mathcal{O}(H_n)$ is observed, whereas for subs. 2 first a higher order co-simulation of $\mathcal{O}(H_n^2)$ is achieved. This can be explained as follows: Constant extrapolation produces an splitting error of $\mathcal{O}(H_n)$. For subs. 2 first, the coupling variable U_{Co} is extrapolated constantly. This variable is coupled to the algebraic node potential U_1 . Using Kirchhoff's Current law it follows for the coupling current $I_{Co} - I_L = 0$. However, the inductor as dynamic element is involved in subs. 2 with differential current:

$$\frac{d}{dt}I_L = \frac{U_L}{L}, \quad \text{with} \quad U_L = U_1 - U_2.$$

Consequently, the coupling current I_{Co} is improved during time-integration and the convergence rate is improved from $\mathcal{O}(H_n)$ to $\mathcal{O}(H_n^2)$. For subs. 1 first, the coupling variable I_{Co} is constantly

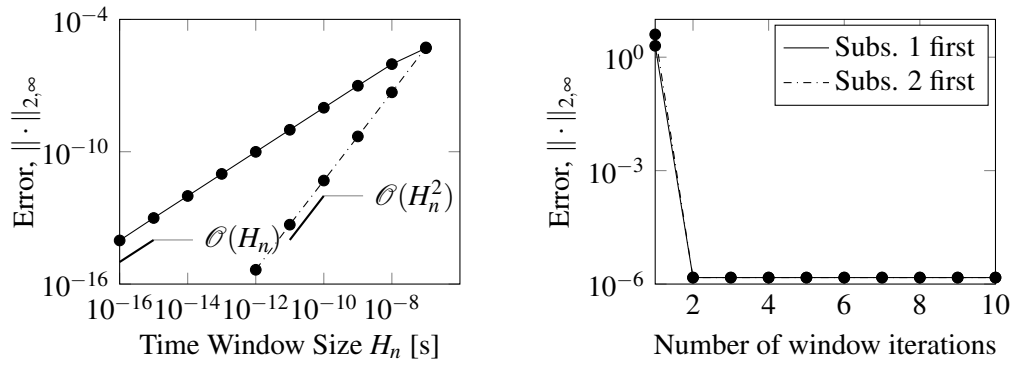


Figure 7.1: Convergence and contraction for the LR-circuit given in Fig.3.6. (Left) lumped error for different time window sizes H_n and one iteration step per window. (Right) lumped error in dependence of iteration steps k on $[0.4, 0.5]$ ms ($H_n = 0.1$ ms). [39]

extrapolated. Due to the fact that subs. 1 is merely a system of algebraic equations, see eq. (3.61), the effect of higher order co-simulation is no longer existent.

Conclusions

The theoretical results of Chapter 3.2.3 are verified by numerical simulations. The co-simulation model is solved without splitting errors after each subsystem is updated only once, whereas the standard recursion estimate (3.20) does not indicate this behavior. Furthermore, we showed that a higher order co-simulation with constant extrapolated coupling variable is feasible by exploiting network topology in the surrounding of the interface (controlled sources). These result is in line with the theoretical result for the LR-coupling approach of Chapter 3.2.2 for DAE-DAE coupled problems, see Chapter 3.1.2, where the order is improved from $\alpha_n + \mathcal{O}(H_n)$ to $\mathcal{O}(H_n)$, see Theorem 10.

7.2 Coupling Interfaces in Application

In this chapter the R-splitting and LR-coupling as introduced in Chapter 3.2.1 and Chapter 3.2.2 are numerically verified with their applications for field/circuit coupled problems, see Chapter 2.3. Both splittings will be compared to the standard way of cutting at the EM device boundaries where co-simulation is unconditional stable for the computational sequence of field first, see Chapter 3.1.3. The computational effort is measured in terms of the number of iteration steps and finally we count the total number of solved linear systems as quantitative measure.

7.2.1 R-Splitting for Field/Circuit Coupled Problems

Now, the advantage of R-splitting as alternative splitting approach for co-simulation is highlighted. Recall that the contraction factor and the constant of the extended recursion deduction

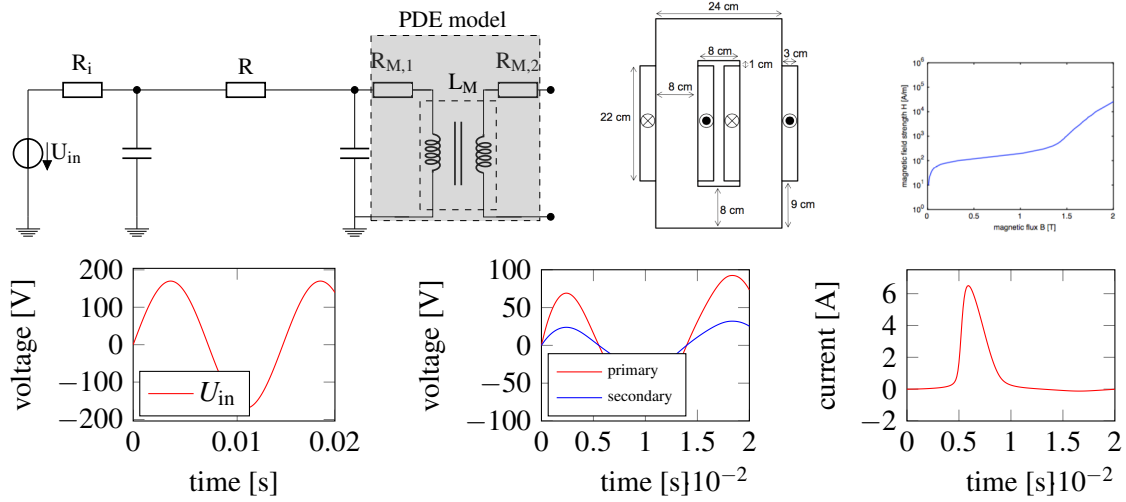


Figure 7.2: (top, left) Field/circuit co-simulation test circuit, (top, middle) specification of the transformer, (top, right) non-linear BH-curve, (bottom, left) sinusoidal input voltage, (bottom, middle) primary and secondary voltage, (bottom, right) inrush current.

read

$$\alpha_n = \frac{G_2}{G_1}, \quad C = G_2 \left(\frac{G_2}{G_1 [1 - L_f(1 + L_\Phi)H_n]} + \frac{\frac{G_2}{1 - L_f(1 + L_\Phi)H} H - 1}{1 - (L_f(1 + L_\Phi) + G_1)H} \right), \quad (7.1)$$

with Lipschitz constants L_Φ , L_f and conductances $G_1 = 1/R_1$, $G_2 = 1/R_2$, see Theorem 8. Therefore, the coupling strength of co-simulation with R-splitting is specified by the ratio of the coupling resistances. That is, for an increasing resistance R_2 the constant C and the contraction factor α_n decreases, i.e., both quantities benefit from resistance splitting. In other words, R-splitting is able to affect co-simulation in an appropriate and positive manner such that the coupling between both subsystems becomes strong.

The electrical engineering test example is shown in Fig. 7.2 (top, left): an electric circuit coupled to a EM device (transformer model).

EM device: The field part is a single phase isolated transformer with a non-linear BH-curve (non-linear model) typically used in low frequency applications, for example, in power supply units of electrical devices. The resistances of the coil windings are extracted from the PDE model in FEMM, see [6]: internal resistance $R_{M,1} = 0.449\Omega$ (primary side) and $R_{M,2} = 0.062\Omega$ (secondary side). This transformer has 260 turns on the primary and 90 turns on the secondary side without load (no-load test). Thus, the secondary voltage is scaled to a third of its primary voltage, see Fig. 7.2 (bottom, middle). Usually, when electrical devices like electrical machines or transformers are turned on, instantaneous an high input current occurs, see Fig. 7.2 (bottom, right). Especially, when transformers are magnetized for the first time a high magnetizing current flows for several cycles, since the magnetizing flux operates in the saturation region of its BH-curve. The occurred current is referred to as the inrush current, input surge current or switch-on surge [62]. Particularly, when the transformer operates in no-load, i.e., with its secondary side open, the current can be 10 to 15 times larger than the mean value and can flow for several cycles. These circumstance makes our test circuit difficult for co-simulation. However, it shows the potential of R-splitting.

Circuit: The circuit part is a two-stages low-pass filter supplied by a sinusoidal voltage source $U_{in} = A \sin(\omega t)$ at low frequency $f = 60\text{Hz}$ with amplitude $A = 170\text{V}$. As resistances and capaci-

tances we choose: $R_i = 10\Omega$, $R = 20\Omega$ and $C_1 = C_2 = 1\text{nF}$. The setting yields corner frequencies of approximately 16 MHz for the first and 8 MHz for the second stage low-pass filter. Consequently, the amplitude is not attenuated at the input (primary side) of the transformer.

Coupling: The coupling of field and circuit is managed by the software package FIDES, see [7]. The data exchange between both subsystem is organized via controlled current and voltage sources (source coupling), see Chapter 3.1.3. In fact, the resistance R is split into two resistances in series: $R_1 := \omega R$, $R_2 := (1 - \omega)R$ with $\omega \in [0, 1]$, see Fig. 7.3. For a splitting ratio $\omega \in (0, 1)$ the controlled sources are directly connected to resistances which ensures the mutual algebraic dependence of the subsystems. As a consequence, source-coupling does not change the differential index property. Note that for the limits of the ratio parameter ($\omega = 0$ and $\omega = 1$) the subsystem which is coupled by the controlled voltage source becomes index-2, if there is a loop of capacitances and at least one voltage source only (CV-loop), see Chapter 2.3. Therefore, it is assumed that the controlled sources are placed in such a way that both (all) subsystems remain an index-1 problem for all $\omega \in [0, 1]$. This leads to the following:

Remark 24 (Avoiding Index-2). *When R-splitting with $\omega \in [0, 1]$ is used for a field/circuit coupled problem (2.14) with arbitrary circuit part (provided index-1), then Ass. 2 (index-1 assumption) is particularly fulfilled for voltage driven MQS devices.*

With respect to (7.1), the ratio parameter has been chosen small such that the contraction factor becomes small. However, the contraction factor must not disappear completely, since the dependence of the circuit and the field shall remain algebraic for both computational sequences (field and circuit first). For our test simulation the ratio parameter of R-splitting has been chosen as $\omega = 0.05$. Consequently, the resistance R is split into parts of $R_1 = 1\Omega$ and $R_2 = 19\Omega$. By following Theorem 8 a contraction factor of $\alpha_n = 1/19$ occurs.

Simulation settings: Co-simulation is performed on $\Gamma = [t_0, t_{end}]$. The strongly coupled system, see Fig. 7.2, is computed on Γ as well and serves as our reference solution. The strongly as well as the weakly coupled system is solved by the same time stepping scheme `ode23s` (Rosenbrock method) with accuracy $\text{AbsTOL} = \text{RelTOL} = 10^{-5}$. The microsteps h_n of time-integration are chosen such that the number of total time points are sufficient to render the dynamics of the respective system. Consequently, only the multi-window propagation of splitting errors within the dynamic iteration process occurs.

Convergence and Computational Effort

Two different test scenarios are performed. First, we investigate the maximum possible window size for convergence which serves as indicator for the strength of the coupling. Secondly, the computational effort is analyzed for the R-splitting and the standard coupling approach.

Window Size: Co-simulation is studied on $\Gamma = [0, 10]\text{s}$ with various time window sizes H_n . Figure 7.3 (right) shows the error in the primary current of the transformer by using the R-splitting and the standard approach (for the computational sequence of field first), for different window sizes H_n and increasing iteration steps k per window. Cutting at the EM device boundaries is already divergent for time window sizes $H_n > 2.5\text{ms}$, whereas R-splitting supports time window sizes up to $H_n = 10\text{s}$. and thus obviously R-splitting accepts much larger time windows. The simulation result proves Theorem 8 numerically: The strength of coupling between field and circuit can be affected by the resistances R_1 and R_2 . For $R_2 \rightarrow \infty$ it follows $\alpha_n = 0$ and $C = 0$ and thus the co-simulation with R-splitting interface is converged for every $H_n \subseteq [t_0, t_{end}]$.

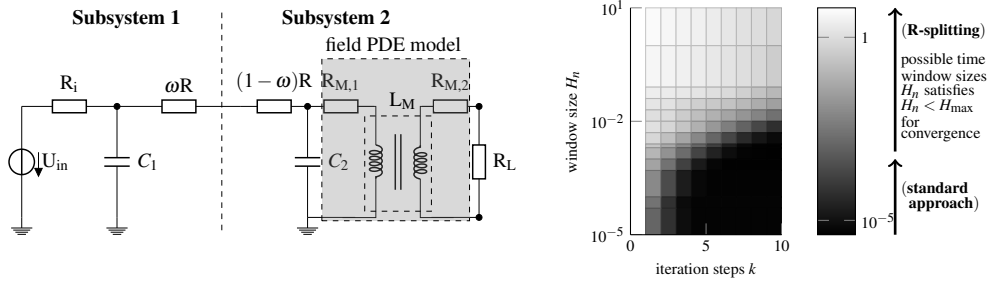


Figure 7.3: (left) Field device coupled to a circuit with R-splitting. Settings: $R_i = 10\Omega$, $R = 20\Omega$, no-load test ($R_L \rightarrow \infty$), $C_1 = C_2 = 1\text{nF}$, $U_{in}(t) = 170\text{V} \cdot \cos(2\pi 60\text{s}^{-1} \cdot t)$. (right) Window sizes H_n versus number of iteration steps k for cutting at the EM device boundaries (field first) and for R-splitting. [37]

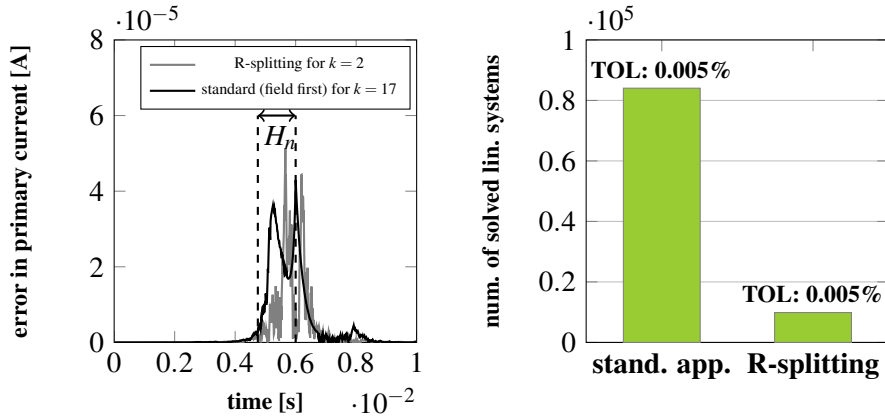


Figure 7.4: (left) Error of co-simulation measured in the primary current of the transformer in the time interval $[0, 0.01]\text{s}$ with $H_n = 10^{-3}\text{s}$. (right) Computational effort for R-splitting and cutting at the EM device boundaries (field first) in terms of solved linear systems of equations. [37]

Computational Effort: Now, co-simulation is studied on a smaller time interval $\Gamma = [0, 0.01]\text{s}$ with constant time window size $H_n = 10^{-3}\text{s}$. Thus, still a half of the first period (switching-on phase) of the input voltage source U_{in} is covered including the high inrush current peak, see Fig. 7.2 (bottom, right). Figure 7.4 compares the computational effort for both approaches (R-splitting and standard approach). With R-splitting the coupling becomes stronger and the number of required iteration steps decreases to achieve a predefined accuracy, see Fig. 7.4 (left). Here, R-splitting only requires two iteration steps to be close to the reference solution. Consequently, the coupled system is solved within the tolerance after the circuit and the field is updated only once, whereas the standard coupling approach requires $k = 17$ iteration steps to be in a similar scale of errors. To assess platform (computer system) independent, the computational effort is measured in terms of linear systems. Cutting at the EM device boundaries requires to solve 84047 linear systems, whereas the R-splitting approach only requires to solve 9872 linear systems, see Fig. 7.4(right). Consequently, the computational effort can be reduced by 90% to achieve the same accuracy in the solution of about 10^{-5} . A further reduction of the ratio parameter ω , i.e., for $\omega < 1/19$, only yields a slightly improvement, since the contraction factor is close to zero and the coupling strength is basically determined by its constant C .

Inrush Currents in Field/Circuit Coupled Problems

The inrush current through the transformer is limited by the equivalent circuit resistance (primary side surrogate resistance). The number of iteration steps which are required within the dynamic iteration process for field/circuit coupled problems are primarily affected by the amplitude of the inrush current during the switch-on phase. By cutting at the EM device boundaries into circuit and field, the information about the surrogate resistance is shifted to the circuit subsystem and thus the field device responds with its maximum inrush current for several cycles. Here, by shifting the resistance from the circuit to the field limits the current through the primary coil.

To demonstrate the impact of inrush currents in co-simulation, we compare two different settings. First, we chose R-splitting with ratio parameter ω as before ($\alpha = 1/19$). Consequently, a major part of resistance R is displaced to the field. Then, we investigate the standard decoupling approach, i.e., $\omega = 1$, with the computational sequence of field first. The co-simulation is tested on $\Gamma = [0, 0.02]$ s with time window size $H_n = 10^{-3}$ s. This is approximately one period of the input voltage source U_{in} and is sufficient to cover the non-linear start-up phase of the transformer.

Figure 7.5 (top) shows the primary (inrush) current through the transformer for the field/circuit test example of Fig. 7.3 for R-splitting (top, left) and for the standard approach (top, right). Figure 7.5 (bottom) shows the related error in the primary current measured by comparing with the reference solution (primary current obtained by solving the monolithic model Fig. 7.2 (bottom, right)) for an increasing number of iteration steps. For R-splitting the primary current is already close to the reference solution after one iteration step has been performed. However, cutting at the boundaries induces a high inrush current of about 2 times of the mean current during the first iteration. Therefore, co-simulation requires further iteration steps to adjust the deviation.

7.2.2 LR-Coupling for Field/Circuit Coupled Problems

In this chapter the LR-coupling approach given in Chapter 3.2.2 is treated as a further alternative to the standard coupling approach of Chapter 3.1.3. To highlight the advantage of LR-coupling, our focus is on the convergence rate and on the speed of contraction of a dynamic iteration process.

The engineering test example for co-simulation is a transformer typically used in low frequency applications connected by a wire to a supply voltage source, see Fig. 7.6(top, left).

EM device: The transformer model is the same as in Chapter 7.2.1. The transformer operates without a load resistance (no-load test) and thus the secondary current becomes zero. Again a high inrush current occurs during the switch-on phase, see Fig. 7.6 (bottom, right). Further technical details can be found in Chapter 7.2.1.

Circuit: The circuit part consists of a first order RC low-pass filter followed by a second order RLC low-pass filter. The circuit is excited by a sinusoidal supply voltage source at low frequency $f = 60$ Hz and amplitude $A = 170$ V. The following parameter values are used: $R_i = 10\Omega$, $R_W = 2\text{k}\Omega$, $C_{W1} = C_{W2} = 1\text{nF}$ and $L = 1\text{nH}$. These particular choice yields corner frequencies of $f_{g1} = 1/(2\pi \cdot R_i \cdot C_{W1}) = 16$ MHz for the first order low-pass filter and $f_{g2} = 1/(R_W \cdot C_{W2}) = 500$ kHz, $f_{g3} = 1/(\sqrt{L_W \cdot C_W}) = 1$ GHz for the second order low-pass filter. Consequently, the supply voltage U_{in} is not attenuated at the input of the transformer.

Coupling: Again, the data exchange between the subsystems is managed by controlled sources (source coupling), where the LR-coupling approach is used to couple field and circuit part. To this

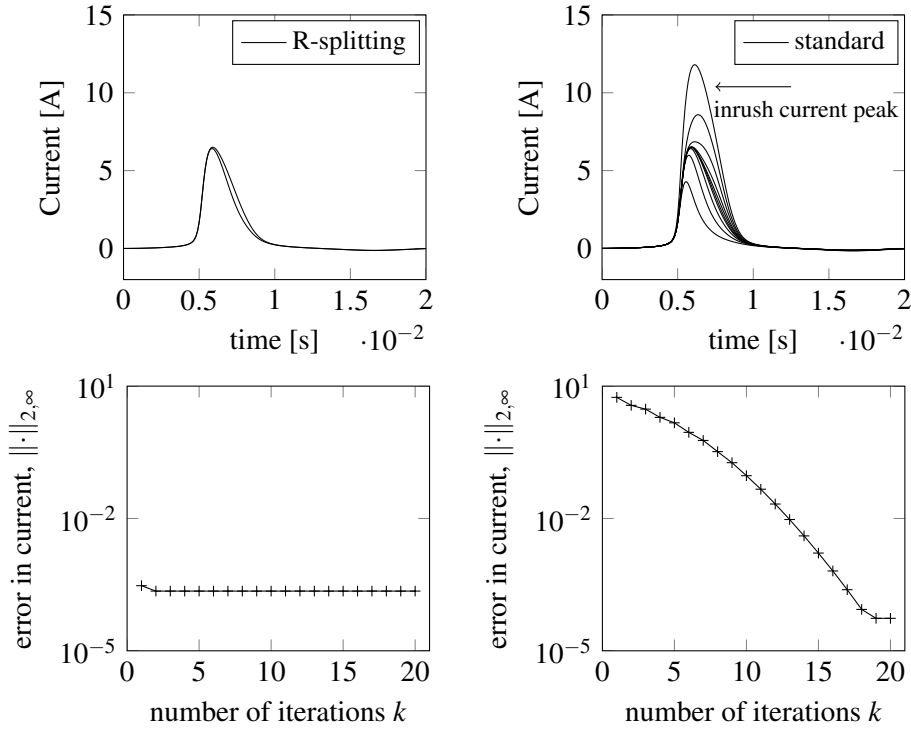


Figure 7.5: (top, left) Inrush current of the transformer using the R-splitting ($\omega = 0.05$) for $k = 1, \dots, 20$ iterations (fast contraction). (top, right) Inrush current using the standard approach and field first for $k = 1, \dots, 20$ iterations (slow contraction). (bottom, left and bottom, right) Contraction of field/circuit co-simulation. Splitting errors in the primary current of the transformer in dependence of the number of iterations.

end, the resistance R_W is split into two resistances $R_{W,1} = 1\text{k}\Omega$ and $R_{W,2} = 1\text{k}\Omega$, where $R_{W,2}$, L_W and C_{W2} is shifted to the field, see Fig. 7.7. For the computational sequence of subs. 1 and subs. 2 first both subsystems are mutual algebraically coupled by eq. (3.49a), see DAE-DAE coupling of Chapter 3.1.2. Therefore, the convergence is not guaranteed by following the standard recursion estimate of Lemma 3. However, our numerical results show the benefit of LR-coupling compared to the (for the computational sequence of field first) unconditionally stable standard approach of Chapter 3.1.3.

Convergence and Contraction

Convergence Analysis: To verify the rate of convergence, we inspect the splitting error in the primary and secondary voltage as well as in the primary current of the transformer. To start up co-simulation, we employ constant extrapolation for the coupling variables I_{Co} and U_{Co} . Starting with zeros as initial setting, we first solve the model (strongly and weakly coupled) up to $t_0 = 0.001\text{s}$ for consistency. Then, convergence is investigated on the first time window $[t_0, t_0 + H_n]$ by using consistent initial values. Since co-simulation is performed on a fixed time window H_n , the error is not propagated to subsequent windows (no error transport), which ensures to measure the pure splitting error.

For LR-coupling as well as for the standard field/circuit coupling approach the same equidistant

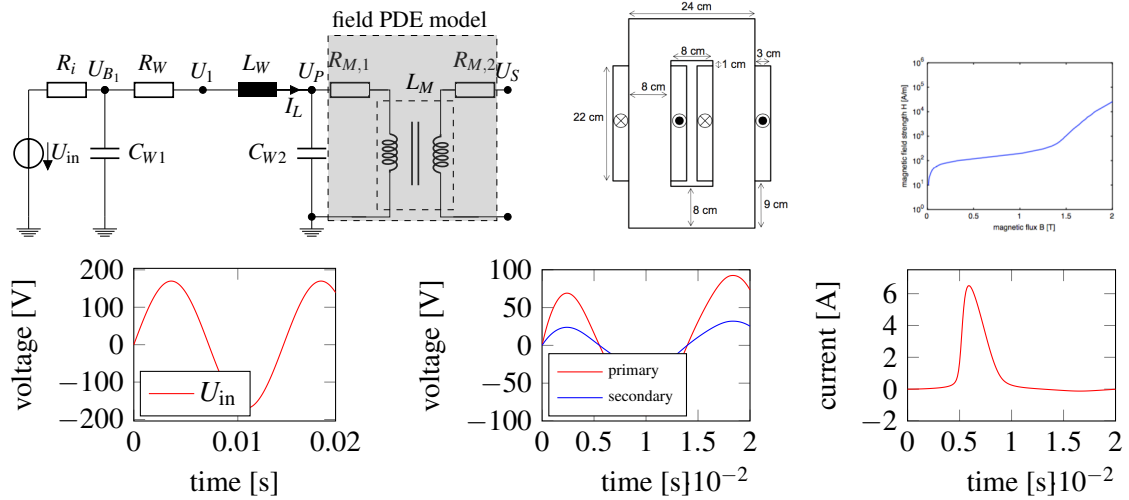


Figure 7.6: (top, left) Field/circuit co-simulation test circuit, (top, middle) specification of the transformer, (top, right) non-linear BH-curve, (bottom, left) sinusoidal input voltage, (bottom, middle) primary and secondary voltage, (bottom, right) inrush current

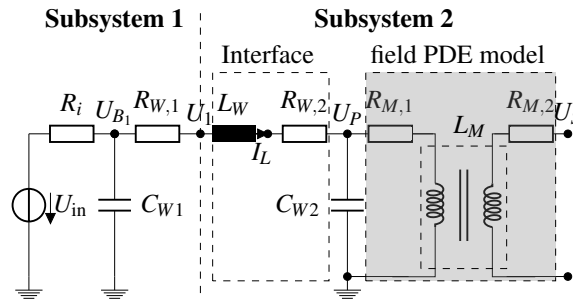


Figure 7.7: LR-coupling for the field/circuit co-simulation model. Subs. 1 is a purely circuit, whereas subs. 2 consists of a circuit and a field device. [36]

microsteps are used for the time stepping methods. Hence no interpolation techniques are required to measure errors, since each time-integrator provides a solution on the same discretized grid $t_0 < t_1 < \dots < t_n < t_{n+1} < \dots < t_e$. The errors are given with respect to the corresponding reference solution computed by a monolithic simulation. Figure 7.8 shows the error in the primary and secondary voltages of the transformer for various time window sizes H_n . It shows that the gain of accuracy per iteration is $\mathcal{O}(H_n)$ for both approaches. This result fits to the predicted convergence rate of Theorem 10 for LR-coupling and proves the convergence rate numerically.

However, a higher convergence rate of $\mathcal{O}(H_n^2)$ is achieved concerning the primary current of the transformer, see Fig. 7.8 (right, solid line). This behavior can be explained as follows: the old iterate $I_{Co}^{(k-1)}$ (coupling current) which enters subs. 1 is directly linked to the induction current $I_L^{(k)}$ by the Kirchhoff's current law $0 = I_{Co}^{(k-1)} + I_L^{(k)}$. Recall that $I_L^{(k)}$ is differential, thus it is obtained by time-integration. With C_{W2} small enough (in our case $C_{W2} = 1\text{ nF}$) the capacitance connected to ground can be neglected and thus only a negligible current is flowing through the capacitor against ground. Again, Kirchhoff's current law for the node U_P yields a direct link between the induction current and the primary current and $0 = I_p^{(k)} + I_L^{(k)}$ holds for slow alternating node potential U_P and C_{W2} small enough. Therefore, the convergence of the primary current is enhanced during time-integration. However, the lumped error in all unknowns will remain unaffected. Consequently,

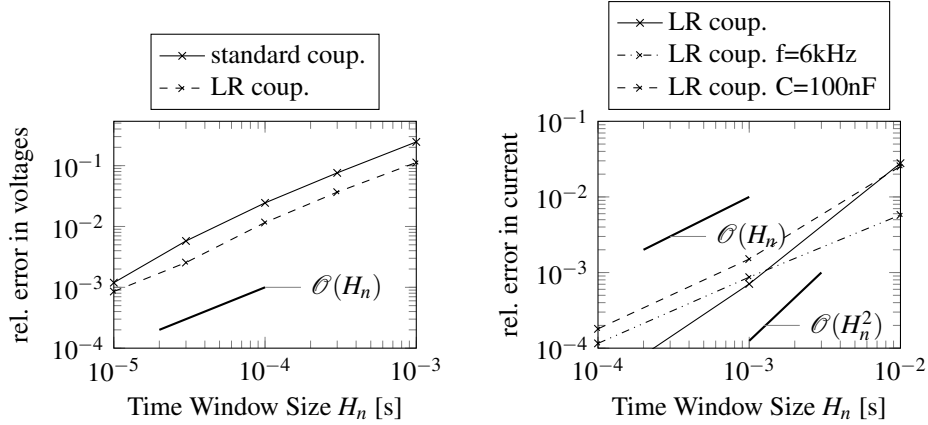


Figure 7.8: (left) Convergence concerning the primary and secondary voltages and (right) convergence concerning the primary current. The figures show the splitting error in dependence of the time window sizes H_n and one iteration per time window. [36]

LR-coupling provides overall convergence rate $\mathcal{O}(H_n)$, while the error behavior of the primary current benefits from its surrounding circuit structure.

To confirm the effect of higher order, we repeat co-simulation with an increased capacitance $C_{W2} = 100\text{nF}$ as well as for an increased frequency $f = 6\text{ kHz}$. Now, the current through the capacitance becomes more important. Notice that this current is governed by the node potential U_P which offers convergence rate $\mathcal{O}(H_n)$, see Fig. 7.8 (left). Consequently, the gain of accuracy per iteration is reduced to $\mathcal{O}(H_n)$, since the primary current of the transformer do not longer benefit from its surrounding network structure.

Window Size: We investigate co-simulation by using the LR-coupling approach with respect to the maximum time window size for convergence. The time-integration of all subsystems is done by the implicit Euler method with a Newton tolerance of 10^{-5} . For the standard coupling approach, co-simulation starts the computation with the field where convergence is guaranteed. However, for the LR-coupling approach, co-simulation starts the computation with the circuit which creates a mutual algebraic dependence between the subsystems. Figure 7.9 shows the solutions (waveforms) of the primary voltage within the first time window H_n , i.e., the time interval $[0, H_n]$, for an increasing number of iteration steps $k = 1, \dots, 10$. By cutting at the EM device

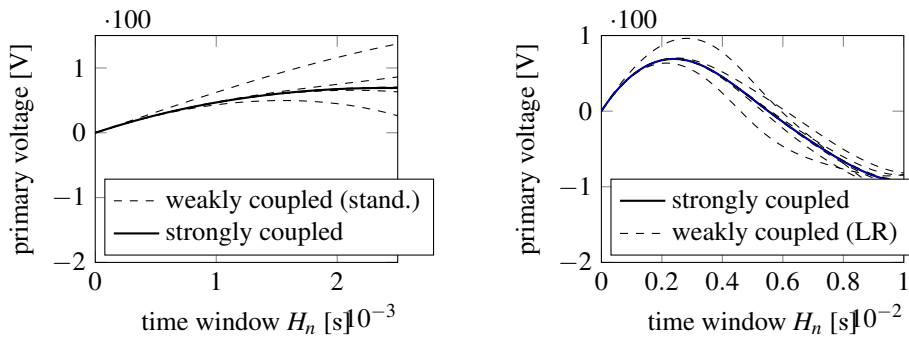


Figure 7.9: Waveforms of the primary voltage on the time interval $[0, H_n]$ for increasing iteration steps $k = 1, \dots, 10$. (left) for the standard approach and $H_n = 2.5\text{ms}$, (right) for the LR-coupling and $H_n = 10\text{s}$ (figure shows only a cutout). [36]

boundaries (standard coupling approach) the maximum possible time window size for convergence (already tested in Chapter 7.2.1) is $H_n = 2.5\text{ms}$, see Fig. 7.9 (left). Now, LR-coupling enables to use time window sizes up to $H_n = 10\text{s}$ (maximum tested window), where Fig. 7.9 (right) shows only a cutout. However, the use of large window sizes usually increases the number of iteration steps to achieve a certain tolerance. Consequently, a fair comparison of both approaches is to use the same window size and count the number of iteration steps which are required to be close to the reference solution (strongly coupled model). This is analyzed next.

Contraction: Now, we analyze contraction on the time interval $\Gamma = [0, 0.02]\text{s}$, which is about one cycle of the supply voltage U_{in} . First, a constant time window size for both types of coupling is applied. Here, the largest possible window size of the standard approach, i.e., where the standard approach is still convergent, is used to proceed sequentially over Γ , i.e., $H = 2.5\text{ms}$.

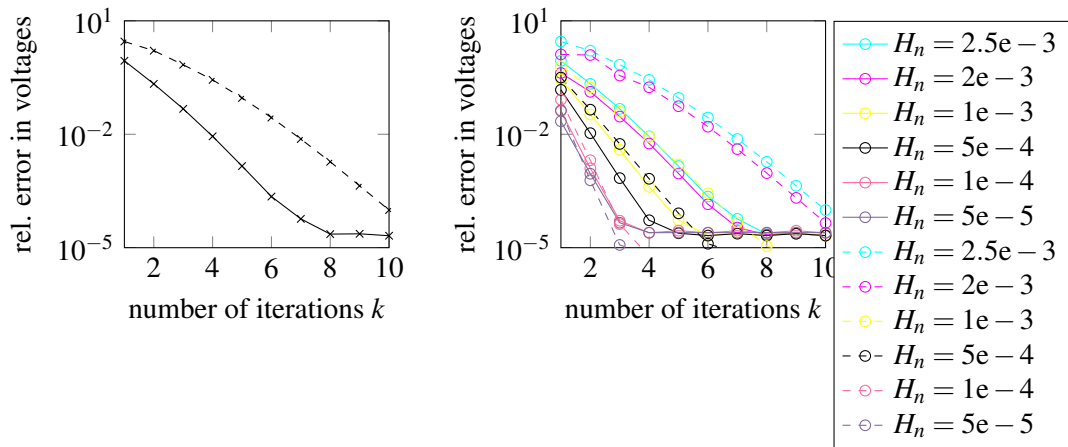


Figure 7.10: (left) Contraction of field/circuit co-simulation. Figures show the splitting error in the primary and secondary voltages on the full time interval $[0, 0.02]\text{s}$ with (left) constant time window size $H_n = 2.5\text{ms}$ for both approaches and (right) different time window sizes. For a given time window H_n the dashed line refers to the standard approach, whereas the solid line refers to the splitting error of the LR-coupling. [36]

Figure 7.10 shows the splitting error in the primary and secondary voltages of the transformer plotted against the number of iteration steps per window. It shows that both approaches are convergent. Again, the speed of contraction of co-simulation with given H_n is affected by the constant $C_{I_{co}}$ of eq. (3.53) reflecting the strength of the coupling. It appears that LR-coupling requires less iteration steps to be close to the reference solution. Recall that the convergence rate which is determined by the diagonal element of the recursion matrix in (3.54) is $C_{I_{co}} \mathcal{O}(H_n)$. Therefore, the number of iteration steps for a given time window H_n and predefined accuracy depends on the size of the constant $C_{I_{co}}$. Consequently, the number of required iterations decreases with decreasing time window sizes, see Fig. 7.10 (right). It shows that the benefit of the LR-coupling approach is lost once the time window size dropped below a certain value of about 10^{-4}s . The balance of the number of iteration steps, the time window size and the time-integration accuracy is a common problem in co-simulation is difficult to determine, since all quantities are mutually affect each other.

Conclusions

The theoretical results of the R-splitting and LR-coupling approach of Chapter 3.2.1 and Chapter 3.2.2 were confirmed by numerical simulations. Their benefit in co-simulation has been tested for field/circuit coupled problems (2.14). Therefore, we analyzed the convergence and contraction for both approaches. We numerically verified the convergence rate of $\mathcal{O}(H_n)$ for the R-splitting and LR-coupling, see Theorem 8 and Theorem 10, respectively. Furthermore, we showed numerically that R-splitting enables to enhance the strength of coupling by shifting a major part of the coupling resistance to the field-subsystem.

Cutting at the EM-device boundaries is always convergent for the computational sequence of field first. However, we showed that both splittings support larger time windows and require much less iteration steps to achieve a predefined accuracy. Consequently, both approaches enable to reduce the computational effort, which we verified in terms of solved linear systems. For our choice of parameter values, R-splitting is more contractive as the LR-coupling. The examples demonstrate the interplay between iteration steps, time window size and desired time-integration accuracy. It shows that a combined window size and sweep control enables to further improve the efficiency.

7.3 Accuracy of the Lower Bound Estimator for Purely Algebraic-to-Algebraic Coupling

Within this chapter, the accuracy of the lower bound estimation of Chapter 6.1 is numerically verified. To this end the R-splitting of Chapter 3.2.1 is used, where the contraction factor is known analytically and given by the ratio of the coupling resistances, see Theorem 8. Recall that R-splitting fulfills the purely algebraic-to-algebraic coupling condition, since only old algebraic unknowns enter algebraic constraints. Consequently, ineq. 6.3 is a lower bound for α_n in (3.35) and thus the accuracy of the estimate can be easily verified by comparison of the exact and approximative value.

Simulation settings: We consider the engineering test example already introduced in Chapter 7.2.1. This is a transformer connected by a two-stage low-pass filter to a sinusoidal voltage source at low frequency ($f = 60$ Hz), see Fig. 7.3. For further technical details see Chapter 7.2.1. As resistances and capacitances we choose $R_i = R = 20\Omega$ and $C_1 = C_2 = 1\text{nF}$, which result in a low-pass filter with two equal corner frequencies of approximately $f_g = 8$ MHz. As a consequence, the signal of the supply voltage U_{in} is not attenuated at the input (primary side) of the transformer. The problem is simulated on the time interval $[0, 0.01]\text{s}$, which is approximately one period of the supply voltage.

Remark 25. *R-splitting gives the opportunity for several test cases with convergence and divergence characteristic by a different choice of the ratio parameter ω . For the limits $\omega = 0$ and $\omega = 1$ the subsystems are coupled via an ordinary differential equation in subs. 1 or subs. 2, since one coupling node becomes differential. Thus, the algebraic-to-algebraic splitting structure (6.2) is not given and (6.3) is not valid.*

The accuracy has been tested for two different time window sizes of $H_n = 10^{-3}\text{s}$ (**T1**) and $H_n = 10^{-6}\text{s}$ (**T2**). Then, co-simulation works sequentially along the time-axis and iteratively on H_n with $k = 5$ iteration steps per window.

T1: Table 7.1 shows that the lower bound (6.3) is fulfilled for each tested α_n . But, the estimation becomes less accurate for larger values of α_n . This can be explained as follows: the contraction factor α_n and the constant C increase for an decreasing resistance R_2 , that is, the coupling becomes weaker. Consequently, the impact of the second term (CH_n) of the diagonal element in (6.4) is getting more important. However, it can be observed that the upper bound estimate meet the convergence/divergence behavior of co-simulation (compare the first, second and last column).

Table 7.1: Accuracy of the lower bound estimation for different values of α_n

Exact α_n	Estimated α_n	abs. Error	window size H_n	co-simulation
0.0502	0.035	0.017	$H_n = 10^{-3}$ s	convergent
0.33	0.280	0.05	$H_n = 10^{-3}$ s	convergent
0.96	0.472	0.488	$H_n = 10^{-3}$ s	convergent
19	5.832	13.168	$H_n = 10^{-3}$ s	divergent

T2: Remember that the coupling for our test example (with R-splitting) becomes weaker for an increasing contraction factor α_n , since the constant C increases, too. Table 7.2 shows the simulation results for a reduced time window $H_n = 10^{-6}$ s, which yields much smaller errors for all tested α_n . These confirms our theoretical result that a smaller time window size H_n increases the accuracy of the lower bound estimator even for weakly coupled problems. The fact that the error of the estimator can be reduced by the size of the time window will be important in Chapter 7.4. Here, the lower bound is used to create a set of samples to measure the PDF of the contraction factor by using the KDE and the spectral approach of Chapter 6.

Table 7.2: Accuracy of the lower bound estimation for different values of α_n

Exact α_n	Estimated α_n	abs. Error	window size H_n	co-simulation
0.0502	0.049	0.001	$H_n = 10^{-6}$ s	convergent
0.33	0.32	0.01	$H_n = 10^{-6}$ s	convergent
0.96	0.89	0.07	$H_n = 10^{-6}$ s	convergent
19	18.1	0.1	$H_n = 10^{-6}$ s	divergent

Conclusions

We numerically verified the way to estimate the contraction factor online (during co-simulation) by using the lower bound estimate of Chapter 6.1. The accuracy of the estimate depends on whether the subsystems are strongly or weakly coupled as well as on the time window size H_n . Here, the choice of a suitable small time window is the only way to control the accuracy. Therefore, the lower bound estimate requires to split the simulation time into small time windows to estimate the contraction factor precisely even for weakly coupled problems. However, the requirement for small time windows is in contradiction with the aim of reducing the computation effort of co-simulation, where it is desirable to enlarge the time windows to proceed faster over the time-axis. To bypass this problem, one can define non-equidistant communication points with small time intervals $[T_n, T_{n+1}]$ only where necessary. Here, an effective time window size control algorithm in co-simulation is of great interest. Furthermore, we showed that the lower bound

is always fulfilled for our test example and convergence/divergence of co-simulation is always detected.

7.4 PDF estimation by using the KDE and Spectral Method

This chapter discusses the KDE and the spectral method introduced in Chapter 6. Both strategies are applied to co-simulation in order to estimate the probability density of the contraction factor. The simulation results are compared with a reference PDF (exact solution) determined as follows:

Following [17], the exact PDF is given by a mapping function $\phi(\mathbf{X})$, where \mathbf{X} is a random input vector with joint density $f_{\mathbf{X}}: \mathbb{R}^n \rightarrow \mathbb{R}_+$, see Def. 6. For univariate stochastic processes the exact PDF of the contraction factor $\alpha_n = \phi(\xi)$, where ξ is a random variable with density $f_{\xi}: [a, b] \rightarrow \mathbb{R}_+$, is given by the transformation formula for densities:

$$\hat{p}(\alpha_n) = f_{\xi}(\phi^{-1}(\alpha_n)) \left| \frac{d}{d\alpha_n} \phi^{-1}(\alpha_n) \right|, \quad \text{with } \phi: \mathbb{R} \rightarrow \mathbb{R}. \quad (7.2)$$

A generalized form for multivariate transformations is also given in [17].

Again we investigate the field/circuit co-simulation example already introduced in Chapter 7.2.1, see Fig. 7.3. The R-splitting approach of Chapter 3.2.1 is applied as coupling strategy. The parameter settings are made in the same way as in the previous Chapter 7.3. Only one parameter in the co-simulation model is assumed to be uncertain, where the coupling resistances R_1 or R_2 are considered to be uniformly distributed (for $i = 1, 2$):

$$R_i \sim \mathcal{U}(10 \Omega - \delta R_i, 10 \Omega + \delta R_i), \quad (7.3)$$

with variations of $\delta R_1 = 1\Omega$ and $\delta R_2 = 7\Omega$. Remember, these two resistances affect the convergence of the co-simulation, see Chapter 3.2.1. As a consequence, the iteration process becomes stochastic with an stochastic contraction factor. For our test example two different cases occur: we have to distinguish between the linear and non-linear case, since resistance R_1 is located in the nominator, whereas resistance R_2 is located in the denominator:

$$\text{(linear) case 1: } \frac{\langle R_1 \rangle}{R_2}, \quad \text{(non-linear) case 2: } \frac{R_1}{\langle R_2 \rangle}, \quad (7.4)$$

where $\langle \cdot \rangle$ denotes the uncertain resistance. Generally, the mapping function ϕ is not explicitly known in co-simulation and the density cannot be derived by (7.2). However, by using the R-splitting approach the mapping function ϕ is given by (7.4). Therefore, eq. (7.2) enables to calculate the exact PDF and allows for an comparison between the KDE (6.9) and the spectral method (6.15).

Now, the task is to estimate the PDF of the contraction factor as precise as possible from a set of samples. Our algorithm works in the following manner:

Step 1: For each sample point, the reference model (strongly coupled system) is solved in time-domain up to t_0 to obtain consistent initial values. Then, co-simulation is started on $[t_0, t_0 + H_n]$ with k iteration steps for each sample point $\mathbf{p}^{(i)} \in \mathcal{G}$ using its corresponding initial values $\mathbf{x}_0(\mathbf{p}^{(i)})$. The error in the solution $\mathbf{X}_n^k(t)$ is measured by Richardson Extrapolation, see Chapter 6.1. Therefore, we restart the computation for each sample point by using two steps with the half of the

window size $[t_0, t_0 + H_{n/2}] \rightarrow [H_{n/2}, H_{n/2} + H_{n/2}]$. The comparison of both solutions $\mathbf{X}_n^k(t)$ and $\mathbf{X}_{n/2}^k(t)$ gives an estimate for the error in $\mathbf{X}_n^k(t)$ by (6.8). Furthermore, constant extrapolation of the initial value is used for the initial guess $\tilde{\mathbf{X}}^{(0)}(t)$ on H_n and $H_{n/2}$. Consequently, the gain of accuracy per iteration within the dynamic iteration process caused by Richardson Extrapolation is $\mathcal{O}(H_n^2)$.

Step 2: Note that R-splitting fulfills the purely algebraic-to-algebraic coupling condition of Ass. 4, since no old differential unknowns occur and thus the lower bound (6.3) holds for our test circuit. Consequently, samples for α_n can be derived by (6.3) using H_n small enough, see Chapter 7.3. Now as a finale step, by using these finite set of samples we estimate the PDF by the KDE (6.9) and the spectral approach (6.15).

Simulation settings: Co-simulation is performed on the time interval $[0.01, 0.01 + H_n]$ s with time window size $H_n = 10^{-6}$ s. The window size was chosen (small enough) with respect to the numerical results of Chapter 7.3. We perform $k = 5$ iteration steps on H_n . The weakly coupled system as well as the strongly coupled system (serves as reference solution for comparison) is solved by using the same time stepping scheme `ode23s` (Rosenbrock method) with an accuracy of $\text{AbsTOL} = \text{RelTOL} = 10^{-4}$. For the spectral approach the Legendre polynomials are used as basis polynomials up to degree one ($p = 1$) for the linear case 1 and degree two ($p = 2$) for the non-linear case 2, see (7.4). Consequently, the number of polynomials in (4.31) is $P = 2$ for the linear and $P = 3$ for the non-linear case. The unknown coefficient functions of the gPC expansion are determined via the stochastic collocation method as introduced in Chapter 4.2.3. The probabilistic integrals are approximated by the Gauss-Legendre quadrature of Chapter 5 with various number of quadrature points. For the KDE method the Gaussian kernel is used as single distribution for each sample, see Chapter 6.2. Here, the parameter space is sampled within the predetermined scope of uncertainty by using equidistant grid points.

Numerical Results

Figure 7.11 shows the estimated PDF by using the spectral and the KDE approach for a various number of samples. The exact PDF is obtained by the transformation formula for densities (7.2). For the linear test case 1 in (7.4), the spectral method becomes more accurate. In fact, due to the linear mapping the spectral method is able to reconstruct the stochastic process precisely when at least linear polynomials are used within the gPC expansion (6.13), see Fig. 7.11 (right). Here, the KDE approach requires a large number of samples to reconstruct uniform distributions. For test case 2 in (7.4), the mapping function ϕ becomes non-linear. Figure 7.11 (left) shows the simulation results. The spectral method is too coarse as long as polynomials up to second order are used as polynomial basis for the approximation in (6.13). However, using polynomials of higher order reduces the error. Here, for a large number of model simulations of (6.1) the KDE approach becomes close to the exact solution and provides much better results. However, it should be remembered that the spectral approach only required to determine the unknown coefficients f_0, f_1 and f_3 belonging to the polynomial of zero, first and second order, which required to solve the model (6.1) three times. This is only a fraction of the computational cost of the KDE approach. However, it gives an idea of how the distribution looks like.

Now, the error in the estimated PDF (linear test case 1) caused by Richardson Extrapolation as well as by an variation of the time window size H_n is investigated. Therefore, Fig. 7.12 compares the accuracy of the PDF obtained by the KDE approach (by using an equidistant grid with 10^4 samples) and by the spectral approach (with maximum polynomial degree $p = 1$) with and without

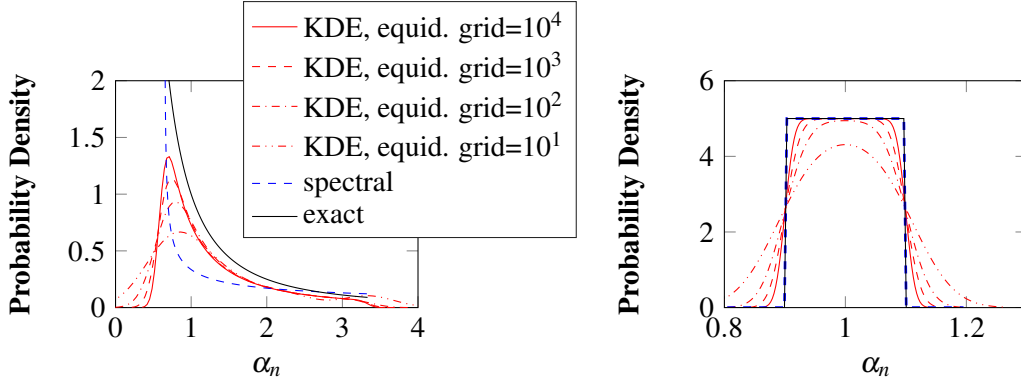


Figure 7.11: (right) Probability density function of the contraction factor α_n for resistance R_1 uncertain (linear mapping function ϕ) and (left) for resistance R_2 uncertain (non-linear mapping function ϕ) by using the spectral approach and the KDE approach for an increasing number of sample points. [37]

Richardson Extrapolation in dependence of the time window size H_n . The accuracy is measured in the expectation value. Naturally, the error of the lower bound estimate (6.3) increases with an increasing time window size, since for larger window sizes the reduction per iteration is governed by the additional term $C \cdot H_n$, see Fig. 7.12 (right).

Furthermore, for an increasing window sizes the inaccuracy of Richardson Extrapolation increases too, that is, Richardson Extrapolation induces an additional error. Here, using Richardson Extrapolation as error estimator, instead of the reference solution obtained by solving the strongly coupled problem (3.1), yields approximately the same scale of error as long as H_n is not too large. For our test example, the estimation provides errors of about 10^{-3} as long as we use time window sizes $H_n \leq 10^{-10}$ s. Notice that there is the possibility to use larger window sizes if one is not interested in such high accuracy of about 10^{-4} .

Conclusions

The way of estimating the contraction factor during the co-simulation with a lower bound was studied. The lower bound was used to calculate a set of sample points from which a probability density function can be reconstructed. Knowledge about the distribution may be help to detect divergence of co-simulation models with uncertain components. We demonstrated that the spectral and the KDE approach are suitable techniques for estimating the distribution of the contraction factor. Especially for the linear test case, the usage of basis polynomials up to the first order in the gPC expansion is sufficient to render the model behavior. Here, the spectral method becomes highly effective and provides exact results. However, for the non-linear test case both methods become computational expensive. Here, again the KDE approach requires a large number of sample points, whereas the spectral approach needs higher order polynomials to meet the exact PDF. Consequently, for the multivariate case (many uncertain parameters) both methods become inefficient with applications in co-simulation. That is, for the spectral method the computational effort to calculate the stochastic momenta of the gPC expansion increases in terms of (15), whereas the KDE method requires a large number of model simulations for a dense sampling of the random parameter space. Hence, the aim is to reduce the computational effort. Therefore, the calculation of the gPC based sensitivity indices, see Chapter 4.2.4, can be used for an efficient random

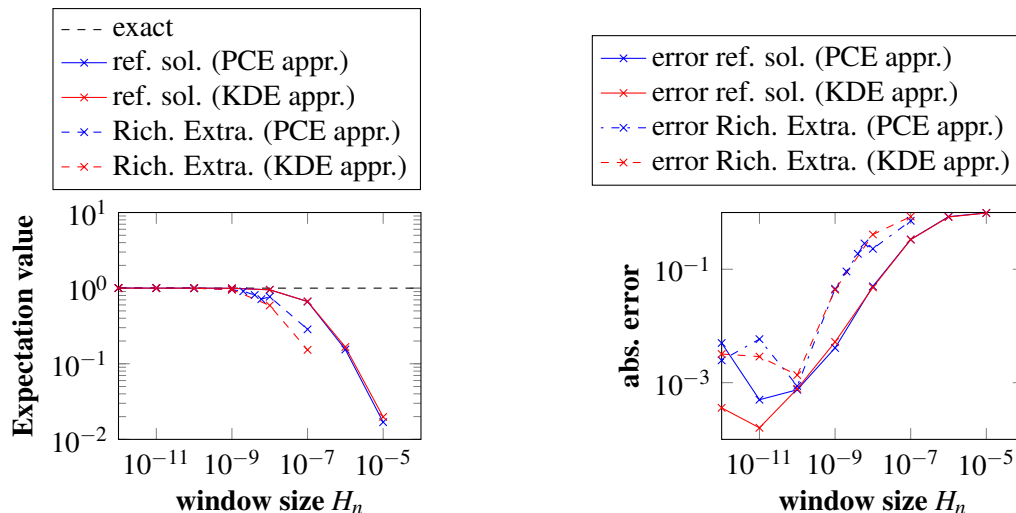


Figure 7.12: (left) Expectation value evaluated with the KDE (with 10^4 samples) and the spectral method (with maximum polynomial degree $p = 1$) with and without Richardson Extrapolation in dependence of the time window sizes H_n . (right) Error in the expectation value for different time window sizes H_n . [37]

parameter space sampling. This seems to be useful especially for the spectral approach, where the coefficient functions of the gPC expansion are already computed and the information about sensitivities can be easily extracted, see Chapter 4.2.4. Of course, the results of both methods are first and foremost dependent on the accuracy of the underlying sample points, i.e., the contraction factor estimate, from which a PDF should be reconstructed. Here, we numerically verified that Richardson Extrapolation is suitable to measure the splitting error in co-simulation. However, we showed that the accuracy of Richardson Extrapolation as well as of the lower bound estimate is strongly dependent on the time window size H_n . For our numerical test example, a window size $H_n \leq 10^{-10}$ s is required to achieve an error of approximately 10^{-3} .

7.5 Uncertainty Quantification in Co-Simulation for Coupled Electric Circuits

We consider a linear test circuit including uncertainties. The decoupling into subsystems is again managed via controlled sources and the splitting is organized such that no algebraic constraint depends on an old algebraic iterate (for the computational sequence of subs. 1 first). Thus, no contraction factor α_n occurs and consequently uncertainties does not destroy the convergence of co-simulation. However, it is of great interest to analyze the stochastic process of the expectation value as well as the variance and their propagation within a dynamic iteration procedure. To this end, again the non-intrusive gPC expansion of Chapter 4.2.2 is applied to analyze the stochastic behavior for our circuit.

Our engineering test circuit is a two-stage RLC lowpass filter given in Fig. 7.13 with uncertain components $\mathbf{p} = (R_1, R_2, C_1, C_2)^\top$, where we consider all components in \mathbf{p} of the i -th subsystem

to be independent and uniformly distributed random variables (for $i = 1, 2$):

$$C_i \sim \mathcal{U}(1\text{pF} - \delta C_i, 1\text{pF} + \delta C_i) \quad \text{and} \quad R_i \sim \mathcal{U}(10\text{k}\Omega - \delta R_i, 10\text{k}\Omega + \delta R_i), \quad (7.5)$$

where $\pm\delta C_i$, $\pm\delta R_i$ denote the variation from their respective nominal values. The variations will be held variable and specified different for the various numerical tests. Furthermore, we assume the inductance $L = 1\text{mH}$ and supply voltage $U_{\text{in}}(t) = 1\text{V} \cdot \cos(\omega t)$ with an angular frequency $\omega = 2\pi \cdot 5 \cdot 10^3 \text{s}^{-1}$. The split into subsystems is organized at node U_3 with corresponding coupling

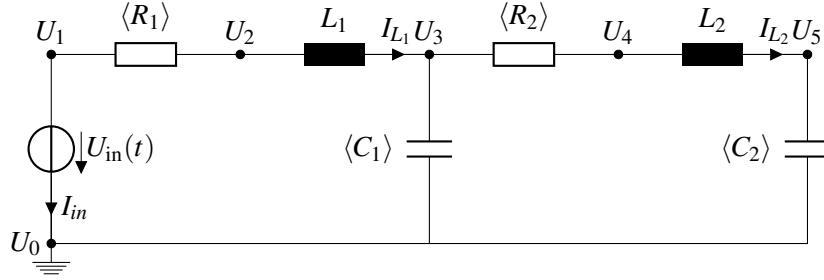


Figure 7.13: Two-stage RLC lowpass filter powered by a supply voltage source $U_{\text{in}}(t)$. The notation $\langle \cdot \rangle$ denotes the uncertain parameters. [38]

node U_{Co} and current I_{Co} . Thus the information transport between both systems is done by these two additional variables only. Figure 7.14 shows the resulting co-simulation model. Notice that for such a splitting, both subsystems can be described by the same (index-1) DAE (5.2) with solution $\mathbf{x} \in [t_0, t_e] \rightarrow \mathbb{R}^n$. The only difference is in the right-hand side $\mathbf{f}(t)$ (including all time dependent sources) of one of the subsystems, since the additional current source $I_{Co}(t)$ is included due to the source coupling approach.

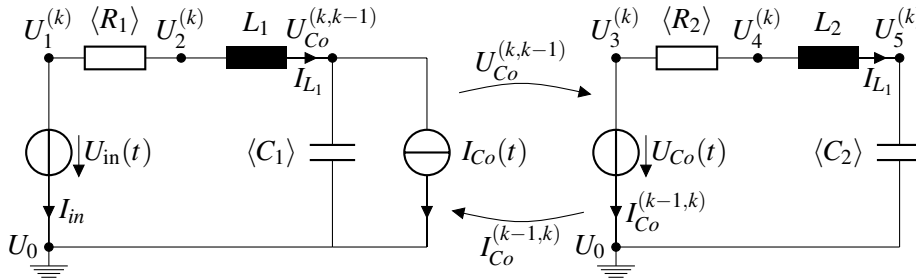


Figure 7.14: Decoupled two-stage RLC lowpass filter using the source-coupling with uncertain components R_1, R_2, C_1, C_2 . The first/second superscript notation $(k)/(k-1)$ denotes the old and new differential/algebraic iterates for subs. 1/subs. 2 first. [38]

Coupling Analysis: Their mutual dependence can be stated as follows: for the computational sequence of subs. 1 first, the algebraic iterate I_{Co} of subs. 2 enters the differential equation to further update subs. 1, since the capacitance C_1 ensures that the coupling node U_{Co} becomes differential. For the reverse computational sequence of subs. 2 first, the algebraic iterate U_{Co} enters the algebraic equation of subs. 2. Instantaneously the DAEs for the split circuits in semi-explicit form read:

$$\begin{aligned} \dot{\mathbf{y}}_1 &= \mathbf{f}_1(\mathbf{y}_1(\mathbf{p}), \mathbf{z}_1(\mathbf{p}), \mathbf{z}_2(\mathbf{p})), & \dot{\mathbf{y}}_2 &= \mathbf{f}_2(\mathbf{y}_2(\mathbf{p}), \mathbf{z}_2(\mathbf{p})), \\ 0 &= \mathbf{g}_1(\mathbf{y}_1(\mathbf{p}), \mathbf{z}_1(\mathbf{p})), & 0 &= \mathbf{g}_2(\mathbf{y}_1(\mathbf{p}), \mathbf{y}_2(\mathbf{p}), \mathbf{z}_2(\mathbf{p})), \end{aligned} \quad (7.6)$$

where the variables of the subsystems are allocated as follows:

$$\mathbf{y}_1 := [U_{Co}, I_{L_1}]^\top, \quad \mathbf{z}_1 := [U_1, U_2, I_{in}]^\top, \quad \mathbf{y}_2 := [U_5, I_{L_2}]^\top, \quad \mathbf{z}_2 := [U_3, U_4, I_{Co}]^\top.$$

Now, when starting the Gauß-Seidel-type dynamic iteration with subsystem 1, the corresponding splitting functions read (for subs. 1 first):

$$\mathbf{F} := \begin{bmatrix} \mathbf{f}_1(\mathbf{y}_1^{(k)}(\mathbf{p}), \mathbf{z}_1^{(k)}(\mathbf{p}), \boxed{\mathbf{z}_2^{(k-1)}(\mathbf{p})}) \\ \mathbf{f}_2(\mathbf{y}_2^{(k)}(\mathbf{p}), \mathbf{z}_2^{(k)}(\mathbf{p})) \end{bmatrix}, \quad \mathbf{G} := \begin{bmatrix} \mathbf{g}_1(\mathbf{y}_1^{(k)}(\mathbf{p}), \mathbf{z}_1^{(k)}(\mathbf{p})) \\ \mathbf{g}_2(\mathbf{y}_1^{(k)}(\mathbf{p}), \mathbf{y}_2^{(k)}(\mathbf{p}), \mathbf{z}_2^{(k)}(\mathbf{p})) \end{bmatrix}. \quad (7.7)$$

Thus, no algebraic constraint depends on old algebraic iterate. Therefore, the simple coupling condition of Cor. 6 (ii) is valid and co-simulation yields the minimum convergence rate $\mathcal{O}(\sqrt{H_n})$. Hence, introducing uncertainties does not manipulate the properties of convergence and the dynamic iteration is unconditionally stable for splitting (7.7) using time window sizes $H_n < H_{\max}$.

Simulation Settings: Co-simulation is studied on a single time window $[t_0, t_0 + H_n]$ with $t_0 = 0.4\text{ms}$. Consequently, no error propagation to subsequent windows occurs and such only the splitting errors of a dynamic iteration appear. The non-intrusive gPC expansion with polynomials up to a maximum polynomial of degree three is used, which enables to detect momenta up to the third order, i.e., interactions up to three parameters can be identified. The gPC coefficient functions (4.36) are computed by the stochastic collocation method of Chapter 4.2.3. Here, the Gauss-Legendre quadrature of order five based on a tensor-product grid $\mathcal{G} = \{\mathbf{p}^{(1)}, \dots, \mathbf{p}^{(m)}\}$ is used, see Chapter 5, which requires to solve the model 81 times. We choose $\text{AbsTOL} = \text{RelTOL} = 10^{-4}$ for the respective time-integrator accuracy.

Our algorithm works in the following way: For each sample-point $\mathbf{p}^{(i)} \in \mathcal{G}$, the reference model (strongly coupled system) is solved in time domain up to t_0 to obtain consistent initial values $\mathbf{x}_0(\mathbf{p}^{(i)})$. Then, constant extrapolation of the initial value is used as initial guess on time window H_n . Now, co-simulation is performed on $[t_0, t_0 + H_n]$ with k iteration steps for each sample $\mathbf{p}^{(i)}$. The output for the different realizations of the parameters is a discrete solution $\tilde{\mathbf{x}}(t_n, \mathbf{p}^{(i)})^{(k)}$. Based on these set of simulation results, we compute the stochastic momenta (depending on step k) by evaluate the multiple integral (4.36) in each microstep $t_0 < t_1 < \dots < t_n < t_{n+1} < \dots < t_e$. Therefore, the expectation value and variance (4.39) as well as sensitivity indices (4.41) can be computed for all discrete time points t_n . Finally, interpolation over the entire timeline yields the (overall) solution $\tilde{\mathbf{x}}(\mathbf{p}^{(i)})^{(k)}$.

Numerical Results

Sensitivity analysis: For the parameter variations in (7.5) we choose $\delta R_1 = \delta R_2 = 1\text{k}\Omega$ (10%) for the resistances and $\delta C_1 = \delta C_2 = 0.5\text{pF}$ (50%) for the capacitances. Figure 7.15 shows the total-order sensitivity indices for the reference model (without co-simulation) in the time interval $[0, 0.5]\text{ms}$. The result serves as the reference solution for comparison. Now, first we compute total-order gPC sensitivity indices (4.43) for the different uncertain parameters in a co-simulation environment. Therefore, a sensitivity analysis is performed on the single time window $H_n = 0.1\text{ms}$ for $k = 1, \dots, 10$ iteration steps with respect to the (output) node U_5 . Figure 7.16 shows the total-order sensitivity indices $SgPC^T$ for R_1 , R_2 , C_1 and C_2 computed on the specified single time window H_n . It shows that the sensitivity index for the capacitance C_2 yields the largest value. Note that the capacitance C_2 is directly connected to the output U_5 . The result is in line with the general observation in UQ, that components which are located close to the output have a higher

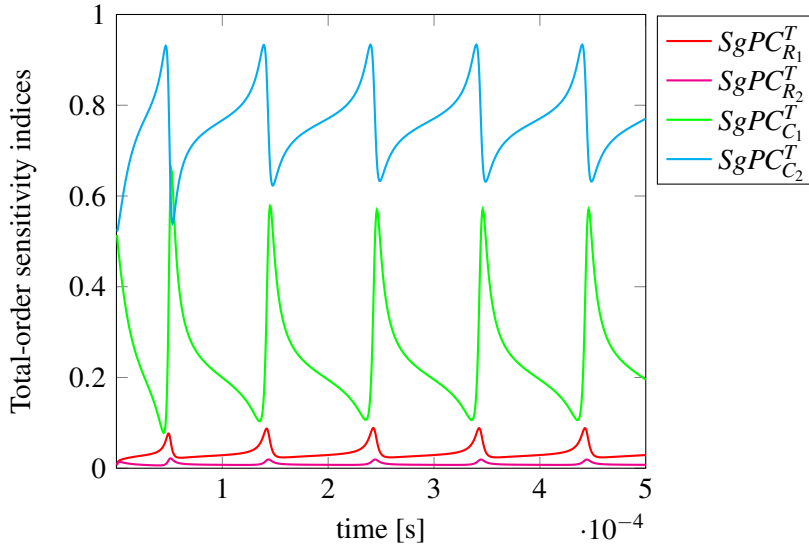


Figure 7.15: Total-order sensitivities indices for R_1, R_2, C_1 and C_2 in $[0, 0.5]$ ms computed by solving the strongly coupled model.

impact than components which are topological located further away. Furthermore, the example shows that the number of iteration steps which are required to be in the same scale of sensitivities as for the reference model differs from each other. That is, the sensitivities related to C_2 requires the most number of iteration steps. However, one can see that order of magnitude of sensitivities can already assessed after few iteration steps.

As a second step the accuracy of the gPC based sensitivity indices obtained within a co-simulation framework is investigated for a set of different variations $\delta R_i, \delta C_i$. To this end, we suppose uniform distributed variables with the following variation for the resistances and the capacitances around the nominal respective values in (7.5):

$$\delta R_i = 1\text{k}\Omega \dots 5\text{k}\Omega \text{ (10\% - 50\%)}, \quad \delta C_i = 0.1\text{pF} \dots 0.5\text{pF} \text{ (10\% - 50\%)}. \quad (7.8)$$

The error in the solution on the n -th time window after k iteration steps $\tilde{x}^{(k)}(t)$ is measured by comparing with the reference solution $x_m(t)$ computed by a monolithic simulation:

$$\Delta_n^{(k)}(t) = x_m(t) - \tilde{x}^{(k)}(t), \quad \delta_n^{(k)} := \|\Delta_n^{(k)}\|_\infty.$$

We focus on the error in the gPC based sensitivity indices. The final error which corresponds to a given variation (7.8) involves all sensitivities, since it is measured by averaging.

Figure 7.17 shows the average error in the total-order sensitivity indices for $k = 1, 3, 5, 10$ iteration steps in co-simulation. One can see that the error increases for an increasing level of variation (uncertainty). This is due to the fact that the number of model simulations is fixed whereas the variation increases. Consequently, the predetermined number of 81 samples is no longer sufficient for a dense sampling of the parameter space. Furthermore, a continuous improvement up to $k = 10$ iteration steps can be observed in cases of high variations (uncertainties) for C_i and R_i . Accordingly, slight variations in the co-simulation model requires less iteration steps, whereas the level of variations in the capacitances mainly control the error size.

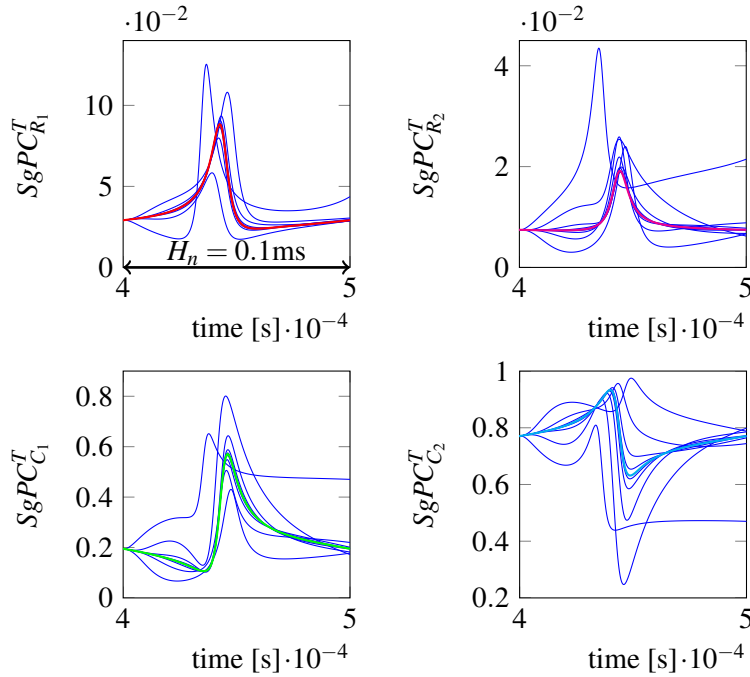


Figure 7.16: Total-order sensitivity indices (solid blue lines) on the time interval $[0.4\text{ms}, 0.4\text{ms} + H_n]\text{ms}$ with $H_n = 0.1\text{ms}$ (single time window) for the parameters R_1 (top, left), R_2 (top, right), C_1 (bottom, left) and C_2 (bottom, right) for $k = 1, \dots, 10$ iteration steps: $R_i \sim \mathcal{U}(9\text{k}\Omega, 11\text{k}\Omega)$, $C_i \sim \mathcal{U}(0.5\text{pF}, 1.5\text{pF})$. The red, magenta, green and cyan solid lines denote the total sensitivity indices computed for the strongly coupled model (without co-simulation).

Convergence and Contraction: Next, we investigate convergence and contraction of the expectation value and standard deviation for splitting (7.7). The results are compared with the monolithic, deterministic solution. Therefore, all nodes U_1, \dots, U_5 are involved, which requires a sensitivity analysis for every single node. However, the number of model simulations remains the same but the number of probabilistic integrals to determine the unknown coefficient functions is five times larger. Based on the coefficient functions, we finally compute expectation value and standard deviation for each node (output).

Again, we suppose variations of $\delta R_1 = \delta R_2 = 1\text{k}\Omega$ (10%) for the resistances and $\delta C_1 = \delta C_2 = 0.5\text{pF}$ (50%) for the capacitances. Figure 7.18 shows the results. All quantities have almost the same rate of convergence of about $\mathcal{O}(H_n)$ for window sizes $10^{-8}\text{s} \leq H_n \leq 10^{-4}\text{s}$, whereas a further reduction of the window size does not further reduce the error in the expectation value and standard deviation, see Fig. 7.18 (left). This behavior differs to the deterministic solution, where an improvement up to the machine precision can be observed. The reasons are different: the accumulation of errors limits the accuracy. The Gauss-Legendre quadrature of order five produces a numerical quadrature error (5.10) in each coefficient function $f_j(t)$ of (4.35). Additionally, the representation of the stochastic process by the gPC expansion is only an approximation, since only a finite number of terms are involved. Consequently, a truncation error (4.40) occurs.

Figure 7.18 (right) shows the splitting error in the expectation value and the standard deviation on $H_n = 10^{-4}\text{s}$ for different iteration steps. It shows that the expectation value requires much less iterations than the standard deviation. Here, the expectation value is already well approximated

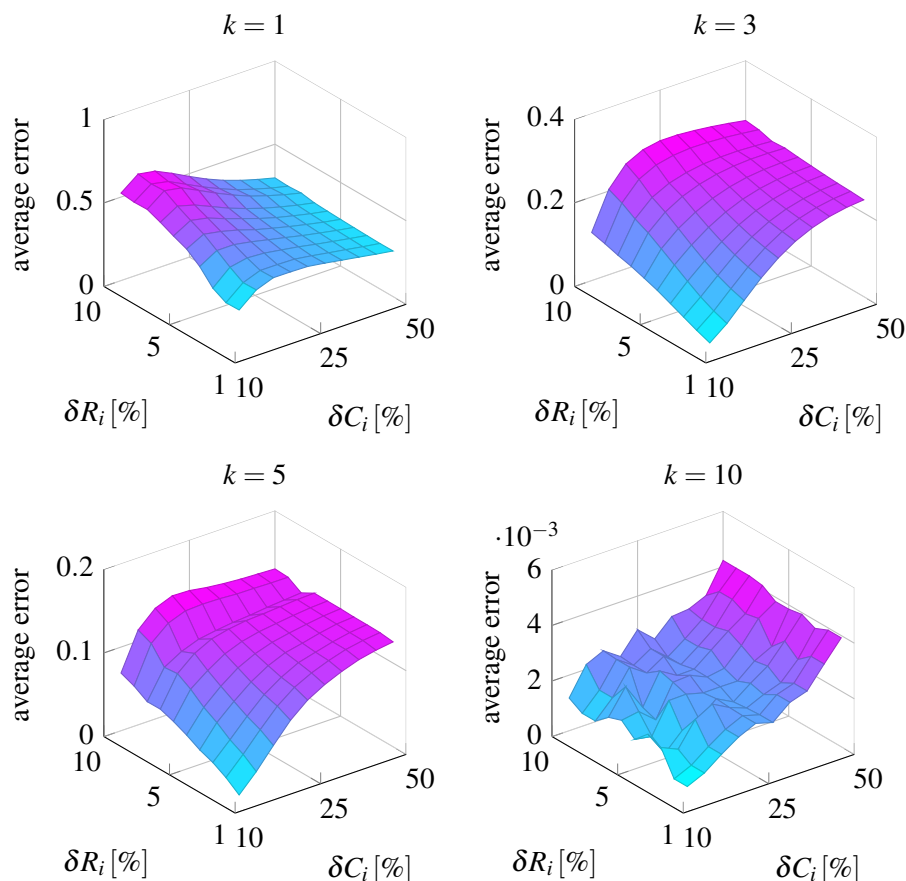


Figure 7.17: Average error over all total-order sensitivity indices $SgPC^T$ for $k = 1, 3, 5, 10$ iteration steps and uniformly distributed random variables: $R_i \sim \mathcal{U}(10\text{k}\Omega - \delta R_i, 10\text{k}\Omega + \delta R_i)$, $C_i \sim \mathcal{U}(1\text{pF} - \delta C_i, 1\text{pF} + \delta C_i)$. [38]

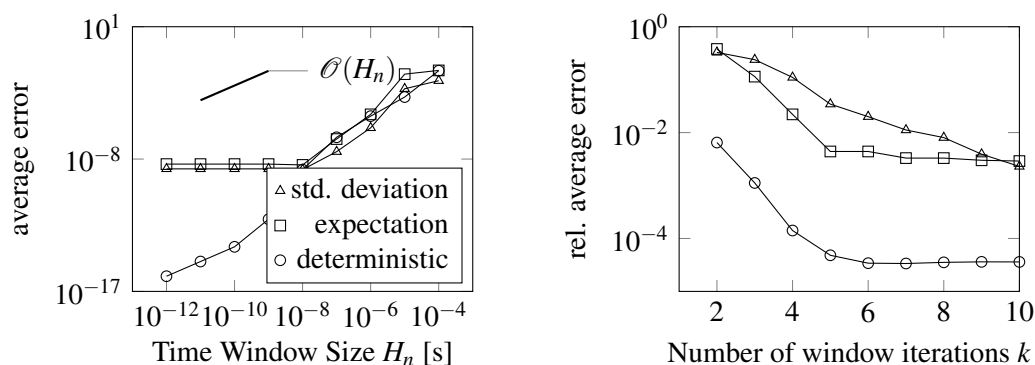


Figure 7.18: (left) Convergence in the expectation value, standard deviation and in the deterministic solution measured in the node potentials U_1, U_2, U_3, U_4, U_5 in dependence of the time window sizes H_n with $k = 4$ iteration steps. (right) Splitting error in the expectation value, standard deviation and in the deterministic solution on the time window $H_n = 10^{-4}$ s in dependence of the number of iterations. [38]

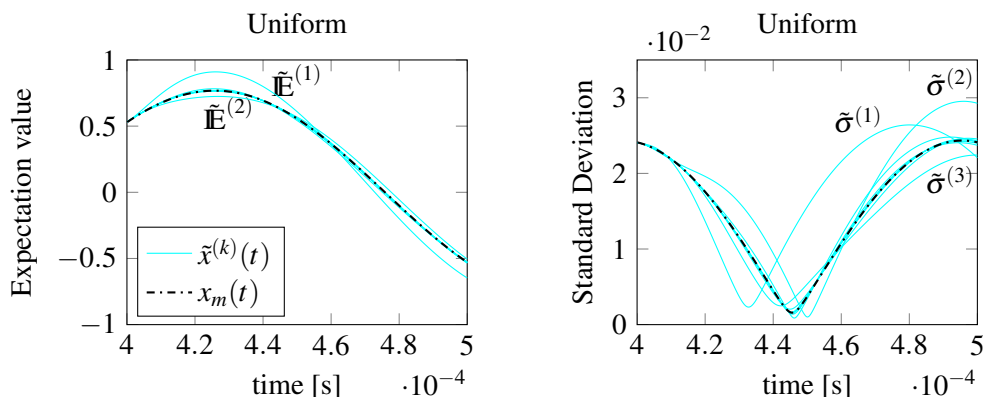


Figure 7.19: Expectation value and standard deviation for the output node U_5 for $k = 1, \dots, 10$ iteration steps on a single time window $H_n = 10^{-4}$ s. [38]

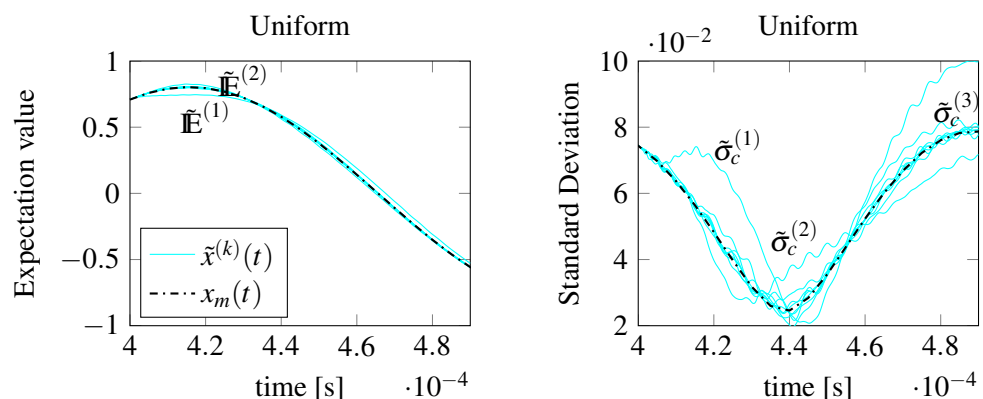


Figure 7.20: Expectation value and standard deviation for the coupling node U_3 for $k = 1, \dots, 10$ iteration steps on a single time window $H_n = 10^{-4}$ s. [38]

after four iteration steps, whereas the standard deviation requires about ten iteration steps to be in the same scale of error. This is because the expectation value is exactly represented by the first coefficient function f_0 of the gPC expansion, see (4.39), and thus no truncation error (4.40) occurs. However, for the standard deviation this is different, since it is obtained by the square root of the variance given by the finite sum over all remaining coefficient functions, see (4.39). Consequently, the truncation error as well as the accumulation of quadrature errors and rounding errors require further iteration steps in co-simulation. Figure 7.19 and Fig. 7.20 show explicitly the waveforms of the expectation value and the standard deviation on $H_n = 10^{-4}$ s for different iteration steps for the coupling nodes U_3 and for the output U_5 .

Conclusions

It has been shown for our test circuit, that the number of iteration steps to achieve a predefined accuracy is mainly controlled by the level of uncertainty. Co-simulation with components including high uncertainties naturally requires a large number of iterations within the dynamic iteration. Regarding the convergence rate of stochastic processes, uncertainties in time-dependent compo-

nents (capacitances and inductances) have a greater impact than static components (resistances). Concerning our test example, the speed of contraction for the expectation value and standard deviation differs from each other, that is different number of iteration steps are required to achieve a suitable accuracy.

8

Chapter 8

Summary

Instead of single-rate time-integration of field devices and circuits, the co-simulation is discussed within this thesis. The standard way of co-simulation for field/circuit coupled problems is the separate treatment of field devices and electric circuits. Here, the Gauß-Seidel-type dynamic iteration is unconditionally stable and convergent for the computational sequence of voltage driven field devices first. However, our numerical investigations show that the coupling strength becomes weak and dynamic iteration may require a lot of iteration steps to attain a certain time-integration accuracy. This is reflected in a high computational effort, since all subsystems have to be solved frequently in time, where the repeated computation of the field problem is the most computational expensive part. This thesis deals with different strategies to couple field devices with circuits in an more efficient way. The new coupling approaches were mathematically investigated, analyzed and simulated with their application in electrical engineering.

It has been shown by numerical simulations that inrush currents generated from field devices, e.g. transformers or electrical machines, during the switching-on phase suffer from co-simulation for the standard decoupling approach (cutting at the field device boundaries). Here the field subsystem responds with a much higher inrush current peak compared to the (exact) computed inrush current when single-rate time-integration is applied. This is because the information about the circuit surrogate resistance, which limits the current through the field device, is missing when the field subsystem is computed iteratively. This becomes crucial in co-simulation, since the information transport from the field to the circuit is managed by a controlled current source excited by the computed inrush current obtained from the previous iteration. Consequently, the number of iteration steps (the computational effort) of co-simulation can be reduced by controlling the inrush current splitting error. Therefore, we proposed the R-splitting and LR-coupling techniques as two alternatives to the standard coupling approach. The basic concept for both is roughly the same: relocate parts of the primary side surrogate resistance from the circuit to the field subsystem such that the field devices are not able to respond with their maximum inrush current (only limited by the internal resistances of the coil windings). In other words: at the beginning of dynamic iteration the field device subsystem should have knowledge about network components that reduce the maximum inrush current peak. From the co-simulation point of view, the number of repeated model simulations is strongly related to the strength of coupling between the subsystems. Naturally, the stronger the coupling, the larger the communication points (time window size) can be set for convergence. It has been proved mathematically that the coupling strength benefits from both splittings, which we numerically verified for a transformer-circuit engineering test example. However, using large time window sizes typically increase the number iteration steps. Using the same time window size allows us to compare the splitting error for the different interfaces with respect to the number of iteration steps. Here, field/circuit co-simulation using the R-splitting and LR-coupling requires much less iterations to attain the same accuracy as the standard approach (cutting at the field device boundaries). Consequently, both interfaces enable the user to reduce the computational effort. In terms of solved linear systems of equations the amount of effort with R-splitting has been reduced by 90%. However, at least for our numerical test example, the error reduction per iteration of co-simulation with R-splitting is much better than for co-simulation with

LR-coupling. Furthermore, by the extended recursion estimate we proved that co-simulation with R-splitting is unconditionally stable and convergent independently of the computational sequence of the subsystems (field or circuit first), i.e., the contraction factor vanishes. The principles were analyzed within a more general framework for coupled (index-1) DAEs such that the (extended) splitting scheme also holds for further applications, e.g. semiconductor/circuit problems, see [24].

Furthermore, we aimed to compute statistics for co-simulation models with uncertain behaviors, i.e., with uncertain parameters. Here, we focused on estimate the probability density function for the contraction factor online to detect divergence during the co-simulation procedure. This enables for a combined window size and sweep control to further improve the efficiency. The KDE and the spectral method were tested as two different concepts for that purpose. Numerical investigations show that both schemes are able to reconstruct density functions from a finite number of samples. For uniformly distributed random processes with linear impact, the spectral method enables to compute the density function exactly and is much more efficient than the KDE approach. However, for random processes with a nonlinear impact, both schemes become computational expensive, since they require a dense sampling pattern of the random parameter space. This becomes difficult in co-simulation for models, whose subsystems are expensive to compute, e.g. field devices.

Outlook

New interesting problems and questions arise: the possibility of estimate the density function of the contraction factor enables to assess the divergence probability of a co-simulation procedure. The measurement of densities becomes difficult for a field/circuit co-simulation, since the treatment of the field subproblem is computational expensive. Here, sensitivity indices should be exploited for an effective parameter space sampling to reduce the number of repeated simulations. Furthermore, the implementation and numerical validation of a macrostep size control within a dynamic iteration process using density functions is obvious and of great interest.

Within this thesis, the LR and R splitting are proposed and analyzed as an extension of source coupling. However, [24] proposes also the parameter coupling which includes still sources for the information transport between the subsystems and additional lumped elements, where it is numerically shown that it provides better convergence and contraction. Thus, our new splittings should be also analyzed within the parameter coupling. Furthermore, the question arises how useful the new splitting techniques are for circuits which are coupled to field devices sharing more than one coupling node (more than one coupling variables are involved). Here, a co-simulation of a circuit coupled to a permanent-magnet synchronous engine seems to be appropriate for this task.

Bibliography

- [1] Li, J. and Marzouk, Y.: Multivariate semi-parametric density estimation using polynomial chaos expansion. In preparation. (2013)
- [2] Arnold, M. and Günther, M.: Preconditioned Dynamic Iteration for Coupled Differential-Algebraic Systems. In: BIT, vol. 41, pp. 1–25. (2001)
- [3] Bartel, A., Brunk, M., Günther, M., and Schöps, S.: Dynamic Iteration for Coupled Problems of Electric Circuits and Distributed Devices. In: SIAM J. Sci. Comput., vol. 35, No. 2, pp. 315–335. (2013)
- [4] Burrage, K.: Parallel Methods for Systems of Ordinary Differential Equations. In: Clarendon Press, Oxford. (1995)
- [5] Verhoeven, A., Tasić, B., Beelen, T. G. J., ter Maten, J., and Mattheij, R. M. M.: Bdf compound-fast multirate transient analysis with adaptive step size control. In: Journal of Numerical Analysis, Industrial and Applied Mathematics, vol. 3, No. 3-4, pp. 275-297. (2008)
- [6] Meeker, D.: Finite Element Method Magnetics - User's Manual. Version 4.2. <http://www.femm.info/Archives/doc/manual42.pdf>
- [7] Schöps, S.: Field Device Simulator: <http://www.schoeps.org/home>
- [8] Schöps, S., De Gersem, H., and Bartel, A.: A Cosimulation Framework for Multirate Time Integration of Field/Circuit Coupled Problems. In: IEEE Transactions on Magnetics, vol. 46, No. 8. (2010)
- [9] Xiu, D.: Numerical Methods for Stochastic Computations – A Spectral Method Approach. In: Princeton University Press. (2010)
- [10] Bartel, A. and Günther, M.: A multirate w-method for electrical networks in state-space formulation. In: Journal of Computation and Applied Mathematics, vol. 147, pp. 411 – 425. (2002)
- [11] Günther, M., Kvaerno, A., and Rentrop, P.: Multirate partitioned runge-kutta methods. In: BIT, vol. 41, pp. 504 – 514. (2001)
- [12] Günther, M. and Rentrop, P.: Multirate row methods and latency of electric circuits. In: Applied Numerical Mathematics, vol. 13, pp. 83 – 102. (1993)
- [13] Scott, A. W.: Wiley Interdisciplinary Reviews. Computational Statistics. In: John Wiley & Sons, Inc. (2010)
- [14] Scott, D. W.: Multivariate Density Estimation. Theory, Practice, and Visualization. Second Edition. In: John Wiley & Sons, Inc, Hoboken, NJ. (2015)

-
- [15] Gerstner, T. and Griebel, M.: Numerical Integration using Sparse Grids. In: Num. Algorithms, vol. 18, pp. 209–232. (1998)
- [16] Kornatka, M.: The wighted kernel density estimation methods for analysing reliability of electricity supply. In: 17th International Scientific Conference on Electric Power Engineering (EPE), pp. 1-4. (2016)
- [17] Graybill, F. A., McFarlane Mood, A., and Boes, D. C.: Introduction to the Theory of Statistics. In: McGraw-Hill Education - Europe. International 3rd Revised ed.. (1974)
- [18] Wiener, N.: The Homogeneous Chaos. In: American Journal of Mathematics, vol. 60, No. 4. (1938)
- [19] Cameron, R. and Martin, W. T.: The orthogonal development of nonlinear functionals in series of Fourier-Hermite functionals. In: Ann. Math., vol. 48, No. 2, pp. 385 – 392. (1947)
- [20] Bartel, A.: Partial Differential-Algebraic Models in Chip Design - Thermal and Semiconductor Problems. PhD thesis. In: VDI Verlag, Fortschritt-Berichte VDI, Reihe 20, Düsseldorf. (2004)
- [21] Estévez Schwarz, D. and Tischendorf C.: Structural analysis of electric circuits and consequences for MNA. In: International Journal of Circuit Theory and Applications, vol. 28, No. 2, pp. 131–162. (2000)
- [22] Feldmann, U. and Günther, M.: CAD-based electric-circuit modeling in industry. Part I: mathematical structure and index of network equations. In: Surveys on Mathematics for Industry, pp. 97–129. (1999)
- [23] Grasser, K.: Mixed-Mode Device Simulation. PhD thesis, TU Wien. (1999).
- [24] Schöps, S.: Multiscale Modeling and Multirate Time-Integration of Field/Circuit Coupled Problems. PhD thesis. Bergischen Universität Wuppertal, Katholische Universität Leuven. (2011)
- [25] Maxwell, J. C.: A Dynamical Theory of the Electromagnetic Field. In: Royal Society Transactions CLV, pp. 459–512. (1864)
- [26] Haus, H. A. and Melcher, J. R.: Electromagnetic Fields and Energy. In: Prentice Hall. (1989)
- [27] Brenan, K. E., Campbell, S. L., and Petzold, L. R.: Numerical solution of initial-value problems in differential-algebraic equations. In: SIAM. (1995)
- [28] Hairer, E., Nørsett, S. P., and Wanner, G.: Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems. In: 2nd ed. Springer Series in Computational Mathematics. Berlin: Springer. (2002)
- [29] Tsukerman, I. A.: Finite Element Differential-Algebraic Systems for Eddy Current Problems. In: Numerical Algorithms, vol. 31, No. 1, pp. 319–335. (2002)
- [30] Schöps, S., Bartel, A., De Gersem, H., and Günther, M.: DAE-Index and Convergence Analysis of Lumped Electric Circuits Refined by 3-D MQS Conductor Models. In: Scientific Computing in Electrical Engineering (SCEE). Ed. by Janne Roos and Luis R. J. Costa.

- Mathematics in Industry 14. Springer, pp. 341–350. (2010)
- [31] Günther, M., Feldmann, U., and ter Maten, J. In: Numerical Methods in Electromagnetics. Ed. by Wil Schilders, Philippe Ciarlet, and Jan ter Maten. vol. 13. Handbook of Numerical Analysis. North-Holland. Chap. Modelling and Discretization of Circuit Problems. (2005)
- [32] Bartel, A., Baumanns, S., and Schöps, S.: Structural analysis of electrical circuits including magnetoquasistatic devices. In: Applied Numerical Mathematics. vol. 61, No. 12, pp. 1257–1270. (2011)
- [33] Quarteroni, A. and Valli, A.: Numerical Approximation of Partial Differential Equations. In: Springer Series in Computational Mathematics. Berlin: vol. 23. Springer. (2008)
- [34] Booton, R. C.: Computational methods for electromagnetics and microwaves. In: John Wiley & Sons, New York. (1992)
- [35] Binns, K. J., Lawrenson, P. J., and Trowbridge, C. W.: The analytical and numerical solution of electric and magnetic fields. In: John Wiley & Sons, Chichester. (1992)
- [36] Gausling, K. and Bartel, A.: Coupling Interfaces and their Impact in Field/Circuit Co-Simulation. In: IEEE Transactions on Magnetics, vol. 52, No. 3. (2016)
- [37] Gausling, K. and Bartel, A.: Density Estimation Techniques in Cosimulation using Spectral- and Kernel Methods. In: Scientific Computing in Electrical Engineering (SCEE). To be published. (2016)
- [38] Gausling, K. and Bartel, A.: Uncertainty Quantification in Co-Simulation for Coupled Electrical Circuits. In: Scientific Computing in Electrical Engineering (SCEE), vol. 23, pp. 243–251. (2014)
- [39] Gausling, K. and Bartel, A.: Analysis of the Contraction Condition in the Co-Simulation of a Specific Electric Circuit. In: Progress in Industrial Mathematics at ECMI 2014. Mathematics in Industry, vol. 22., pp. 853–860. (2014)
- [40] Brunner, A.: Polynomielles Chaos für zufällige Volatilität und zufälligem Zins in der Optionsbewertung. Diplomarbeit. In: Universität Bayreuth. (2010)
- [41] Sobol, I. M.: Sensitivity Estimates for Nonlinear Mathematical Models. In: Matematicheskoe Modelirovanie, vol. 2, pp. 112–118. (1990). translated in English: Sensitivity Estimates for Nonlinear Mathematical Models. In: Mathematical Modeling & Computational Experiment, vol. 1, pp. 407–414. (1993)
- [42] Cacuci, D. G.: Sensitivity and Uncertainty Analysis. Theory. I. In: Chapman & Hall
- [43] Cacuci, D. G., Ionescu-Bujor, M., and Navon, M.: Sensitivity and Uncertainty Analysis. Application to Large-Scale Systems. II. In: Chapman & Hall. (2005)
- [44] Schwieger, V.: Nicht-lineare Sensitivitätsanalyse gezeigt an Beispielen zu bewegten Objekten. Habilitation. Fakultät für Bau- und Umweltingenieurwissenschaften, Universität Stuttgart. (2005)
- [45] Saltelli, A., Ratto, M., Andres, T., Campolongo, F., Cariboni, J., Gatelli, D., Saisana, M., and Tarantola S.: Global Sensitivity Analysis. The Primer. Wiley. (2008)

- [46] Saltelli, A.: Making best use of model evaluations to compute sensitivity indices. In: *Computer Physics Communication*, vol. 145, No. 2, pp. 280–293. (2002)
- [47] Bentele, M., Lavrik, I., and Ulrich, M.: Mathematical Modeling reveals threshold mechanism in CD95-induced apoptosis. In: *J. Cell Biol.*, vol. 166, No. 6, pp. 839–851. (2004)
- [48] Cukier, R. I., Fortuin, C. M., Shuler, K. E., Petschek, A. G., and Schaibly, J. H.: Study of the sensitivity of coupled reaction systems to uncertainties in rate coefficients. In: *Journal of Chemical Physics*, vol. 59, pp. 3873–3878. (1973)
- [49] Bullard, C. W. and Sebald, A. V.: Monte Carlo Sensitivity Analysis of Input-Output Models. In: *The Review of Economics and Statistics*, vol. 70, No. 4, pp. 708–712. (1988)
- [50] Augustin, F., Gilg, A., Paffrath, M., Rentrop, P., and Wever, U.: Polynomial Chaos for the approximation of uncertainties. In: *Euro. Jnl. of Applied Mathematics*, vol. 19, pp. 149–190. (2008)
- [51] Pulch, R., ter Maten, J., and Augustin, F.: Sensitivity analysis and model order reduction for random linear dynamical systems. In: *Mathematics and Computers in Simulation*, vol. 111, pp. 80–95. (2015)
- [52] Schöps, S., De Gerssem, H., Bartel, A.: Higher-Order Cosimulation of Field/Circuit Coupled Problems In: *IEEE Transactions on Magnetics*, vol. 48, No. 2, pp. 535–538. (2012)
- [53] Abramowitz, M. and Stegun, I. A., eds. (1983) [June 1964]. "Chapter 25.4, Integration". *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*. Applied Mathematics Series. 55 (Ninth reprint with additional corrections of tenth original printing with corrections (December 1972); first ed.).
- [54] Stoer, J. and Bulirsch, R.: *Introduction to Numerical Analysis* (3rd ed.), Springer. (2002)
- [55] Cools, R.: Advances in multidimensional integration. In: *J. Comput. Appl. Math.*, vol. 149, pp. 1–12. (2002)
- [56] Wasilkowski, G. W. and Wozniakowski, H.: Explicit cost bounds of algorithms for multivariate tensor product problems. In: *J. Complexity*, vol. 11, pp. 1–56. (1995)
- [57] Smolyak, S. A.: Quadrature and interpolation for tensor products of certain classes of functions. In: *Soviet Math. Dokl.*, vol. 4, pp. 240–243. (1963)
- [58] Novak, E. and Ritter, K.: Simple cubature formulas with high polynomial exactness. In: *Constructive Approximation*, vol. 15, pp. 499–522. (1999)
- [59] Bellman, R.: *Dynamic Programming*. In: University Press, Princeton. (1957)
- [60] Gauss, C.: *Methodus nova integralium valores per approximationem inveniendi*. In : Cambridge University Press, Cambridge Library Collection - Mathematics, pp. 165–196.
- [61] Richter, T. and Wick, T.: *Einführung in die Numerische Mathematik*. Skript. <http://www.numerik.uni-hd.de/~richter/pdf-files/numerik0.pdf>. (2012)
- [62] Yacamini, R. and Abu-Nasser, A.: The calculation of inrush current in three-phase transformers. In: *IEE Proceedings B - Electric Power Applications*, vol. 133, No. 1, pp. 31–40. (1986)