An aggregation model reduction method for one-dimensional distributed systems.
PhD Thesis

Andreas Linhart

Norwegian University of Science and Technology
Faculty of Science and Technology
Department of Chemical Engineering

September 8, 2009

Abstract

This thesis discusses a model reduction method for one-dimensional distributed systems. These systems can be spatially discretely distributed, like staged distillation columns, or spatially continuously distributed, like packed distillation columns, tubular reactors etc. The method takes its starting point in the aggregated modeling method of Lévine and Rouchon (1991) for simple staged distillation column models. The derivation of the method is substantially simplified, making it possible to apply the method to more complex models that include mass and energy balances. The basic principle of the method is to divide the system into intervals of steady-state systems, which are connected by dynamic aggregation elements. In discrete systems, the aggregation elements are some selected units, and the remaining units form the steady-state systems. In continuous systems, the aggregation elements are dynamic equations connecting the steady-state systems described by boundary value problems.

In order to obtain models of lower complexity than the original models, the steady-state systems have to be eliminated from the reduced models. This can be achieved by solving them either numerically or analytically as functions of the independent variables of the aggregation elements. The resulting functions can be substituted into the dynamic equations, yielding reduced models that retain the original structure of the original model, but have a much lower number of states.

As a case study, a complex distillation column including mass and energy balances and complex hydraulic and thermodynamic relationships is investigated. The steady-state equations in the reduced model are replaced by tabulated functions that are implemented using linear multi-dimensional interpolation. The reduced model is compared with a fast implementation of the original model, and is shown to be several times faster at comparable accuracy.

For spatially continuous systems, the method is a new approach to derive discretized equations from the original partial differential equations. It requires more implementation effort, but yields more accurate models than alternative discretization methods such as finite difference and finite element approximations. The method is demonstrated on a fixed bed reactor model and a heat exchanger model.

Acknowledgments

I want to thank Sigurd Skogestad for supervising this thesis and helping me in many occasions with advice and knowledge. It is impressive that his door is always open and he is ready to discuss matters at almost any time.

I would further like to thank Pål Kittilsen from Cybernetica AS for good cooperation and offering help when needed. Thanks to my colleagues from the PROMATCH project for good cooperation and company in various places in Europe.

Many thanks to Lisbeth Roel for help in administrative matters in the department.

I feel grateful to my former colleagues Bjørn Tore and Olav for a lot of technical help and advice in general PhD matters. Thanks to my colleagues and friends of the process control group for help and good company.

Most important for me is to thank my family and the good people and friends I met during this time. I thank all of you for your friendship and help.

This work has been supported by the European Union within the Marie-Curie Training Network PROMATCH under the grant number MRTN-CT-2004-512441.

Contents

Chapter 1

Introduction

1.1 Motivation

Mathematical models have become an indispensable tool for the analysis, prediction, control and optimization of many phenomena and processes in the world. Examples are technical systems, such as chemical processes, robots or vehicles, geophysical systems such as weather, climate or reservoir systems, hydraulic, ecological, economical and astronomical systems. A mathematical model for a certain system can be used for analyzing the system and for predicting the system behavior. In some cases, especially for technical systems, knowledge about the systems behavior can be used for designing or manipulating a system to achieve a desired behavior.

Mathematical models can quickly become very complex, when for a realistic prediction a large number of details or subsystems need to be included in the model. This can make the simulation or the analysis of a model difficult. This is the motivation for deriving less complex "reduced" models from complex models, where a reduced model is supposed to approximate the original behavior sufficiently well for a given application.

This thesis discusses a model reduction method for one-dimensional systems. These systems extend spatially in one direction, and appear frequently in chemical engineering and other technical areas. Typical exam-

ples are columns for distillation, absorption and cleaning, tubular or fixed bed reactors, heat exchangers, etc.
One important class of application mathematical models of these processes are needed for is real-time optimizing control. In these applications, the models are used to simulate the behavior of the system when certain variables of the system are manipulated. This way, the system can be controlled in a desired way. The required changes of the manipulated variables can be found in an automatic optimization procedure using multiple simulations and an objective function that evaluates the behavior of the system. The values of the manipulated variables are determined such that they minimize this objective function.

This work was part of the European Union project PROMATCH, which focused on developing model reduction methods for application in real-time optimizing process control applications. The project was divided into groups, each of which investigated different model reduction approaches. This work focused on physically-based model reduction methods, whereas other groups focused on more mathematical approaches. As a case study, an industrial distillation column was employed.
The work in this study started out with the development of a model reduction procedure for staged distillation models. The purpose was to derive reduced models that reduce the computational complexit when the model is simulated in model predictive control and real-time optimization applications. The method was subsequently extended to general one-dimensional systems. This broadens the range of applications of the method.

1.2 Thesis overview

Chapter 2 introduces some of the basic concepts and terminology that is needed to understand the later chapters of the thesis. An introduction into the numerical aspects of integrating differential-algebraic equations is given, with the aim of explaining the computational complexity of mathematical models and how reduced models can reduce it.

Chapter 3 reviews the aggregated modeling method of Lévine and Rouchon (1991) and presents a clarified derivation of the reduced models. From a simple distillation model, several reduced models are derived. It is shown that a gain in computational performance can be achieved when the algebraic equations resulting from the reduction procedure are eliminated from the reduced models. Optimized and non-optimized reduced model parameters are tested with several numerical integrators, and the incorporation of the regulatory control layer of the system is investigated.

Chapter 4 presents an extension of the reduction method to complex distillation models with mass and energy balances and complex hydraulic and thermodynamic relationships. Fast implementations of both full and reduced models are described. The elimination procedure of the algebraic equations and the tabulation of the required function values are described in detail. Reduced models with optimized and non-optimized reduced model parameters are tested, and their performance is compared with the original model. A detailed analysis of the computational complexity of both full and reduced models is given. Alternative approaches concerning implementation details of the reduction method and the relationship to alternative model reduction methods are discussed.

Chapter 5 presents a generalization of the reduction method to one-dimensional distributed systems. For discrete systems, the reduction procedure is the same as for the distillation models discussed before. For continuous systems, the reduction procedure is derived as a limit case of the discrete procedure, resulting in a combination of aggregation elements described by ordinary differential equations and steady-state systems described by boundary value systems. While a simple distillation model serves as example for discrete systems, a heat exchanger model and a fixed bed reactor model are used as examples for continuous systems. The approximation quality of the reduced models is compared with the approximation quality of models obtained by finite difference approximations. The method is compared with the standard singular perturbation procedure to derive slow reduced models, and the differences and similarities are discussed.

Chapter 6 contains conclusions and directions for future research.

1.3 Publications

Chapter 3: Linhart, A., and Skogestad, S., 2009. Computational performance of aggregated distillation models. Computers & Chemical Engineering 33, 296-308.

Chapter 4: Linhart, A., and Skogestad, S., 2009. Reduced distillation models via stage aggregation. Submitted to Chemical Engineering Science.

Chapter 5: Linhart, A., and Skogestad, S., 2009. An aggregated model reduction method for one-dimensional distributed parameter systems. Submitted to Chemical Engineering Science.

Chapter 2

Preliminaries

2.1 Model reduction

Generally, the term *model reduction* refers to a procedure where a reduced model of lower complexity is derived from a more detailed *original* or *full model*. The reduced model is supposed to be capable to reproduce the behavior of the full model with a certain *approximation quality*. Usually, the target application of the reduced model determines the accuracy requirements for the approximation quality of the original behavior. Similarly, a lower complexity of the reduced model means that the reduced model can be handled easier by the target application. For example, if the reduced model is intended for design purposes, a low structural complexity is desired. If the model is intended for real-time applications, a low *computational complexity* is desired. The term *performance of a reduced model* will in the following be used for the relationship of complexity and approximation quality compared to the original model.

Often, model reduction is understood as *model order reduction*, which means that the reduced model has a lower number of dynamic states than the original model. While a lower number of dynamic states is often a key requirement for a reduced computational complexity, a model order reduction can also come at the cost of complicating the structure of the original model or increasing the complexity of the remaining model equations. An example are *projection-based methods* such as balanced truncation (Scherpen, 1993) or proper orthogonal decomposition (Astrid, 2004), where the dynamic state x of the original model is projected on a reduced state space (becoming $\tilde{x}$) by a certain projection matrix T. However, to evaluate the right-hand sides of the equations of the reduced model, the full state has to be reconstructed by applying some inverse transformation matrix $\tilde{T}$:

$$\dot{\tilde{x}} = Tf(\tilde{T}\tilde{x}). \tag{2.1}$$

Here, although $\tilde{x}$ is the reduced state vector of reduced length, the full right-hand side $f(x)$ is evaluated. Furthermore, the transformation T possibly destroys all structure that was present in the original model.

Another example are models derived using *quasi-steady-state assumptions.* In these models, a number of dynamic equations are transformed into algebraic equations by setting their left-hand sides to 0. If the resulting algebraic equations are not eliminated from the model somehow, they have to be solved during the integration of the dynamic equations. In section 2.2, it is discussed in detail that this is computationally approximately as costly as integrating the original system.

Generally, model reduction is a complex field. There exist numerous approaches for a vast variety of systems. Mainly, it can be differentiated between methods for linear (Antoulas, 2005) and nonlinear systems (Ersal et al., 2008, Marquardt, 2001, van den Berg, 2005). Naturally, methods for linear systems can be much more generic than methods for nonlinear systems, which are usually developed for more narrow classes of systems.

For nonlinear model reduction of distillation models, several approaches have been used in the past. One successful approach are orthogonal collocation models (Cho and Joseph, 1983, Stewart et al., 1984), which inherits the concept of finite element methods for partial differential equations (Hundsdorfer and Verwer, 2008). There, variable profiles along the spatial dimension of the system are approximated by piecewise polynomial basis functions. Successful applications regarding experimental validation (Karlström and Breitholtz, 1992) and computational performance (Dalaouti and Seferlis, 2006) have been reported. Another approach are wave propagation methods (Hwang, 1991, Kienle, 2000, Marquardt, 1990), where the dynamics of the distillation column are approximated by moving wave profiles. While the method was originally restricted by assumptions like constant holdups and flows, extensions to models including enthalpy and holdup effects were presented lately (Hankins, 2007).

The method of balancing of empirical gramians (Hahn and Edgar, 2002) uses a state transformation based on empirical gramians such that the states of the transformed system are approximately ordered according to their significance for the input-output behavior of the system. It is an extension of the balanced realization methods for linear systems as described in Skogestad and Postlethwaite (2005). Hahn and Sun (2004) describe an application of the balancing method with subsequent reduction of the algebraic equations in the balanced model.

A singular perturbation approach is presented by Kumar and Daoutidis (2003). The resulting model is more suitable for nonlinear controler design than for fast simulations due to its implicit nature. An overview of further reduction and simplification methods for distillation column is given by Skogestad (1997).

This thesis focuses on a model reduction method for one-dimensional dynamic systems, which inherits the concepts of the aggregated modeling method of Lévine and Rouchon (1991) for simple staged distillation column models. This method is a further development of the compartmental modeling method of Benallou et al. (1986), where the distillation column is divided into compartments, the dynamics of which are governed by a single differential equation for each compartment. The aggregated modeling method has been applied lately by Bian et al. (2005) and Khowinij et al. (2005) for deriving reduced models for real-time optimizing control applications.

The aggregated modeling method will be described in more detail later in this thesis. The remaining sections of the introduction are intended to provide an overview of important aspects of theory and applications to be able to understand the performance of the reduction method investigated in this thesis.

2.2 Numerical solution of differential-algebraic equations

This section gives an introduction into the numerical integration of ordinary differential equations and differential-algebraic equations. The theory is very well described in classic books such as Hairer et al., 2000, Hairer and Wanner, 2002, and Ascher and Petzold, 1998. The reason why some of the basic concepts are repeated here is that the computational aspect of any model reduction can only be understood if the principal functionality of the numerical algorithm used to integrate the model in time is known. In particular, two numerical aspects will be addressed in this section:

1. The concept and implication of *stiff* systems are explained. This is important for understanding singular perturbation systems and differential-algebraic solvers.

2. The numerical tasks of the integration that most of the computational effort is spent in are described; these are the residual evaluation, Jacobian evaluation and the solution of the linear systems of equations arising from the integration algorithm.

2.2.1 Basic integration in time

The starting point is the integration (numerical solution) of ordinary differential equations of the form

$$\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, \mathbf{u}, t), \tag{2.2}$$

where $\mathbf{x}$ is the state vector, $\mathbf{u}$ is the input vector, and t is time. The simplest integration scheme possible is the *Euler forward scheme* (*"explicit Euler"*)

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta t \mathbf{f}(\mathbf{x}_i, \mathbf{u}_i, t_i), \tag{2.3}$$

where $\mathbf{x}_i$ is the numerical solution at time t_i, and $\Delta t = t_{i+1} - t_i$ is a time step of a certain length. This is called an *explicit* scheme, because the equations for the new states $\mathbf{x}_{i+1}$ at time t_{i+1} are explicit in the variables known at

time t_i. Explicit schemes have the advantage that the computation of the new states requires the least possible effort.
In contrast to this, *implicit* schemes require the solution of a system of nonlinear equations at every time step. The simplest implicit integration scheme is the *Euler backward scheme* (*"implicit Euler"*)

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \Delta t \mathbf{f}(\mathbf{x}_{i+1}, \mathbf{u}_{i+1}, t_{i+1}), \tag{2.4}$$

where the new states $\mathbf{x}_{i+1}$ are implicitly defined. What is the advantage of the computationally more expensive implicit schemes?
Figure 2.1 shows the solution of the simple test equation

$$\dot{x} = -\lambda x \tag{2.5}$$

with $\lambda = 1$ using the explicit and implicit Euler schemes for different time steps. It can be seen that for the smallest time step used, $\Delta t = 0.5$, both schemes yield qualitatively correct solutions. For larger time steps, the explicit Euler scheme yields oscillating solutions, which become unstable for $\Delta t > 2$, whereas the implicit Euler scheme yields stable, yet less accurate solutions. This is an elementary result from stability theory of numerical differential equations. The consequence is that explicit schemes are restricted to time steps smaller than a certain value that depends on the time-constants of the system.

2.2.2 Stiff systems

In case of the example above, it seems appropriate for accuracy reasons to choose a time step that is smaller than the stability requirement for the explicit Euler scheme. The situation is different, however, for *stiff* systems. A trivial example for such a system is the following linear system

$$\dot{x}_1 = -x_1, \tag{2.6}$$
$$\dot{x}_2 = -100x_2. \tag{2.7}$$

If this system was to be integrated with the explicit Euler scheme, the time step Δt would have to be chosen smaller than 0.02 to obtain a stable

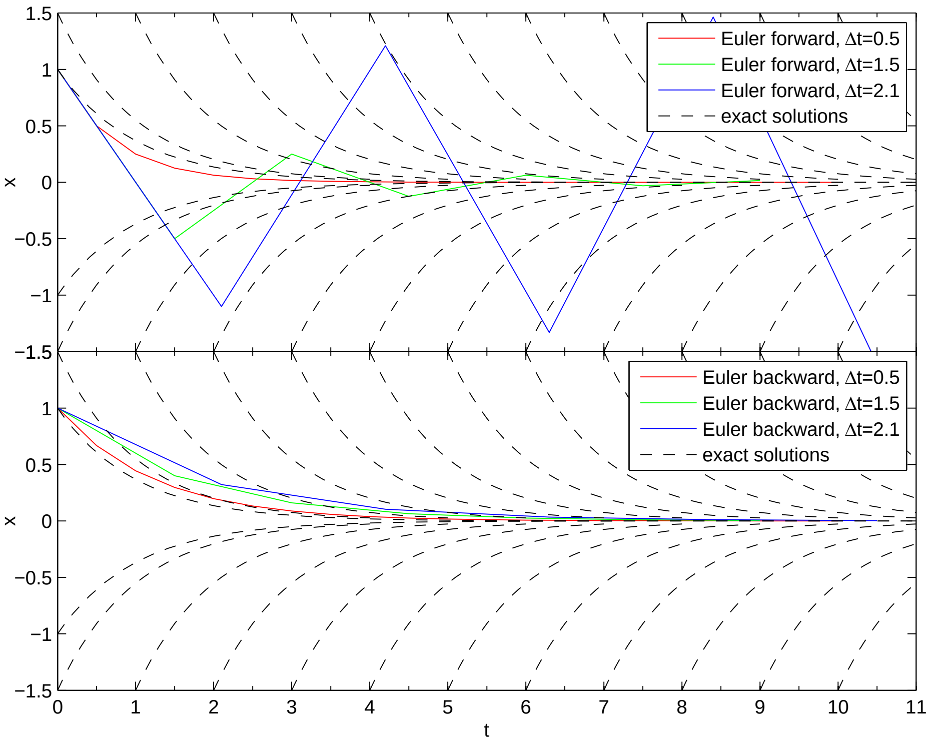

Figure 2.1: Solutions of test equation $\dot{x} = -\lambda x$ with Euler forward scheme (upper plot) and Euler backward scheme (lower plot) with fixed step sizes 0.5, 1.5 and 2.1.

solution for equation (2.7). This time step would be, however, quite small for the integration of equation (2.6), resulting in a large number of steps to obtain a solution for the complete system.

Basically, stiff systems are systems where dynamics on different time-scales are present. In the above example, the two time-scales are clear. In general, it is possible for all linear systems to decompose the system into eigenmodes. The time-scale of each eigenmode is determined by its eigenvalue. A linear system can be said to be stiff if there coexist large and small eigenvalues in the system. However, Hundsdofer and Verwer (2007) remark that "Stiffness is not a mathematical definition, since no quantification is given for 'large' or 'moderate'. Instead it is an operational concept, indicating the class of problems for which implicit methods can perform (much) better than explicit methods". Similarly, Hairer and Wanner (2002) give the definition "Stiff systems are systems where explicit methods don't work."

A simple example for a stiff nonlinear system is the Van der Pol oscillator (Khalil, 2000)

$$\dot{x}_1 = x_2, \tag{2.8}$$

$$\dot{x}_2 = \mu(1 - x_1^2)x_2 - x_1, \tag{2.9}$$

when $\mu >> 1$. This system is a nonlinear modification of the harmonic oscillator, and performs cyclic relaxation oscillation. Figure 2.2 shows phase plots of the two variables x_1 and x_2 for simulations with $\mu = 10$. The dots on the trajectory in the upper plot mark time intervals of 0.2. The total period time is around 20. It can be seen that the dynamics of the system are much slower along the horizontal parts of the trajectory than on the remaining parts. The middle and lower plots show MATLAB simulations with the explicit solver ode45 and the implicit solver ode15s, respectively. The dots on the trajectory mark the time steps the solver has taken. It can be seen that the explicit solver ode45 uses much more time steps on the horizontal parts of the trajectory with slow dynamics than on the rest of the trajectory, whereas the step size for the implicit solver ode15s is only determined by accuracy requirements, which leads to large step sizes on the parts with slow dynamics.

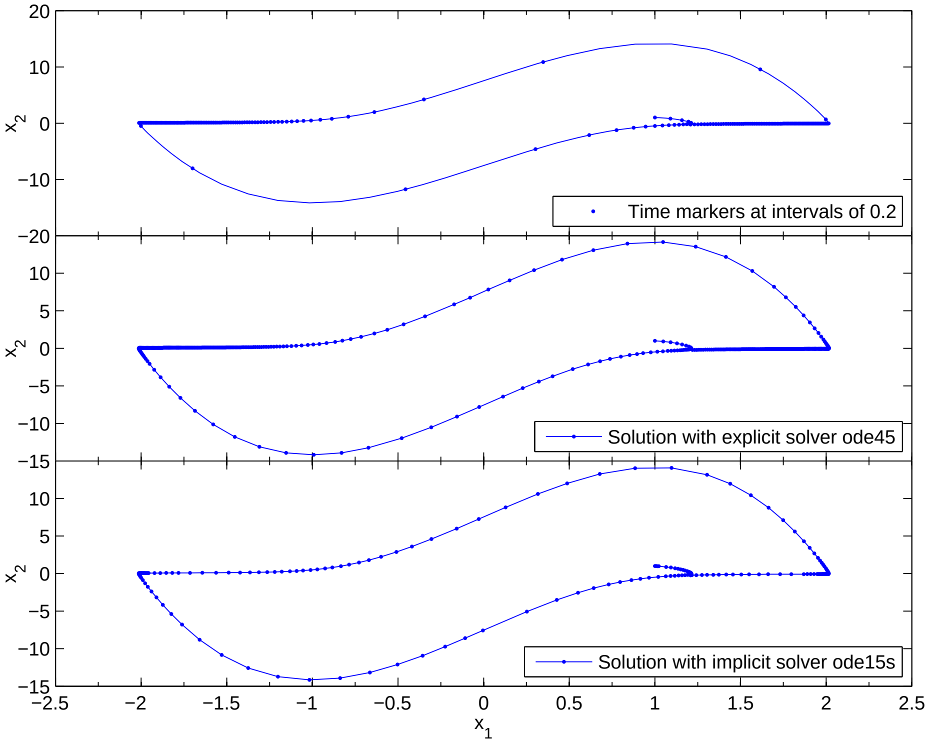

Figure 2.2: Phase plot of the van der Pol equation. The upper plot shows time markers on the solution at intervals of 0.2. The middle plot shows the solution and the steps of the explicit solver ode45. The lower plot shows the solution and the steps of the implicit solver ode15s.

2.2.3 Singular perturbation systems

The time-scale separation of the van der Pol oscillator can be made visible if the system is transformed into the *standard form of singular perturbations* (Kokotovic et al., 1986). For this, a state transformation

$$\begin{aligned} y &= -\frac{1}{\mu}x_1 + x_2 - \frac{1}{3}x_2^3, && (2.10)\\ z &= x_1, && (2.11) \end{aligned}$$

is introduced, which, together with the time scaling $t' = t/\mu$ and the parameter $\epsilon = 1/\mu^2$, yields the system

$$\begin{aligned} \frac{dy}{dt'} &= z, && (2.12)\\ \varepsilon\frac{dz}{dt'} &= -y + z - \frac{1}{3}z^3. && (2.13) \end{aligned}$$

In this form, the variable y is called the *slow variable*, and the variable z is called the *fast variable*. In the standard form of singular perturbations, the time derivatives of the fast variables are multiplied with a small *singular perturbation parameter* ε. The transformation that puts the system into the standard form of singular perturbations is chosen such that the right-hand sides of the equations are well balanced in terms of magnitude, and the small singular perturbation parameter causes the difference in dynamics between the slow and the fast variables. This can be seen in figure 2.3, where the dots on the blue trajectory mark time intervals of 0.01 (in the t' time-scale). The dynamics of the system are fast in the z variable and slow in the y variable. This example demonstrates how the two-time-scale dynamics of a nonlinear system can be analyzed by transforming the system into the singular perturbation standard form. As can be seen already for this simple example, the transformation to do this is not obvious. This is one of the difficulties when working with systems on two or more time-scales. However, once the system has been successfully transformed, the time-scale separation can be exploited for, for example, controler design or model reduction.

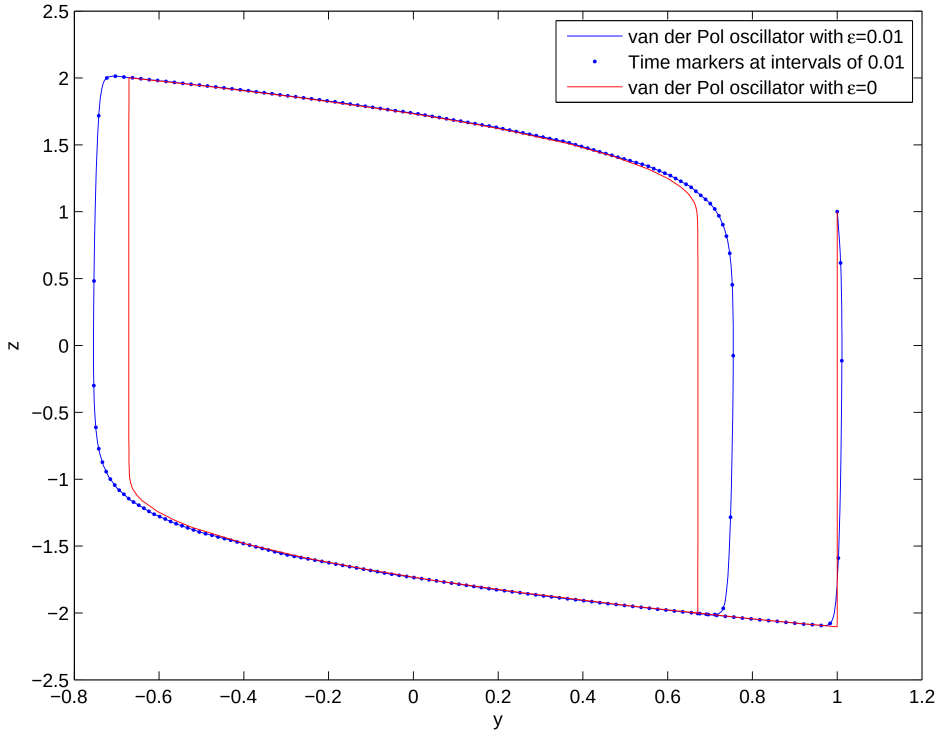

Figure 2.3: Phase plot of the transformed van der Pol equation. y and z are the slow and fast variables after the transformation, respectively. The blue trajectory is the system with $\varepsilon = 0.01$, the red trajectory is the system with quasi-steady-state approximation of the fast dynamics. The dots on the blue trajectory mark time intervals of 0.01.

In the latter case, the system can be approximated by a *quasi-steady-state approximation*, where the singular perturbation parameter ε goes to 0. The resulting system reads

$$\begin{aligned} \dot{y} &= z, & (2.14)\\ 0 &= -y + z - \frac{1}{3}z^3. & (2.15) \end{aligned}$$

The idea is to approximate the fast dynamics of the system by instantaneous dynamics. The fast dynamic equations consequently are converted into algebraic equations. This is one reason for the expression "singular perturbation", because the perturbation parameter ε changes the nature of the system when going to 0. This approximation will only yield satisfactory approximation quality if the time-scales are sufficiently distinct. In figure 2.3, the red trajectory shows the solution of this system. The vertical trajectories signify instantaneous movement to the next slow interval. It can be seen that the slow movement of the system is approximated quite well. The model reduction has, however, only succeeded in reducing the dynamic order of the system by one. The system consists now of dynamic and algebraic equations, and is therefore described by *differential-algebraic equations (DAE)*. Their numerical treatment does not differ much from the treatment of the stiff system they were derived from. This will be discussed next.

2.2.4 Differential-algebraic equations

Systems of the form

$$\begin{aligned} \dot{\mathbf{y}} &= \mathbf{f}(\mathbf{y}, \mathbf{z}, t), & (2.16)\\ 0 &= \mathbf{g}(\mathbf{y}, \mathbf{z}, t), & (2.17) \end{aligned}$$

are called *differential-algebraic equations (DAE)*, because they consist of differential equations (2.16), and algebraic equations (2.17). For simplicity of notation, the input variable $\mathbf{u}$ is left out in the equations. The algebraic equations may result from a quasi-steady-state approximation of a fast dynamic equation, as described in the previous section, or can be constitutive equations, which implicitly define variables used in the dynamic equations. Especially in mechanical systems, the algebraic equations can be constraints, which force the motion of the dynamic system on certain paths. In these cases, depending on the choice of coordinates for the dynamic variables, the algebraic equations can give rise to *high-index systems*, which are numerically difficult to solve. There exists extensive literature on these kind of systems (Hairer and Wanner, 2002, Ascher and Petzold, 1998). In the following, only the relatively simple *index-1 systems* are discussed. For these systems, basically two solution strategies are possible:

1. The algebraic equations are solved separately from the dynamic equations. The system (2.16) and (2.17) then takes the form

   $$\dot{\mathbf{y}} = \mathbf{f}(\mathbf{y}, \mathbf{h}(\mathbf{y}, t), t), \tag{2.18}$$

   where $\mathbf{h}(\mathbf{y}, t)$ is the solution of the algebraic equations (2.17) for $\mathbf{z}$. This can be done in two ways:

   (a) The algebraic equations are solved each time the algebraic variables are required using a numerical solver for nonlinear systems of equations (*nested approach*);

   (b) The algebraic equations are pre-solved, that means the algebraic variables that are needed in the dynamic equations are calculated as functions of the relevant dynamic variables. In simple cases, analytic solutions for the algebraic equations can be found, but generally, the equations have to be solved numerically on a sufficiently large domain of input variables and the solutions have to be approximated in a suitable way. This can be done by, for example, interpolation in tables or polynomial approximations.

2. The algebraic equations and dynamic equations are solved simultaneously using a *differential-algebraic solver* (*simultaneous approach*).

Strategy 1a is usually the least efficient. It can be used, however, if the algebraic equations can be solved with little effort, and if a system of ordinary differential equations is desired for certain reasons. For example, in some cases an explicit solver can be used to integrate the differential equations, which is usually faster than using an implicit solver in case the differential equations are not stiff. If, however, an implicit solver is used to integrate the differential equations, the efficiency is significantly reduced. This is mainly because the original algebraic equations (2.17) have to be solved every time the right-hand side of equation (2.18) is required. Then, the two iterations for solving the nonlinear equations (2.17) and the equations from the implicit integration scheme are nested.

Strategy 1b can be very efficient, when the pre-solution of the algebraic equations results in accurate approximation functions of acceptable complexity. In general, pre-solution is only possible if the number of variables the solution of the algebraic equations depend on is sufficiently low. However, in case this strategy can be applied successfully, a real reduction of the model order can be achieved.

Most frequently, however, the most efficient and convenient way is strategy 2. In this case, all algebraic equations are solved simultaneously. For example, the implicit Euler scheme is applied to the dynamic equations (2.16), and the algebraic equations (2.17) of the original systems are solved at the new time step $i+1$:

$$\begin{aligned} \mathbf{y}_{i+1} &= \mathbf{y}_i + \Delta t \mathbf{f}(\mathbf{y}_{i+1}, \mathbf{z}_{i+1}, t_{i+1}), & (2.19) \\ 0 &= \mathbf{g}(\mathbf{y}_{i+1}, \mathbf{z}_{i+1}, t_{i+1}). & (2.20) \end{aligned}$$

This way, at each time step, the nonlinear system of equations

$$\begin{aligned} 0 &= \mathbf{y}_i + \Delta t \mathbf{f}(\mathbf{y}_{i+1}, \mathbf{z}_{i+1}, t_{i+1}) - \mathbf{y}_{i+1}, & (2.21) \\ 0 &= \mathbf{g}(\mathbf{y}_{i+1}, \mathbf{z}_{i+1}, t_{i+1}) & (2.22) \end{aligned}$$

has to be solved.

2.2.5 Computational complexity of DAE systems

Nonlinear equations

It is important to understand the distribution of the computational effort when solving a DAE system in order to evaluate the effectiveness of a certain model reduction method. In the following, the implicit Euler scheme is used to discuss the basic computational aspects.
The implicit Euler scheme in (2.21)-(2.22) can be written as

$$0 = \gamma(\mathbf{x}_{i+1}), \tag{2.23}$$

where $\gamma = \{\gamma_{\mathbf{1}}, \gamma_{\mathbf{2}}\}$ are the combined right-hand sides of equations (2.21) and (2.22), and $\mathbf{x}_{i+1} = \{\mathbf{y}_{i+1}, \mathbf{z}_{i+1}\}$ are the combined unknown variables. For the solution of this system, usually some nonlinear equation solver based on the Newton scheme is used:

$$0 = \frac{\partial}{\partial \mathbf{x}}\gamma\bigg|_{\mathbf{x}^j} (\mathbf{x}^j - \mathbf{x}^{j+1}) - \gamma(\mathbf{x}^j), \tag{2.24}$$

where $\mathbf{x}^j$ is the j-th iteration variable. The superscript index ^{j} is used here to distinguish the iteration index from the time index $_i$ in the Euler scheme (2.21) and (2.22). $\frac{\partial}{\partial \mathbf{x}}\gamma\Big|_{\mathbf{x}^j}$ is here the Jacobian matrix of eq. (2.23) evaluated at the current iterate $\mathbf{x}^j$. In order to obtain the next iterate $\mathbf{x}^{j+1}$, the linear equation system

$$\frac{\partial}{\partial \mathbf{x}}\gamma\bigg|_{\mathbf{x}^j} \mathbf{x}^{j+1} = \frac{\partial}{\partial \mathbf{x}}\gamma\bigg|_{\mathbf{x}^j} \mathbf{x}^j - \gamma(\mathbf{x}^j) \tag{2.25}$$

has to be solved at each iteration step. Since the Newton method is a local method that requires an initial guess for the solution that is close to the real solution, the Jacobian usually does not change much during the iteration steps. A common simplification is therefore to use a *quasi-Newton scheme* using a constant Jacobian $\frac{\partial}{\partial \mathbf{x}}\gamma\Big|_{\mathbf{x}_i}$ that is evaluated at the last known time step. In practice, the same Jacobian is often used for several time steps, and is only updated when the speed of convergence of the Newton scheme is significantly reduced.

Linear algebra

Eq. (2.25) is of the form $\mathbf{Ax} = \mathbf{b}$ and can be solved using a suitable linear algebra method. For the efficiency of the solution, it is crucial to consider the structure of the Jacobian matrix, which basically reflects the interaction structure of the original system. Three different structures are frequent:

1. Dense matrices contain a large number of non-zero elements distributed over the whole matrix;

2. Sparse matrices contain a small number of non-zero elements distributed over the whole matrix, while most of the matrix elements are zero;

3. Banded matrices contain elements only on a diagonal band of a certain width around the main diagonal.

For each structure, specialized algorithms can be used to efficiently solve the system. Sparse matrices are usually solved with *iterative solvers*, while dense and banded matrices are usually solved with *direct solvers*. Sophisticated DAE solvers like DASPK (Li and Petzold, 2000) offer the opportunity to choose between several types of linear algebra solver, or to use an externally provided solver. Because the model reduction method described in this work can be applied to systems with a banded Jacobian structure, the solution of banded linear equation systems is discussed here in more detail. A good reference is Strang (1986).

Direct solution by lower-upper decomposition

The linear equation system

$$\mathbf{Ax} = \mathbf{b} \tag{2.26}$$

arising from equation (2.25) can be solved using the lower-upper (LU) decomposition of the matrix a, $\mathbf{A} = \mathbf{LU}$, where $\mathbf{L}$ and $\mathbf{U}$ are lower and upper triangular matrices, respectively. The system can then be written as

$$\mathbf{LUx} = \mathbf{b}, \tag{2.27}$$

and can be solved sequentially using the LU-solution

$$\mathbf{L}\mathbf{c} = \mathbf{b}, \tag{2.28}$$

$$\mathbf{U}\mathbf{x} = \mathbf{c}. \tag{2.29}$$

These triangular systems can be solved by simple line-by-line substitution. The LU-decomposition is usually done using the Gauss elimination algorithm which involves the elimination of the elements below the main diagonal by subtracting the corresponding lines from each other. The LU-decomposition is independent of the right-hand side of eq. (2.26), and has therefore to be done only once for the matrix $\mathbf{A}$. Using the LU-decomposition, changing right-hand sides can be handled efficiently. This is of advantage when applied to the Newton iterations (2.25), where the Jacobian matrix is constant, but the right-hand side changes at each iteration step.

The number of operations of both LU-decomposition and LU-solution is proportional to n^2 for dense matrices $\mathbf{A}$, where $\mathbf{A}$ is an $n \cdot n$ matrix. If $\mathbf{A}$ is a banded matrix, the number of operations is proportional to $n \cdot w$, where w is the width of the band of non-zero elements of the matrix. The solution of banded systems with a width that is significantly less than that of the original system is therefore computationally significantly cheaper (with factor n/w) than the solution of dense systems.

Main sources of computational complexity

The main sources of the computational complexity of the numerical integration of a differential equation system using the implicit Euler scheme and LU-decomposition are therefore

1. the evaluation of the right-hand sides $\gamma(\mathbf{x}^j)$ of equation (2.23) at each iteration step of (2.25), which consists mainly of the evaluation of the right-hand sides of the original differential equations (2.16) and (2.17);

2. the evaluation of the Jacobian of the right-hand side of equations (2.21) and (2.22), which is done once for several time steps; this is

mainly the evaluation of the Jacobian of the right-hand side of the original differential equations (2.16) and (2.17);

3. the LU-decomposition of the Jacobian, which is done once for every Jacobian update;
4. the LU-solution of equations (2.25) at each iteration step.

Their relative contribution depends on the system that is integrated. The right-hand side calculations can be computationally very intensive if complex calculations such as thermodynamic properties have to be done. The same will affect the calculation of the Jacobian. On the other hand, the system size and the structure of the Jacobian influences the computational complexity of the linear algebra algorithms.
Generally, the computational complexity of the integration of the system can only be significantly reduced if the complexities of the computationally most intensive contributors are significantly reduced.

2.3 Model predictive control and real-time optimization

One important industrial application for reduced models as investigated in this thesis are real-time optimizing control applications such as *nonlinear model predictive control (NMPC)* (Allgöwer and Zheng, 2000, Qin and Badgwell, 2003) and *dynamic real-time optimization (D-RTO)* (Schlegel, 2005). In these applications, the manipulated variables of the process are determined such that a certain objective function is minimized:

$$\min_{\mathbf{u}(t)} J(\mathbf{u}(t)), \tag{2.30}$$

where $\mathbf{u}(t)$ is the vector of the manipulated variables of the process (inputs), and J is the objective function. In NMPC, the objective function usually contains terms to penalize the deviation of the important output trajectories from corresponding reference trajectories, and terms that assign a cost to the manipulated variables of the process. In D-RTO, the objective function often contains economic terms that relate the trajectories of the process to the revenues that can be gained from the product of the process, and the costs of operation. Often, with D-RTO reference trajectories that allow an economically efficient operation of the process are generated, whereas NMPC is used to keep the process close to the reference trajectories in the present of disturbances. D-RTO therefore often operates on longer time horizons than NMPC.

NMPC and D-RTO work by using a model of the process to predict the future behavior of the process given a certain choice of input variable trajectories. In order to obtain a solvable minimization problem, the time span the future inputs for the process are computed for is limited to a certain *control horizon*. On this control horizon, the input signal is discretized on intervals, such that the input trajectory of each input on each interval is parametrized by one variable. This yields the discrete inputs $\mathbf{u}_i, i = 1, ..., N$, where N is the number of intervals on the control horizon. The length of an interval is referred to as the *sampling time*. The trajectories of the model are considered in the objective function (2.30). This means that the model equations are evaluated repeatedly during the iteration to find the minimum of the objective function.

The advantage of NMPC and D-RTO over classical control schemes is the ability to handle constraints in the inputs and the process trajectories, and in the ability to handle processes with strong nonlinearities. However, the application to large-scale systems is limited, since the algorithms are computationally very intensive. Therefore, reduced models that can be simulated faster than the original process models are of high interest.

The reduced models derived with the method described in this thesis approach the steady-state asymptotically with time. This means that the approximation quality is best when the inputs of the system are changing slowly and the system is close to steady-state, or when the inputs are constant. The dynamic order of the reduced model can be varied. The higher the dynamic order of the reduced model, the better is the approximation quality after or during fast input changes. This means that the choice of the reduced model should be adapted to the requirements of the application. In this respect, the reduced model has most likely to be of higher dynamic order when applied in a fast MPC with short sampling times than in a slow D-RTO on a long time horizon. Similarly, the reduced models are likely to perform better with slow smooth input signals than with fast step signals. If possible, this should be considered when designing the MPC or D-RTO application.

Bibliography

[1] Allgöwer, F., and Zheng, A., 2000. Nonlinear Model Predictive Control. Progress in Systems Theory 26, Birkhäuser, Basel.

[2] Antoulas, A.C., 2005. Approximation of Large-Scale Dynamical Systems. Cambridge University Press, Cambridge.

[3] Astrid, P., 2004. Reduction of process simulation models: a proper orthogonal decomposition approach. PhD Thesis, TU Eindhoven.

[4] Ascher, U.M., and Petzold, L.R., 1998. Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. SIAM, Philadelphia.

[5] Bian, S., Khowinij, S., Henson, M.A., Belanger, P., and Megan, L., 2005. Compartmental modeling of high purity air separation columns. Computers & Chemical Engineering 29, 2096-2109.

[6] Dalaouti, N., and Seferlis, P., 2006. A unified modeling framework for the optimal design and dynamic simulation of staged reactive separation processes. Computers & Chemical Engineering 30, 1264-1277.

[7] Ersal, T., Fathy, H.K., Rideout, D.G., Louca, L.S., and Stein, J.L., 2008. A review of proper modeling techniques. Journal of Dynamic Systems, Measurement, and Control 130, article number 061008.

[8] Hairer, E., Nørsett, S.P., and Wanner, G., 2000. Solving Ordinary Differential Equations I - nonstiff problems. Springer, Berlin.

[9] Hairer, E., and Wanner, G., 2002. Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems. Springer, Berlin.

[10] Hankins, N.P., 2007. A nonlinear wave model with variable molar flows for dynamic behavior and disturbance propagation in distillation columns. Chemical Engineering Research and Design 85, 65-73.

[11] Hundsdorfer, W., and Verwer, J.G., 2007. Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations. Springer, Berlin.

[12] Hahn, J., and Edgar, T.F., 2002. An improved method for nonlinear model reduction using balancing of empirical gramians. Computers & Chemical Engineering 26, 1379-1397.

[13] Hwang, Y.L., 1991. Nonlinear-wave theory for dynamics of binary distillation columns. AICHE Journal 37, 705-723.

[14] Karlström, A., and Breitholtz, C., 1992. Experimental Validation of a Packed Bed Distillation Model. Chemical Engineering Technology 15, 406-416.

[15] Khalil, H.K., 2000. Nonlinear Systems, third edition. Prentice Hall, New Jersey.

[16] Khowinij, S., Henson, M.A., Belanger, P., and Megan, L., 2005. Dynamic compartmental modeling of nitrogen purification columns. Separation and Purification Technology 46, 95-109.

[17] Kienle, A., 2000. Low-order dynamic models for ideal multicomponent distillation processes using nonlinear wave propagation theory. Chemical Engineering Science 55, 1817-1828.

[18] Kokotovic, P., Khalil, H.K., and O'Reilly, J., 1986. Singular Perturbation Methods in Control: Analysis and Design. SIAM classics in applied mathematics 25, SIAM, London.

[19] Kumar, A., and Daoutidis, P., 2003. Nonlinear model reduction and control for high-purity distillation columns. Industrial and Chemistry Research 42, 4495-4505.

[20] Lévine, J., and Rouchon, P., 1991. Quality Control of Binary Distillation Columns via Nonlinear Aggregated Models. Automatica 27, 463-480.

[21] Li, S., and Petzold, L.R., 2000. Software and Algorithms for Sensitivity Analysis of Large-Scale Differential Algebraic Systems. Journal of Computational and Applied Mathematics 125, 131-145.

[22] Marquardt, W., 2001. Nonlinear model reduction for optimization based control of transient chemical processes. Proceedings Chemical Process Control VI, 30-60.

[23] Marquardt, W., 1990. Traveling waves in chemical processes. International Chemical Engineering 30, 585-606.

[24] Qin, S.J., and Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control Engineering Practice 11, 733-764.

[25] Scherpen, J.M.A., 1993. Balancing for nonlinear systems. Systems & Control Letters 12, 143-153.

[26] Schlegel, M., 2005. Adaptive discretization methods for the efficient solution of dynamic optimization problems. VDI-Verlag, Düsseldorf.

[27] Strang, G., 1988. Linear algebra and its applications. Third edition. Thomson Learning.

[28] Skogestad, S., 1997. Dynamics and Control of Distillation Columns: A Critical Survey. Modeling, Identification and Control 18, 177-217.

[29] Skogestad, S., and Postlethwaite, I., 2005. Multivariable feedback control - Analysis and design, Second Edition. John Wiley & Sons, Chichester.

[30] Sun, C., and Hahn, J., 2005. Reduction of Differential-Algebraic Equation Systems via Projections and System Identification. Journal of Process Control 15, 639-650.

[31] Stewart, W.E., Levien, K.L., and Morari, M., 1984. Simulation of fractionation by orthogonal collocation. Chemical Engineering Science 40, 409-421.

[32] van den Berg, J., 2005. Model reduction for dynamic real-time optimization for chemical processes. PhD Thesis, TU Delft.

Chapter 3

Computational performance of aggregated distillation models

published in Computers & Chemical Engineering 33, pages 296-308 (2009).

Abstract

Compartmental and aggregated modeling is used to derive low-order (reduced) dynamic models from detailed models of staged processes. In this study, the aggregated modeling method of Lévine and Rouchon (1991) is revised with the objective of deriving computationally efficient models for real-time control and optimization applications. A simple implementation of the original method not requiring the specification of compartments is presented. The resulting DAE models are converted into ODE models by pre-solving and substituting the algebraic equations resulting from the reduction procedure, which is the key step to increase simulation speed. To study this, the performances of several full and reduced distillation models, with and without base-layer controllers, are compared using different numerical integrators. It is found that while the reduced DAE models are computationally not advantageous, the reduced ODE models decrease the simulation time by a factor of 5 to 10.

Keywords:

Model reduction; Distillation; Aggregated modeling; Computational performance; Real-time optimization; Singular perturbation

3.1 Introduction

With the establishment of dynamic real-time optimization and model predictive control as state-of-the-art methods to efficiently operate industrial processes, reduced models with low computational complexity are in the focus of current research (Allgöwer and Zheng, 2000, Marquardt, 2001, van den Berg, 2005). In particular, reduced nonlinear physically-based models are of high interest for the prediction of the system behavior over a wide range of operating conditions. Many model reduction techniques have been developed for nonlinear systems (Marquardt, 2001, van den Berg, 2005), most of which produce models of lower order. This, however, does not guarantee that the reduced models show a computationally better performance than the original models they were derived from (van den Berg, 2005). This is because a reduced model is most likely less accurate than the original full model, and that the numerical complexity of the full model is often retained in the equations of the reduced model.

For nonlinear model reduction of distillation columns, several model reduction and simplification methods have been developed in the past (Benallou et al., 1986, Cho and Joseph, 1983, Khowinij et al., 2004, 2005, Kienle, 2000, Kumar and Daoutidis, 2003, Lévine and Rouchon, 1991, Marquardt 1990, Skogestad, 1997). Among these are the method of compartmental modeling (Benallou et al., 1986) and later the improved variant of aggregated modeling by Lévine and Rouchon (1991). The latter method is used for deriving the reduced models investigated in this study. It is based on partitioning a distillation column into "compartments" consisting of "steady-state" trays

and dynamic "aggregation" trays, and using a singular perturbation argument (Kokotovic et al., 1986) to derive a reduced-order model. Among its advantages is the perfect steady-state agreement with the original model, a simple derivation, and the good control of the reduced model complexity. Originally, these methods were intended for nonlinear controller design, for which a low-order model is necessary. More recently, they have been used to reduce the simulation time in real-time applications (Bian et al., 2005, Khowinij et al., 2004, 2005). However, it is shown in this study that while only transforming the original system into a reduced system in differential-algebraic equation (DAE) form does not improve the simulation speed of the reduced model, a subsequent elimination of the algebraic equations is necessary to obtain a reduced model in ordinary differential equation (ODE) form, which shows a significantly improved computational performance compared to the original model. On a more fundamental level, it is shown that the notion of compartments is not necessary in the derivation of the reduction method. This greatly simplifies the derivation and makes the extension of the method to more complex systems straightforward.
The paper is organized as follows: In section 3.2, the full model for a binary distillation column, and the derivation of reduced models from this using aggregated modeling is described. Important implementation details and properties of the models are given. In section 3.3, the framework for testing the computational performance of the models is explained, discussing the input signal, the model parameters and the numerical solvers used for simulating the models. The results of the simulations are given in section 3.4. Section 3.5 discusses the results of the simulations, and details and possible extensions of the reduction method. A summary and conclusions are given in section 3.6.

3.2 Models

3.2.1 System and modeling assumptions

The system investigated is a binary distillation column with 72 trays plus reboiler and condenser. Two variants of this system are studied:

1. An "uncontrolled" column with level controllers for condenser and reboiler, but with no temperature or composition control. The reflux L and the boil-up V remain as degrees of freedom ("LV-configuration").

2. A "controlled" column with an additional composition controller in the lower column part that manipulates the boil-up rate V.

The controlled column is shown schematically in figure 3.1. In the following, the uncontrolled system is used to explain the system equations and the reduction procedure. Later, in section 3.2.4, the inclusion of the composition controller is explained.

All assumptions made in this simplified distillation model are discussed in detail by Skogestad (1997). The major modeling assumptions are: Ideal trays, which means that liquid and vapor are in equilibrium at each tray; ideal mixture, which means that the vapor composition y can be expressed as a function of the liquid composition x assuming the constant relative volatility

$$y = k(x) = \frac{\alpha x}{1 + (\alpha - 1)x}, \tag{3.1}$$

where α is the relative volatility; constant molar flows, which means that the energy balance is simplified; constant molar holdup on each tray and negligible mass in the vapor phase.

The assumption of constant molar flows may not be good if the model is to be used for control purposes (Skogestad, 1997), but the focus here is on longer time scales.

The column has one feed flow F at tray number n_F. z_F denotes the concentration of the first (light) component in the feed. A liquid flow L (or $L + F$ for trays below the feed tray) and a vapor flow V enter and leave each tray. The condenser and reboiler levels are assumed to be controlled using the distillate flow D and bottom flow B, respectively. For simplicity, perfect level control is assumed, such that $D = V - L$ and $B = L + F - V$. Note that the assumption of perfect level control is not important with the LV-configuration. The concentrations in these flows determine the purity of the distillation products and are therefore the most important output variables in the process. The feed flow rate F and the feed concentration z_F can be seen as disturbance variables, and the flows L and V are manipulated variables for control.

3.2.2 Full uncontrolled model

The full model consists of one component material balance for each tray and the condenser and reboiler. For ease of notation, the condenser and reboiler are written as tray 1 and N:

$$\begin{aligned}
H_1\dot{x}_1 &= Vy_2 - Vx_1, && (3.2)\\
H_i\dot{x}_i &= Lx_{i-1} + Vy_{i+1} - Lx_i - Vy_i, && (3.3)\\
& \quad i = 2, ..., n_F - 1, && \\
H_{n_F}\dot{x}_{n_F} &= Lx_{i-1} + Vy_{i+1} - (L+F)x_i - Vy_i + Fz_F, && (3.4)\\
H_i\dot{x}_i &= (L+F)x_{i-1} + Vy_{i+1} - (L+F)x_i - Vy_i, && (3.5)\\
& \quad i = n_F + 1, ..., N - 1, && \\
H_N\dot{x}_N &= (L+F)x_{N-1} - (L+F-V)x_N - Vy_N, && (3.6)
\end{aligned}$$

where H_i is the total liquid molar holdup, x_i and $y_i = k(x_i)$ are the concentrations of the first component in the liquid and vapor phase, respectively, of tray i, N is the number of trays including the condenser and reboiler, n_F is the index of the feed tray, and V, L, F, z_F are as described above. The parameters for the system used in this study are given in table 3.2.2. For the input variables, nominal values are given. The parameters are scaled average values of a real industrial distillation column, such that the dynamic behavior and the time constants are similar. SI units (mol, mol/s, s, etc.) are used for all variables.

3.2.3 Reduced models

The reduced models investigated in this study are derived via the aggregated modeling method of Lévine and Rouchon (1991). The method is based on partitioning the column into N_c "compartments", where each

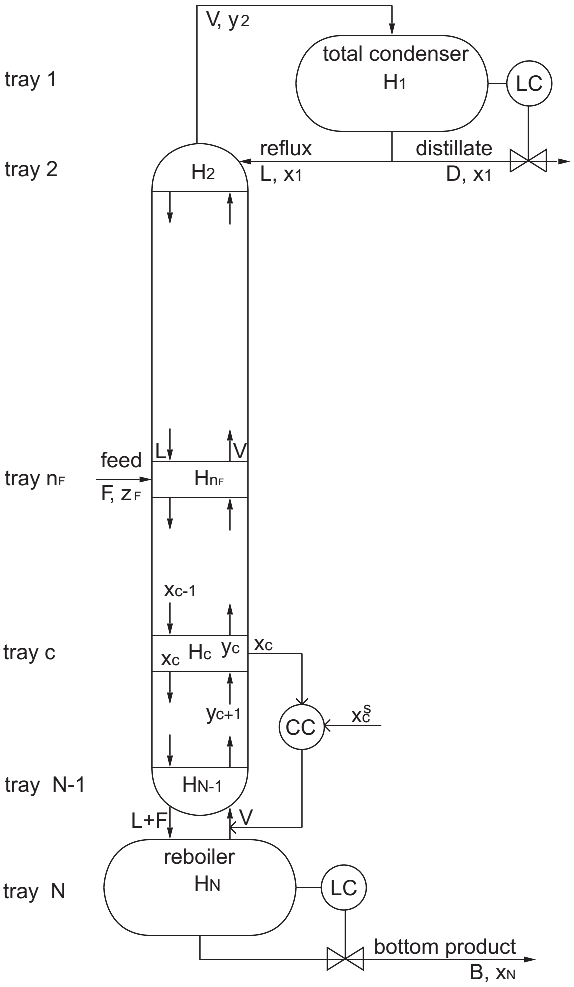

Figure 3.1: Schematic diagram of a binary distillation column with a composition controller in the lower column section.

Parameter	Value
N	74
n_F	36
H_1	20 mol
H_N	20 mol
$H_i, i = 2, ..., N-1$	1 mol
F	0.04 mol/s
α	1.33
Input	Nominal value
z_F	0.45
L	0.12 mol/s
V	0.14 mol/s

Table 3.1: Parameters of the full model.

compartment comprises a number of consecutive trays. By assigning the total holdup of all trays of a compartment to a single "aggregation" tray within the compartment, a time-scale separation caused by the difference between the large compartment holdup and the small tray holdups is made explicit in the model. This time-scale separation can be used for reducing the dynamic order of the model by applying quasi-steady-state assumptions to the trays within each compartment.

In their derivation of the method, Lévine and Rouchon (1991) perform one uncommented simplification step, which results from a mathematically inaccurate treatment of a term when applying the quasi-steady-state assumption. In remark 3 in their paper, they note that the reduction procedure can be described much simpler. This simplification step has the major consequence for the resulting reduced model that only the compartment holdups, but not their boundaries appear in the model equations. Due to this simplification, the reduction procedure is not a true singular perturbation method (Kokotovic et al., 1986). However, the structure of the reduced model, which retains the tridiagonal Jacobian structure of the original model, is numerically favorable. Extensions to more complex models, as will be discussed in section 3.5, are also much more straightforward. In the following, the derivation of the method is therefore described in some detail. First, the original derivation of Lévine and Rouchon is explained for one compartment. Then, the alternative, much simpler derivation is explained, and the relationship to the original derivation is shown. A “true” singular perturbation model, resulting from the derivation without the simplification step, is then compared to the reduced model of Lévine and Rouchon.

Original derivation of reduced compartment model by Lévine and Rouchon

The following derivation considers a certain compartment j, which is formed from a number m of consecutive trays (in this case not including a feed). For simplicity of notation, the numbering of the trays starts with 1 and ends with m. It is assumed here for simplicity that the holdups H of all individual trays are equal. Then, $H = \frac{1}{m}\bar{H}$, where $\bar{H}$ is the total compartment holdup. The original model is

$$\begin{aligned} \frac{1}{m}\bar{H}\dot{x}_1 &= Lx_0 + Vk(x_2) - Lx_1 - Vk(x_1), && (3.7)\\ \frac{1}{m}\bar{H}\dot{x}_2 &= Lx_1 + Vk(x_3) - Lx_2 - Vk(x_2), && (3.8)\\ &\dots \\ \frac{1}{m}\bar{H}\dot{x}_m &= Lx_{m-1} + Vk(x_{m+1}) - Lx_m - Vk(x_m), && (3.9)\end{aligned}$$

with the same notation as in equations (3.1)-(3.6).

A coordinate transformation is applied by introducing the average concentration of the compartment $\bar{x}_s$, where s is the index of the ”aggregation tray” of the compartment, and leaving the remaining $m-1$ variables unchanged:

$$\begin{aligned} \bar{x}_s &= \frac{1}{m}\sum_{i=1}^{m} x_i, && (3.10)\\ \bar{x}_i &= x_i,\ i = 1,...,m,\ i \neq s. && (3.11)\end{aligned}$$

The transformed system now reads

$$\frac{1}{m}\bar{H}\dot{\bar{x}}_1 = Lx_0 + Vk(\bar{x}_2) - L\bar{x}_1 - Vk(\bar{x}_1), \tag{3.12}$$

$$\frac{1}{m}\bar{H}\dot{\bar{x}}_2 = L\bar{x}_1 + Vk(\bar{x}_3) - L\bar{x}_2 - Vk(\bar{x}_2), \tag{3.13}$$

$$\ldots$$

$$\frac{1}{m}\bar{H}\dot{\bar{x}}_{s-1} = L\bar{x}_{s-2} + Vk\Big(m\bar{x}_s - \sum_{i\neq s}\bar{x}_i\Big) - L\bar{x}_{s-1} - Vk(\bar{x}_{s-1}), \tag{3.14}$$

$$\bar{H}\dot{\bar{x}}_s = Lx_0 + Vk(x_{m+1}) - L\bar{x}_m - Vk(\bar{x}_1), \tag{3.15}$$

$$\frac{1}{m}\bar{H}\dot{\bar{x}}_{s+1} = L\Big(m\bar{x}_s - \sum_{i\neq s}\bar{x}_i\Big) + Vk(\bar{x}_{s+2}) - L\bar{x}_{s+1} - Vk(\bar{x}_{s+1}), \tag{3.16}$$

$$\ldots$$

$$\frac{1}{m}\bar{H}\dot{\bar{x}}_m = L\bar{x}_{m-1} + Vk(x_{m+1}) - L\bar{x}_m - Vk(\bar{x}_m). \tag{3.17}$$

This system is now in the standard form of singular perturbation (Kokotovic et al., 1986) with $\varepsilon := \frac{1}{m}$ as a small parameter, and a reduced model can be derived by applying the quasi-steady-state assumption $\varepsilon \to 0$. In this way, all trays except for the aggregation tray are converted into "steady-state" trays, and are described by algebraic equations.
The term $m\bar{x}_s - \sum_{i\neq s}\bar{x}_i$ that appears in equation (3.14) and (3.16) is replaced by $\bar{x}_s$ as a result of the quasi-steady-state assumption $\varepsilon \to 0$. Lévine and Rouchon (1991) use a slightly more complex formulation that admits nonuniform tray holdups. The derived reduced system then reads

$$0 = Lx_0 + Vk(\bar{x}_2) - L\bar{x}_1 - Vk(\bar{x}_1), \tag{3.18}$$

$$\ldots$$

$$0 = L\bar{x}_{s-2} + Vk(\bar{x}_s) - L\bar{x}_{s-1} - Vk(\bar{x}_{s-1}), \tag{3.19}$$

$$\bar{H}\dot{\bar{x}}_s = Lx_0 + Vk(x_{m+1}) - L\bar{x}_m - Vk(\bar{x}_1), \tag{3.20}$$

$$0 = L\bar{x}_s + Vk(\bar{x}_{s+2}) - L\bar{x}_{s+1} - Vk(\bar{x}_{s+1}), \tag{3.21}$$

$$\ldots$$

$$0 = L\bar{x}_{m-1} + Vk(x_{m+1}) - L\bar{x}_m - Vk(\bar{x}_m). \tag{3.22}$$

Simplified derivation

Lévine and Rouchon observe that the derivation of a reduced compartment model can be done much simpler in less mathematical form: The trays $i \neq s$ have no holdup ($H_i \to 0$), and the aggregation tray s has the compartment holdup ($H_s = \sum_{i=1}^{m} H_i$).

This can be seen from equations (3.18) to (3.22) by subtracting all remaining equations from equation (3.20). Then, equation (3.20) can be replaced by equation

$$\bar{H}\dot{\bar{x}}_s = L\bar{x}_{s-1} + Vk(\bar{x}_{s+1}) - L\bar{x}_s - Vk(\bar{x}_s). \tag{3.23}$$

This is the same as shifting the compartment boundaries to the aggregation trays, as illustrated in figure 3.2.

The whole model reduction procedure can therefore be simplified to multiplying the left-hand sides of the aggregation tray equations with a constant $\gg 1$, and setting the remaining left-hand sides to 0. As it will be explained later, the scalar multiplying the left-hand side of the aggregation tray equation can be chosen arbitrarily without changing the steady-state of the model. Because of this and the fact that the compartment boundaries do not appear in the model anymore, the notion of "compartments" is not needed in the derivation of the method. It is sufficient to select some aggregation trays and assign them a large holdup.

This is somewhat surprising, since the original derivation starts with the transformation (3.10)-(3.11), where the compartment boundaries are clearly specified (here trays 1 and m). The reason that these compartment boundaries dissappear during the derivation is the treatment of the term $m\bar{x}_s - \sum_{i\neq s}\bar{x}_i$ in equation (3.14) and (3.16) while performing the quasi-steady-state approximation $\varepsilon \to 0$. Mathematically, since $m\bar{x}_s - \sum_{i\neq s}\bar{x}_i = x_s$, this term is invariant under the operation $\varepsilon \to 0$. If this term is left unchanged, the resulting model differs from the above model in equations (3.19) and

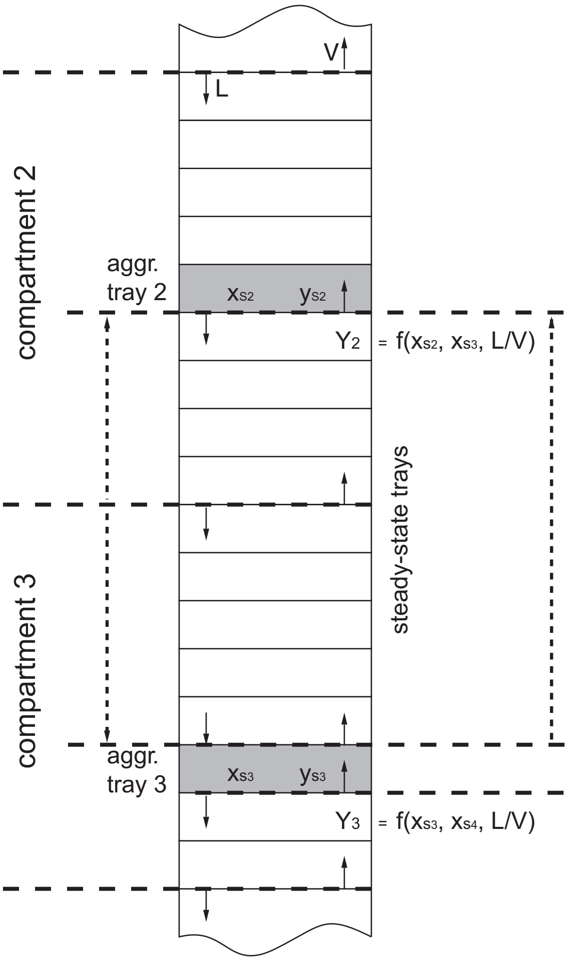

Figure 3.2: Schematic diagram of two compartments. Since the trays between two aggregation trays are in steady-state, balance boundaries can be moved arbitrarily between the aggregation trays without changing the balances.

(3.21), which then read

$$0 = L\bar{x}_{s-2} + Vk\Big(m\bar{x}_s - \sum_{i \neq s} \bar{x}_i\Big) - L\bar{x}_{s-1} - Vk(\bar{x}_{s-1}) \tag{3.24}$$

and

$$0 = L\Big(m\bar{x}_s - \sum_{i \neq s} \bar{x}_i\Big) + Vk(\bar{x}_{s+2}) - L\bar{x}_{s+1} - Vk(\bar{x}_{s+1}). \tag{3.25}$$

In this model, the compartment boundaries appear explicitly in the summation terms. This model variant is a “true” singular perturbation model in a sense that first the model is transformed into the standard form of singular perturbations, and then the quasi-steady-state assumption is applied. In the derivation of Lévine and Rouchon, an additional step simplifying the transformed model is taken. This, however, has major advantages: first, the derivation of the reduced model is greatly simplified, as described above. Secondly, the resulting reduced model retains the tridiagonal structure of the Jacobian of the original model. This is advantageous from a numerical point of view, and allows for the elimination of the algebraic equations from the model as will be described in section 3.2.3.

Figure 3.3 shows the response of both models to a step change in the feed concentration z_F. The “true” singular perturbation model shows an inverse response which is not present in the response of the full model. Except for this, the approximation quality of the “true” singular perturbation model is better than that of the model of Lévine and Rouchon. Both models yield the same steady-state values of the output variables. This is obvious from the facts that the right-hand side of the model of Lévine and Rouchon coincides with the right-hand side of the full model, and that the “true” singular perturbation model results from a variable transformation that leaves the top and bottom variables unchanged.

Because of the favorable structure of the model of Lévine and Rouchon, and the wrong inverse response of the “true” singular perturbation model, only the reduced model as derived by Lévine and Rouchon will be investigated in the rest of this paper. Instead of the term “compartment”, the term

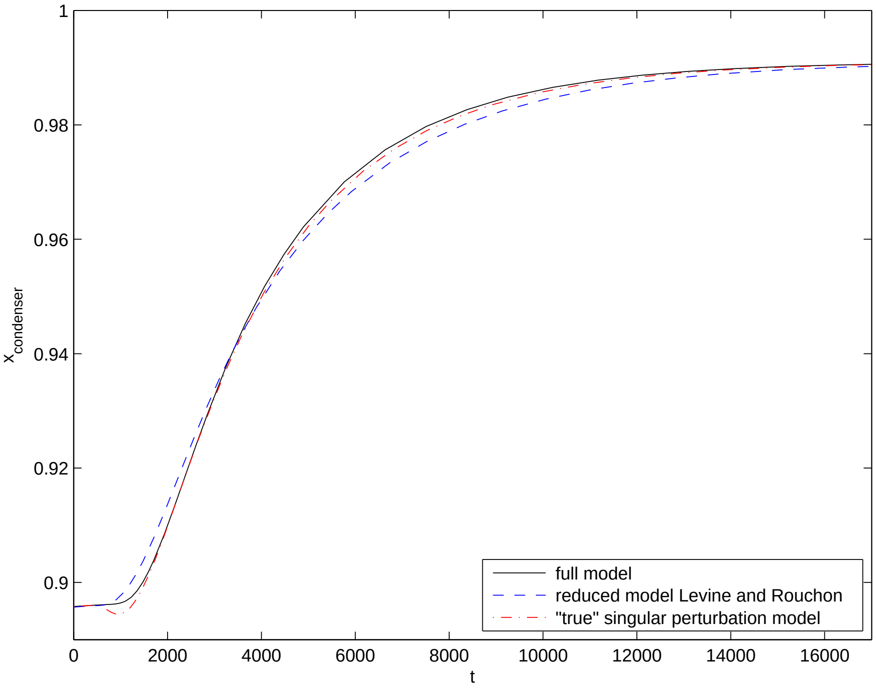

Figure 3.3: Responses of the top concentration of the full model, the model of Lévine and Rouchon (dashed line), and the “true” singular perturbation model (dash-dotted line) with five compartments to a step change in z_F from 0.45 to 0.55.

"aggregation tray" is used, since the method is basically "compartment-free", as explained above. Although the term "stage" is more general than "tray", the term "aggregation tray" is used throughout the paper to stay consistent with the terminology used by Lévine and Rouchon (1991).

Reduced model in DAE form

In this study, reduced models with between three and seven aggregation trays are used. The reboiler and condenser are always chosen as aggregation trays due to their large holdups. Applying the reduction method described above, the resulting DAE system for a reduced model with five aggregation trays reads

$$\begin{aligned}
\bar{H}_1\dot{\bar{x}}_1 &= Vk(\bar{x}_2) - V\bar{x}_1, && (3.26)\\
0 &= L\bar{x}_{i-1} + Vk(\bar{x}_{i+1}) - L\bar{x}_i - Vk(\bar{x}_i), && (3.27)\\
&\quad i = 2, ..., s_2 - 1, \\
\bar{H}_2\dot{\bar{x}}_{s_2} &= L\bar{x}_{s_2-1} + Vk(\bar{x}_{s_2+1}) - L\bar{x}_{s_2} - Vk(\bar{x}_{s_2}), && (3.28)\\
0 &= L\bar{x}_{i-1} + Vk(\bar{x}_{i+1}) - L\bar{x}_i - Vk(\bar{x}_i), && (3.29)\\
&\quad i = s_2 + 1, ..., s_3 - 1, \\
\bar{H}_3\dot{\bar{x}}_{s_3} &= L\bar{x}_{s_3-1} + Vk(\bar{x}_{s_3+1}) - (L+F)\bar{x}_{s_3} - Vk(\bar{x}_{s_3}) \\
&\quad +Fz_F, && (3.30)\\
0 &= (L+F)\bar{x}_{i-1} + Vk(\bar{x}_{i+1}) - (L+F)\bar{x}_i - Vk(\bar{x}_i), && (3.31)\\
&\quad i = s_3 + 1, ..., s_4 - 1, \\
\bar{H}_4\dot{\bar{x}}_{s_4} &= (L+F)\bar{x}_{s_4-1} + Vk(\bar{x}_{s_4+1}) - (L+F)\bar{x}_{s_4} - Vk(\bar{x}_{s_4}), && (3.32)\\
0 &= (L+F)\bar{x}_{i-1} + Vk(\bar{x}_{i+1}) - (L+F)\bar{x}_i - Vk(\bar{x}_i), && (3.33)\\
&\quad i = s_4 + 1, ..., N - 1, \\
\bar{H}_5\dot{\bar{x}}_N &= (L+F)\bar{x}_{N-1} - (L+F-V)\bar{x}_N - Vk(\bar{x}_N), && (3.34)
\end{aligned}$$

where $\bar{H}_j$ and s_j are the aggregated holdup and the index of aggregation tray j, respectively. The $\bar{x}$ notation is used to distinguish the variables in equations (3.26)-(3.34), which are partially algebraic and partially dynamic, from the purely dynamic variables in the full model (3.2)-(3.6).

Reduced model in ODE form

The equations of the steady-state trays between two aggregation trays contain only the algebraic variables of the steady-state trays, the state variables of the two aggregation trays, and either the ratio L/V or $(L+F)/V$. By writing the mass balance around each aggregation tray and all "steady-state" trays above it, the reduced model can be formulated such that the only algebraic variables appearing in the dynamic equations are the vapor concentrations of the trays directly below each aggregation tray, i.e. $y_{s_j+1}, j = 1, ..., N_c$ (figure 3.2). By pre-solving the systems of algebraic equations for these variables as a function of the dynamic variables and the ratio V/L, $y_{s_j+1} = Y_j(x_{s_j}, x_{s_{j+1}}, V/L)$, and by substituting them into equations (3.26)-(3.34), the DAE system can be cast in ODE form as

$$\begin{aligned}
\bar{H}_1\dot{\tilde{x}}_1 &= VY_1(\tilde{x}_1, \tilde{x}_2, V/L) - V\tilde{x}_1, && (3.35)\\
\bar{H}_2\dot{\tilde{x}}_2 &= L\tilde{x}_1 + VY_2(\tilde{x}_2, \tilde{x}_3, V/L) \\
&\quad -L\tilde{x}_2 - VY_1(\tilde{x}_1, \tilde{x}_2, V/L), && (3.36)\\
\bar{H}_3\dot{\tilde{x}}_3 &= L\tilde{x}_2 + VY_3(\tilde{x}_3, \tilde{x}_4, V/(L+F)) \\
&\quad -(L+F)\tilde{x}_3 - VY_2(\tilde{x}_2, \tilde{x}_3, V/L) + Fz_F, && (3.37)\\
\bar{H}_4\dot{\tilde{x}}_4 &= (L+F)\tilde{x}_3 + VY_4(\tilde{x}_4, \tilde{x}_5, V/(L+F)) \\
&\quad -(L+F)\tilde{x}_4 - VY_3(\tilde{x}_3, \tilde{x}_4, V/(L+F)), && (3.38)\\
\bar{H}_5\dot{\tilde{x}}_5 &= (L+F)\tilde{x}_4 - (L+F-V)\tilde{x}_5 \\
&\quad -VY_4(\tilde{x}_4, \tilde{x}_5, V/(L+F)). && (3.39)
\end{aligned}$$

This system is now an ODE system. The $\tilde{x}$ notation is used to distinguish the variables in equations (3.35)-(3.39) from the variables in the full model (3.2)-(3.6). The functions $Y_j, j = 1, ..., N_c$ can be calculated off-line and stored and later retrieved in a suitable way. In this study, look-up tables as described in section 3.2.7 are used for this purpose.

3.2.4 Model with composition controller

To stabilize the composition profile, it is common in distillation control to use the boil-up or reflux rate to control the temperature at a certain location inside the column. The setpoint can serve as a manipulated variable for a higher control or optimization layer. In this study, composition is controlled instead of temperature, which is equivalent for a binary mixture. If the boil-up rate is used, a proportional-integral control law for the concentration at the controlled tray c reads

$$V(t) = K_c(x_c(t) - x_c^s(t) + x_I(t)/T_I), \tag{3.40}$$

$$x_I(t) = \int_{-\infty}^{t} (x_c(\tau) - x_c^s(\tau))d\tau, \tag{3.41}$$

where K_c is the gain of the controller, x_c^s is the composition setpoint, x_I is the state of the integral part of the controller, and T_I is the integral time. This control law can be used directly in the full model equations (3.2)-(3.6) and the reduced models equations (3.26)-(3.34) and (3.35)-(3.39).
In this study, $K_c = 0.5$ and $T_I = 400$s were chosen using the Skogestad IMC tuning rules for PI-controllers (Skogestad, 2003) to achieve a closed-loop response of the controlled concentration with a time-constant of 100s, assuming an effective delay of 0s for the boil-up.

The reduced model can be further simplified when perfect composition control, that is

$$x_c = x_c^s, \tag{3.42}$$

where the setpoint is (piecewise) constant, $\dot{x}_c = \dot{x}_c^s = 0$, is assumed. Then, the equilibrium relation

$$0 = (L + F)x_{c-1} + Vk(x_{c+1}) - (L + F)x_c^s - Vk(x_c^s), \tag{3.43}$$

$$\Leftrightarrow V/(L + F) = \frac{(x_c^s - x_{c-1})}{k(x_{c+1}) - k(x_c^s)} \tag{3.44}$$

can be derived. Here, the liquid flow is $(L+F)$, because the controlled tray (c) is assumed to be in in the lower column section (below the feed).
Equations (3.42) and (3.44) can be used in the reduced DAE model (3.26)-(3.34) to eliminate one concentration and either the reflux rate L or boil-up rate V. If the controlled tray is chosen as an aggregation tray, its concentration variable will change from dynamic to algebraic, thereby reducing the

dynamic model order by one. Compared to the corresponding PI-controlled reduced model, the dynamic order is reduced by two, since the dynamic I-part of the controller is not needed.

The model can be converted again into ODE form as described in section 3.2.3 by solving the algebraic equations of the model off-line. For the part of the model containing the perfectly-controlled tray, a separate table containing

$$Y_c = Y_c(x_k, x_N, x_c^s) \tag{3.45}$$

and

$$V/(L + F) = f(x_k, x_N, x_c^s), \tag{3.46}$$

where k is the index of the aggregation tray above the controlled aggregation tray, is needed.

Figure 3.4 shows the responses of the PI-controlled model and the perfectly-controlled model, both with seven equally-sized compartments, to step changes in z_F and in x_c^s. While the PI-controlled model is closer to the full model response for the step change in z_F, it shows a fast response with large overshoot and succeeding oscillations for the setpoint change in x_c^s. Note that the PI-controller was not retuned, leading to a much faster response of the reduced system because of the neglected dynamics.

3.2.5 Stiffness analysis

An approximate stiffness analysis is done by linearising the uncontrolled full and a five-aggregation tray model around a steady-state for a constant z_F, and by comparing the largest and smallest eigenvalue λ of the linearised systems. Figure 3.5 shows the largest and smallest eigenvalues (plot a), and their ratio (plot b), of the full and the reduced model for the full range $0 \leq z_F \leq 1$.

It can be seen that the full model is quite stiff with a maximal eigenvalue ratio (condition number) at $z_F = 0.5$ of $7.7 \cdot 10^4$. The reduced system does not contain the fast modes of the full system, which is indicated by the fact that the absolute values of the smallest eigenvalues of the full system are

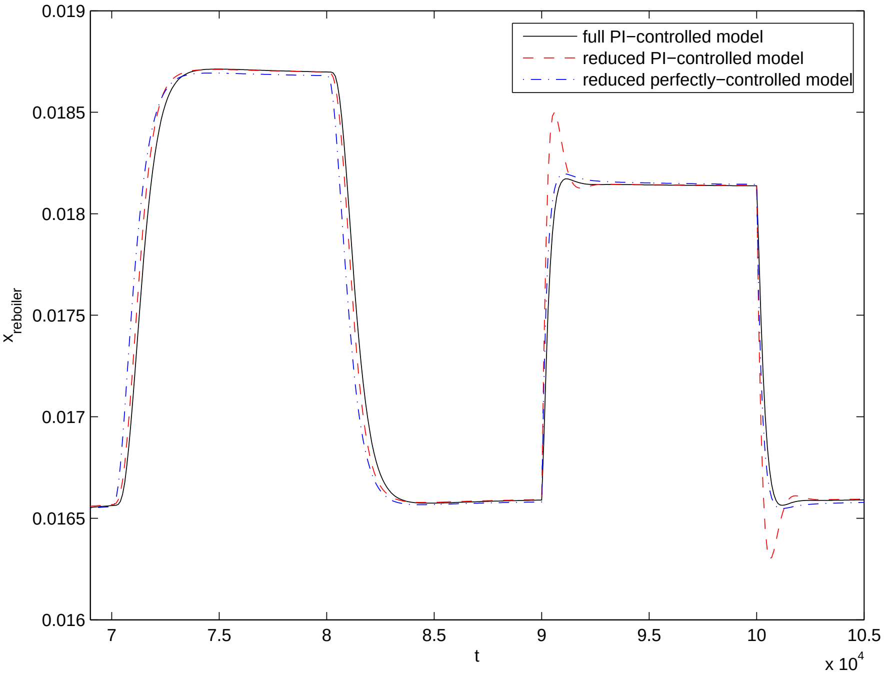

Figure 3.4: Responses of the reboiler concentration of the full PI-controlled model (full line), a reduced PI-controlled model (dashed line) and a perfectly-controlled model (dashed-dotted line), with seven and six aggregation trays, respectively, to a step change in z_F from 0.35 to 0.3 and back (left part of the plot), and in x_c^s from 0.105 to 0.1125 and back (right part of the plot). While the PI-controlled model shows a more accurate response to the change in z_F than the perfectly-controlled model, it displays a too large overshoot as response to the change in x_c^s.

about 300 times larger than of the reduced system. The maximal eigenvalue ratio of the reduced system is therefore reduced to 198, which indicates that there is still some stiffness present in the system. This remaining stiffness is created by a sharp minimum for the smallest eigenvalue at $z_F = 0.5$, which is present in both full and reduced system. This phenomenon is described by Skogestad (1987). However, this slowest time-scale is actually not interesting for control, because a temperature or composition controller will cause the system to move to steady-state much faster. Therefore, for the numerical performance only the fastest eigenvalue, which dictates the step size of the integration, is of interest.

3.2.6 Choice of reduced model parameters

To the knowledge of the authors, no systematic procedure to determine compartment parameters that are optimal for a certain application is available. Khowinij et al. (2005) report that the accuracy of the compartmental models is not strongly affected by the location of the aggregation trays. They place the aggregation trays at the middle of the respective compartments. In order to investigate the influence of the reduced model parameter choice on the accuracy of the reduced model, a straightforward choice and an optimized choice of parameters are compared in this study.

It is worth noticing that although requiring that the sum of the aggregation tray holdups is equal to the sum of the tray plus condenser and reboiler holdups is physically meaningful, it is not necessary for obtaining dynamically functional models. Any choice of the aggregation tray holdups (that leads to a stable model) will yield a model that approaches the same steady-state as the full model. Thus the holdups can be used as degrees of freedom to optimize the dynamic behavior of the reduced model. In this study, however, models are restricted to the "physical" choice of aggregation tray holdups such that the reduced column model has the same holdup as the full column model.

For the study of the uncontrolled model, two versions of models with five and seven aggregation trays are compared: a straightforward version with equally-sized and spaced aggregation trays, and an optimized version, where

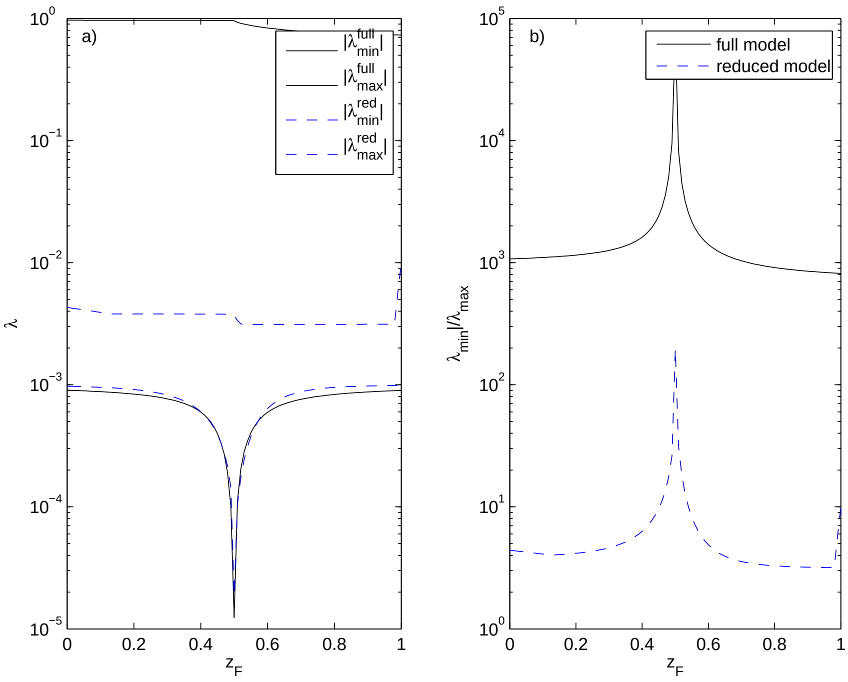

Figure 3.5: Plots of the absolute values of the largest and smallest eigenvalues (plot a) of the uncontrolled full and a five-aggregation tray reduced model linearised around a constant z_F, and of their ratios (plot b).

the parameters are determined by an optimization procedure to minimize the reduced model error.
In all models, the feed tray was chosen to be an aggregation tray. In this way, the input appears only in the dynamic part of the model, but not in the algebraic equations.
With this choice, the three-aggregation tray model is given, because there is no freedom of choice for the aggregation tray parameters: condenser, reboiler, and the feed tray are the aggregation trays. For the study of the controlled model, the location of the controlled tray was fixed to be 62, and the last aggregation tray before the reboiler was positioned there as well, allowing a straightforward implementation of the controller. The parameter values for the five- and seven-aggregation tray models used in this study are given in table 3.2.6.

3.2.7 Implementation details of reduced models

The steady-state functions $Y_j(x_{s_j}, x_{s_{j+1}}, V/L)$ were calculated off-line and stored in a look-up table with 30x40x200 entries, and the functions
$Y_c(x_{s_l}, x_{reboiler}, x^s_{i_c})$ and $V/(L + F) = f(x_{s_l}, x_{reboiler}, x^s_{i_c})$ in tables with 60x20x200 entries. From the tables, function values are obtained by applying three-dimensional interpolation. In this study, multi-linear and multi-cubic interpolation was used (Press et al., 2007). Investigations on the table interpolation accuracy showed deviations in the 4th digit (less than 1%) for linear interpolation and in the 6th digit (less than 0.01%) for cubic interpolation. The table formats are the outcome of an optimization procedure, where the minimum average interpolation error was identified among possible entry number combinations all yielding the same table size. The obtained optimal combinations reflect the fact that the function shows stronger curvature in some dimensions than in others. The function $Y_j(x_{s_j}, x_{s_{j+1}}, V/L)$ is almost linear in the first variable, because of the dominant influence of the liquid concentration from above on the function value.

aggregation tray index j	1	2	3	4	5	6	7
uncontrolled models:							
3 aggregation trays							
s_j	1	36	74				
$\bar{H}$	20	72	20				
optimized							
s_j	1	14	36	60	74		
$\bar{H}$	20	21	28	23	20		
s_j	1	8	20	36	53	67	74
$\bar{H}$	20	10	15	19	18	10	20
equally-sized							
s_j	1	13	36	62	74		
$\bar{H}$	20	24	24	24	20		
s_j	1	8	22	36	53	67	74
$\bar{H}$	20	14	14	16	14	14	20
controlled models:							
optimized							
s_j	1	13	36	62	74		
$\bar{H}$	20	21	45	10	20		
equally-sized							
s_j	1	13	36	62	74		
$\bar{H}$	20	24	24	24	20		
s_j	1	8	22	36	53	62	74
$\bar{H}$	20	14	14	16	14	14	20

Table 3.2: Positions and holdups of the aggregation trays of the reduced models.

3.3 Framework for evaluation of reduced model performance

In order to assess the computational performance of the different models, simulation runs were performed, where the models were simulated with different solvers at varying simulation tolerances. The simulation time for a given model and solver depends on the simulation tolerance, which also influences the simulation error compared to the exact solution (section 3.3.3). The performance of the models can therefore only be compared by considering both simulation time and simulation accuracy.

In this study, only the open-loop performance of the various models was assessed. This means that the input signals (z_F, L and V; or z_F, L and x_c^s for the controlled column) to the system were fixed at predetermined profiles during the simulations.

The input profile was selected with the aim of exposing the system to step changes in the disturbance and control variables over a wide range of operating conditions. The simulation accuracy was measured using the average two-norm of the output deviations from the exact solution.

3.3.1 Selection of input profile

The input profiles for z_F, L and V to test the performance of the uncontrolled models are shown in figure 3.6. They include changes of two different magnitudes in the feed concentration z_F and the reflux rate L and vapor flow V. The two latter are changed simultaneously, such that their difference is constantly half of the feed flow F. This kind of simultaneous change is likely to appear in closed-loop optimal control of a high-purity column, because then the split is approximately constant (Shinskey, 1984). Nevertheless, we see from the simulations that the variations in product compositions (x_D, x_B) are large, because there is no compensation for changes in the feed composition (z_F). In the case of the controlled model, the same input signal was used, but the boil-up rate V was determined by the composition controller. The variation in product composition is here much less (figure 3.7). The time interval between changes in each variable

was chosen to 10^5s (about 3 hours).
To take into account fast and slowly varying input signals, both a stepwise changes and slowly changing cubic B-spline signals have been used (Schlegel, 2005). The cubic B-splines yield a continuously differentiable input signal, where the rate of change form one stationary value u_t to the next u_{t+1} is determined by a time constant δ. The total transition time is 2δ with a maximum slope of $(u_{t+1} - u_t)/\delta$. A value of 3000s was chosen for δ.
The input profiles for z_F, L and x_c^s for the controlled column are shown in figure 3.7. Since the setpoint of the composition controller can be used as an additional manipulated variable for control, two sets of input profiles, one with step changes in z_F and L (plot a), and one with step changes in z_F and x_c^s were used (plot b).

3.3.2 Selection of accuracy measure

As a measure for model accuracy, the condenser and reboiler concentrations, which are the most important variables for control, are evaluated by taking the average of the 2-norm of the deviations from the exact solution (see 3.3.3) over the whole simulation period:

$$\varepsilon = \frac{1}{t_{end}} \int\limits_{0}^{t_{end}} \sqrt{\left(x_{condenser}^{exact} - x_{condenser}\right)^2 + \left(x_{reboiler}^{exact} - x_{reboiler}\right)^2} dt, \quad (3.47)$$

where $x_{condenser}$ and $x_{reboiler}$ are the condenser and reboiler concentrations, respectively, of either the full or the reduced model, and of the exact solution.

3.3.3 Simulation details

Solvers

In this study, the following four solvers have been used to simulate the full model and the different versions of the reduced models:

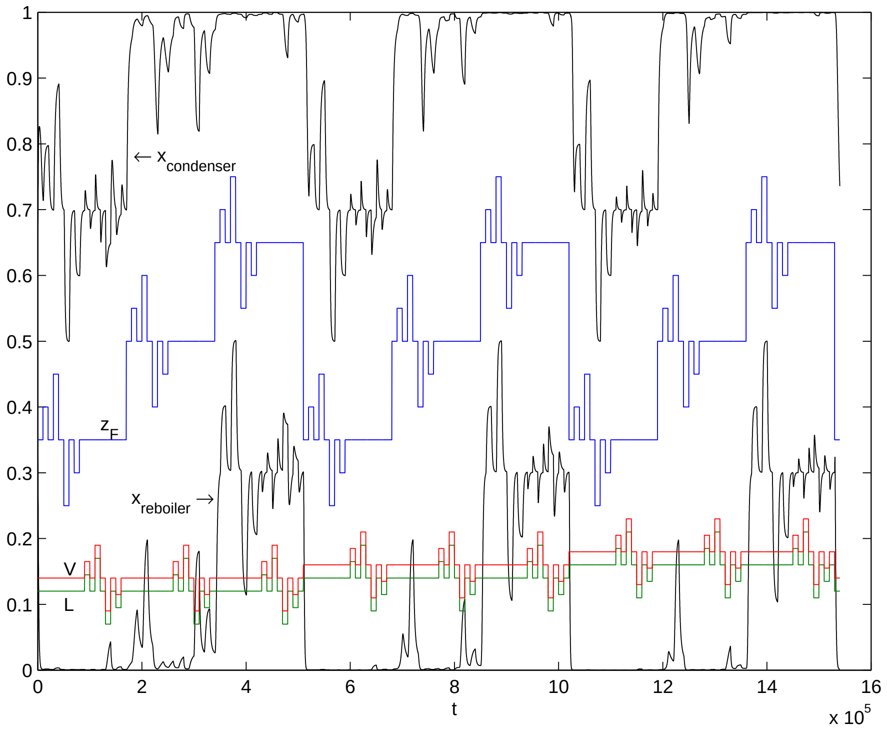

Figure 3.6: Step input signals used for simulating the *uncontrolled* models and output trajectories of the full model to this input. The stepwise changing lines are the feed concentration z_F, the reflux L and the boil-up range V. V is always 0.02 higher that L. The curves on top and bottom are the concentrations of condenser and reboiler, respectively.

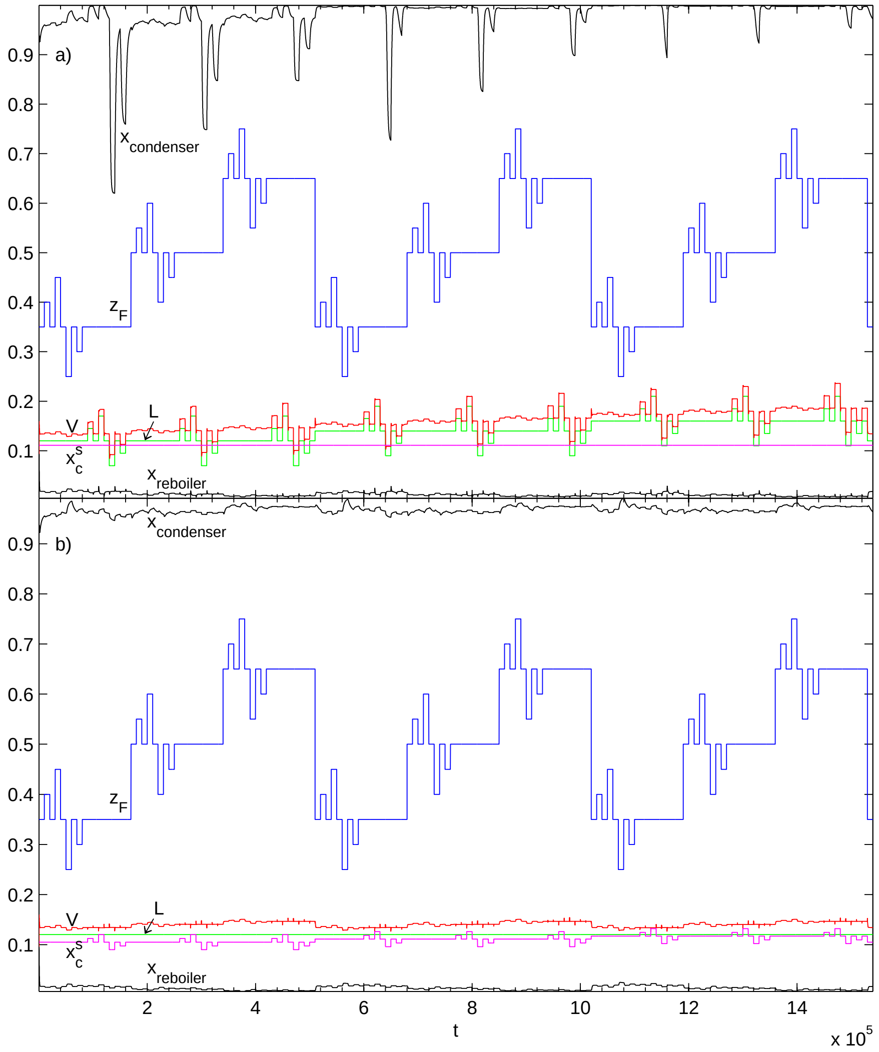

Figure 3.7: Step input signals used for simulating the *controlled* models and output trajectories of the full model to this input. Plot a) shows the input signal where z_F and V is varied, and plot b) shows the input signal where z_F and x_c^s is varied. The curves on top and bottom of both plots are the concentrations of condenser and reboiler, respectively. In both cases, V is manipulated by the composition controller.

- LIMEX: This solver is an extrapolation scheme for the solution of linearly-implicit differential-algebraic systems (Hairer and Wanner, 2002, Ehrig and Nowak, 2000). In this study, version 4.2A by Ehrig and Nowak (2000) for systems with dense or banded Jacobian was used. With this solver, all full and reduced model versions have been simulated. This solver is attractive for the use in real-time optimization, because of its good restarting qualities for problems involving frequent discontinuities (Brenan et al., 1996). It is used, for example, in the real-time optimization software DyOS (Schlegel et al., 2005, Würth et al., 2007).

- RADAU: This solver is an implicit Runge-Kutta scheme with coefficients based on Radau IIA quadrature formulas (Hairer and Wanner, 2002). The code used in this study is the classic 5th-order code RADAU5 by Hairer and Wanner (1996). It is suitable to solve stiff equations and was used to integrate the full model and the reduced models in ODE form.

- DOPRI: This solver is an embedded explicit Runge-Kutta scheme based on the Dormand-Prince pair for step control (Hairer et al., 2000). The code used in this study is the classic 5th-order code DOPRI5 by Hairer and Wanner (1996). It is a solver for non-stiff equations and was used to integrate the reduced models in ODE form. This is possible because of the reduced stiffness of the reduced models. The same integration scheme is used in the MATLAB solver ode45 (Shampine and Reichelt, 1997). The higher-order code DOPRI853 has also been tested, but showed comparable performance to the 5th-order solver and was therefore not included into the detailed study.

- DASPK: This solver is an implicit multistep integrator based on the backward differentiation formula (Ascher and Petzold, 1998). Version 3.0 by Li and Petzold (2000) was used in this study. It was used to integrate the full and the reduced controlled models in this study. A similar integration scheme is used in the MATLAB solver ode15s

(Shampine and Reichelt, 1997).

All solvers provide dense output, that is, interpolated solution values can be obtained at any desired point within the simulation period (Hairer at al., 2000). The LIMEX and DASPK codes were restarted after every change in the input signal. The RADAU and DOPRI codes were only started once at the beginning of a simulation, because restarting did not have any effect on their execution time.

The numerical properties of the controlled and the uncontrolled models are significantly different. The reason for this is a change of the Jacobian structure, which is caused by the composition controller. Figure 3.8 shows the Jacobian structure of the full controlled model. The two vertical columns of entries on the right side of the plot reflect the influence of the state of the controlled tray (left column) and the I-state of the PI-controller (right column) on the equations of each tray via the vapor flow V. In contrast to this, the Jacobian of the uncontrolled model is purely tridiagonal because of the absence of the composition controller. This has important numerical consequences. The Jacobian of the uncontrolled models and the solution of the linear equations arising from the implicit integration schemes using this Jacobian can be done numerically efficient by exploiting the tridiagonally banded structure of the Jacobian. The Jacobian of the controlled models is, however, no longer banded. This implies that the calculation of a numerical Jacobian and the linear algebra is much slower. To avoid this loss in performance, the Jacobian for the controlled models was calculated analytically, and a specialized solver for the linear algebra which takes into account both banded structure and vertical lines in the Jacobian was used. To simulate the controlled systems, the solver DASPK (Li and Petzold, 2000) was used, since this solver allows for the straightforward use of external linear algebra routines. For all solvers, integration tolerances $AbsTol = RelTol$ in the range 10^{-1} to 10^{-5} were used.

Calculation of "exact" solution

As “exact” solutions, simulations with very tight simulation tolerances (10^{-13}) were performed, using the RADAU code for the uncontrolled mod-

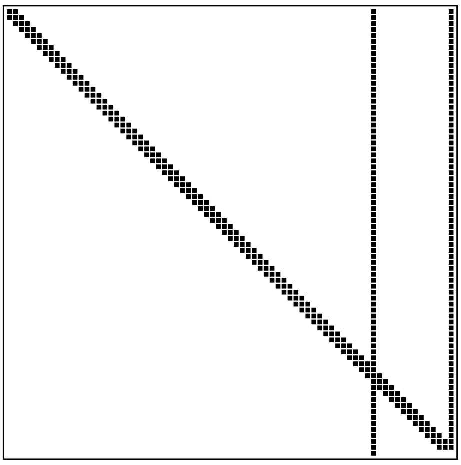

Figure 3.8: Plot of the Jacobian structure of the full controlled model.

els and the DASPK code for the controlled models.

3.4 Results

3.4.1 Accuracy vs. simulation time

Uncontrolled models and general observations

Figures 3.9 and 3.10 show the performance of the uncontrolled full and different optimized reduced models with linear table interpolation in combination with different solvers, simulated at varying tolerances, for the "slow" and the "fast" input signals, respectively. For better readability, the plots were split into three subplots, each of which shows the results for one solver. For the "fast" input signal, the following observations can be made:

- The reduction error limits the best achievable accuracy of the reduced models. As expected, these accuracies increase with increasing reduced model order. For crude simulation tolerances (low accuracy simulations), the accuracy of the reduced models is dominated by the simulation error. For tight simulation tolerances, the reduction error dominates and no further gain in accuracy can be achieved by increasing the simulation accuracy. Typically, for the LIMEX solver, tighter tolerances than 10^{-3} will mostly increase the simulation time, but not the accuracy.

- While the reduced models in DAE form do not show any improvement in computational speed, the reduced models in ODE form can be simulated much faster. This is explainable with the fact that a general-purpose DAE solver works basically like a stiff ODE solver (Hairer and Wanner, 2002), where the algebraic equations are treated similar to fast dynamic equations. This means that there is no computational gain to be expected from converting dynamic equations into algebraic equations, although the use of specialized sparse DAE solvers and control algorithms might change this picture (Bian et al., 2005, Khowinij et al., 2005).

- For every model and solver, the accuracy curve intersects at a certain accuracy with the curve of the model of the next lower order. From this accuracy on downwards, it is computationally favorable to use the model of lower order. However, the seven-aggregation tray model is still performing well at low accuracies in some cases (LIMEX solver), such that it could be a more robust choice when there is doubt about the proper choice of simulation accuracy.

- Compared to the full model at the same accuracy, the highest achievable gain in simulation speed is between eight and ten.

For the "slow" input signal, similar observations can be made, but the performance of the individual solvers is different and the maximum gain in simulation speed is not as significant as in the stepwise case.

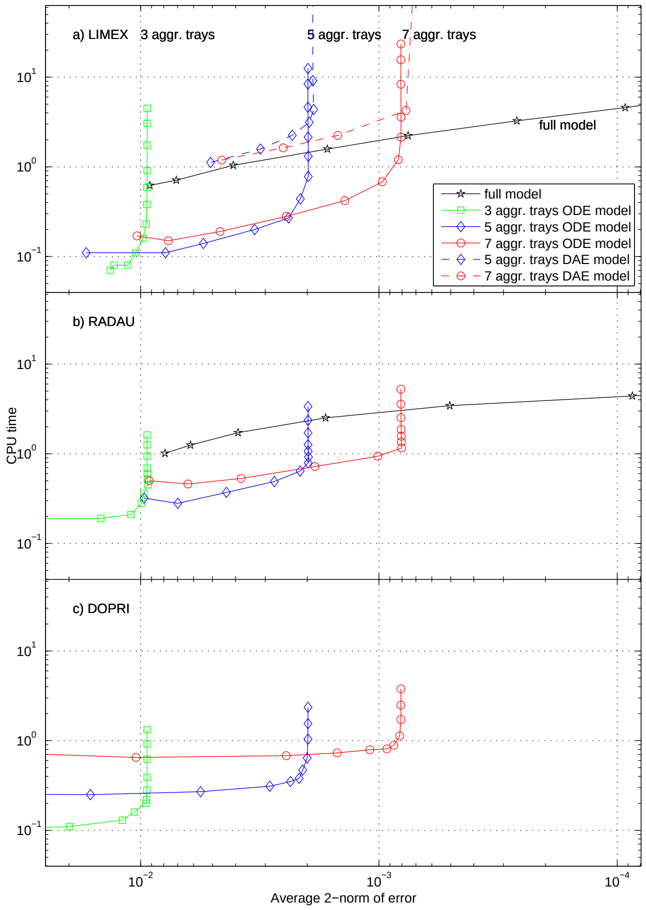

Figure 3.9: Comparison of the performance of the uncontrolled models, simulated with various solvers and at various simulation tolerances, and with a stepwise input signal (figure 3.6). Part a) shows simulations with the LIMEX solver, b) with the RADAU solver and c) with the DOPRI solver. Simulations of the same model with different simulation tolerances are connected with a line. Full lines are used for ODE models, and dashed lines for DAE models. In all simulations, linear table interpolation was used. Due to its high stiffness, the full model could not be simulated with the DOPRI-solver.

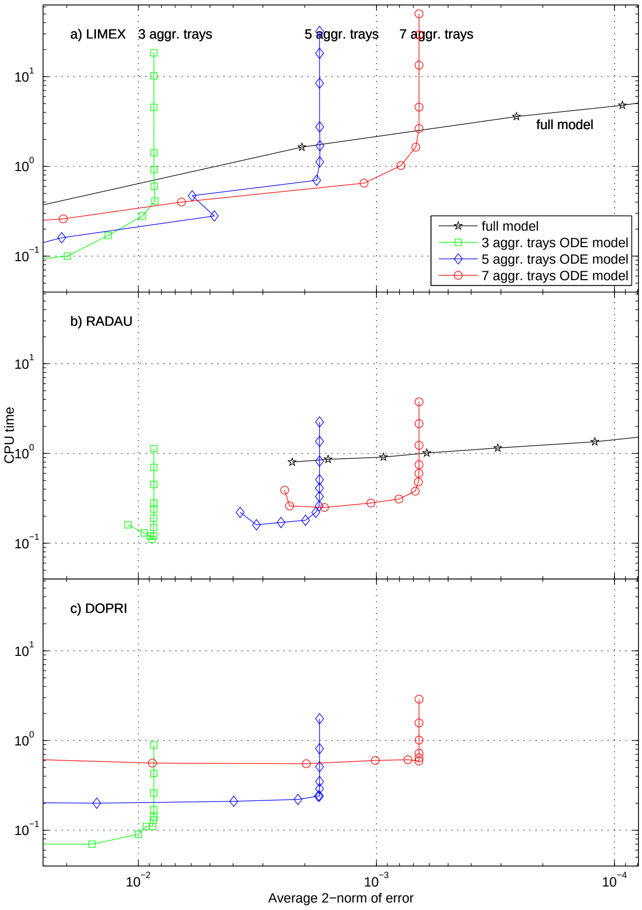

Figure 3.10: Comparison of the performance of the uncontrolled models, simulated with various solvers and at various simulation tolerances, and with the “slow” input signal.

Controlled models

Figure 3.11 shows the performances of four reduced models for stepwise input changes in z_F and L (part a), and in z_F and x_c^s (part b). The reduced models are models with five and seven equally-sized aggregation trays, respectively, an optimized five-aggregation tray model, and a perfectly-controlled model with six aggregation trays, where the reboiler and the controlled aggregation tray are merged together.

The following observations can be made:

- The general trends of the performance of the controlled model are the same as for the uncontrolled models.
- The optimal five-aggregation tray model has a lower reduction error than the equally-sized model, but shows slightly worse performance at lower accuracies for changes in z_F and L.
- The perfectly-controlled model shows slightly better performance than the seven-aggregation tray model at higher accuracies. The situation reverses at lower accuracies.
- At intermediate accuracies, the gain in computational speed is roughly of factor six.

Figure 3.12 shows numerical statistics of the simulation runs for the full, the PI-controlled seven-aggregation tray model, and the perfectly-controlled model with six aggregation trays. The numerical behavior of the PI-controlled model is very similar to that of the full model. The perfectly-controlled model shows significantly lower numbers of steps for intermediate and tight tolerances, and accordingly low numbers for the other values as well. The number of residuum calculations is increased, since the Jacobian is calculated numerically. A possible explanation for this behavior is that the perfect model does not include the comparably fast dynamics of the composition controller in the bottom, allowing the integrator to take larger integration steps.

Figure 3.13 shows the percentage of simulation time that is spent in the

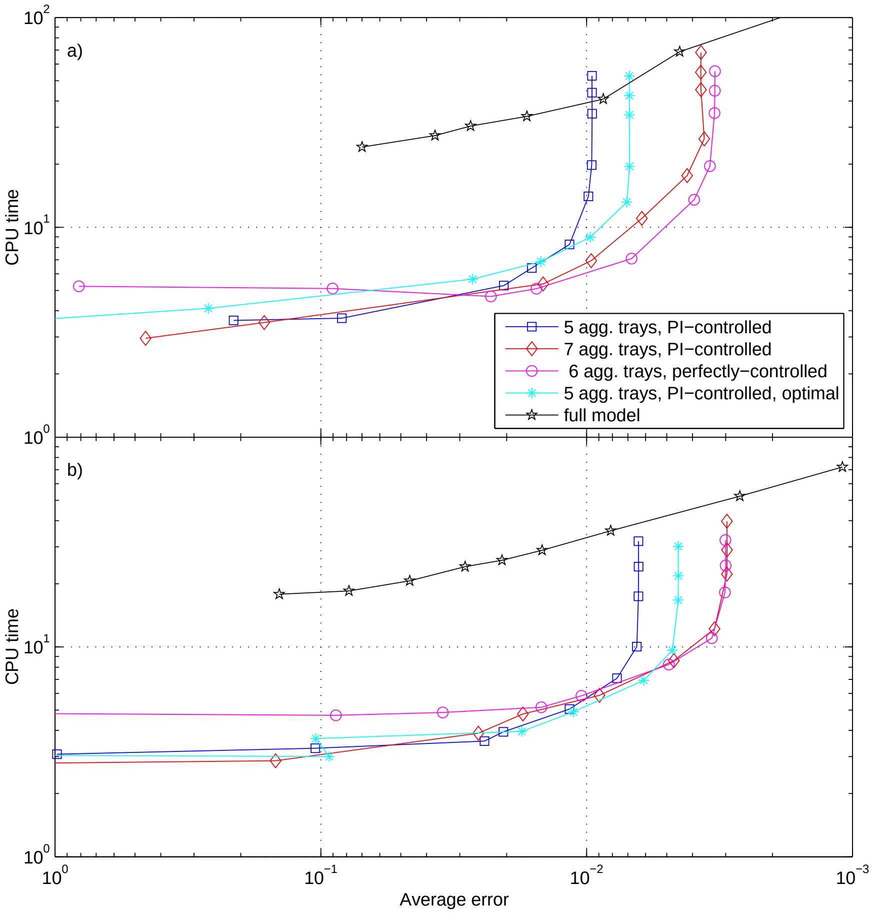

Figure 3.11: Comparison of the performance of the full and four reduced controlled models, simulated with various simulation tolerances and input signals with step changes in z_F and L (a), and in z_F and x_c^s (b) (figure 3.7).

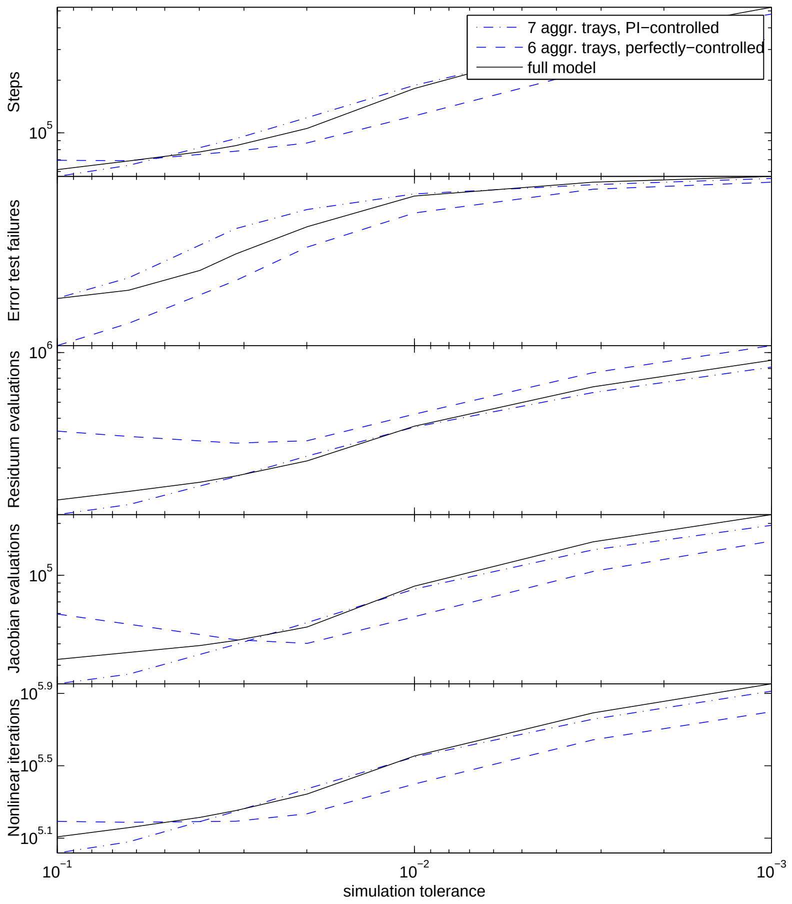

Figure 3.12: Comparison of the numerical behavior of the full, the seven-aggregation tray PI-controlled reduced model, and the six-aggregation tray perfectly-controlled model.

computationally most intensive parts of the integration for the full, the seven-aggregation tray model and the perfectly-controlled model with six aggregation trays. For the full model, the functions for the solution of the linear equations dominate the simulation time. For the reduced model, residual and Jacobian evaluation time dominate. This is because of the computationally expensive look-up table interpolation that is used in the reduced models, and the reduction of the number of dynamic variables of the models. The perfectly-controlled model is simulated with numerically calculated Jacobians, therefore the time of the Jacobian calculation is included in the residual calculation time. The remaining 40-50% of the simulation time is spent in other parts of the solver. In the reduced models, the time spent for the linear look-up table interpolation amounts to about half of the time spent for the residual and Jacobian computations, which is about 20% of the total simulation time.

3.4.2 Choice of compartments

In table 3.3, the reduction error of the reduced model variants are shown. The error of the optimized models is roughly between 50% and 80% of the models with equally-sized aggregation trays. For higher-order reduced models, the ratio gets better in favor of the optimized variants, reflecting the increased number of degrees of freedom for optimization.

3.4.3 Solver performances

It has been observed before that the LIMEX solver performs well in simulations with low accuracy, while it is usually outperformed by other solvers at high accuracies (Schlegel, 2005). The same tendency can be observed in this study when comparing the full model being simulated with LIMEX and RADAU codes, respectively.

For the uncontrolled reduced models at low accuracies, in case of a stepwise input signal, the LIMEX solver shows a performance that is far superior to that of the RADAU solver. The DOPRI code shows comparable performance to the LIMEX code for high accuracies. While the LIMEX solver

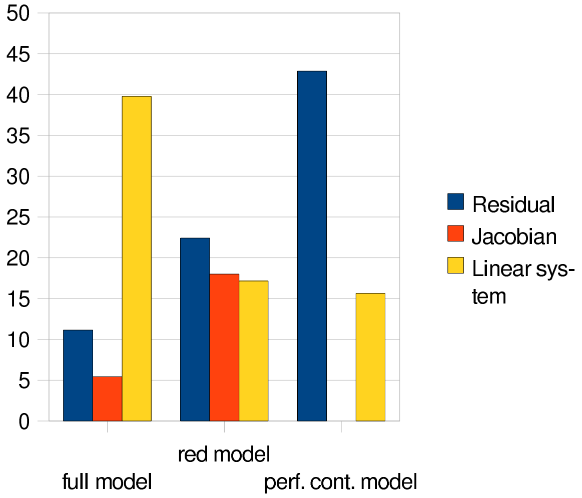

Figure 3.13: Comparison of the percentage of simulation time of the computationally most intensive parts of the integration for the full, the seven-aggregation tray model and the perfectly-controlled model with six aggregation trays.

Table 3.3: Reduction errors of the reduced models for input signals with step changes in z_F and L. Column A gives the number of aggregation trays, column B gives the choice of position and size of the aggregation trays (equally-sized or optimal), column C gives the type of interpolation used (linear or cubic), and column D gives the type of concentration control (PI or perfect control).

A	B	C	D	reduction error
uncontrolled models				
3		lin.		9.55 1e-3
3		cub.		9.33 1e-3
5	eqs.	lin.		2.57 1e-3
5	opt.	lin.		2.00 1e-3
5	eqs.	cub.		2.45 1e-3
5	opt.	cub.		1.91 1e-3
7	eqs.	lin.		1.35 1e-3
7	opt.	lin.		8.13 1e-4
7	eqs.	cub.		1.26 1e-3
7	opt.	cub.		7.08 1e-4
controlled models				
4	eqs.	lin.	pc	6.07 1e-3
4	eqs.	cub.	pc	5.34 1e-3
5	eqs.	lin.	PI	9.54 1e-3
5	opt.	lin.	PI	6.09 1e-3
5	eqs.	cub.	PI	8.09 1e-3
5	opt.	cub.	PI	5.23 1e-3
6	eqs.	lin.	pc	3.29 1e-3
6	eqs.	cub.	pc	2.44 1e-3
7	eqs.	lin.	PI	3.72 1e-3
7	eqs.	cub.	PI	2.64 1e-3

shows a gradual trade-off between accuracy and simulation time, the DOPRI solver rather abruptly looses performance when the simulation accuracy is not close to the maximal achievable accuracy. During simulations with crude tolerances, the DOPRI solver usually finds the problem to be stiff. This is probably the cause for the loss in performance.

In the "slow" input case, the performance of the RADAU solver is superior to the other solvers. A possible explanation is that the advantage of the LIMEX code when applied to discontinuous input signals is not visible in this case. The DOPRI code again looses performance abruptly because of the problem becoming stiff for crude simulation tolerances.

The performance of the DASPK solver as the only solver used for integrating the controlled models cannot be compared to the other solvers. However, the same gradual trade-off between accuracy and simulation time as with the LIMEX solver for the uncontrolled models can be observed.

3.5 Discussion

3.5.1 Reduction method

It was shown in section 3.2.3 that due to a simplification step in the original derivation of Lévine and Rouchon (1991), the reduction method is a very simple procedure that can be applied without specifying any compartments, but only aggregation trays. Furthermore, it was shown that the algebraic equations resulting from the quasi-steady-state approximation can be eliminated from the model, resulting in significantly more compact reduced models that can be simulated several times faster than the original model. The main bottleneck of the method lies in the second step to convert the DAE system into an ODE system. The off-line computed solutions of the algebraic equations have to be stored and retrieved as functions of the state variables of the aggregation trays encompassing the steady-state trays in a suitable way. In this study, this was done in a rather straightforward way by tabulating the relevant solutions and retrieve them by multi-dimensional interpolation of the table values. Since both table size and interpolation intensiveness increases exponentially with the number of independent variables, a high number of independent variables will lead to computationally expensive right-hand sides of the reduced models. In this study, the time spent for the linear look-up table interpolation is about 20% of the total simulation time. With every additional independent variable, this number will double, in case of linear interpolation, and increase four times, in case of cubic interpolation. The gain in computational performance will therefore depend on how computationally intensive the table interpolation is compared to the computational intensiveness of the residual and Jacobian calculation of the full model.

3.5.2 Relationship to singular perturbation methods

As described in section 3.2.3, the model reduction method was originally derived as a singular perturbation method. However, since an additional modification of the equations is performed, the derivation is different from a standard singular perturbation approach, where the model is taken into the standard form of singular perturbation by some state transformation, and then a quasi-steady-state approximation is performed (Kokotovic, 1986). In section 3.2.3, these "true" singular perturbation systems have been compared the reduced models by Lévine and Rouchon (1991). Their reduced models still show a typical "slow model" behavior. This means that the model is getting the more accurate, the closer it is to steady-state. The initial response to a fast input change is typically faster than the response of the original model, because the fast dynamics of the original model are replaced by "instantaneous dynamics" by the quasi-steady-state approximations in the reduced model. The method of Lévine and Rouchon can therefore be described as a "singular perturbation-related" method. The physical interpretation of the method is also slightly different: Instead of exploiting the time-scale separation between slow compartments and fast trays, the method works by making the signal transport between the aggregation trays infinitely fast, and slowing the dynamics down again by increasing the time-constants of the aggregation trays.

3.5.3 Validity of investigated system

The distillation model used in this study is very simple. The constant molar holdup assumption is generally not a good assumption for the initial response to a change in reflux and for control purposes (Skogestad, 1997), but is acceptable in this study where only one control loop in the bottom is closed. The model is, however, well suited to demonstrate the application and performance of the model reduction method. It can be expected that the obtained results possess some predictive quality for more complex cases, since the dynamics and time-constants are similar.

3.5.4 Extension and generalization of reduction method

The simple derivation of reduced models presented in this paper makes it straightforward to extend the reduction method to more complex distillation models, and to generalize it to general one-dimensionally distributed systems.

In more realistic distillation models, each tray comprises a total mass balance (tray hydraulics), an energy balance, and possibly several component balances. These can be treated in the same way as the single component mass balance equations used in this study. The resulting algebraic equations are treated in the same way as well. However, the number of independent variables, which determine the solution of the algebraic equations, is larger than in the models in this study, yielding more complex functions that have to be approximated. Indeed, the reduction method has been successfully applied to a binary distillation column with energy balances, complex thermodynamic and hydraulic relationships. These results will be published separately.

Furthermore, it is possible to generalize the reduction method to general one-dimensionally spatially distributed systems, such as plug-flow reactors or heat exchangers. If the system consists of discrete units, such as the trays in a distillation column, or a model resulting from the discretization of a partial differential equation, the reduction method can be applied in exactly the same way. A detailed description of this method is in preparation.

3.5.5 Computational performance of reduced models

It has been found that for a given accuracy a gain in simulation speed of a factor five to ten can be achieved using the reduced models proposed in this study. The accuracy measure is the integrated deviation from a reference trajectory, when the model is exposed to step changes in the inputs, where the interval between two step changes is long enough to allow the system to come close to steady-state again. However, the performance of a reduced model will strongly depend on the application. In model predictive control, for example, the model will be exposed to step changes in the inputs at intervals of a few hundred seconds. Since the reduced models presented here are relatively slow models, their response in the initial period after an input change is possibly rather inaccurate (compare figure 3.3). However, the reduced models might be well suited for an application that emphasizes the long-term behavior of the system, such as in optimal trajectory planning. This is confirmed by a study by Würth et al., 2007, who found that reduced aggregated models yield significant savings in computation time for dynamic real-time optimization.

3.5.6 Tabulation of steady-state tray solutions

Multi-linear and multi-cubic interpolation has been used in this study to obtain continuous function values from the look-up tables of the steady-state tray solutions. Multi-linear interpolation is recommended because multi-cubic interpolation will be computationally expensive for a high number of independent variables as can be expected for models with more components. For a low number of independent variables, an increased accuracy (see table 3.3) can instead be achieved by multi-linear interpolation with a higher table resolution.

3.5.7 Comparison with other model reduction methods for distillation columns

Besides tray aggregation, wave propagation theory (Hankins, 2007, Kienle, 2000, Marquardt, 1990) and orthogonal collocation methods (Cho and

Joseph, 1983, Dalouti and Seferlis, 2006) have been proposed as alternative methods for deriving reduced distillation models. Wave propagation approximates the system dynamics by traveling wave profiles in response to input changes. Kienle (2000) reports very good dynamic and steady-state approximation qualities of the wave models, assuming constant molar flow rates, holdups and pressure in the system. Hankins (2007) presents extensions of the method to systems with energy balances and hydraulic effects. Khowinij et al. (2004) compare the performance of a wave model to that of a compartment model. They report that the approximation quality of the wave model is rather poor compared to that of the compartment model, which, however, might be due to the different assumptions incorporated in the two models.

Orthogonal collocation approximates the state variable profile as sums of orthogonal polynomials. For this, a discrete column model is first transformed into continuous partial differential equations. By enforcing the approximating profile to satisfy the differential equations at a number of collocation points, discrete dynamic equations are obtained. The approximation quality of collocation methods depend on the choice and number of collocation points. Dalaouti and Seferlis (2006) apply the approach to complex staged reactive separation processes. Their reduced models show good approximation quality, while a gain in simulation speed of several times is reported.

Of the three methods, tray aggregation is conceptually probably the simplest. However, the implementation requires some effort, as is described in this study. In terms of accuracy, each method has its strengths and weaknesses. The advantage of tray aggregation compared to the other methods is the perfect match of steady-states, making it interesting for applications where the long-term behavior is important. Wave propagation, on the other hand, seems to yield a good approximation of initial system responses to fast changing inputs. Collocation methods are probably the most widely used methods to obtain reduced distillation models. Since the method can be applied to complex systems without many restrictive assumptions, the accuracy is suitable for most applications. Of the three methods, wave propagation is computationally probably the least intensive. However, a detailed comparison of the three methods with careful implementations taking the application of the models into account has yet to be done in order to assess their strengths and weaknesses for that particular application.

3.6 Summary and conclusions

In this study, the tray aggregation method of Lévine and Rouchon (1991) has been used to derive reduced models of a binary distillation column model. It has been shown that the original derivation, which partitions the column into “compartments” of consecutive trays, can be simplified to a procedure which needs only the specification of aggregation trays and is independent of the choice of compartment boundaries. This gives the resulting reduced model a favorable numerical structure and makes the extension to more complex models straightforward. The original method yields a DAE model with the same number of variables as the original model. Because of the banded Jacobian structure of the reduced model, the algebraic equations of the DAE models can be eliminated from the model by off-line solution and substitution of the tabulated solution values as functions of the dynamic variables into the dynamic equations, yielding low-order ODE models. The numerical performance of these models has been assessed by simulations using four different numerical integrators. While simulation of the DAE models showed no gain in simulation speed, the reduced models yielded an improvement of up to factor ten. Due to the reduced stiffness of the models, a non-stiff integrator can be used for integration. The inclusion of a controller in the bottom section of the column has been investigated. While the application of a PI-controller to the reduced model is easily possible, the controller can also be directly incorporated into the reduced model assuming perfect control. Both variants show good performance, decreasing simulation time about six times.

The computational gains that can be achieved by the reduced models form make these kind of models attractive for use in real-time optimization applications. In more complex models, the performance of the reduced models will depend on the complexity of the right hand sides as well as the number of independent variables for the tabulation of the steady-state tray solutions.

Acknowledgement

This work has been supported by the European Union within the Marie-Curie Training Network PROMATCH under the grant number MRTN-CT-2004-512441.

Bibliography

[1] Allgöwer F., & Zheng, A. (2000). Nonlinear Model Predictive Control. Progress in Systems Theory, 26. Basel: Birkhäuser.

[2] Ascher, U. M., & Petzold, L. R. (1998). Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. Philadelphia: SIAM.

[3] Benallou, A., Seborg, D. E., & Mellichamp, D. A. (1986). Dynamic Compartmental Models for Separation Processes. AIChE J., 32, 1067.

[4] Bian, S., Khowinij, S., Henson, M. A., Belanger, P., & Megan, L. (2005): Compartmental modeling of high purity air separation columns. Comput. Chem. Eng., 29, 2096.

[5] Brenan, K. E., Campell, S. L., & Petzold, L. R. (1996). Numerical Solution of Initial-Value Problems in Differential-Algebraic Equations. Philadelphia: SIAM.

[6] Cho, Y. S., & Joseph, B. (1983). Reduced-Order Steady-State and Dynamic Models for Separation Processes. Part I. Development of the Model Reduction Procedure. AIChE Journal, 29, 261.

[7] Dalaouti, N., & Seferlis, P. (2006). A unified modeling framework for the optimal design and dynamic simulation of staged reactive separation processes. Comput. Chem. Eng., 30, 1264.

[8] Ehrig R., & Nowak, U. (2000). Konrad-Zuse-Zentrum für Informationstechnik Berlin, http://www.zib.de.

[9] Hairer, E., Nørsett, S. P., & Wanner, G. (2000). Solving Ordinary Differential Equations I - nonstiff problems. Berlin: Springer.

[10] Hairer, E., & Wanner, G. (2002): Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems. Berlin: Springer.

[11] Hairer, E., & Wanner, G. (1996). http://www.unige.ch/∼hairer/software.html.

[12] Hankins, N. P. (2007). A non-linear wave model with variable molar flows for dynamic behavior and disturbance propagation in distillation columns. Chem. Eng. Res. Des., 85, 65.

[13] Khowinij, S., Henson, M. A., Belanger, P., & Megan, L. (2005). Dynamic compartmental modeling of nitrogen purification columns. Sep. Purif. Technol., 46, 95.

[14] Khowinij, S., Bian, S., Henson, M. A., Belanger, P., & Megan, L. (2004). Reduced Order Modeling of High Purity Distillation Columns for Nonlinear Model Predictive Control, Proc. Am. Contr. Conf., Boston, Massachusets, 4237.

[15] Kienle, A. (2000). Low-order dynamic models for ideal multicomponent distillation processes using nonlinear wave propagation theory. Chem. Eng. Sci., 55, 1817.

[16] Kokotovic, P., Khalil, H. K., & O'Reilly, J. (1986). Singular Perturbation Methods in Control: Analysis and Design. SIAM classics in applied mathematics 25, London: SIAM.

[17] Kumar, A., & Daoutidis, P. (2003). Nonlinear model reduction and control for high-purity distillation columns. Industrial and Chemistry Research, 42, 4495.

[18] Lévine, J., & Rouchon, P. (1991). Quality Control of Binary Distillation Columns via Nonlinear Aggregated Models. Automatica, 27, 463.

[19] Li, S., & Petzold, L. R. (2000). Software and Algorithms for Sensitivity Analysis of Large-Scale Differential Algebraic Systems, J. Comput. Appl. Math., 125, 131.

[20] Marquardt, W. (1990). Traveling waves in chemical processes. Int. Chem. Eng., 30, 585.

[21] Marquardt, W. (2001). Nonlinear model reduction for optimization based control of transient chemical processes. Proc. CPC VI, 30.

[22] Press, W. H., Teukolsky, S. A., Vetterling, W. T., & Flannery, B. P. (2007). Numerical Recipes: The Art of Scientific Computing, Third Edition. Cambridge: Cambridge University Press.

[23] Schlegel, M. (2005). Adaptive discretization methods for the efficient solution of dynamic optimization problems. Fortschritt-Berichte VDI, 829, Düsseldorf: VDI-Verlag.

[24] Schlegel, M., Stockmann, K., Binder, T., & Marquardt, W. (2005). Dynamic optimization using adaptive control vector parameterization. Comput. Chem. Eng., 29, 1731.

[25] Shampine, L. F., & Reichelt, M. W. (1997). The MATLAB ODE Suite. SIAM J. Sci. Comput., 18, 1.

[26] Shinskey, F. G. (1984). Distillation Control For Productivity and Energy Conservation. 2nd Edition, New York: McGraw-Hill.

[27] Skogestad, S., & Morari, M. (1987). The dominant time constant for distillation columns. Comput. Chem. Eng., 11, 607.

[28] Skogestad, S. (1997). Dynamics and Control of Distillation Columns: A Critical Survey. Modeling, Identification and Control, 18, 177.

[29] Skogestad, S. (2003). Simple analytic rules for model reduction and PID controller tuning, J. Proc. Contr., 13, 291.

[30] van den Berg, J. (2005). Model Reduction for Dynamic Real-Time Optimization for Chemical Processes. PhD Thesis, TU Delft.

[31] Würth, L., Linhart, A., Preisig, H., & Marquardt, W. (2007). Performance of reduced distillation models in dynamic real-time optimization. ECCE, Copenhagen.

Chapter 4

Reduced distillation models via stage aggregation

submitted to Chemical Engineering Science.

Abstract

A method for deriving computationally efficient reduced nonlinear distillation models is proposed, which extends the aggregated modeling method of Lévine and Rouchon (1991) to complex models. The column dynamics are approximated by a low number of slow dynamic aggregation stages connected by blocks of steady-state stages. This is achieved by simple manipulation of the left-hand sides of the differential equations. In order to reduce the computational complexity of the resulting DAE system, the algebraic equations resulting from the reduction procedure are replaced by interpolation in tables or polynomial approximations. The resulting reduced model approximates the original dynamic model very accurately, and increases the simulation speed several times. This makes the reduced models interesting for real-time applications. The numerical properties of the models and possible improvements are discussed.

Keywords

Model reduction; Distillation; Aggregated modeling; Mathematical modeling; Dynamic simulation

4.1 Introduction

This study describes and analyzes a model reduction method for staged distillation column models. Reduced models are used for system analysis and controller design, and for speeding up simulations. The latter is much desired for model predictive control (Allgöwer and Zheng 2000, Qin and Badgwell, 2003) and dynamic real-time optimization (Schlegel, 2005) applications. Numerous model reduction methods for linear (Antoulas, 2005) and nonlinear systems (Marquardt, 2001, van den Berg, 2005) have been described in literature.

The model reduction method presented in this work extends the aggregated modeling method of Lévine and Rouchon (1991). The original method was developed as an improvement of the compartmental modeling method of Benallou et al. (1986), and used a very simple distillation model assuming constant stage holdups, constant molar flows and constant relative volatility. When applying the method, the column is partitioned into compartments, each of which comprises a number of consecutive stages. The dynamics of a compartment is approximated by one dynamic stage, which is assigned the total compartment holdup, while the remaining stages of the compartment are treated as in steady-state. This is achieved by a state transformation for the compartment variables, and a subsequent quasi-steady-state approximation of the fast equations of the transformed system. This approach has been used recently by Khowinij et al. (2004, 2005), and Bian et al. (2005) to derive reduced models of a distillation column with variable stage holdups, with the objective of obtaining reduced models that increase the simulation speed. They conclude that a tailor-made DAE solver is necessary to significantly speed up the simulations.

However, it was shown by Linhart and Skogestad (2009) that applying the original method in combination with an ordinary DAE solver does not increase the simulation speed. The reason for this is that the method converts the majority of the dynamic equations of the full model into algebraic equations, which does not change the overall size of the system. Since in a DAE solver, dynamic and algebraic equations are treated very similarly (Ascher and Petzold, 1998, Hairer and Wanner, 2002), no gain in computation speed can be expected. Alternatively, Linhart and Skogestad (2009) show that the algebraic equations resulting from the reduction procedure can be eliminated from the model due to the banded Jacobian structure of the reduced model, which yields a much smaller ODE or DAE system. This model now yields a significant improvement in computational performance. In addition, it was shown that the method of Lévine and Rouchon can be interpreted to be basically compartment-free. This means that only the dynamic aggregation stages have to be specified, but no partition of the column into compartments is necessary.

The model reduction method presented in this study extends the original method of Lévine and Rouchon (1991) and the extension of Khowinij et al. (2005) and Bian et al. (2005) in the following aspects:

1. The notion of "compartments" is abandoned; only the specification of "aggregation stages" is necessary. Their "holdup" is a free tuning parameter of the reduced model.

2. The method can be applied to all kinds of staged processes with mass and energy balances, and complex hydraulic and thermodynamic relationships. No simplification of the original system prior to the model reduction procedure is needed.

3. The algebraic equations resulting from the reduction procedure are eliminated from the reduced model and replaced by functions, which are substituted in the dynamic aggregation stage equations. This is the crucial step to obtain computationally superior reduced models.

4. The physical interpretation of the reduction principle is different. Instead of comparing the different time-scales of the slow compartment

dynamics and fast stage dynamics as in the original method of Lévine and Rouchon (1991), the method works by making signal transport through the steady-state stages infinitely fast, and slowing down the column dynamics by assigning large "aggregated holdup factors " to the aggregation stages. Although related, the method is no true singular perturbation method (Kokotovic et al., 1986, Linhart and Skogestad, 2009).

This paper is organized as follows. In section 4.2, a full distillation model that is used to demonstrate the reduction method is introduced. Important structural and implementation issues of the model are discussed. Section 4.3 describes the derivation of the reduced model from the full model. In a first step, by a simple manipulation of the left-hand sides of the differential equations of the full column, a reduced model of the same size as the original model is obtained. In a second step, the resulting algebraic equations are eliminated from the model and replaced by more efficient approximations such as table interpolations. As the second step is crucial for the performance of the reduced model, this part is described in more detail. In section 4.4, the approximation quality and computational performance of the reduced model is investigated. For this purpose, simulations with fast changes in the input variables of the models over a range of simulation tolerances are performed. The accuracy of the reduced models is compared with the original model, and is set into relation with the simulation speed. An analysis of the numerical behavior and of the distribution of computational complexity in the models and the solver is given. Finally, the advantages and disadvantages of the stage aggregation method, possible improvements and applications, and a brief comparison to other model reduction methods for distillation models are discussed in section 4.5.

4.2 Full model

4.2.1 System and modeling assumptions

The distillation column used in this study to demonstrate the reduction method is a high-purity distillation column with 92 stages, a reflux drum with a total condenser, and a reboiler. The case-study model in this study uses a binary mixture, but the model description and the model reduction procedure is for a multi-component mixture. Ideal stages with perfect mixing and vapor-liquid equilibrium on each stage are assumed.

4.2.2 Mathematical description

For notational convenience, the reflux drum and reboiler are written as stages 1 and N, respectively.
For a mixture with N_c components, the state of each stage is described by $N_c + 1$ dynamic variables: M_i^{tot} (total mole number on stage i), $\mathbf{M}_i$ (vector of $N_c - 1$ component moles on stage i), and U_i^{tot} (total internal energy on stage i). Since the sum of the N_c components gives the total holdup M_i^{tot}, this formulation is equivalent to including all N_c components in the $\mathbf{M}_i$ vector. The dynamic evolution of each state is governed by a differential balance equation. In addition, there is a large number of algebraic equations, including thermodynamic relationships for the vapor-liquid equilibrium.

Dynamic balance equations

On each stage, $N_c + 1$ balance equations can be formulated. The balance equations for the stages except the reboiler, condenser and feed stage ($2 \leq i \leq N-1, i \neq i_F$) read

$$\begin{aligned}
\dot{M}_i^{tot} &= L_{i-1} + V_{i+1} - L_i - V_i, && (4.1)\\
\dot{\mathbf{M}}_i &= L_{i-1}\mathbf{x}_{i-1} + V_{i+1}\mathbf{y}_{i+1} - L_i\mathbf{x}_i - V_i\mathbf{y}_i, && (4.2)\\
\dot{U}_i^{tot} &= L_{i-1}h_{i-1}^L + V_{i+1}h_{i+1}^V - L_i h_i^L - V_i h_i^V - Q_i^{hl}. && (4.3)
\end{aligned}$$

The balance equations for the feed stage i_F read

$$\begin{aligned}
\dot{M}_{i_F}^{tot} &= L_{i_F-1} + V_{i_F+1} - L_{i_F} - V_{i_F} + F, && (4.4)\\
\dot{\mathbf{M}}_{i_F} &= L_{i_F-1}\mathbf{x}_{i_F-1} + V_{i_F+1}\mathbf{y}_{i_F+1} - L_{i_F}\mathbf{x}_{i_F} - V_{i_F}\mathbf{y}_{i_F} + F\mathbf{z}_F, && (4.5)\\
\dot{U}_{i_F}^{tot} &= L_{i_F-1}h_{i_F-1}^L + V_{i_F+1}h_{i_F+1}^V - L_{i_F}h_{i_F}^L - V_{i_F}h_{i_F}^V - Q_{i_F}^{hl} && (4.6)\\
&\quad +Fh_F.
\end{aligned}$$

The balance equations for the reflux drum with total condenser ($i = 1$) read

$$\begin{aligned}
\dot{M}_1^{tot} &= V_{top} - (R + D), && (4.7)\\
\dot{\mathbf{M}}_1 &= V_{top}\mathbf{y}_2 - (R + D)\mathbf{x}_1, && (4.8)\\
\dot{U}_1^{tot} &= V_{top}h_2^V - (R + D)h_1^L + Q_c. && (4.9)
\end{aligned}$$

The balance equations for the reboiler ($i = N$) read

$$\begin{aligned}
\dot{M}_N^{tot} &= L_{N-1} - B - V_N, && (4.10)\\
\dot{\mathbf{M}}_N &= L_{N-1}\mathbf{x}_{N-1} - B\mathbf{x}_N - V_N\mathbf{y}_N, && (4.11)\\
\dot{U}_N^{tot} &= L_{N-1}h_{N-1}^L - Bh_N^L - V_Nh_N^V + Q_r. && (4.12)
\end{aligned}$$

The variables used in the above equations are explained in tables 4.1 and 4.2. Note that $\mathbf{M}_i$, $\mathbf{x}_i$, $\mathbf{y}_i$ and $\mathbf{z}_F$ are vectors of length $N_c - 1$, except in the binary case, where they are scalars.

Algebraic relations for sum of phases

The intensive variables for the individual phases $\mathbf{x}_i$, $\mathbf{y}_i$, h_i^L, h_i^V must satisfy some algebraic relations, since the sum of the phases makes up the total holdup. The sum of the total mass, component masses, energy and volume of the phases on stage i can be written as

$$\begin{aligned}
M_i^{tot} &= M_i^L + M_i^V, && (4.13)\\
\mathbf{M}_i &= M_i^L\mathbf{x}_i + M_i^V\mathbf{y}_i, && (4.14)\\
U_i^{tot} &= M_i^Lh_i^L + M_i^Vh_i^V - p_iV_i, && (4.15)\\
V_i &= M_i^Lv_i^L + M_i^Vv_i^V, && (4.16)
\end{aligned}$$

Table 4.1: Full model variable description.

Variable	Description
i	stage index
i_F	index of feed stage
M_i^{tot}	total mole number
M_i	total mole number of component 1
U_i^{tot}	total internal energy
x_i	liquid concentration of component 1
y_i	vapor concentration of component 1
h_i^L	liquid enthalpy
h_i^V	vapor enthalpy
L_i	liquid outflow
V_i	vapor outflow
V_{top}	vapor flow from top stage into reflux drum
D	liquid distillate outflow of reflux drum
B	liquid bottom product outflow of reboiler
R	reflux flow out of reflux drum
F	feed flow into feed stage
z_F	concentration of component 1 in feed
h_F	feed enthalpy
Q_c	heat flow into condenser
Q_r	heat flow into reboiler
Q_i^{hl}	stage heat loss from to environment

Table 4.2: Full model variable values.

Variable	SI-Unit	typical values		
		stage	reflux drum	reboiler
i				
i_F				
M_i^{tot}	mol	3500	49000	100000
M_i	mol	3300	45500	1500
U_i^{tot}	kJ	-5.7·10^4	-1.0·10^6	-1.7·10^6
x_i		0.34	0.94	0.015
y_i		0.40	0.96	0.019
h_i^L	kJ/mol	-18	-22	-18
h_i^V	kJ/mol	1.2	1.6	1.8
L_i	mol/s	500		
V_i	mol/s	500		500
V_{top}	mol/s		425	
D	mol/s		55	
B	mol/s			100
R	mol/s		370	
F	mol/s	155		
z_F		0.34		
h_F	kJ/mol	-22		
Q_c	kW		-9500	
Q_r	kW			10000
Q_i^{hl}	kW	2.4		

where V_i is the total volume of stage i (which is assumed constant), and M_i^L and M_i^V are the stage liquid and vapor masses, respectively. To reduce the number of algebraic equations that need to be solved by the DAE solver, equations (4.13) and one of equations (4.14)-(4.16) can be solved to obtain explicit expressions for M_i^L and M_i^V. The simplest choice is to combine equations (4.13) and (4.16) to get

$$\begin{aligned} M_i^L &= (V_i - M_i^{tot} v_i^V)/(v_i^L - v_i^V), && (4.17) \\ M_i^V &= M_i^{tot} - M_i^L, && (4.18) \end{aligned}$$

where v_i^L and v_i^V are the specific volumes of liquid and vapor phase, respectively. The remaining N_c algebraic equations that need to be solved are then (4.14) and (4.15).

Algebraic thermodynamic relationships

The vapor-liquid equilibrium on stage i gives N_c algebraic relations (one for each component):

$$f_i^L(\mathbf{x}_i, p_i, T_i) = f_i^V(\mathbf{y}_i, p_i, T_i). \qquad (4.19)$$

In this study, the thermodynamic quantities f_i^L, f_i^V, $\mathbf{x}_i$, $\mathbf{y}_i$, h_i^L, h_i^V, v_i^L, v_i^V, ρ_i^L and ρ_i^V are obtained by means of the Soave-Redlich-Kwong equations of state (Reid et al., 1997).

Algebraic hydraulic relationships

The liquid flows L_i are calculated by means of a modified Francis weir equation (Green and Perry, 2007)

$$L_i = \gamma \rho_i^L |l_i/\beta - h_{W,i}|^{1.5}, \qquad (4.20)$$

where β and γ are geometry-dependent factors, l_i the liquid level, and $h_{W,i}$ is the weir height of stage i, respectively.
The vapor flows V_i are calculated by

$$V_i = \gamma \sqrt{|p_i - p_{i-1} - \rho_{i-1}^L g l_{i-1}| \rho_i^V}, \qquad (4.21)$$

where g is the standard gravity.

Table 4.3: Base-layer PI-controllers.

Controller	CV	MV
Level controller reflux drum	l_1	D
Pressure controller top stage	p_2	V_{top}
Temperature controller stage 76	T_{76}	Q_r
Level controller reboiler	l_N	B

Algebraic equations for heat loss

The heat loss of a tray to the environment is modeled by a linear heat transfer equation

$$Q_i^{hl} = \alpha_i(T_i - T_{environment}), \tag{4.22}$$

where α_i is the heat transduction coefficient through the outer wall of stage i. The heat loss is frequently neglected ($\alpha_i = 0$) in distillation modeling.

Algebraic equations for condenser cooling

The cooling of the condenser is modeled as

$$Q_c = -V_{top}(h_2^V + \beta), \tag{4.23}$$

where β an adjustable parameter.

Dynamic equations for controllers

The column is stabilized by four base-layer PI-controllers. The controllers with their controlled variables (CV) and manipulated variables (MV) are listed in table 4.3.

4.2.3 Alternative strategies for solution of the algebraic equations

As mentioned, there are $(N_c+1)\cdot N$ dynamic balance equations, where N_c is the number of components and N is the number of stages in the column. The associated dynamic state variables on stage i are M_i^{tot}, $\mathbf{M}_i$ (vector of length N_c-1) and U_i^{tot}. In addition, there is a large number of algebraic equations which are generally not explicit in the dynamic state variables (M_i^{tot}, $\mathbf{M}_i$, U_i^{tot}), and therefore need to be solved. Several approaches for solving the algebraic equations are possible:

- **Approach 1.** The algebraic equations are solved separately at each evaluation of the derivatives of the dynamic state variables (right hand side of dynamic balance equations). This is in general not numerically efficient.

- **Approach 2.** The differential and algebraic equations (DAEs) are solved simultaneously using a DAE solver. Here, one generally tries to minimize the number of algebraic equations and associated algebraic state variables used in the DAE solver by finding the lowest number of algebraic equations that must be solved to make the rest of algebraic equation set explicit. In most cases, the algebraic vapor-liquid equilibrium (VLE) relations (4.19) are explicit in the variables pressure p, temperature T, liquid composition $\mathbf{x}$ (vector with N_c-1 independent variables) and vapor composition $\mathbf{y}$ (vector with N_c-1 independent variables). In total, this gives $2N_c$ algebraic state variables. Thus, on each stage the $2N_c$ algebraic equations given in (4.14), (4.15) and (4.19) (for p, T, $\mathbf{x}$, $\mathbf{y}$) need to be included in addition to the N_c+1 differential equations (for M_i^{tot}, $\mathbf{M}_i$, U_i^{tot}).

- **Approach 3.** Solve the algebraic equations off-line and represent the solution in terms of functions of suitable independent variables. Note that one can use the same functions for all stages in the column.

  There are many possibilities for the choice of the independent variables:

- **3.1 Use the dynamic state variables also for functions:** A straightforward choice are the $N_c + 1$ dynamic state variables (M_i^{tot}, $\mathbf{M}_i$, U_i^{tot}). Then, the algebraic equations can be solved offline to generate the required functions in terms of these $N_c + 1$ variables. However, this set of variables may not be the best, since high-dimensional functions with independent variables which can assume values on a large domain are difficult to implement.
- **3.2 Introduce special algebraic variables for functions:** It is desirable to minimize the number of variables to reduce the dimension of the functions. From the Gibbs phase rule, it is actually sufficient to specify N_c (rather than $N_c + 1$) intensive variables for a system in vapor-liquid equilibrium. For the binary mixture in this study ($N_c = 2$), pressure p and temperature T are chosen as independent variables.

4.2.4 Final DAE equation set for full model

In this study, a binary mixture is considered and approach 3.2 with T and p (on each stage) as algebraic state variables is used. On each of the N stages, the DAE set includes three differential equations (see equations (4.1)-(4.12)), plus one algebraic equation for the sum of phases holdup of component 1 (4.14) and one algebraic equation for the sum of phases internal energy (4.15). The 5 associated state variables X_i on each stage are

$$X_i = \{M_i^{tot}, M_i, U_i^{tot}, p_i, T_i\}. \tag{4.24}$$

Note that for a binary mixture, M_i is the scalar holdup of component 1. In addition to these $3N$ dynamic and $2N$ algebraic equations, the full DAE model has one dynamic equation for each controller with integral action. The resulting full set of equations solved by the DAE solver can be written in the form

$$M\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}, \mathbf{u}), \tag{4.25}$$

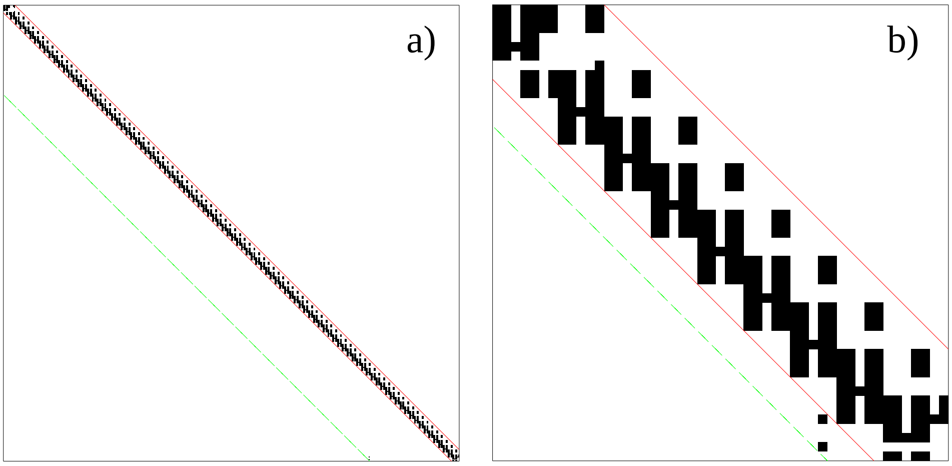

Figure 4.1: Jacobian structures of the full (plot a) and reduced (plot b) models. Shown are the dependencies of the right-hand sides **F** on the states **X**. The solid lines mark the width of the non-zero Jacobian elements, when the elements corresponding to the temperature controller are excluded. The dashed lines mark the width when these elements are included.

where **X** are the $5N + 4$ state variables used by the DAE solver, **u** is an input vector, and M is the diagonal mass matrix with a 1 on the diagonal for a differential equation and a 0 for an algebraic equation.

The remaining algebraic equations, including equations (4.17) and (4.18), and also the flash equations which are represented by tables, are explicit in **X** and are solved at each evaluation of the right hand side $\mathbf{F}(\mathbf{X}, \mathbf{u})$.

4.2.5 Jacobian structure

The DAE set for the full model described by (4.25) in section 4.2.4 is highly structured as can be seen from figure 4.1a, which shows the Jacobian structure ($d\mathbf{F}/d\mathbf{X}$) of the full model. The Jacobian is basically a banded matrix. However, the temperature controller in the bottom section, which has influence on the temperature of stage 76 in the bottom section, introduces elements into the Jacobian, which correspond to the proportional and integral action of the controller and which lie outside the narrow band. By this, the width of the Jacobian band is increased several times. The level and pressure controllers at the top of the column also increase the width of the Jacobian band, but to a much less extent, as the manipulated and controlled variables are positioned spatially close to each other. The system including all controllers except the temperature controller in the bottom section has a Jacobian non-zero entry band of width 19. If the temperature controller is included in the system with the temperature measurement located at stage 76 (19 stages from the bottom), the width is increased to 94. The special structure of the Jacobian has to be taken into account for efficiently solving the linear equations arising during the integration of the model. This is described in section 4.2.6.

4.2.6 Implementation of full model

System size

In the present case, with $N = 94$ stages in the column, the full differential-algebraic model contains $94 \cdot 3$ dynamic and $94 \cdot 2$ algebraic variables for the stages, and 4 dynamic variables for the states of the PI-controllers, adding up to a total of 474 variables in the state vector $\mathbf{X}$.

Numerical solution

For simulation, the DAE solver DASPK 3.0 (Li and Petzold, 2000) was used. This solver is implemented in FORTRAN 77. The residual $\mathbf{F}$ and the analytic Jacobian $d\mathbf{F}/d\mathbf{X}$ of the model were programmed in C-code. This ensures a fast implementation for a realistic evaluation of the computational performance.

Tabulation of thermodynamic properties

The thermodynamic VLE relations and property relations were programmed as two-dimensional lookup tables. From these 8 tables, the thermodynamic quantities x, y, h^V, h^L, v^V, v^L, ρ^V and ρ^L are obtained as functions of T and p by cubic spline interpolation (Press et al., 2007) of the table entries. Each table has 1 000 x 1 000 entries, where 273 K $< T <$ 350 K and 1 bar $< P <$ 8 bar.

LU-decomposition of Jacobian

The Jacobian is evaluated analytically. Each time the Jacobian is recomputed, it is decomposed into a lower and an upper triangular matrix. For this, a modified banded Gaussian LU-decomposition is used. The LINPACK (LINPACK, 1978) routine DGBFA as used in DASPK to LU-decompose a banded matrix was modified to work with the narrow banded Jacobian matrix as described in section 4.2.5. An efficient special treatment of the off-band elements was introduced, where the rows containing the off-band elements are included in the elimination steps of the in-band rows above them. Correspondingly, the LINPACK routine DGBSL was modified to solve the linear equation system arising at each integration step using the previously generated LU-decomposition.

4.3 Reduced model

4.3.1 Summary of reduction method

The model reduction method used in this study is based on the stage aggregation method of Lévine and Rouchon (1991). Their original method was developed for a simple column model with only one mass balance per stage. The column is partitioned into a number of “compartments” of consecutive stages. In each compartment, an “aggregation stage” is selected. By a state variable transformation, the concentration of each aggregation stage is replaced by the average concentration of the corresponding compartment stages. By this, the holdups of the aggregation stages are increased significantly compared to the remaining stages. Using this difference in magnitudes, the model can be written as a singular perturbation model. Then, a quasi-steady-state approximation can be applied, rendering all but the aggregation stages as steady-state stages. The resulting model is therefore of the same size as the original model, but the majority of the dynamic stage equations are converted into algebraic equations. It assumes the same steady-state as the original model.

It was shown in a previous study by Linhart and Skogestad (2009) that the method can actually be derived without the notion of compartments. This is due to an undocumented simplification step in the original derivation of the method. Although this simplification step deviates from the standard procedure for deriving singular perturbation models (Kokotovic et al., 1986), it greatly simplifies the derivation of the reduced model and its structure. For the application of the method to a given full model, it is sufficient to select some stages as aggregation stages and multiply the left-hand sides of their dynamic equations with some factors, while all remaining stages are modeled as "steady-state stages" by setting their left-hand sides to zero. This way, the original method can easily be generalized to more complex models including mass and energy balances; see Linhart and Skogestad (2009) for details.

It was found previously by Linhart and Skogestad (2009) that only applying the reduction procedure as described above does not necessarily improve the computational performance of the reduced model compared to the original model, since the number of equations in the reduced model is the same as in the original model. Therefore, in a second step, the algebraic equations resulting from the reduction procedure are eliminated from the reduced model. For this, the block-wise structure of the reduced model is exploited. The complete reduction method can therefore be described as a two-step procedure:

Step 1. Select a number of aggregation stages and multiply their left hand sides by certain factors, which will be called "aggregated holdup factor" in the following. Convert all remaining stages to steady-state stages by setting their left-hand sides to zero. The resulting model is a DAE model of the same size as the original model. It has, however, reduced dynamics.

Step 2. Eliminate the algebraic equations of each block of steady-state stages by replacing them with precomputed functions. This yields a model with a reduced number of variables and equations, which can be simulated faster than the original model. This step can be divided into two sub-steps:

a) Replace all variables of the steady-state stages that appear in the aggregation stage equations by functions obtained from the solutions of the steady-state equations in dependence of the variables of the aggregation stages;

b) Eliminate some of the functions and independent variables to obtain a final reduced model that is as compact as possible.

Step 1 can be applied immediately to the full model by simple manipulation of the left-hand sides of the differential equations. This procedure will be described in section 4.3.2. Step 2a is necessary to produce a reduced-order model that increases the simulation speed. The basic procedure is described in section 4.3.3. The key idea is to replace the algebraic equations resulting from the reduction procedure by precomputed functions. Due to the complexity of the model, these functions can become very complex themselves. To obtain efficient reduced models, in step 2b the number of functions and the number of independent variables is reduced to a minimum. This is described in section 4.3.3.

Note on notation: In order to stay consistent with the notation used in Linhart and Skogestad (2009), the variables in the reduced model after step 1, where the states are partially dynamic and partially algebraic, are marked by the bar notation $\bar{M}_i^{tot}$, $\bar{\mathbf{M}}_i$ and $\bar{U}_i^{tot}$. This is to distinguish them from the variables in the full model (4.1)-(4.12), where the states M_i^{tot}, $\mathbf{M}_i$ and U_i^{tot} are purely dynamic. In order to simplify notation, the variables of the reduced model after step 2 are not marked in a special way. The numbering of the variables is now different from the full model, since the final form of the reduced model consists only of aggregation stages. The functions that replace the algebraic equations of the steady-state stages are marked by $^{()}$. The final form of the reduced model is shown in table 4.4.

Table 4.4: Final form of reduced model.

reflux drum (aggregation stage 1):

$$\begin{aligned}
H_1\dot{M}_1^{tot} &= V_{top} - (R+D) \\
H_1\dot{\mathbf{M}}_1 &= V_{top}\mathbf{y}_2 - (R+D)\mathbf{x}_1 \\
H_1\dot{U}_1^{tot} &= V_{top}h_2^V - (R+D)h_1^L + Q_c
\end{aligned}$$

aggregation stage 2 (below reflux drum):

$$\begin{aligned}
H_2\dot{M}_2^{tot} &= R - L_2 + V_3^{()} - V_{top} \\
H_2\dot{\mathbf{M}}_2 &= R\mathbf{x}_1 - L_2\mathbf{x}_2 + V_3^{()}\mathbf{y}_3^{()} - V_{top}\mathbf{y}_2 \\
H_2\dot{U}_2^{tot} &= Rh_1^L - L_2h_2^L + V_3^{()}h_3^{V()} - V_{top}h_2^V - Q_2^{hl}
\end{aligned}$$

aggregation stage j:

$$\begin{aligned}
H_j\dot{M}_j^{tot} &= L_{j-1} - L_j + V_{j+1}^{()} - V_j^{()} \\
H_j\dot{\mathbf{M}}_j &= L_{j-1}\mathbf{x}_{j-1} - L_j\mathbf{x}_j + V_{j+1}^{()}\mathbf{y}_{j+1}^{()} - V_j^{()}\mathbf{y}_j^{()} \\
H_j\dot{U}_j^{tot} &= L_{j-1}h_{j-1}^L - L_jh_j^L + V_{j+1}^{()}h_{j+1}^{V()} - V_j^{()}h_j^{V()} - Q_j^{hl()} - Q_j^{hl}
\end{aligned}$$

feed stage j_F:

$$\begin{aligned}
H_{j_F}\dot{M}_{j_F}^{tot} &= L_{j_F-1} - L_{j_F} + V_{j_F+1}^{()} - V_{j_F}^{()} + F \\
H_{j_F}\dot{\mathbf{M}}_{j_F} &= L_{j_F-1}\mathbf{x}_{j_F-1} - L_{j_F}\mathbf{x}_{j_F} + V_{j_F+1}^{()}\mathbf{y}_{j_F+1}^{()} - V_{j_F}^{()}\mathbf{y}_{j_F}^{()} + F\mathbf{z}_F \\
H_{j_F}\dot{U}_{j_F}^{tot} &= L_{j_F-1}h_{j_F-1}^L - L_{j_F}h_{j_F}^L + V_{j_F+1}^{()}h_{j_F+1}^{V()} - V_{j_F}^{()}h_{j_F}^{V()} - Q_{j_F}^{hl()} \\
&\quad -Q_{j_F}^{hl} + Fh_F
\end{aligned}$$

aggregation stage $n-1$ (before reboiler):

$$\begin{aligned}
H_{n-1}\dot{M}_{n-1}^{tot} &= L_{n-2} - L_{n-1} + V_n - V_{n-1}^{()} \\
H_{n-1}\dot{\mathbf{M}}_{n-1} &= L_{n-2}\mathbf{x}_{n-2} - L_{n-1}\mathbf{x}_{n-1} + V_n\mathbf{y}_n - V_{n-1}^{()}\mathbf{y}_{n-1}^{()} \\
H_{n-1}\dot{U}_{n-1}^{tot} &= L_{n-2}h_{n-2}^L - L_{n-1}h_{n-1}^L + V_nh_n^V - V_{n-1}^{()}h_{n-1}^{V()} - Q_{n-1}^{hl()} \\
&\quad -Q_{n-1}^{hl}
\end{aligned}$$

reboiler (aggregation stage n):

$$\begin{aligned}
H_n\dot{M}_n^{tot} &= L_{n-1} - B - V_n \\
H_n\dot{\mathbf{M}}_n &= L_{n-1}\mathbf{x}_{n-1} - B\mathbf{x}_n - V_n\mathbf{y}_n \\
H_n\dot{U}_n^{tot} &= L_{n-1}h_{n-1}^L - Bh_n^L - V_nh_n^V + Q_r
\end{aligned}$$

4.3.2 Reduction step 1: Introducing aggregation stages and steady-state stages

Figure 4.2 illustrates the reduction method: A number n stages on positions with the indices s_j, $j = 1...n$, in the column are selected dynamic aggregation stages. s_j is the stage index of aggregation stage j in the reduced model containing both aggregation and steady-state stages; see figure 4.3a. For example, $s_3 = 10$ means that aggregation stage 3 corresponds to stage 10 in the original model.

The dynamics of the aggregation stages are slowed down by multiplying the left-hand sides of the corresponding dynamic equations of each aggregation stage j by the aggregated holdup factor $H_j >> 1$:

$$\begin{aligned} H_j \dot{M}^{tot}_{s_j} &= \bar{L}_{s_j-1} + \bar{V}_{s_j+1} - \bar{L}_{s_j} - \bar{V}_{s_j}, & (4.26)\\ H_j \dot{\mathbf{M}}_{s_j} &= \bar{L}_{s_j-1}\bar{\mathbf{x}}_{s_j-1} + \bar{V}_{s_j+1}\bar{\mathbf{y}}_{s_j+1} - \bar{L}_{s_j}\bar{\mathbf{x}}_{s_j} - \bar{V}_{s_j}\bar{\mathbf{y}}_{s_j}, & (4.27)\\ H_j \dot{U}^{tot}_{s_j} &= \bar{L}_{s_j-1}\bar{h}^L_{s_j-1} + \bar{V}_{s_j+1}\bar{h}^V_{s_j+1} - \bar{L}_{s_j}\bar{h}^L_{s_j} - \bar{V}_{s_j}\bar{h}^V_{s_j} - \bar{Q}^{hl}_{s_j}. & (4.28) \end{aligned}$$

The equations for the feed stage , the reflux drum and the reboiler are treated correspondingly.

The remaining stages $i = 1...N, i \neq s_j$ $(j = 1...n)$, are converted into steady-state stages by setting the left hand sides of the respective dynamic equations to 0:

$$\begin{aligned} 0 &= \bar{L}_{i-1} + \bar{V}_{i+1} - \bar{L}_i - \bar{V}_i, & (4.29)\\ 0 &= \bar{L}_{i-1}\bar{\mathbf{x}}_{i-1} + \bar{V}_{i+1}\bar{\mathbf{y}}_{i+1} - \bar{L}_i\bar{\mathbf{x}}_i - \bar{V}_i\bar{\mathbf{y}}_i, & (4.30)\\ 0 &= \bar{L}_{i-1}\bar{h}^L_{i-1} + \bar{V}_{i+1}\bar{h}^V_{i+1} - \bar{L}_i\bar{h}^L_i - \bar{V}_i\bar{h}^V_i - \bar{Q}^{hl}_i. & (4.31) \end{aligned}$$

4.3.3 Reduction step 2: Elimination of steady-state stages

In the second step of the reduction procedure, the algebraic equations of the steady-state stages are eliminated from the model. Despite the large number of algebraic equations, this is possible because of the structure of the reduced model, where the steady-state stages are grouped in blocks between the dynamic aggregation stages.

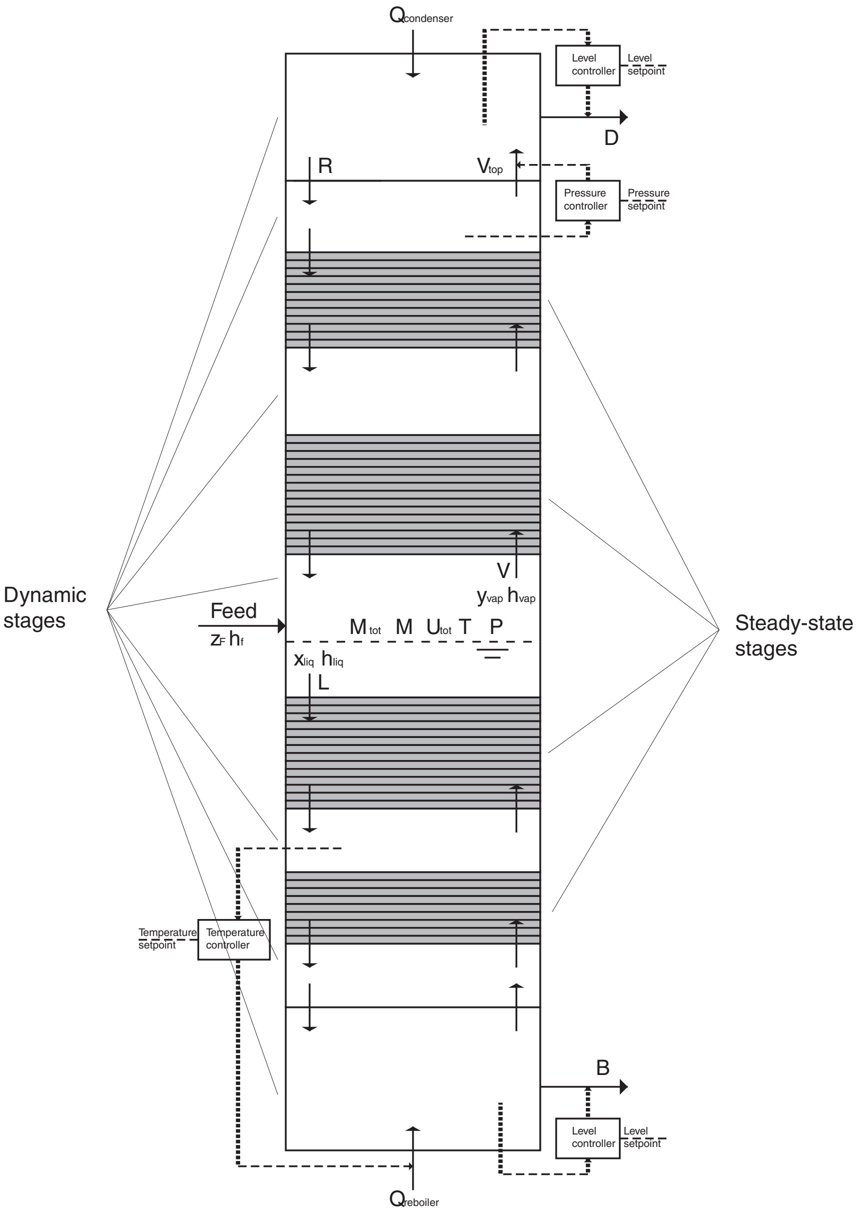

Figure 4.2: Schematic diagram of reduced column model.

Step 2a: Replacement of steady-state equations by functions

Figure 4.3 illustrates the principle. To avoid complicated notation, aggregation stages 2 and 3 are used for demonstration. Table 4.4 can be used as a reference for the general form of the equations. A block of steady-state stages is located between aggregation stages 2 and 3 (figure 4.3a). It is referred to in the following as steady-state block 3. It constitutes a system of algebraic equations, consisting of a set of equations (4.29)-(4.31) for each of the steady-state stages with the indices $i = s_2 + 1$ to $i = s_3 - 1$. It can be solved in dependence on a certain set $\mathbf{z}_3$ of variables of aggregation stages 2 and 3.

In order to eliminate the equations of steady state block 3, the variables $\bar{\mathbf{y}}_{s_2+1}, \bar{h}^V_{s_2+1}$ and $\bar{V}_{s_2+1}$ in the dynamic equations of aggregation stage 2 are replaced by the functions

$$\begin{aligned} \bar{\mathbf{y}}_{s_2+1} &= \mathbf{y}_3^{()}(\mathbf{z}_3), && (4.32)\\ \bar{h}^V_{s_2+1} &= h_3^{V()}(\mathbf{z}_3), && (4.33)\\ \bar{V}_{s_2+1} &= V_3^{()}(\mathbf{z}_3), && (4.34) \end{aligned}$$

and the variables $\bar{\mathbf{x}}_{s_3-1}, \bar{h}^L_{s_3-1}, \bar{L}_{s_3-1}$ and $\bar{V}_{s_3}$ in the dynamic equations of aggregation stage 3 are replaced by the functions

$$\begin{aligned} \bar{\mathbf{x}}_{s_3-1} &= \mathbf{x}_3^{()}(\mathbf{z}_3), && (4.35)\\ \bar{h}^L_{s_3-1} &= h_3^{L()}(\mathbf{z}_3), && (4.36)\\ \bar{L}_{s_3-1} &= L_3^{()}(\mathbf{z}_3), && (4.37)\\ \bar{V}_{s_3} &= V_3^{b()}(\mathbf{z}_3). && (4.38) \end{aligned}$$

The $\mathbf{y}_3^{()}$ notation signifies that the respective variable is a function of the variables of the neighboring aggregation stages 2 and 3; see figure 4.3b. These functions will be called steady-state functions in the following.

The variables above correspond to the flow rates and intensive properties of the flows from the steady-state block into the aggregation stages. In addition, the vapor flow rate $\bar{V}_{s_3}$ from aggregation stage 3 depends on the

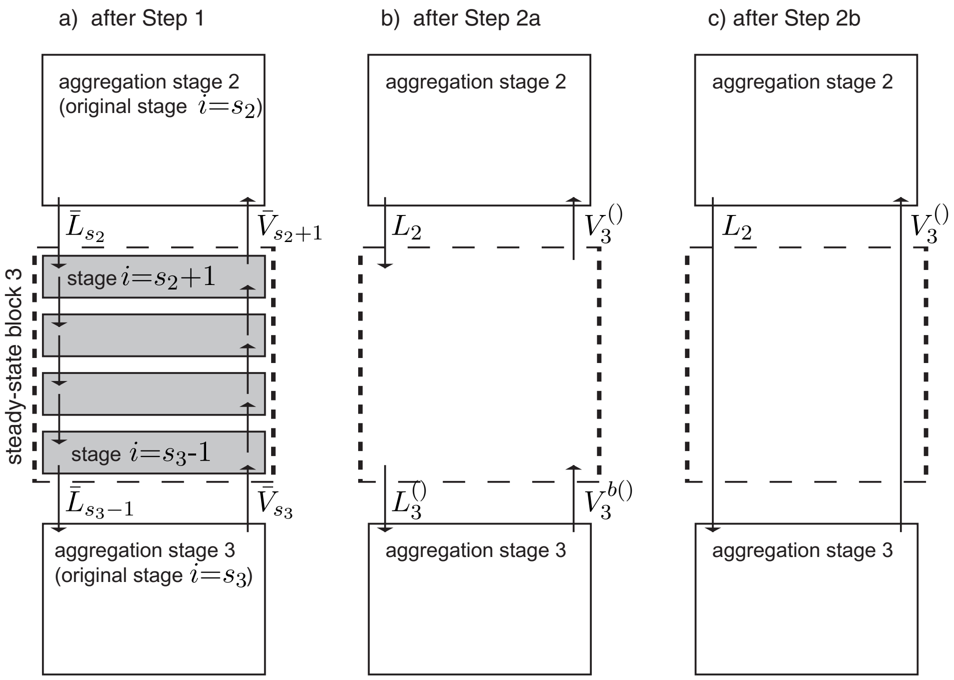

Figure 4.3: Schematic illustration of a block of consecutive steady-state stages between aggregation stages 2 and 3. Part a) shows the structure after reduction step 1. Part b) shows the structure after elimination of the steady-state stages by substitution of functions (4.32)-(4.34) and (4.35)-(4.38). Part c) shows the structure after elimination of the flows on the bottom of the steady-state block by mass conservation.

variables of the bottom stage of the steady-state block. As a consequence, it is replaced by the function $V_3^{b()}$. The ^{b} indicates that this vapor flow is located at the bottom of the steady-state block 3, in contrast to the vapor flow $V_3^{()}$, which is located at the top.
It is assumed here that the liquid flows only depend on the variables of the departing stage, otherwise the liquid flow $\bar{L}_{s_2}$ departing from aggregation stage 2 would have to be replaced by a function as well.
Aggregation stage 3 is used for illustration of the the dynamic equations of the reduced model after the substitution:

$$\begin{aligned} H_3\dot{M}_3^{tot} &= L_3^{()} - V_3^{b()} + V_4^{()} - L_3, && (4.39)\\ H_3\dot{\mathbf{M}}_3 &= L_3^{()}\mathbf{x}_3^{()} - V_3^{b()}\mathbf{y}_3 + V_4^{()}\mathbf{y}_4^{()} - L_3\mathbf{x}_3, && (4.40)\\ H_3\dot{U}_3^{tot} &= L_3^{()}h_3^{L()} - V_3^{b()}h_3^V + V_4^{()}h_4^{V()} - L_3h_3^L - Q_3^{hl}. && (4.41)\end{aligned}$$

Here, the notation is simplified, and M_3, L_3 etc. signify that the reduced model after step 2 consists only of equations and variables corresponding to aggregation stages.
A possible set of independent variables for the functions (4.32)-(4.34) and (4.35)-(4.38) is the set

$$\mathbf{z}_3 = \{M_2^{tot}, \mathbf{M}_2, U_2^{tot}, M_3^{tot}, \mathbf{M}_3, U_3^{tot}\}, \qquad (4.42)$$

consisting of $2N_c + 2$ variables. However, the complexity of the steady-state functions depends strongly on the number and selection of the independent variables. A suitable minimal selection of functions and independent variables is therefore discussed in the next section.

Step 2b: Minimal selection of steady-state functions and independent variables

The functions (4.32)-(4.34) and (4.35)-(4.38) are $2N_c + 3$ functions, while the variable set (4.42) contains $2N_c + 2$ variables. However, the functions are not completely independent of each other. Furthermore, not all state variables of both aggregation stages are needed as independent variables. In the following, it will therefore be shown that

1. The number of independent variables needed is $2N_c + 1$ (instead of $2N_c + 2$),

2. The number of functions needed is $N_c + 1$ (instead of $2N_c + 3$).

Minimal number of independent variables:
The following variables that are present in the system of algebraic equations of steady-state block 3, consisting of the set of equations (4.29)-(4.31) for each of the steady-state stages, depend on the variables of the aggregation stages 2 and 3:

$$\bar{L}_{s_2}, \bar{\mathbf{x}}_{s_2}, \bar{h}^L_{s_2} \text{ and } \bar{V}_{s_2+1} \tag{4.43}$$

depend on variables of aggregation stage 2, and

$$\bar{V}_{s_3}, \bar{\mathbf{y}}_{s_3} \text{ and } \bar{h}^V_{s_3} \tag{4.44}$$

depend on variables of aggregation stage 3 (compare figure 3a). The vapor flow $\bar{V}_{s_2+1}$ is a variable of steady-state stage $s_2 + 1$, but appears here because of its dependence on the variables of aggregation stage 2. Except for the liquid flow $\bar{L}_{s_2}$, all variables depend only on N_c intensive variables on the respective aggregation stage. $\bar{L}_{s_2}$ depends on an additional extensive variable due to its dependence on the liquid level on aggregation stage 2.
A suitable set of $2N_c + 1$ independent variables is therefore, for example,

$$\mathbf{z}_3 = \{\mathbf{x}_2, T_2, L_2, \mathbf{y}_3, T_3\}. \tag{4.45}$$

Here, the liquid flow from aggregation stage 2, L_2, is directly used as an independent variable for the functions of steady-state block 3.
For the case-study model with a binary mixture in the present work, it is convenient to use set of independent variables

$$\mathbf{z}_3 = \{T_2, p_2, L_2, T_3, p_3\}. \tag{4.46}$$

Minimal number of steady-state functions:
In the steady-state blocks, mass is conserved. Considering the total and

$N_c - 1$ component mass balances around steady-state block 3 (compare figure 4.3b),

$$\begin{aligned} 0 &= L_2 - V_3^{()} - L_3^{()} + V_3^{b()}, \qquad (4.47) \\ 0 &= L_2\mathbf{x}_2 - V_3^{()}\mathbf{y}_3^{()} - L_3^{()}\mathbf{x}_3^{()} + V_3^{b()}\mathbf{y}_3, \qquad (4.48) \end{aligned}$$

N_c additional equations are obtained. They can be used to reduce the number of functions that need to be substituted in the dynamic equations of the aggregation stages (4.39)-(4.41). Energy is, however, only conserved if the heat loss occurring at each stage is neglected:

$$0 = L_2 h_2^L - V_3^{()} h_3^{V()} - L_3^{()} h_3^{L()} + V_3^{b()} h_3^V - Q_3^{hl()}. \qquad (4.49)$$

Here, $Q_3^{hl()}$ is the accumulated heat loss of steady-state block 3.
Equations (4.47)-(4.49) can be rearranged to

$$\begin{aligned} L_3^{()} - V_3^{b()} &= L_2 - V_3^{()}, \qquad (4.50) \\ L_3^{()}\mathbf{x}_3^{()} - V_3^{b()}\mathbf{y}_3 &= L_2\mathbf{x}_2 - V_3^{()}\mathbf{y}_3^{()}, \qquad (4.51) \\ L_3^{()} h_3^{L()} - V_3^{b()} h_3^V &= L_2 h_2^L - V_3^{()} h_3^{V()} - Q_3^{hl()}, \qquad (4.52) \end{aligned}$$

and can then be used to eliminate the corresponding terms in the dynamic equations of aggregation stages. The equations for aggregation stage 3 (4.39)-(4.41) then read

$$\begin{aligned} H_3\dot{M}_3^{tot} &= L_2 - L_3 + V_4^{()} - V_3^{()}, \qquad (4.53) \\ H_3\dot{\mathbf{M}}_3 &= L_2\mathbf{x}_2 - L_3\mathbf{x}_3 + V_4^{()}\mathbf{y}_4^{()} - V_3^{()}\mathbf{y}_3^{()}, \qquad (4.54) \\ H_3\dot{U}_3^{tot} &= L_2 h_2^L - L_3 h_3^L + V_4^{()} h_4^{V()} - V_3^{()} h_3^{V()} - Q_3^{hl()} - Q_3^{hl}, \ (4.55) \end{aligned}$$

where only the vapor flow variables $\mathbf{y}_3^{()}$, $h_3^{V()}$, $V_3^{()}$, and the accumulated heat loss $Q_3^{hl()}$ remain as functions of steady-state block 3 (compare figure 4.3c). Note that equation (4.55) also includes the heat loss term Q_3^{hl} for aggregation stage 3.

A further reduction of the number of steady-state functions can be achieved

by using the fact that the vapor flow rate $V_3^{()}$ depends only on intensive variables of the topmost steady-state stage $s_2 + 1$ (compare figure 4.3a and b). It is therefore sufficient to know N_c intensive variables on this stage, for example $\mathbf{y}_3^{()}(\mathbf{z}_3)$ and $p_3^{()}(\mathbf{z}_3)$, to calculate all other intensive variables of the vapor flow (i.e. $h_3^{V()}$), and the vapor flow rate $V_3^{()}$.
If the heat loss on each tray is not neglected, an additional function

$$Q_3^{hl()} = Q_3^{hl()}(\mathbf{z}_3) \tag{4.56}$$

has to be included in the set of functions.
In the case of a binary mixture, it is practical to use the set of functions

$$T_3^{()}(\mathbf{z}_3),\ p_3^{()}(\mathbf{z}_3),\ Q_3^{hl()}(\mathbf{z}_3), \tag{4.57}$$

because then $y_3^{()}$ (which is scalar in this case) and $h_3^{V()}$ can be conveniently calculated from the tabulated thermodynamics as described in section 4.2.6.

4.3.4 Jacobian structure

The Jacobian of the reduced model as given in table 4.4 has exactly the same structure as the Jacobian of the full model, but the reduced model has fewer stages (see figure 4.1a and b). Since the temperature controller in the bottom now only spans over two stages, the width of the Jacobian of the model including the controller does not differ much from that of the reduced model without temperature controller.

4.3.5 Reduced model structure and parameters

The reduced model in this study consists of nine dynamic aggregation stages and $94 - 9 = 85$ steady-state stages. The locations and aggregated holdup factors of the aggregation stages are free tuning parameters. However, the following recommendations can be given (compare figure 4.2):

- The sum of all aggregated holdup factors should approximately amount to the number of stages in the system to obtain similar time constants of the reduced model.

- Reboiler and reflux drum should be chosen as aggregation stages because of their large capacities. Their aggregated holdup factors H_1 and H_n should be close to 1.
- The feed stage should be chosen as aggregation stage for an easy inclusion the feed variables in the reduced model equations.
- The stages where a controller is applied, i.e. the pressure-controlled topmost stage and the temperature-controlled stage in the bottom section, should be chosen as aggregation stages. This way, the controllers can be included in the reduced model exactly as in the full model.
- In the full model, the temperature control loop in the bottom section spans over a relatively large number of stages. To achieve a good approximation of the control loop behavior in the reduced model, it is therefore advisable to increase the dynamic order by including one additional aggregation stage inside the control loop.

In this study, two sets of parameters for the reduced model with nine aggregation stages are used for evaluating the performance of the reduced models:

1. An "equally-distributed" choice of parameters, where the free aggregation stages are distributed between the fixed aggregation stages at equal distances. The aggregated holdup factor of each aggregation stage corresponds to half of the number of steady-state stages between the aggregation stage and the adjacent aggregation stages on both sides plus one for the aggregation stage.
2. An "optimized" choice of parameters, where the free parameters were determined by a least-squares fit of the top concentration trajectory of the reduced model on the full model trajectory using the input signal described in section 4.4.1. The parameter optimization can be performed conveniently using the reduced model in DAE form that is obtained after reduction step 1 as described in section 4.3.2.

Table 4.5: Positions and aggregated holdup factors of the aggregation stages of the reduced models. A model with equally-distributed aggregation stages and holdups, and a model with optimized aggregation stage positions and holdups is shown.

	Equally-distributed:		Optimized:	
aggregation stage	s_j	H_j	s_j	H_j
1 (Reflux drum)	1	1	1	1
2	2	8	2	8.42
3	17	14.5	13	10.93
4	31	14.5	26	16.22
5 (Feed)	46	15	46	19.51
6	61	15	65	12.34
7 (Temp. controlled)	76	16	76	14.46
8	93	9	93	6.99
9 (Reboiler)	94	1	94	1

To find the (locally) optimal parameter set, discrete and continuous optimizations were performed iteratively.

The equally-distributed parameter set is used to demonstrate the approximation quality of a reduced model, where no particular effort is undertaken to determine favorable reduced model parameters. This can be considered the least accurate approximation quality that can be expected from a reduced model. On the other hand, the optimized parameter set gives an indication of the best possible approximation quality, which is, however, specific for the given case the parameters were optimized for. The equally-distributed and optimized parameter sets are given in table 4.5.

4.3.6 Implementation of steady-state functions by table interpolation

In the reduced model, the steady-state functions (4.57) are used to calculate the vapor flow variables in the dynamic equations of the aggregation stages (4.53)-(4.55). These functions are the solutions of the steady-state blocks described by equations (4.29)-(4.31) between the aggregation stages that depend on the set of independent variables (4.46) as described in section 4.3.3. The solutions can only be obtained numerically due to the nonlinear nature of the equations. The continuous functions (4.57) have therefore to be generated from discrete numerical solutions on the domain of the independent variables.

In this study, a five-dimensional lookup table is used for this purpose. The function values are calculated numerically on a grid of a certain resolution spanning the input domain. Function values at arbitrary points on the input domain can then be retrieved by interpolating between neighboring table entries.

The following issues are important when generating and using the table:

1. The simplest way to obtain continuous function values is multi-dimensional linear interpolation (Press et al., 2007) between the discrete table entries. For a five-dimensional interpolation, $2^5 = 32$ table lookup operations and proportionally many calculations are needed. This is computationally relatively expensive, compared to other calculations in the column model. Possible simplifications are discussed later in section 4.5.

2. The table needs a certain resolution to achieve a sufficient approximation accuracy using linear interpolation. It is therefore advisable to restrict the domain of the independent variables. This can be done by determining the maximal and minimal values of these variables during a suitable simulation.

3. Some safety margin should be added to the domain of the independent variables to take situations into account, when the independent

variables leave their previously calculated operating domain due to unexpected dynamic behavior of the system.

4. There are many possibilities for choosing the set of independent variables. A good choice may yield a significant decrease in table size for a given accuracy. This is illustrated in figure 4.4. Depicted are trajectories of the temperatures T and pressures p of two neighboring aggregation stages. While the temperatures assume values on large parts of the domain, the pressures are tighter correlated and move only on a narrow band of the whole domain. This can be explained by the fast nature of the pressure dynamics, which is due to the immediate dependence of the vapor flow on the pressure difference between two stages. It is therefore advisable to choose the pressure p_j of one dynamic stage j, and the pressure difference $\Delta p = p_{j+1} - p_j$ as independent variables, instead of the two pressures p_j and p_{j+1}. This reduces the domain of the independent variables and thereby the size of the table several times.

5. In order to make optimal use of the available memory, the table resolution along each dimension and thereby the total table size can be adapted to the accuracy requirements. This can be done in two steps:

    (a) The interpolation error for a given table resolution is estimated. For this, the function value at a test point is calculated numerically. Symmetrically around this point, 2^5 grid points with a distance in each dimension corresponding to the table resolution are calculated numerically, and the interpolated function value at the test point is determined. This can be repeated for a number of test points to scan the domain of independent variables systematically, because the degree of curvature of the functions might vary over the domain. Either the average or the maximum of the absolute differences between exact and interpolated function values can be taken as a measure for the interpolation error.

Table 4.6: Dimensions of lookup tables for approximation of the steady-state stage functions.

Steady-state block j	T_j	p_j	L_j	T_{j+1}	Δp_j
2	10	11	14	20	55
3	15	14	17	15	44
4	23	17	25	15	92
5	22	15	29	15	70
6	12	13	16	12	78
7	12	13	17	10	200

(b) The effect of the interpolation error on the outputs of interest in steady-state is estimated. The two outputs of primary interest of the model are the top and bottom product concentrations of component 1. The sensitivity of these concentrations to the error in one function can be calculated by perturbing the corresponding function value and calculating the finite-difference quotient. It was found that the sensitivities do not change significantly when different steady-states (corresponding to different constant inputs) are used to calculate the difference quotient.

The interpolation error of a function multiplied by the corresponding sensitivity gives an estimate for the effect of the interpolation error on the outputs. Appropriate table dimensions can now be found by minimizing a certain norm of the vector of the interpolation error effects for a given total storage space. The resulting dimensions of the tables used in this study are shown in table 4.6.

4.4 Reduced model performance

In this section, the performance of the reduced model is compared with the performance of the original model. The performance of a model always

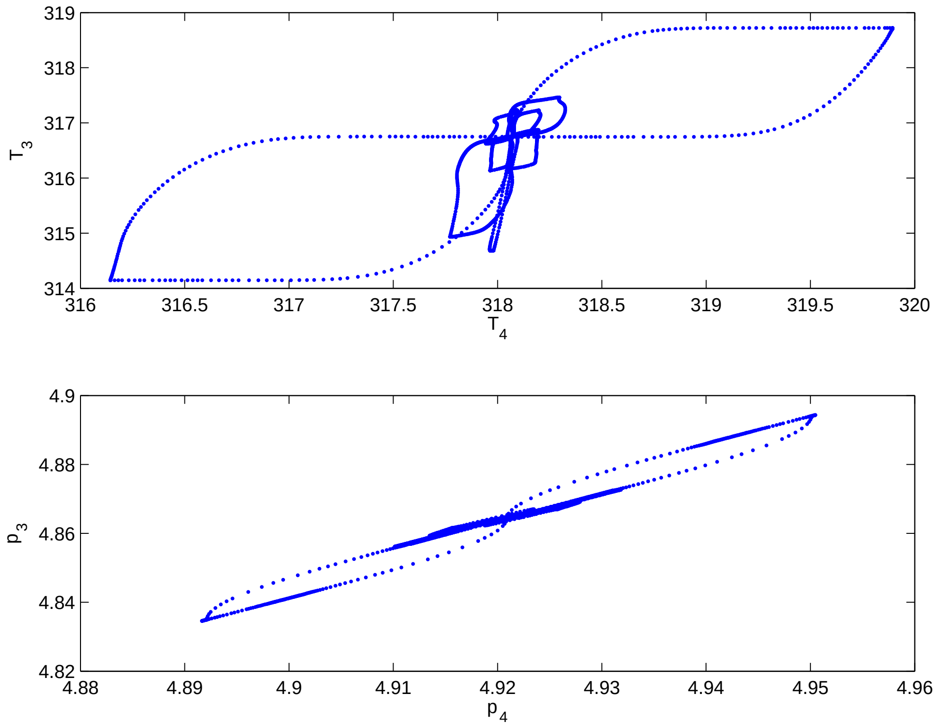

Figure 4.4: Temperature and pressure correlations of aggregation stages 2 and 3.

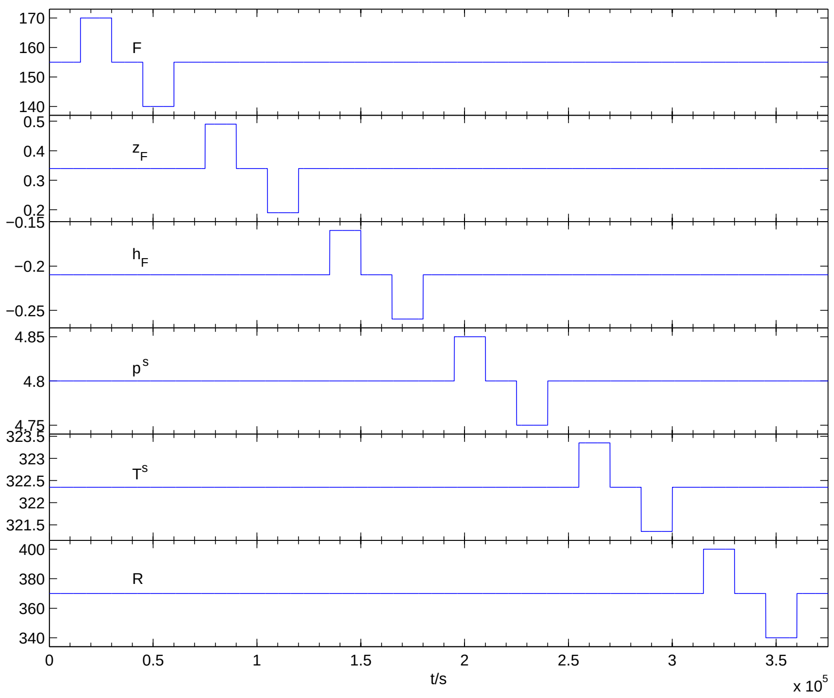

Figure 4.5: Input trajectories used for model performance assessment.

depends on the application the model is intended for. The objective of the performance assessment in this study is to give general insight into the approximation quality and the numerical performance of the reduced model in comparison with the original model. For this, simulations with fast continuous changes in the different inputs are performed.

4.4.1 Test input trajectories

Figure 4.5 shows the input trajectories used for the performance assessment. The inputs F, z_F and h_F describe the feed into the column, and can be seen from a control perspective as disturbance variables. The inputs p^s, T^s and

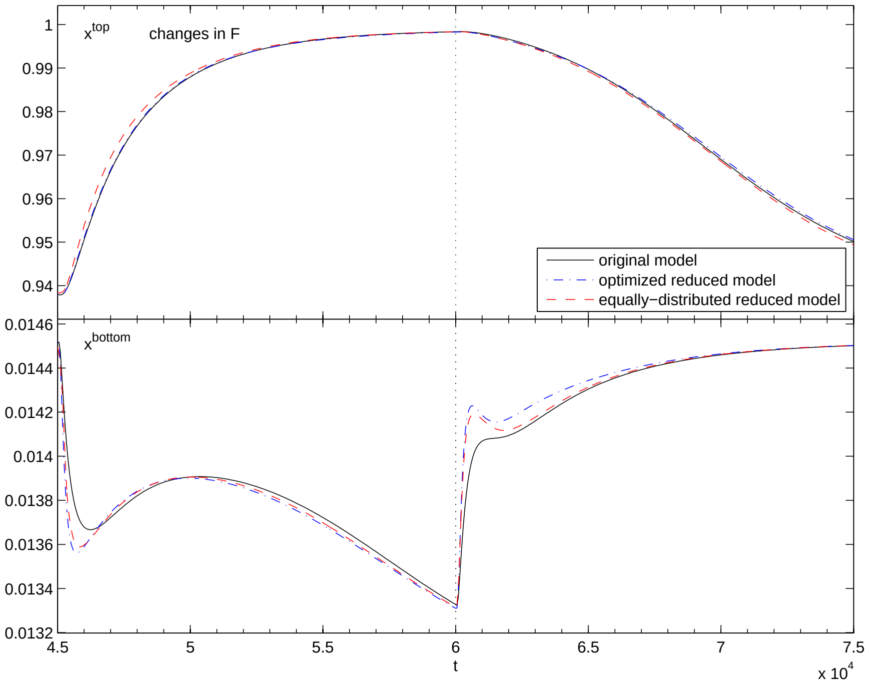

Figure 4.6: Top and bottom concentration trajectories of the full and the reduced models. The feed flow rate F is changed from 155 to 140 (left part) and back (right part).

R are the pressure controller setpoint, the temperature controller setpoint, and the reflux rate, respectively. They can be used as manipulated variables for higher-level control of the column. The input changes are implemented as continuous cubic-spline functions with a transition time of 10s. After each change, the inputs are kept constant for 15 000s, allowing the system to approach steady state again.

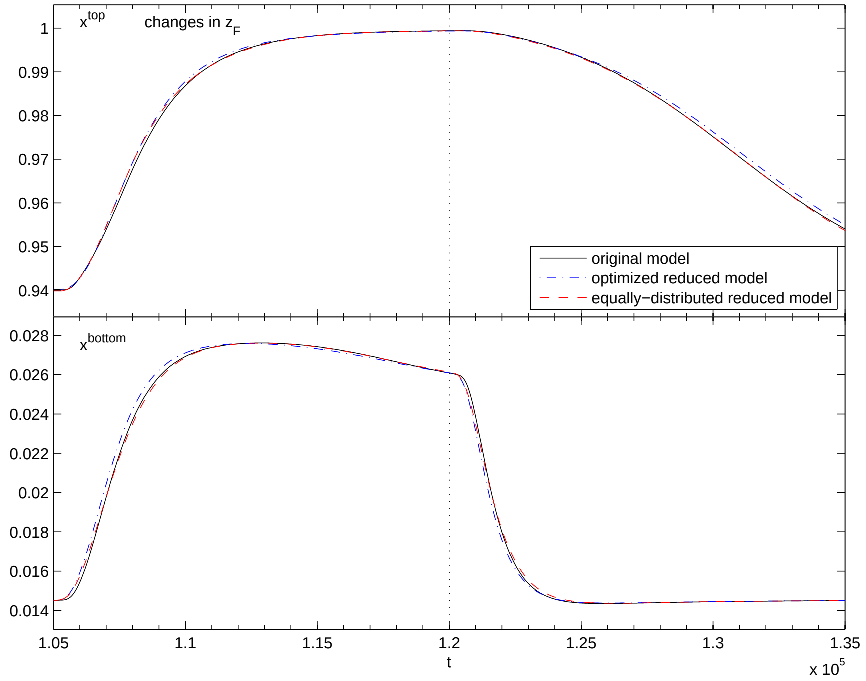

Figure 4.7: Top and bottom concentration trajectories of the full and the reduced models. The feed concentration z_F is changed from 0.34 to 0.19 (left part) and back (right part).

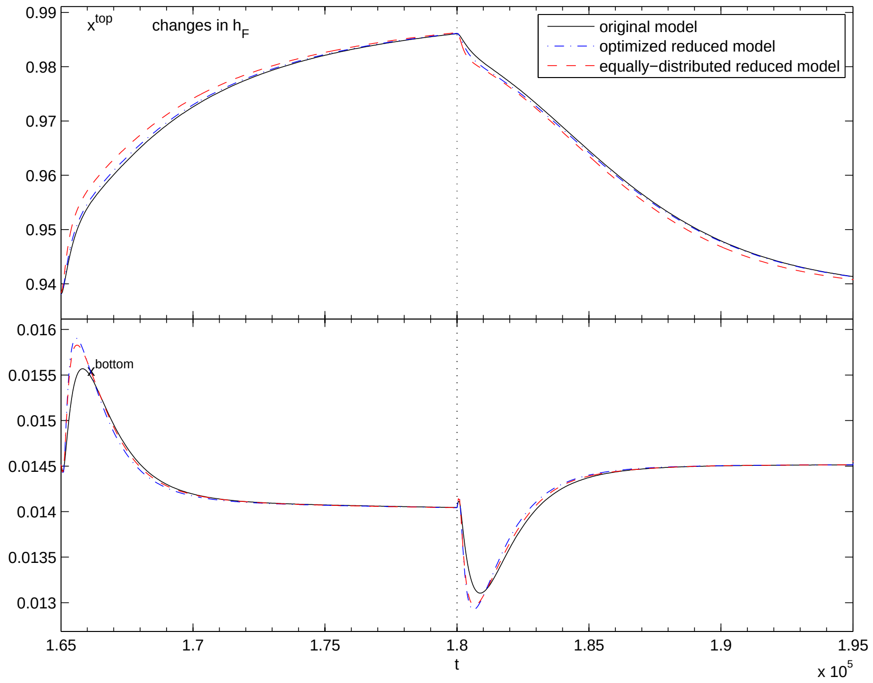

Figure 4.8: Top and bottom concentration trajectories of the full and the reduced models. The feed enthalpy z_h is changed from 0.2098 to 0.2598 (left part) and back (right part).

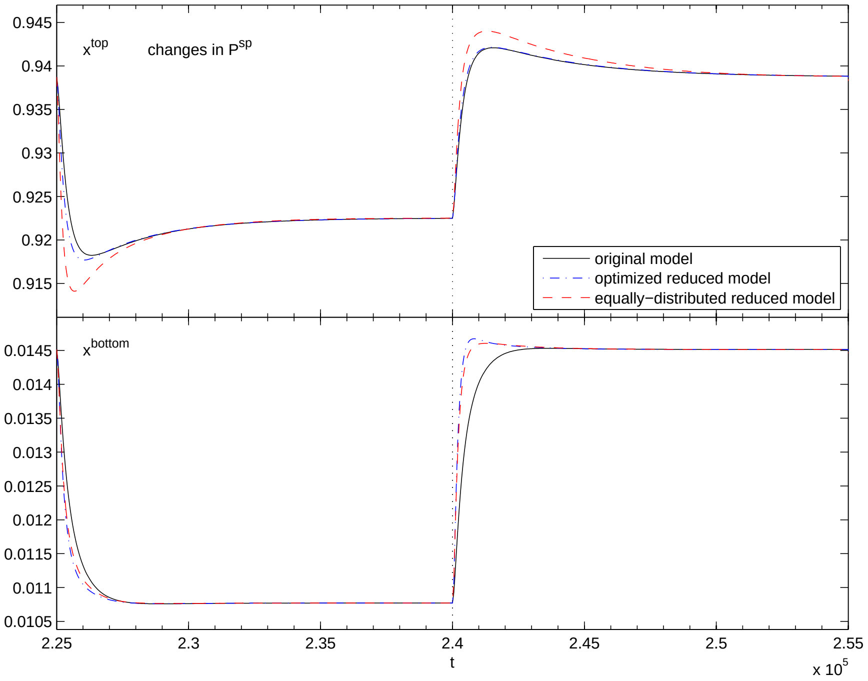

Figure 4.9: Top and bottom concentration trajectories of the full and the reduced models. The pressure setpoint p^{sp} is changed from 4.8 to 4.75 (left part) and back (right part).

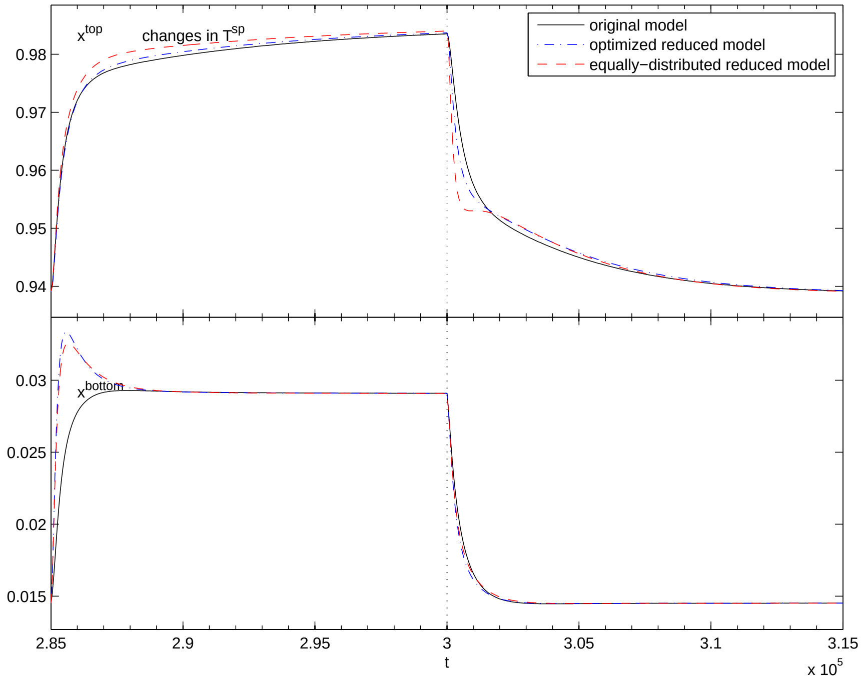

Figure 4.10: Top and bottom concentration trajectories of the full and the reduced models. The temperature setpoint T^{sp} is changed from 322.35 to 321.35 (left part) and back (right part).

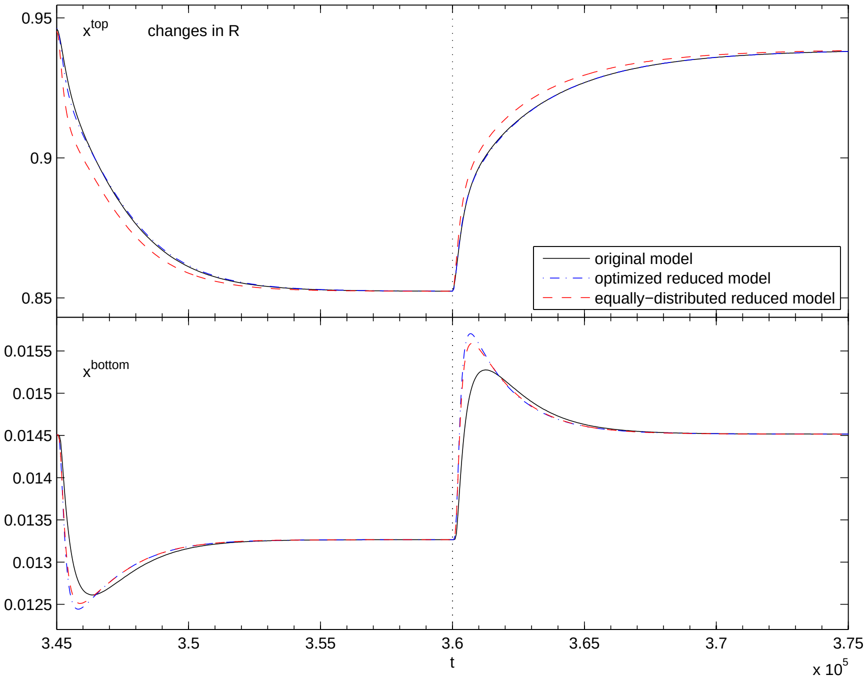

Figure 4.11: Top and bottom concentration trajectories of the full and the reduced models. The reflux rate R is changed from 370 to 340 (left part) and back (right part).

4.4.2 Accuracy of reduced model

Figures 4.6-4.11 show sections of the responses of the top and bottom concentrations of the full and the reduced models to changes in the different inputs. In relative terms, the deviations of the bottom concentrations of the reduced models are larger than the deviations of the top concentrations from the original model. In absolute terms, however, the bottom concentration deviations are small compared to the top concentration deviations, due to the action of the temperature controller in the bottom section. The parameters of the optimized reduced model have been determined by fitting the top concentration trajectories only. This explains the fact that the approximation of the bottom concentration is not more accurate for the optimized reduced model than for the equally-distributed model.

Generally, in terms of top concentration approximation accuracy, the optimized reduced model is superior to the equally-distributed reduced model. This is not the case for input changes in the feed concentration (figure 4.7), where both models are approximating the original dynamics very accurately, but the equally-distributed model is slightly more accurate. This is because the optimized reduced model has been optimized to approximate the original model over the whole simulation domain, which lowers the approximation quality at some points to gain a larger improvement at others. It can be observed that the equally-distributed reduced model is generally faster than the full model. This suggests that the infinite fast signal transport through the steady-state blocks in the reduced model is not fully compensated by the large aggregated holdup factors of the aggregation stages, such that the reduced model can possibly be improved by slightly increasing the aggregated holdup factors. Interestingly, the aggregated holdup factors of the optimized reduced model are even smaller (compare table 4.5), indicating that the locations of the aggregation stages have a considerable influence on the approximation accuracy.

4.4.3 Computational performance of reduced model

In order to compare the original and the reduced model, both were simulated at the simulation tolerances

$$\theta^{abs} = \theta^{rel} = 10^{i/2},\ i = 2, ..., 8, \tag{4.58}$$

where θ^{abs} and θ^{rel} are the absolute and relative simulation tolerances, respectively. For simplicity, the same value was used for both during one simulation.

Simulation time versus tolerance

Figure 4.12 shows a logarithmic plot of the simulation (CPU) times for simulations of 370 000s (real time) using the input trajectory described in section 4.4.1 of the full and the reduced model in dependence of the simulation tolerance θ. At the same simulation tolerance, the reduced model can be simulated about 5 times faster at crude tolerances ($\theta = 10^{-1}$), and about 8 times faster at tight tolerances ($\theta \leq 10^{-2}$). At the average tolerance of $\theta = 10^{-2.5}$, the simulation time is about 0.056s for the reduced model, and 0.43s for the full model.

Simulation time versus error

To obtain a measure for the accuracy of a certain model, the trajectories of the model can be compared with trajectories of the original model simulated at very tight tolerances ($\theta = 10^{-8}$). The latter can be seen as the "exact" trajectories of the model. In this study, the average deviation of the top concentrations from the exact trajectory is used as a measure for the different models:

$$\varepsilon = \frac{1}{t_{end}} \int\limits_0^{t_{end}} \left| x_1^{exact}(t) - x_1^{model}(t) \right| dt, \tag{4.59}$$

where $x_1^{exact}(t)$ is the top concentration trajectory of the full model simulated at very tight tolerances, and $x_1^{model}(t)$ is the top concentration trajectory of the model the error of which is to be quantified. This error is called

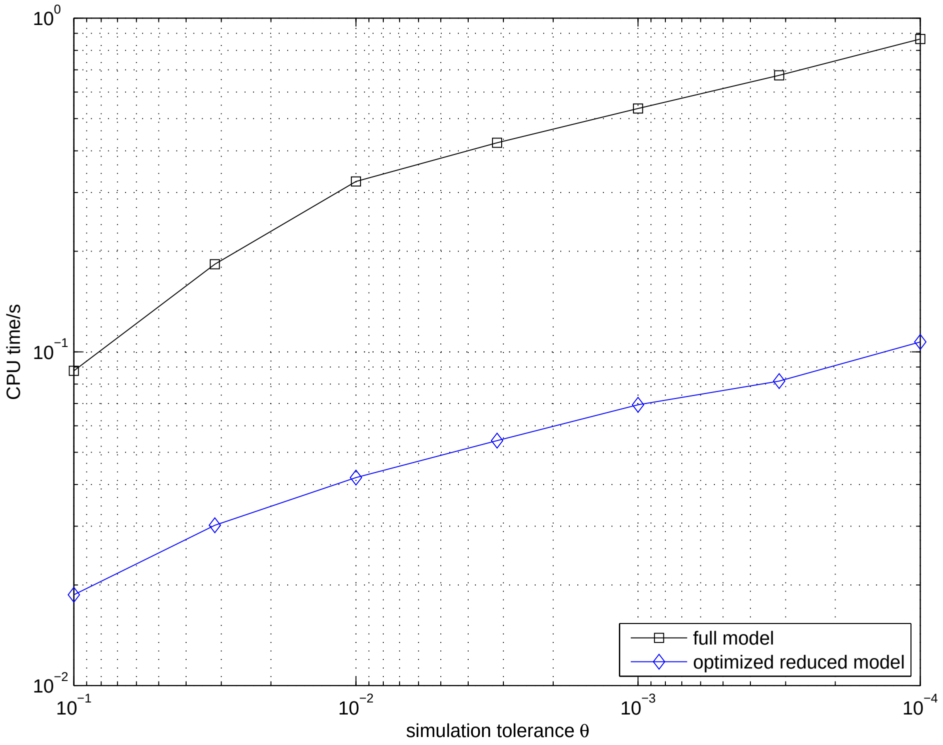

Figure 4.12: Simulation times of full and reduced models at different tolerances.

the overall error in the following. In practice, the integral is replaced by the average of sample points at intervals of 50s. Since the bottom concentration is varying little compared to the top concentration due to the temperature controller action, it is not included in the accuracy measure.

The accuracy of the full model is only determined by the simulation error, which is caused by the trade-off between simulation time and simulation accuracy governed by the simulation tolerance. The accuracy of a reduced model is in addition to the simulation error affected by the reduction error, which results from the difference between the full and the reduced dynamics. For a given reduced model and error measure, the reduction error is constant. A third error affecting the overall error of a reduced model is the implementation error, which results from the inexact implementation of the mathematically derived reduced model equations. For the reduced models in this study, an implementation error is caused by the implementation of the steady-state functions by interpolated tables. However, this error is small compared to the reduction error. This can be seen in figures 4.6-4.11, where practically no deviation between full and reduced model is visible when close to steady-state. In the following, the implementation error is therefore neglected.

Figure 4.13 shows the simulation time of the full and the optimized reduced model versus the overall error. It can be seen that the simulation times of both the full and reduced model increase with increasing simulation accuracy (decreasing simulation error). The reduction error of the reduced model is limiting the maximal achievable accuracy for the simulation with tight tolerances, where increasing the simulation accuracy does not lead to an increase of the overall accuracy. The reduction error starts to dominate the overall error from tolerances of around $\theta = 10^{-2.5}$ and on. At the maximal achievable accuracy, the overall error is around $4.7 \cdot 10^{-4}$. Below this tolerance, the simulation time of the reduced model is considerably lower than that of the full model, with a factor of approximately 6.5 at $\theta = 10^{-2.5}$.

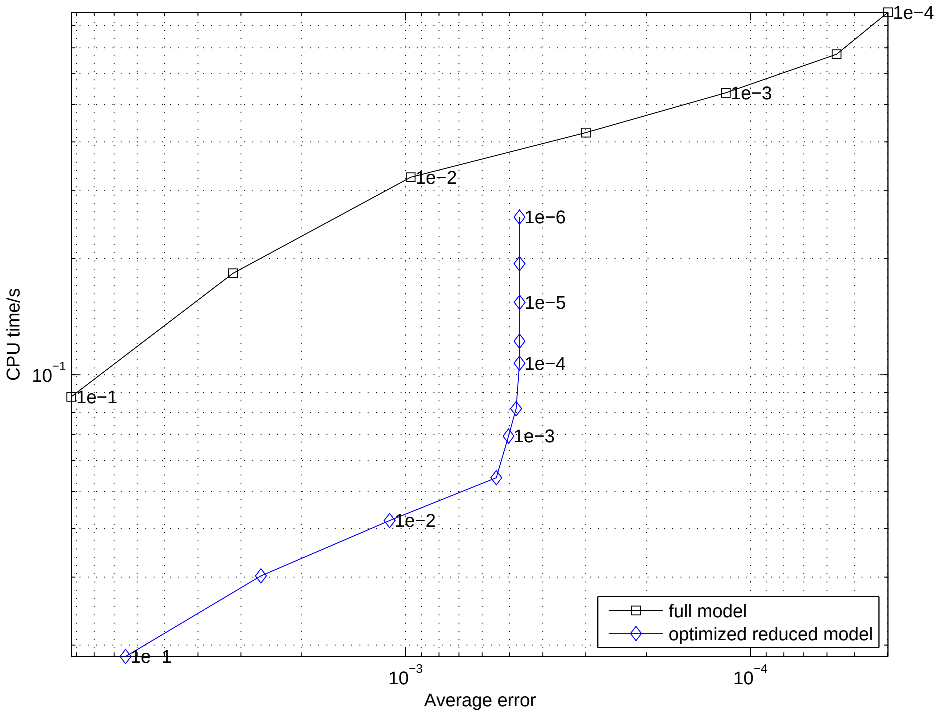

Figure 4.13: Simulation time versus average error. The numbers along the data points are the simulation tolerances used during the corresponding simulations.

Table 4.7: Percentage of simulation time spent in the main solver functions and the most important subfunctions.

		full model		reduced model	
main functions (A)	subfunctions (B)	A	B	A	B
residual	thermodynamics	16.3%	11.4%	24.0 %	8.1%
	hydraulics		3.1%		1.1%
	lookup table				10.9%
Jacobian	thermodynamics	18.9%	5.4%	16.6%	5.5%
	hydraulics		10.4%		1.8%
	lookup table				5.9%
LU-decomposition	row scaling/addition	28.4%	12.6%	22.00%	9.8%
LU-solution	row scaling/addition	13.9%	6.7%	12.7%	6.6%

Computational complexity of model and solver

Table 4.7 shows the contributions of the main model and solver functions and of their most important subfunctions to the total simulation times of the full and the reduced model. The numbers were obtained from simulations with the simulation tolerance $\theta = 10^{-2.5}$. At this tolerance, the reduced model shows the best performance (see figure 4.13).

It can be seen that for the full model, residual and Jacobian evaluation are computationally less intensive than the LU-decomposition and LU-solution functions. The execution time of the residual evaluations is dominated by the thermodynamic calculations on every stage, whereas in the Jacobian calculations, the execution times of the functions for computing the hydraulic quantities and their derivatives are higher than the thermodynamic calculations.

No function uses much more of the execution time than the other functions. This means that no significant increase in simulation speed can be achieved by reducing the execution time of a single function. The most expensive functions are the linear algebra functions (LU-decomposition and LU-solution). A doubling of execution speed here would only lead to a 22%

decrease in total simulation time.

For the reduced model, the percentage of the execution time of the residual evaluations is significantly higher than for the full model. This is due to the computationally expensive steady-state function lookup tables and interpolations. They require with $\sim 11\%$ almost half of the execution time of the function. The situation is similar for the Jacobian evaluations, where the derivative calculation of the tabulated functions account for $\sim 6\%$ of $\sim 17\%$. The hydraulic calculation execution times are not significant in residual and Jacobian calculations anymore, because the vapor flow is obtained from the steady-state functions. This is especially the case for the Jacobian, where the computationally intensive calculations of the vapor flow derivatives are not necessary anymore.

4.5 Discussion

4.5.1 Model reduction method

The main theoretical aspects of the model reduction method used in this study compared to the original method of Lévine and Rouchon (1991) have been discussed previously by Linhart and Skogestad (2009). As shown in section 4.4, the reduced model is capable of reproducing the dynamic behavior with good accuracy, and almost perfectly reproduces the steady-states, except for some negligible implementation error. The computational complexity is several times lower than that of the original full model.

The simplified derivation using aggregation stages instead of compartments makes the method applicable in a straightforward fashion to all kinds of staged processes. Since in step 1 of the reduction procedure only simple manipulations of the left-hand sides of the dynamic equations of the original model are needed, it is easy to quickly derive a model with reduced dynamics to test the suitability for a given application, and to determine a suitable parametrization and perform a dynamic analysis of the reduced model.

Step 2 of the reduction procedure is conceptually straightforward, but requires more implementation effort. Due to the high dimensionality of the steady-state functions that are substituted into the dynamic equations, the method is restricted to systems with a low number of state variables on each stage. This is the main bottleneck of the method. However, the lookup table with linear interpolation used in this study is a relatively simple and straightforward approach, which works very well for the example system. Possible improvements are discussed in the next section.

In the original method of Lévine and Rouchon (1991), a fast time-scale of the stage dynamics and a slow time-scale of the compartment dynamics is identified. Such a time-scale separation is typical in singular perturbation systems. However, in this case the time-scales are somewhat constructed, since the compartments are not present in the real system, but are artificially introduced into the model. It was shown in Linhart and Skogestad (2009) that only by some undocumented simplification step that deviates from the normal singular perturbation procedure, a reduced model of the same form as the models in this paper is obtained. The compartment boundaries do not appear in the model anymore, which makes the notion of compartments useless. It is therefore misleading to use compartments and time-scale separations to explain the principle of the method.

To understand and classify the model reduction method of the present work, it is therefore important to emphasize that the method does not rely on any time-scale separation in the column, and is therefore no real singular perturbation method. Instead, a different physical interpretation can be given: The transport of "signals“ (changes of mass and energy and intensive quantities) through the steady-state stages is made infinitely fast, which is compensated by the slow dynamics of the dynamic aggregation stages which are distributed over the column. The method described in the present work is therefore a specialized model reduction method for one-dimensionally distributed staged systems.

4.5.2 Implementation of steady-state functions

The implementation of the steady-state functions as described in section 4.3.3 is difficult because of the large number of independent variables. In the example distillation column in this study, the number of independent variables is five. This is one less than the total number of dynamic states of the aggregation stages on both sides of each block of steady-state stages. This is due to the unsymmetrical flows in the column, where the vapor flow only depends on the intensive quantities on each stage.

Extension to systems with more components

In this study, the steady-state functions were implemented using five-dimensional lookup tables with multi-linear interpolation. The interpolation is computationally expensive, because each interpolation dimension increases the computational complexity by factor two. Thus, $2^5 = 32$ operations are needed to obtain one interpolated value from a table. However, as shown in section 4.4.3, the table lookup and interpolation takes only about 17% of the total simulation time. This means that the reduced model is only insignificantly slowed down by the additional complexity resulting from the elimination of the algebraic equations. From a computational performance point of view, it is therefore possible to apply the method to more complex systems. If, for example, the reduction method is applied to a system with three components, one dynamic and one algebraic state per stage is added, increasing the number of states from 5 to 7. In addition, two more independent variables corresponding to one additional state on each side of each steady-state block have to be included in the table and the interpolation. This means that about 17% of the simulation time which is spent in the table lookup and interpolation will increase by factor 4, while the remaining about 83% of the simulation time will increase proportionally to increase in the number of states by factor 7/5. Then, the table lookup and interpolation will take about 37% of the overall time. Since the computation time of the full model will also increase by factor 7/5, the reduced model will still be several times faster. For example, if the reduced model was 8 times faster at the same simulation tolerance, the extended model with three components will still be 6 times faster.

Reduction of table complexity

To reduce the complexity of the tables, the following ideas can be considered:

- The function to be approximated can be partially linearized in the following way:

  $$f(x_1, x_2, x_3) \approx f_1(x_1, x_2) + f_2(x_1, x_2)x_3. \tag{4.60}$$

  This can be done when, for example, the function depends on the concentration of a component that has a very low concentration compared to the other components. Then, the nonlinear function that has to be tabulated is of lower dimension.

- Cubic spline interpolation can be used instead of linear interpolation along dimensions which require a high resolution. For example, table 4.6 shows that the table dimension corresponding to the independent variable δP requires a high resolution. This is due to a more nonlinear dependence of the function values on this variable. Cubic spline interpolation is easy to implement, but requires four lookup operations per dimension. If one table dimension is interpolated with cubic splines, the computational complexity of the interpolation will therefore double. However, since the interpolation error is of higher order, the complexity of the tables can be reduced several times.

- The table resolution can be adapted locally to the curvature of the tabulated function. A simple way to do this is to use non-uniform table grids. A more sophisticated method is the use of sparse grids, where the table resolution is adapted locally (Barthelmann et al., 2000).

Polynomial functional approximation

As an alternative to tables, functional approximation using polynomials or other suitable basis functions can be used. Their coefficients can be determined by, for example, least-square fits to sample data on a certain domain of the independent variables. However, also here the resulting expressions can be rather complex due to the high number of independent variables, and the approximation accuracy can be unsatisfactory due to the global nature of the approximation. If, for example, polynomials up to third order are used, the resulting expression will consist of 56 terms.

Model robustness considerations

For all implementations of the steady-state functions, it is advisable to choose the domain of the independent variables as small as possible to achieve a high function accuracy. However, the independent variable domain has to be large enough to cover the operating domain the reduced model is intended for. Approximate domain boundaries can be determined by simulation, where maximal and minimal values of the different inputs are used. To increase the model robustness, the independent variable domain should be increased by a certain safety margin to account for situations where the model states leave the expected operating domain. In case the safety margin of a certain steady-state function is not sufficient, the function could be temporarily replaced by a less accurate steady-state function that has a larger independent variable domain. Since the functions affect only the flows between the aggregation stages, continuity of the state evolutions is guaranteed. If even this is not enough, a reduced model after step 1 of the reduction method, that means a model which still explicitly contains the steady-state stages, can be used. Since there is no computational advantage of such a reduced model over a full model, it should be used only as backup.

4.5.3 Application of reduced model in real-time optimizing control

It was shown in section 4.4.3 that the reduced models can increase the simulation speed by a factor of about 7.5 when the same tolerance $\theta = 10^{-2.5}$ is used. This makes the models interesting for model predictive control and dynamic real-time optimization applications. However, the performance of the reduced models was assessed only in open-loop simulations with long intervals between changes in the inputs. In real-time optimizing control applications, input changes occur at much higher frequencies. Due to their structure, the reduced models approximate the long-term dynamics, which asymptotically approach the correct steady-state, with good accuracy. The short-term dynamics are not necessarily approximated equally well. The suitability of a reduced model of this kind for MPC and other real-time optimization applications will largely depend on how well the time-scales of the application and the model are matched, that means if the reduced model is capable to follow the changes in control and disturbance inputs at the frequency and speed they occur in the closed loop application. This issue has to be addressed in a separate study, where the reduced model is applied in a closed-loop optimizing control application.

4.5.4 Alternative model reduction methods for distillation models

There exist a number of alternative model reduction methods for distillation models. A short discussion has been provided in Linhart and Skogestad (2009). The conclusions basically also hold for the extension of the method to more complex distillation models as described in this study. However, to obtain solid criteria to decide which reduction method is best suited for a given application, a rigorous comparison of the most promising methods with the method in the present work has to be made.

Collocation methods (Cho and Joseph, 1983, Dalaouti and Seferlis, 2006) are probably the most similar methods in terms of approximation accuracy and gain in simulation speed. While they are not restricted to a low number of components as the method described in the present study, they possibly lose some approximation accuracy by approximating staged columns by continuous equations and applying collocation methods to the resulting partial differential equations.

Wave propagation methods (Hankins, 2007, Kienle, 2000, Marquardt, 1990) are so far restricted to distillation models with rather strict assumptions such as constant molar flows, since they make use of analytic solution of wave profile equations. The resulting models can therefore be expected to have limited approximation accuracy when used as reduced models for complex distillation models. However, they result in models of very low order, which promise very fast simulations.

Other methods are more suitable for nonlinear controller design than for fast simulations (Kumar and Daoutidis, 2003). An overview of further reduction and simplification methods for distillation column is given by Skogestad (1997).

4.6 Conclusions

A simplification of the aggregated modeling method of Lévine and Rouchon (1991) and an extension to complex distillation models is presented. The method is applicable in a straightforward fashion by manipulating the left-hand sides of the differential equations. It was shown that if the resulting algebraic equations are eliminated from the reduced model, the reduced model yields a gain in computational speed of a factor of around 7.5. The elimination of the algebraic equations is conceptually straightforward, but requires the approximation of functions of five independent variables. In this study, lookup tables combined with multi-linear interpolation were used for this purpose. The approximation quality of the reduced models was shown by simulations to be very accurate. In this study, a binary distillation model was investigated. The extension of the method to systems with a larger number of components is possible, but is limited by the increasing complexity of the function approximations. For systems with a low number of components, the resulting fast and accurate reduced models are promising for real-time optimizing control applications.

Acknowledgments

The authors thank Pål Kittilsen of Cybernetica AS for fruitful discussions. Furthermore, the authors thank Uwe Weimann of Zuse Institut Berlin for providing help using the non-linear equations solver NLEQ (Novak and

Weimann, 1990) that was used to calculate the lookup tables. This work has been supported by the European Union within the Marie-Curie Training Network PROMATCH under the grant number MRTN-CT-2004-512441.

Bibliography

[1] Allgöwer, F., Zheng, A., 2000. Nonlinear Model Predictive Control. Progress in Systems Theory 26, Birkhäuser, Basel.

[2] Antoulas, A.C., 2005. Approximation of Large-Scale Dynamical Systems. Cambridge University Press, Cambridge.

[3] Ascher, U.M., Petzold, L.R., 1998. Computer Methods for Ordinary Differential Equations and Differential-Algebraic Equations. SIAM, Philadelphia.

[4] Barthelmann, V., Novak, E., Ritter, K., 2000. High dimensional polynomial interpolation on sparse grids. Advances in Computational Mathematics 12, 273-288.

[5] Benallou, A., Seborg, D.E., Mellichamp, D.A., 1986. Dynamic Compartmental Models for Separation Processes. AIChE Journal 32, 1067-1078.

[6] Bian, S., Khowinij, S., Henson, M.A., Belanger, P., Megan, L., 2005. Compartmental modeling of high purity air separation columns. Computers & Chemical Engineering 29, 2096-2109.

[7] Cho, Y.S., Joseph, B., 1983. Reduced-Order Steady-State and Dynamic Models for Separation Processes. Part I. Development of the Model Reduction Procedure. AIChE Journal 29, 261-269.

[8] Dalaouti, N., Seferlis, P., 2006. A unified modeling framework for the optimal design and dynamic simulation of staged reactive separation processes. Computers & Chemical Engineering 30, 1264-1277.

[9] Green, D.W., Perry, R.H., 2007. Perry's Chemical Engineers' Handbook, Eighth Edition. McGraw-Hill, New York.

[10] Hairer, E., Wanner, G., 2002. Solving Ordinary Differential Equations II - Stiff and Differential-Algebraic Problems. Springer, Berlin.

[11] Hankins, N.P., 2007. A non-linear wave model with variable molar flows for dynamic behavior and disturbance propagation in distillation columns. Chemical Engineering Research & Design 85, 65-73.

[12] Khowinij, S., Henson, M.A., Belanger, P., Megan, L., 2005. Dynamic compartmental modeling of nitrogen purification columns. Separation and Purification Technology 46, 95-109.

[13] Khowinij, S., Bian, S., Henson, M.A., Belanger, P., Megan, L., 2004. Reduced Order Modeling of High Purity Distillation Columns for Nonlinear Model Predictive Control. Proceedings of the 2004 American Control Conference 5, 4237-4242.

[14] Kienle, A., 2000. Low-order dynamic models for ideal multicomponent distillation processes using nonlinear wave propagation theory. Chemical Engineering Science 55, 1817-1828.

[15] Kokotovic, P., Khalil, H.K., O'Reilly, J., 1986. Singular Perturbation Methods in Control: Analysis and Design. SIAM classics in applied mathematics 25. SIAM, London.

[16] Kumar, A., Daoutidis, P., 2003. Nonlinear model reduction and control for high-purity distillation columns. Industrial and Chemistry Research 42, 4495-4505.

[17] Lévine, J., Rouchon, P., 1991. Quality Control of Binary Distillation Columns via Nonlinear Aggregated Models. Automatica 27, 463-480.

[18] Li, S., Petzold, L.R., 2000. Software and Algorithms for Sensitivity Analysis of Large-Scale Differential Algebraic Systems. Journal of Computational and Applied Mathematics 125, 131-145.

[19] Linhart, A., Skogestad, S., 2009. Computational performance of aggregated distillation models. Computers & Chemical Engineering 33, 296-308.

[20] LINPACK, 1978. http://www.netlib.org/linpack/

[21] Marquardt, W., 2001. Nonlinear model reduction for optimization based control of transient chemical processes. Proceedings Chemical Process Control VI, 30-60.

[22] Marquardt, W., 1990. Traveling waves in chemical processes. International Chemical Engineering 30, 585-606.

[23] Nowak, U., Weimann, L., 1990. A Family of Newton Codes for Systems of Highly Nonlinear Equations - Algorithm, Implementation, Application. ZIB, Technical Report TR 90-10.

[24] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P., 2007. Numerical Recipes: The Art of Scientific Computing, Third Edition. Cambridge University Press, Cambridge.

[25] Qin, S.J., Badgwell, T.A., 2003. A survey of industrial model predictive control technology. Control Engineering Practice 11, 733-764.

[26] Reid, R.C., Prausnitz, J.M., Poling, B.E., 1997. The properties of gases & liquids (4th edition). McGraw-Hill, New York.

[27] Schlegel, M., 2005. Adaptive discretization methods for the efficient solution of dynamic optimization problems. VDI-Verlag, Düsseldorf.

[28] Skogestad, S., 1997. Dynamics and Control of Distillation Columns: A Critical Survey. Modeling, Identification and Control 18, 177-217.

[29] van den Berg, J., 2005. Model reduction for dynamic real-time optimization for chemical processes. PhD Thesis, TU Delft.

4.7 Appendix (not in journal paper)

4.7.1 Functional approximation of steady-state functions

As mentioned in the paper, the main difficulty of applying the reduction method is to express the numerical solutions of the steady-state stage systems as functions of the variables of the aggregation stages. The main problem with look-up tables is the high dimensionality of the tables. It is therefore interesting to search for alternatives to lookup tables.

One possible approach is to approximate the functions by polynomials. Here, polynomials of the form

$$v(x_1, x_2, x_3, x_4, x_5) = \sum_{a+b+c+d+e<4} \alpha_{abcde} x_1^a x_2^b x_3^c x_4^d x_5^e \tag{4.61}$$

are used. $x_1, ..., x_5$ are the independent variables, v is the approximated function, a, b, c, d and e are the integer exponents ≥ 0 of the variable $x_1, ..., x_5$ in the monomials $x_1^a x_2^b x_3^c x_4^d x_5^e$, and α_{abcde} are the coefficients of the monomials with the exponents $a, ..., e$. The sum of the exponents is less than four, which means that the polynomial is of order three.

The coefficients of the polynomial can be determined by a least-squares fit to n precomputed values v_i

$$\min_{\alpha_{abcde}} \sum_{i=1,n} \left(\sum_{a+b+c+d+e<4} \alpha_{abcde} x_{1i}^a x_{2i}^b x_{3i}^c x_{4i}^d x_{5i}^e - v_i \right)^2 \tag{4.62}$$

using singular value decomposition or another suitable method. To test this approach, three sets of precomputed values v_i have been used:

1. Values computed on a uniform grid spanning the domain of the lookup tables. In order to avoid to fit the functions to extreme values located in the corners of the rectangular table domain, only values in an ellipsoid inside the domain touching the domain boundaries have been used;

2. Values computed from a simulation using the input signals as used in the paper;

3. A combination of 1 and 2.

The functions computed with set 2 did not yield stable models. The reason for this is probably the poor extrapolation quality of the functions if fit only to data of a single trajectory. Therefore, in order to obtain more globally valid functions, in set 3 the trajectory values are combined with set 1.
The sets 1 and 3 produced stable models. The top concentration responses to the inputs used in the paper are shown in figure 4.14. The figure shows trajectories of a reduced DAE model, where the algebraic equations are solved during integration of the model (green curve), and of reduced models, where the algebraic equations were solved and replaced by look-up tables (blue curve) and polynomial approximations (red and green curves). It can be seen that the trajectory of the reduced model with tables is virtually indistinguishable from the trajectory of the DAE model, which means that using the tables yields a very high accuracy. The trajectory of the reduced model with polynomials determined with value set 1 shows some relatively large deviations from the DAE model, which are around 0.006 (0.6%) at maximum. The trajectory of the reduced model with polynomials determined with value set 2 shows much smaller deviations, with around 0.002 (0.2%) at maximum. To validate the polynomial approximations obtained with value set 3 on a trajectory that has not been used to determine the polynomial coefficients, the inputs as shown in figure 4.15 were used. They contain simultaneous changes in the disturbance variables F, z_F and h_F, and the control variables p^{sp}, T^{sp} and R. Figure 4.16 shows the top concentration responses to these inputs. It can be seen that the deviations of the reduced model with polynomial approximations using set 3 are still relatively small, and those obtained using set 1 are comparatively larger.

The polynomial approximations can certainly be improved by

- optimizing the choice of monomials or basis functions of the polynomials;
- optimizing the set of precomputed values used to determine the polynomial coefficients. In case of set 1, the values domain is possible too large, forcing the polynomials to approximate the functions for

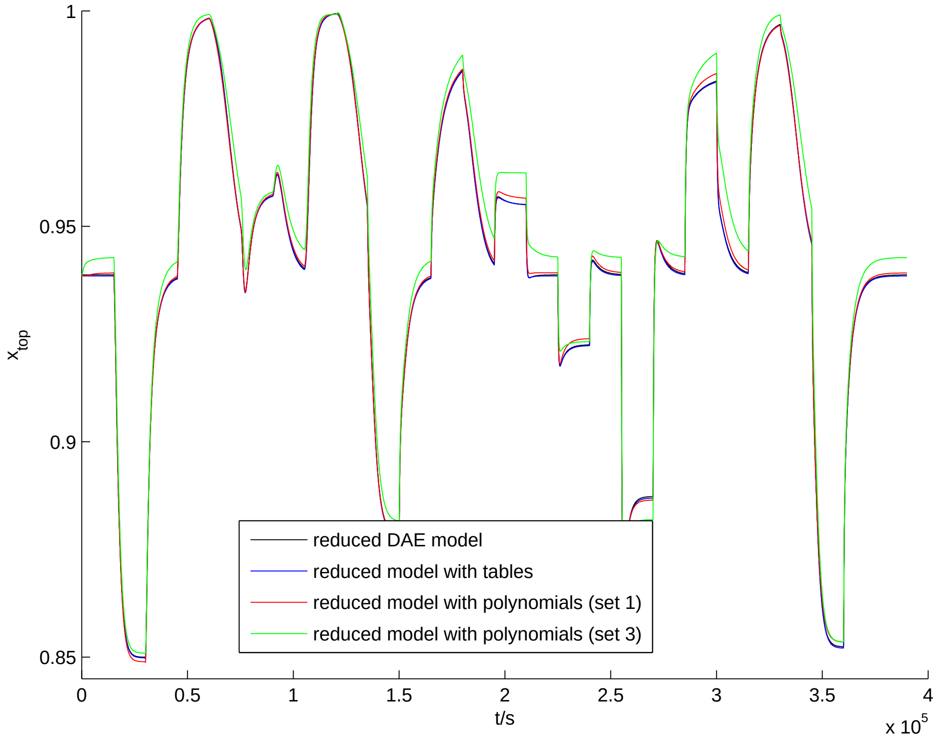

Figure 4.14: Top concentration responses of the reduced DAE model, the reduced model using tables, and the reduced models using polynomial approximations generated using set 1 and 3 of precomputed values.

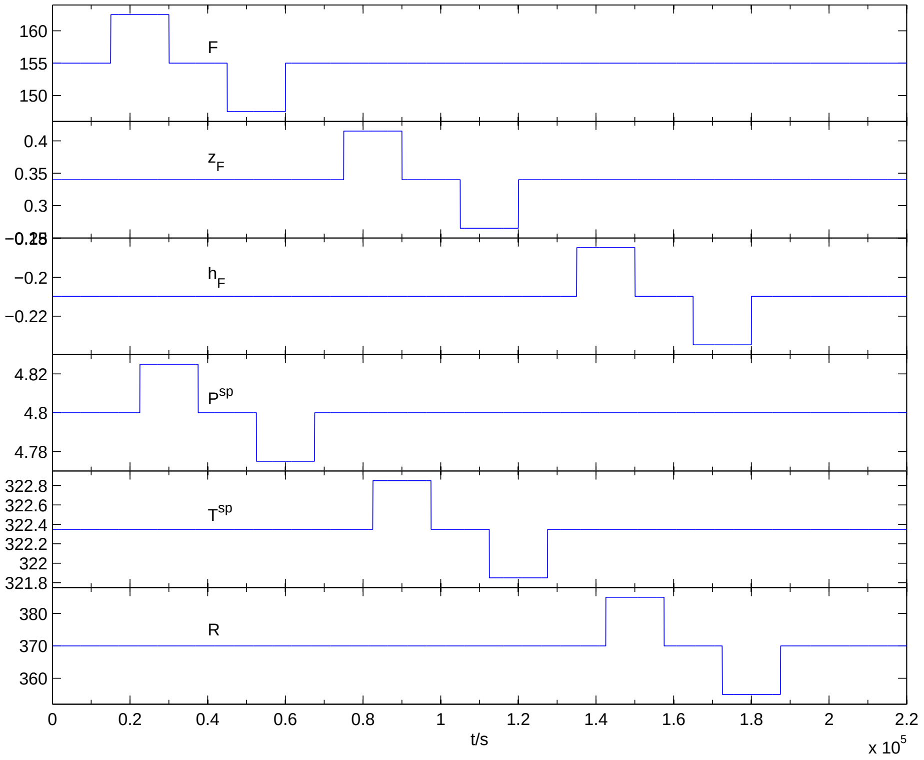

Figure 4.15: Inputs for validation of polynomial approximations.

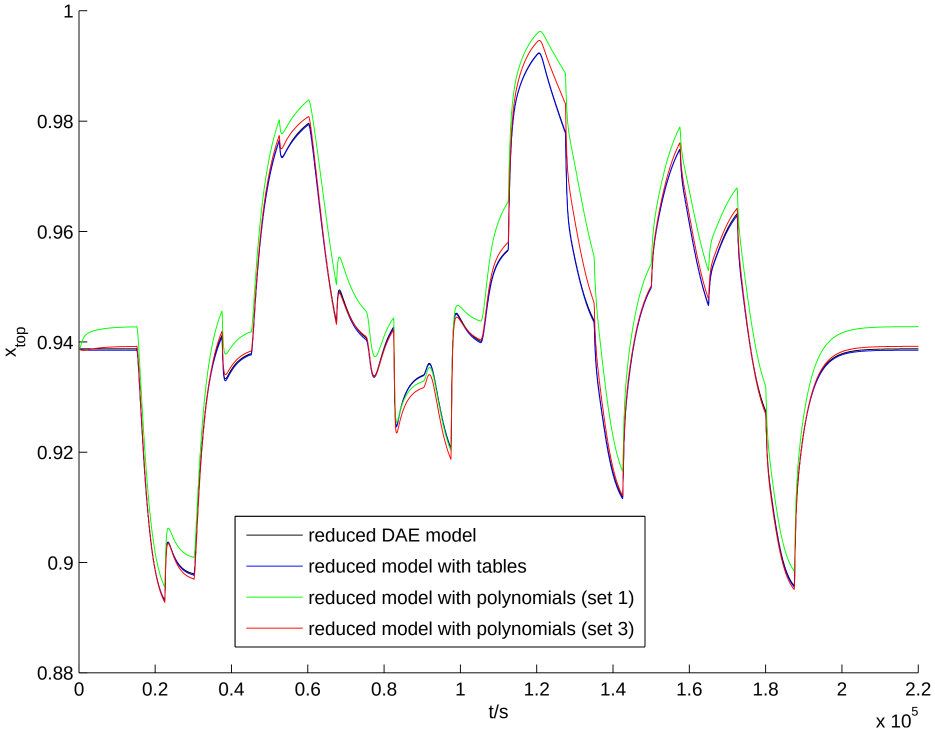

Figure 4.16: Top concentration responses of the reduced DAE model, the reduced model using tables, and the reduced models using polynomial approximations generated using set 1 and 3 of precomputed values. The input signal shown in figure 4.15 were used in the simulations.

independent variable combinations that possibly never occur during simulation of the model or even are unphysical.

If by the above measures, the polynomial can be made sufficiently accurate over a sufficiently large operating domain, a compact and fast reduced model can be obtained.
To get an impression of the complexity of one polynomial, the C-code of one function is shown below. The formula has been formulated such that the number of multiplications is minimal.

```
values[0]=(8302.871581)+ x[0]*((-51.34962423) + x[0]*((0.1183129784)
 +x[0]*(-6.744018695e-05)+x[1]*(-0.003538347307)+x[2]*(-8.844796191e-06)
 +x[3]*(-0.0001131827166)+x[4]*(0.0121784099))+ x[1]*((3.640607428)
 +x[1]*(-0.3074722478)+x[2]*(0.0001563842787)+x[3]*(0.005378120171)
 +x[4]*(-4.291683087))+ x[2]*((0.00701254359)
 +x[2]*(3.290599513e-07)+x[3]*(-6.911572432e-06)+x[4]*(-0.008678613214))
 +x[3]*((0.02796344744) +x[3]*(2.603275094e-05)+x[4]*(0.04903132101))
 +x[4]*((-0.01716118103)+x[4]*(20.39740589)))+ x[1]*((-820.1842509)
 +x[1]*((92.23974184)+x[1]*(1.80946475)+x[2]*(0.001841225865)
 +x[3]*(-0.08293531431)+x[4]*(39.41124615))+x[2]*((-0.02960154583)
 +x[2]*(-1.468357656e-06)+x[3]*(-0.000115030174)+x[4]*(0.009630403784))
 +x[3]*((-1.288562865) +x[3]*(0.0006923232074)+x[4]*(0.7779871165))
 +x[4]*((685.0774355) +x[4]*(202.5959761)))+ x[2]*((-0.6282230372)
 +x[2]*((7.663513239e-05) +x[2]*(-3.682851028e-09)+x[3]*(-5.487549325e-07)
 +x[4]*(6.714960292e-05))+x[3]*((-0.002819335659) +x[3]*(9.125629655e-06)
 +x[4]*(0.00713109264))+x[4]*((0.5388774869) +x[4]*(-1.503768968)))
 +x[3]*((-13.98609228) + x[3]*((0.03987401527)+x[3]*(-5.367430208e-05)
 +x[4]*(-0.1022428853))+ x[4]*((42.35943349) +x[4]*(-13.83671765)))
 +x[4]*((-8139.743094) + x[4]*((-3613.383398) +x[4]*(7621.053285)));
```

4.7.2 Simulation call graphs

Figures 4.17 and 4.18 show the call graphs of the simulations of the full and the reduced model, respectively. They visualize how the different numerical functions of the solver and the model code call each other, and show the percentage of the total execution time that is spent in each function. In the call graphs, each block represents a function, with the following information:

- name of the function (see below for details);
- percentage of total execution time spent in this function including execution times of subfunctions;

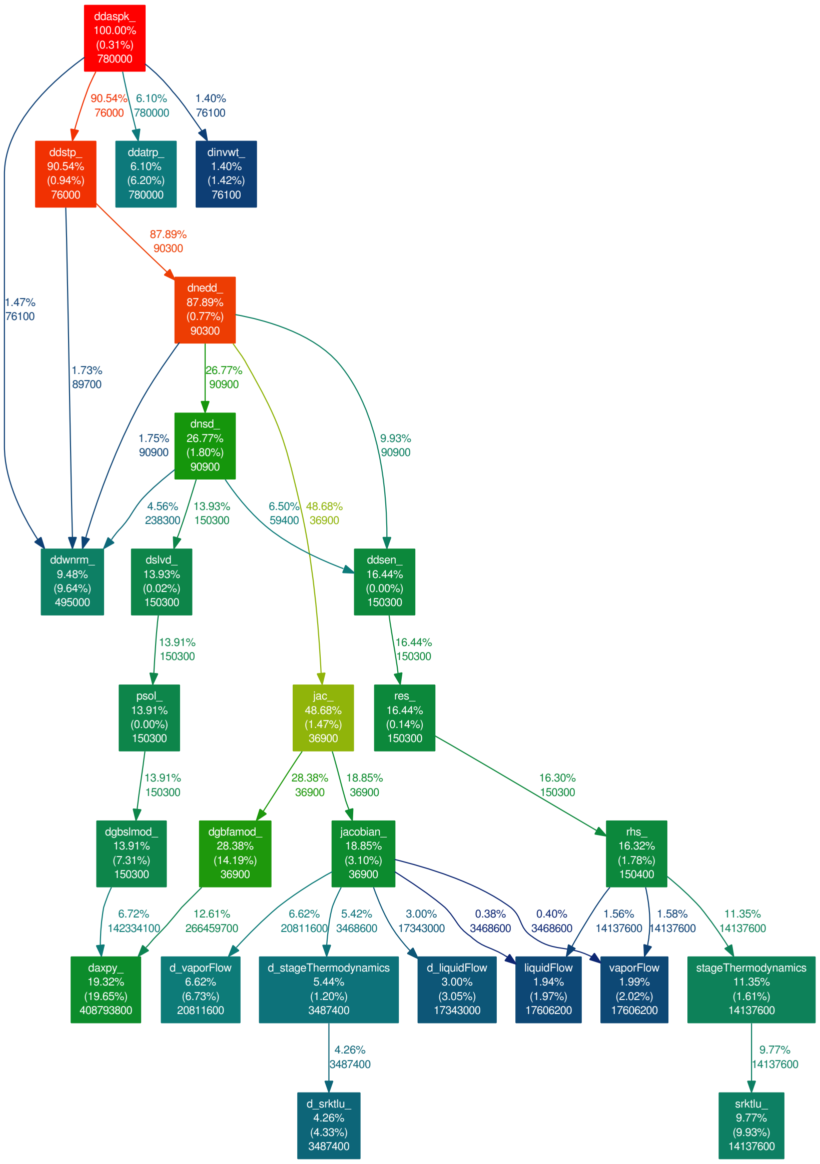
ddaspk_ 100.00% (0.31%) 780000
90.54% 76000
6.10% 780000
1.40% 76100
ddstp_ 90.54% (0.94%) 76000
ddatrp_ 6.10% (6.20%) 780000
dinvwt_ 1.40% (1.42%) 76100
87.89% 90300
1.47% 76100
dnedd_ 87.89% (0.77%) 90300
1.73% 89700
26.77% 90900
1.75% 90900
9.93% 90900
dnsd_ 26.77% (1.80%) 90900
4.56% 238300
13.93% 150300
6.50% 59400
48.68% 36900
ddwnrm_ 9.48% (9.64%) 495000
dslvd_ 13.93% (0.02%) 150300
ddsen_ 16.44% (0.00%) 150300
13.91% 150300
16.44% 150300
psol_ 13.91% (0.00%) 150300
jac_ 48.68% (1.47%) 36900
res_ 16.44% (0.14%) 150300
13.91% 150300
28.38% 36900
18.85% 36900
16.30% 150300
dgbslmod_ 13.91% (7.31%) 150300
dgbfamod_ 28.38% (14.19%) 36900
jacobian_ 18.85% (3.10%) 36900
rhs_ 16.32% (1.78%) 150400
6.72% 142334100
12.61% 266459700
6.62% 20811600
5.42% 3468600
3.00% 17343000
0.38% 3468600
0.40% 3468600
1.56% 14137600
1.58% 14137600
11.35% 14137600
daxpy_ 19.32% (19.65%) 408793800
d_vaporFlow 6.62% (6.73%) 20811600
d_stageThermodynamics 5.44% (1.20%) 3487400
d_liquidFlow 3.00% (3.05%) 17343000
liquidFlow 1.94% (1.97%) 17606200
vaporFlow 1.99% (2.02%) 17606200
stageThermodynamics 11.35% (1.61%) 14137600
4.26% 3487400
9.77% 14137600
d_srktlu_ 4.26% (4.33%) 3487400
srktlu_ 9.77% (9.93%) 14137600

Figure 4.17: Call graph of a simulation of the full model with simulation tolerance $\theta = 10^{-2.5}$.

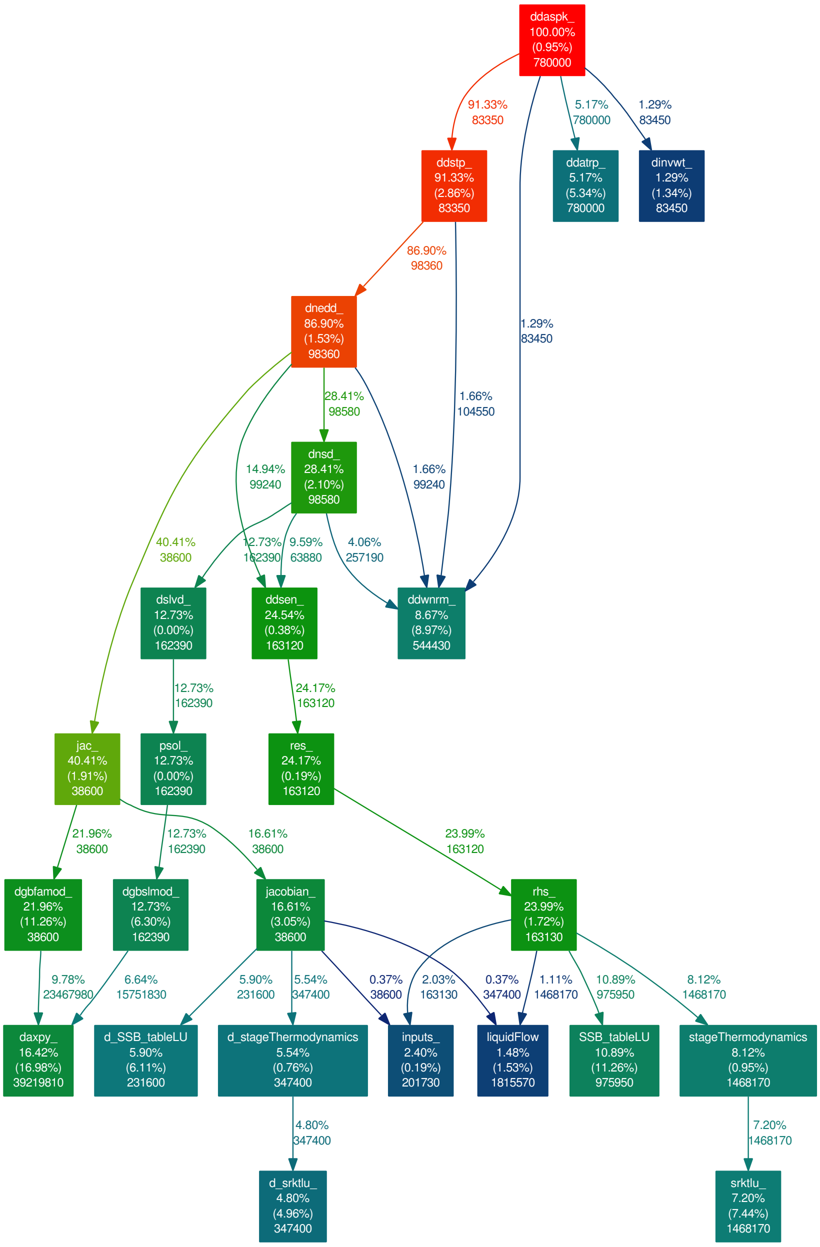

Figure 4.18: Call graph of a simulation of the reduced model with simulation tolerance $\theta = 10^{-2.5}$.

- (in brackets) percentage of total execution time spent only in this function (excluding execution times of subfunctions);
- total number of function executions. To obtain better statistics, each simulation was run 100 times.

Each arrow represents calls of a subfunction, with the following information:

- percentage of total execution time spent in the called function when called from the calling function, including execution times of subfunctions;
- total number of function calls from the calling function.

Only functions and calls that account for more than 1% of the total execution time are shown in the graphs. Both graphs were generated from simulations with the simulation tolerance $\theta = 10^{-2.5}$. At this tolerance, the reduced model shows the best performance (see figure 4.13).
The following are the main functions of the full and reduced model implementations:

- Model functions:
  - **rhs_**: computes the right-hand side (residual) of the DAE systems. It calls subfunctions for calculating the thermodynamic and hydraulic quantities in the column.
  - **jacobian_**: computes the analytic Jacobian of the DAE systems.
  - **stageThermodynamics**: computes the thermodynamic quantities on each stage.
  - **srktlu**: obtains the thermodynamic properties from tabulated SRK correlations using cubic spline interpolation.
  - **liquidFlow**: computes the liquid flow between two stages.
  - **vaporFlow**: computes the vapor flow between two stages.
  - **SSB_tableLU**: computes the tabulated variables of the steady-state blocks by multi-linear interpolation.

  - **d_...**: derivatives of the above functions.

- Solver functions:
  - **ddaspk_**: is the main function of the DASPK solver [18]. It is called iteratively by the main function to integrate the system starting from a certain start time for a certain interval.
  - **res_**: provides the residual of the DAE systems by directly calling the function **rhs_**.
  - **jac_**: provides the Jacobian of the DAE systems by calling the function **jacobian_**. In addition, it calls the function **dgbfamod** to decompose the Jacobian into lower and upper triangular matrices.
  - **psol_**: solves the nonlinear system of equations arising from the integration algorithm.
  - **ddwnrm**: computes the weighted root-mean-square norm of a vector.
  - **ddatrp**: performs interpolation to get an output solution.

- Linear algebra functions:
  - **dgbfamod_**: is a modified version of the LINPACK [20] LU-decomposition routine *dgbfa_* for banded matrices. The modification allows for the inclusion of off-band elements arising from decentralized control loops, as described in section 4.2.6.
  - **dgbslmod_**: is a modified version of the LINPACK LU-solution routine *dgbsl_* for banded matrices.
  - **daxpy_**: is a LINPACK routine for adding scaled matrix rows. It is the most intensively used subroutine of **dgbfamod** and **dgbslmod_**.

The remaining functions are mostly internal functions of the DASPK solver. They are, for example, used for step size control of the integration time step, and are documented in the DASPK code (Li and Petzold, 2000).

Chapter 5

An aggregation model reduction method for one-dimensional distributed parameter systems

submitted to Chemical Engineering Science.

Abstract

A new method for deriving reduced dynamic models of one-dimensional distributed parameter systems is presented. It inherits the concepts of the aggregated modeling method of Lévine and Rouchon (1991) for simple staged distillation models, and can be applied to both spatially discrete and continuous systems. It is based on partitioning the system into intervals of steady-state systems, which are connected by dynamic aggregation elements. By pre-solving and substituting the steady-state systems, a discrete low-order dynamic model is obtained, which asymptotically approaches the true steady-state of the original model. The method is an alternative to discretization methods like finite-difference and finite-element methods for spatially continuous systems, and collocation and wave propagation methods for spatially discrete systems. Implementation details of the method are discussed, and the principle is illustrated on three example systems, namely a distillation column, a heat exchanger and a fixed-bed reactor.

Keywords

Model reduction; Dynamic simulation; Mathematical modeling; Distributed systems; Aggregated modeling; Distillation

5.1 Introduction

This paper presents a new approach for deriving reduced dynamic models of spatially discrete or continuous one-dimensional distributed parameter systems. The reduced models are low-order systems of ordinary differential equations or differential-algebraic equations. For continuous systems, the method can be seen as an alternative to common spatial discretization methods such as finite-difference, finite-volume and finite-element methods (Hundsdorfer and Verwer, 2007).

The method is based on the concept of aggregation, which was used by Lévine and Rouchon (1991) for deriving reduced distillation models. Linhart and Skogestad (2009a) showed that this method can be used to increase the simulation speed several times, and extended the method to complex distillation models (Linhart and Skogestad, 2009b). In this case, the method is an alternative to other model reduction methods for this kind of one-dimensional separation processes, namely orthogonal collocation methods (Cho and Joseph, 1983, Stewart et al., 1984) and wave propagation methods (Marquardt, 1990, Kienle, 2000).

The method presented here is a generalization from distillation columns to one-dimensional spatially distributed parameter systems. These systems can be discrete in space, like stage-wise processes such as staged distillation columns, or continuous, like packed distillation columns, fixed-bed reactors, heat exchangers etc. A special class of discrete systems are spatial discretizations, for example using finite-differences, of continuously distributed systems. The reduction method can be applied to these systems in the same way as it is applied to spatially discrete systems. The reduction procedure for continuous systems can be derived as the limit case of these systems, where the reduction method is first applied to the discretized system, and then the limit case when the discretization interval goes to zero is considered. For continuous systems, the method is limited to spatially second-order systems.

The method is based on choosing several “aggregation points” on the spatial domain of the distributed system. To each of these aggregation points, dynamic “aggregation elements” are assigned. The partial differential equation or the discretely distributed system on the intervals between the aggregation points is treated as in steady-state. The values on the boundaries of the steady-state systems, which appear in the dynamic equations of the adjacent aggregation elements, are computed as functions of the states of the aggregation elements on both sides of each steady-state system. The thus obtained system is discrete and low-order in nature.

The main principle of the method is to replace the signal transport through the system by instantaneous transport through the steady-state intervals from aggregation element to aggregation element, where the transport is slowed down by the large capacities of the aggregation elements.

5.2 Method

In the following, the mathematical structure of the classes of spatially distributed systems that the method can be applied to is described. Subsequently, the reduction procedure, the conceptual steps of which are the same for all systems, is described. These are basically one-dimensional systems, which are spatially either discrete or continuous.

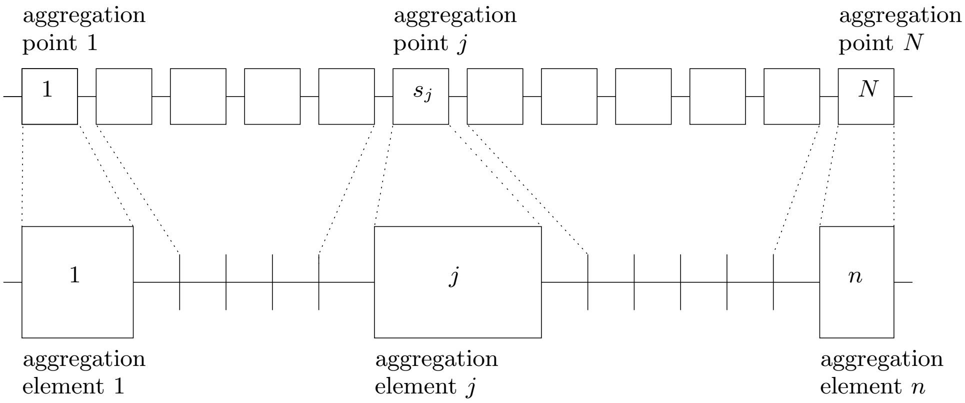

Figure 5.1: Schematic illustration of the reduction method for discrete distributed systems.

5.2.1 Discrete distributed parameter systems

The first class of systems the reduction method can be applied to are discrete one-dimensional distributed systems. Figure 5.1 shows the principal structure of these systems.

The main characteristic of these systems is that they consist of a number of consecutive similar units that communicate with the respective neighboring units along one dimension. For a mathematically convenient notation, the dynamic and algebraic equations of each unit are expressed in vector notation:

$$\begin{aligned}
\mathbf{M}_1\dot{\mathbf{x}}_1(t) &= \mathbf{f}_1(\mathbf{x}_1(t), \mathbf{x}_2(t), \mathbf{p}, t), && (5.1)\\
\mathbf{M}_i\dot{\mathbf{x}}_i(t) &= \mathbf{f}_i(\mathbf{x}_{i-1}(t), \mathbf{x}_i(t), \mathbf{x}_{i+1}(t), \mathbf{p}, t), && (5.2)\\
& \quad 2 \le i \le N-1, \\
\mathbf{M}_N\dot{\mathbf{x}}_N(t) &= \mathbf{f}_N(\mathbf{x}_{N-1}(t), \mathbf{x}_N(t), \mathbf{p}, t), && (5.3)
\end{aligned}$$

where i is the index of the unit, N is the total number of units, t is the time variable, $\mathbf{x}_i$ is the vector consisting of the dynamic and algebraic variables of unit i, $\mathbf{M}_i$ is a diagonal "mass" matrix that can be used to render

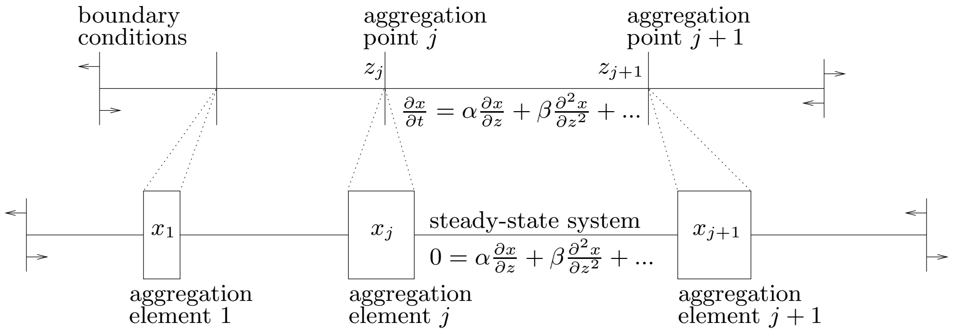
boundary conditions
aggregation point j
aggregation point $j+1$
z_j
z_{j+1}
$\frac{\partial x}{\partial t} = \alpha\frac{\partial x}{\partial z} + \beta\frac{\partial^2 x}{\partial z^2} + ...$
x_1
x_j
steady-state system
$0 = \alpha\frac{\partial x}{\partial z} + \beta\frac{\partial^2 x}{\partial z^2} + ...$
x_{j+1}
aggregation element 1
aggregation element j
aggregation element $j+1$

Figure 5.2: Schematic illustration of the reduction method for continuous distributed systems.

some of the equations algebraic by setting the corresponding value to 0, $\mathbf{f}_i$ is a vector-valued function of the variable vectors of the current and the neighboring units, and $\mathbf{p}$ is a parameter vector. External inputs to the system are included in the notation above by the time-dependency of the $\mathbf{f}_i$.

5.2.2 Continuous distributed parameter systems

The second class of systems are one-dimensional continuous distributed parameter systems, where the spatial order is restricted to a maximum of two. These systems can be written as vector-valued partial differential equations:

$$\frac{\partial \mathbf{x}(z,t)}{\partial t} = \mathbf{D}_z\mathbf{x}(z,t) + \mathbf{R}(\mathbf{x}(z,t),z,t), \qquad 0 \leq z \leq 1, \tag{5.4}$$

where $\mathbf{x}(z,t)$ is the vector of the distributed state variables, z is the spatial variable, t is the time, $\mathbf{D}_z$ is a spatial differential operator acting on the state vector $\mathbf{x}(z,t)$, and $\mathbf{R}(\mathbf{x}(z,t),z,t)$ is a local source term. A certain set of boundary conditions is needed to complete the description, which can also be time-dependent and thus contain external inputs to the system. For simplicity, the spatial domain of the partial differential equation is here chosen to be $[0;1]$. This is not a restriction, since any other spatial domain can be transformed to this by simple scaling of the spatial variable z.

5.2.3 General reduction procedure

Figures 5.1 and 5.2 illustrates the principle of the method. The procedure can be divided into the following steps, which are the same for both the discrete and continuous case:

1. **Selection of aggregation points**

   On the spatial domain of the system, n "aggregation points" are chosen. For discrete systems, these are n distinct indices of units $s_j, j = 1, ..., n$. For continuous systems, these are n points z_j with $0 \leq z_j \leq 1, j = 1, ..., n$.

   The number and position of the aggregation points will affect the dynamic approximation quality of the reduced system, but not the steady-states, and all choices will lead to a functional system.

2. **Introduction of aggregation elements**

   At every aggregation point, an "aggregation element" is positioned. In the discrete case, these elements are just the units at the aggregation points with a modified "capacity" H. In the continuous case, at every aggregation point, an aggregation element is positioned. Their dynamics are governed by simple differential equations that are derived from the original partial differential equation. The "capacity" H of an aggregation element refers to a factor that multiplies the left-hand sides of the dynamic equations of the element.

3. **Steady-state approximation of systems between aggregation elements**

   The equations on the intervals between the aggregation points are treated as in steady-state. In the discrete case, the left hand sides of all equations of the units that are not aggregation elements are set to 0. This results in systems of algebraic equations that depend on certain variables of the aggregation elements on both sides. In the continuous case, the partial differential equation on the intervals between the aggregation elements is treated as steady-state boundary value problems, where certain variables of the aggregation elements serve as boundary conditions.

4. **Precomputed solution of steady-state systems**

   The steady-state systems are solved either numerically or analytically for a range of possible values of the states of the aggregation elements on both sides of each system. For the integration of the aggregation element equations, the solutions on the boundaries of the steady-state systems have to be known. They are therefore expressed as functions of the state variables of the neighboring aggregation elements, and substituted into the aggregation element equations.

5. **Substitution of steady-state solutions**

   The functions computed in step 4 are substituted into the equations of the capacity elements. The resulting system is a set of ODEs (or DAEs, if algebraic equations are present).

Steps 1 to 3 yield a model with reduced dynamics. It is, however, of the same complexity as the original model. In the discrete case, a large number of dynamic equations have been converted into algebraic equations. In the continuous case, the continuous system has been partitioned into dynamic aggregation elements and boundary value problems, which have to be solved simultaneously. A real reduction in model complexity and computational effort is therefore obtained only after steps 4 and 5.
In the following, details specific for either discrete or continuous systems are described.

5.2.4 Discrete systems

After step 3, the equations of the reduced system read

$$H_1\mathbf{M}_1\dot{\mathbf{x}}_1(t) \quad = \quad \mathbf{f}_1(\mathbf{x}_1(t), \mathbf{x}_2(t), \mathbf{p}, t), \tag{5.5}$$

$$\begin{aligned} H_j \mathbf{M}_{s_j} \dot{\mathbf{x}}_{s_j}(t) &= \mathbf{f}_{s_j}(\mathbf{x}_{s_j-1}(t), \mathbf{x}_{s_j}(t), \mathbf{x}_{s_j+1}(t), \mathbf{p}, t), && (5.6)\\ &\quad j = 2, ..., n-1, \\ \mathbf{0} &= \mathbf{f}_i(\mathbf{x}_{i-1}(t), \mathbf{x}_i(t), \mathbf{x}_{i+1}(t), \mathbf{p}, t), && (5.7)\\ &\quad i = 2, ..., N-1, i \neq s_j, j = 1, ..., n, \\ H_n \mathbf{M}_N \dot{\mathbf{x}}_N(t) &= \mathbf{f}_N(\mathbf{x}_{N-1}(t), \mathbf{x}_N(t), \mathbf{p}, t). && (5.8) \end{aligned}$$

Here, to simplify notation, a case is written where unit 1 and N are aggregation elements. Either of these could be steady-state systems as well.

Step 4 involves solving the systems (5.7) for the variables $\mathbf{x}_{s_j-1}$ and $\mathbf{x}_{s_j+1}$, $j = 2, ..., n-1$. These are needed in the equations of the aggregation elements (5.5), (5.6) and (5.8). The variables are expressed as functions of the variables of the aggregation elements on both sides. This means that, for example, for aggregation element j, the functions

$$\mathbf{x}_{s_j+1} = \phi_j(\mathbf{x}_{s_j}, \mathbf{x}_{s_{(j+1)}}, \mathbf{p}) = \phi_j(\bar{\mathbf{x}}_j, \bar{\mathbf{x}}_{j+1}, \mathbf{p}), \tag{5.9}$$

and

$$\mathbf{x}_{s_j-1} = \psi_j(\mathbf{x}_{s_{(j-1)}}, \mathbf{x}_{s_j}, \mathbf{p}) = \psi_j(\bar{\mathbf{x}}_{j-1}, \bar{\mathbf{x}}_j, \mathbf{p}) \tag{5.10}$$

are required. Here, the variable $\mathbf{x}_{s_j+1}$ is a function of the variables $\mathbf{x}_{s_j}$ and $\mathbf{x}_{s_{(j+1)}}$ of aggregation elements s_j and $s_{(j+1)}$. Note the difference between the variables $\mathbf{x}_{s_j+1}$ and $\mathbf{x}_{s_{(j+1)}}$. The first denotes the first unit after the aggregation element unit j, whereas the latter denotes the aggregation element unit $j+1$. To make this difference clear, the notation $\bar{\mathbf{x}}_j$ is introduced, where the bar denotes the state variables of the aggregation elements.

Generally, these functions contain numerical solutions and have to be implemented in a suitable way. A straightforward way is the tabulation of the solution values over a certain domain of the independent variables, and the retrieval of the function values by interpolation of the table values. Whether the functions are implemented as lookup tables or in another way, they will be complex if the dimensionality of the $\mathbf{x}_i$ variables is high. It is therefore advisable to choose the independent variables carefully, since not all variables necessarily are needed to compute the function values. In addition, often not all variables in the vectors $\mathbf{x}_{s_j-1}$ and $\mathbf{x}_{s_j+1}$ are needed in the aggregation element equations.

Step 5 implies the substitution of the functions (5.9) and (5.10) into the aggregation element equations (5.5), (5.6) and (5.8). The resulting system then reads

$$\begin{aligned} H_1\bar{\mathbf{M}}_1\dot{\bar{\mathbf{x}}}_1(t) &= \bar{\mathbf{f}}_1(\bar{\mathbf{x}}_1(t), \phi_1(\bar{\mathbf{x}}_1(t), \bar{\mathbf{x}}_2(t), \mathbf{p}), \mathbf{p}, t), & (5.11) \\ H_j\bar{\mathbf{M}}_j\dot{\bar{\mathbf{x}}}_j(t) &= \bar{\mathbf{f}}_j(\psi_j(\bar{\mathbf{x}}_{j-1}, \bar{\mathbf{x}}_j, \mathbf{p}), \bar{\mathbf{x}}_j(t), \phi_j(\bar{\mathbf{x}}_j, \bar{\mathbf{x}}_{j+1}, \mathbf{p}), \mathbf{p}, t), & (5.12) \\ & \quad j = 2, ..., n-1, & \\ H_n\bar{\mathbf{M}}_n\dot{\bar{\mathbf{x}}}_n(t) &= \bar{\mathbf{f}}_n(\psi_n(\bar{\mathbf{x}}_{n-1}, \bar{\mathbf{x}}_n, \mathbf{p}), \bar{\mathbf{x}}_n(t), \mathbf{p}, t). & (5.13) \end{aligned}$$

Here, the notation $\bar{\mathbf{M}}$, $\bar{\mathbf{x}}$ and $\bar{\mathbf{f}}$ is used to indicate a change of index of the variables and functions due to the elimination of the steady-state variables and equations. For every j, $\bar{\mathbf{x}}_j = \mathbf{x}_{s_j}$ etc. holds.

5.2.5 Continuous systems: second order systems

The differential equations of the aggregation elements for continuous systems can be derived by applying the reduction procedure to a finite-difference discretization of the partial differential equation, and considering the limit case of $\Delta z \to 0$, where Δz is the length of the finite-difference intervals. The result of this operation depends on the nature of the continuous distributed system, more specifically, on the order of the spatial differential operator. The main derivation is demonstrated here for a system with second-order spatial derivatives, which represents a typical convection-diffusion-reaction system. The differences in the procedure for systems with first-order spatial derivatives are discussed in the next section.
The system discussed in this section reads

$$\frac{\partial x}{\partial t} = -\alpha\frac{\partial x}{\partial z} + \beta\frac{\partial^2 x}{\partial z^2} + R(x), \qquad (5.14)$$

with a certain set of boundary conditions, and α and β being dimensionless numbers. For simplicity of notation, a scalar system is used for the deriva-

tion of the reduced model equations.
A finite-difference discretization of the spatial derivatives yields

$$\frac{dx_i}{dt} = -\alpha\frac{x_i - x_{i-1}}{\Delta z} + \beta\frac{x_{i-1} - 2x_i + x_{i+1}}{\Delta z^2} + R(x_i), \tag{5.15}$$

where x_i are the states of the discretized system at the N distinct discretization points $z_i, i = 1, ..., N$, which span the spatial domain over intervals of length $\Delta z = 1/(N-1)$.
According to step 1 and 2, a number n of aggregation points $z_{s_j}, j = 1, ..., n$, is chosen among all discretization points, and the differential equations of the corresponding states are modified by multiplying the left hand side with a "capacity" H_j:

$$\begin{aligned} H_j\frac{dx_{s_j}}{dt} &= -\alpha\frac{x_{s_j} - x_{s_j-1}}{\Delta z} + \beta\frac{x_{s_j-1} - 2x_{s_j} + x_{s_j+1}}{\Delta z^2} + R(x_{s_j})(5.16) \\ & \quad j = 1, ..., n. \end{aligned}$$

Step 3 implies that the remaining equations are treated as in steady-state:

$$\begin{aligned} 0 &= -\alpha\frac{x_i - x_{i-1}}{\Delta z} + \beta\frac{x_{i-1} - 2x_i + x_{i+1}}{\Delta z^2} + R(x_i), \\ & \quad i = 1, ..., N, i \neq s_j, j = 1, ..., n. \end{aligned} \tag{5.17}$$

The resulting model has the same steady-state as the original discretized model. The capacities H_j can be chosen freely, but should compensate for the missing capacities of the steady-state elements. A straightforward choice for a reduced model with equidistant aggregation points is therefore $H_j = N/n$, which distributes the capacities of the discretized states of the original discretized model equally among the aggregation points of the reduced model. N is expressed in terms of Δz as $N = 1/\Delta z + 1$, such that the equations of the aggregation elements read

$$\begin{aligned} \frac{\frac{1}{\Delta z} + 1}{n}\frac{dx_{s_j}}{dt} &= -\alpha\frac{x_{s_j} - x_{s_j-1}}{\Delta z} + \beta\frac{\frac{x_{s_j+1} - x_{s_j}}{\Delta z} - \frac{x_{s_j} - x_{s_j-1}}{\Delta z}}{\Delta z} \\ & \quad + R(x_{s_j}), \end{aligned} \tag{5.18}$$

The second-order finite-difference approximation is here written as the finite-difference of two first-order finite-differences. Multiplying with Δz yields

$$\begin{aligned}\frac{1+\Delta z}{n}\frac{dx_{s_j}}{dt} &= -\alpha(x_{s_j}-x_{s_j-1}) \\ &+\beta\left(\frac{x_{s_j+1}-x_{s_j}}{\Delta z}-\frac{x_{s_j}-x_{s_j-1}}{\Delta z}\right) \\ &+R(x_{s_j})\Delta z.\end{aligned} \tag{5.19}$$

$\Delta z \to 0$ yields the continuous equations. Since the system discussed here is a continuous second-order system, $x_{s_j-1} \to x_{s_j}$ for $\Delta z \to 0$. This is not the case if the system is first-order. This case will be discussed separately below. Thus, $\Delta z \to 0$ results in

$$\frac{1}{n}\frac{d\bar{x}_j}{dt} \; := \; \frac{1}{n}\frac{dx_{s_j}}{dt} = \beta\left(\left.\frac{\partial x}{\partial z}\right|_{z_j}^{+} - \left.\frac{\partial x}{\partial z}\right|_{z_j}^{-}\right). \tag{5.20}$$

The notation $\bar{x}_j$ is introduced here to express that the only remaining state variables are the states at the aggregation points, and $\bar{x}_j = x_{s_j}$.

In step 4, the right derivative $\left.\frac{\partial x}{\partial z}\right|_{z_j}^{+}$ is calculated from the boundary value systems between the aggregation points z_j and z_{j+1},

$$0 \; = \; -\alpha\frac{\partial x}{\partial z} + \beta\frac{\partial^2 x}{\partial z^2} + R(x), \quad z_j \leq z \leq z_{j+1}, \tag{5.21}$$

with the boundary conditions

$$x(z_j) \; = \; \bar{x}_j, \tag{5.22}$$

$$x(z_{j+1}) \; = \; \bar{x}_{j+1}, \tag{5.23}$$

and the left derivative $\left.\frac{\partial x}{\partial z}\right|_{z_j}^{-}$ is calculated from the boundary value systems between the aggregation points z_{j-1} and z_j correspondingly. The solution can be obtained, for example, by using a finite-difference approximation as in equations (5.17). From the solution of a steady-state system (5.21) between the aggregation points z_j and z_{j+1} with the boundary conditions (5.22) and (5.23), the derivatives $\frac{\partial x}{\partial z}\Big|_{z_j}^{+}$ and $\frac{\partial x}{\partial z}\Big|_{z_{j+1}}^{-}$ can be calculated as functions of the states of the aggregation elements:

$$\frac{\partial x}{\partial z}\bigg|_{z_j}^{+} = \phi_j(\bar{x}_j, \bar{x}_{j+1}), \tag{5.24}$$

$$\frac{\partial x}{\partial z}\bigg|_{z_{j+1}}^{-} = \psi_{j+1}(\bar{x}_j, \bar{x}_{j+1}), \tag{5.25}$$

$$j = 2, ..., n-1.$$

For $j = 1$ or $j = n$, the boundary conditions of the original system can be used to solve equation (5.21). The resulting left and right derivatives depend then either only on one aggregation element variable, for example

$$\frac{\partial x}{\partial z}\bigg|_{1}^{+} = \phi_N(\bar{x}_n) \tag{5.26}$$

for independent boundary conditions on the right side, or, when the boundary conditions are cyclic, on the states of the aggregation elements on both ends of the system:

$$\frac{\partial x}{\partial z}\bigg|_{0}^{-} = \psi_1(\bar{x}_1, \bar{x}_n). \tag{5.27}$$

Step 5 implies the substitution of these functions into equations (5.20) to yield the final reduced model

$$\frac{1}{n}\frac{d\bar{x}_j}{dt} = \beta\left(\phi_j(\bar{x}_j, \bar{x}_{j+1}) - \psi_j(\bar{x}_{j-1}, \bar{x}_j)\right). \tag{5.28}$$

5.2.6 Continuous systems: first order systems

A partial differential equation with first-order spatial derivatives reads

$$\frac{\partial x}{\partial t} = -\alpha\frac{\partial x}{\partial z} + R(x), \tag{5.29}$$

with a certain set of boundary conditions, and α being a dimensionless number. The same procedure for steps 1, 2 and 3 as in section 5.2.5 is applied. The equations for the steady-state systems (5.17) now read

$$\begin{aligned} 0 &= -\alpha\frac{x_i - x_{i-1}}{\Delta z} + R(x_i), \\ & \quad i = 1, ..., N, i \neq s_j, j = 1, ..., n. \end{aligned} \tag{5.30}$$

These are the discretizations of the continuous steady-state systems

$$0 = -\alpha\frac{\partial x(z)}{\partial z} + R(x(z)), \quad z_j \leq z \leq z_{j+1}, \tag{5.31}$$

with the single boundary condition on the left side

$$x(z_j) = \bar{x}_j, \tag{5.32}$$

where $x(z)$ denotes the spatially distributed states of the steady-state system j between the aggregation points z_j and z_{j+1}, and $\bar{x}_j$ is the state of aggregation element j on the left side of the system. This implies that the values of the variables on the right side of the steady-state systems are generally not the same as the variable values of the adjacent aggregation element, but depend on the left boundary condition:

$$x(z_{j+1}) = \psi_{j+1}(\bar{x}_j). \tag{5.33}$$

The limit case of equation (5.19) of section 5.2.5, which now reads

$$\frac{1 + \Delta z}{n}\frac{dx_{s_j}}{dt} = -\alpha(x_{s_j} - x_{s_j-1}) + R(x_{s_j})\Delta z,$$

is therefore

$$\frac{1}{n}\frac{d\bar{x}_j}{dt} = -\alpha(\bar{x}_j - \psi_j(\bar{x}_{j-1})). \tag{5.34}$$

5.2.7 Steady-state preservation property

The characteristic property of the aggregation model reduction method is that both the original and the reduced models assume identical steady-states. This means that

1. if the states of the reduced model assume the values of the steady-state profile of original system at the corresponding positions of the aggregation points, the reduced model is in steady-state, and
2. if the reduced model is in steady-state, the profile of the aggregation elements with the interconnecting steady-state systems coincides with the unique steady-state profile of the original system.

To show this, it is assumed that there exists a unique steady-state for the original system. For continuous systems, the **argument** is restricted to systems with spatial derivatives of order up to two, and the steady-state profile of the original system is assumed to be differentiable.

The discrete case is trivial to show, since at steady-state, the equations of the original system (5.1)-(5.3) and the equations of the reduced system (5.5)-(5.8) are identical. Since uniqueness of the solution is assumed, the solutions are identical as well.

In the continuous case, the two parts can be shown separately. The argument is given for a second-order system; first-order systems follow as a special case.

1. Since the states of the aggregation elements lie on the unique steady-state profile of the original system (5.14), the profiles of the steady-state systems between the aggregation elements coincide with the the corresponding parts of the steady-state profile of the original model. Differentiability of the profile of the original system implies that the left and right derivatives at each aggregation element as in equation (5.20) coincide, and the equations are at steady-state.
2. On the steady-state systems between the aggregation points of the reduced model (5.21), the equations of the original system (5.14) are

satisfied at steady-state. Since the boundary conditions of the steady-state systems are the states of the aggregation elements, the profile of the connected steady-state systems is continuous. Since the reduced model is in steady-state, equation (5.20) implies that the first-order spatial derivatives of the steady-state systems on both sides of each aggregation points assume the same values. Then, by equation (5.21), the second-order derivatives of the steady-state systems assume the same values on both sides of each aggregation point. This means that the profile resulting from connecting all steady-state profiles satisfies the original system (5.14) at steady-state on the complete domain and is therefore the unique solution of the original system (5.14) at steady-state.

5.3 Examples

Three simple examples are discussed to illustrate the method.

5.3.1 Distillation column

Model

As an example for a discrete system, a staged distillation column is considered. This example has been discussed in detail in Linhart and Skogestad (2009a). Therefore, the derivation of the model is described here only very briefly.
The original model reads

$$\begin{aligned}
H_1\dot{x}_1 &= Vy_2 - Vx_1, && (5.35)\\
H_i\dot{x}_i &= Lx_{i-1} + Vy_{i+1} - Lx_i - Vy_i, && (5.36)\\
&\quad i = 2, ..., i_F - 1, &&\\
H_{i_F}\dot{x}_{i_F} &= Lx_{i-1} + Vy_{i+1} - (L+F)x_i - Vy_i &&\\
&\quad +Fz_F, && (5.37)\\
H_i\dot{x}_i &= (L+F)x_{i-1} + Vy_{i+1} - (L+F)x_i - Vy_i, && (5.38)
\end{aligned}$$

$$\begin{aligned} & i = i_F + 1, ..., N-1, & \\ H_N \dot{x}_N &= (L+F)x_{N-1} - (L+F-V)x_N - Vy_N, & (5.39) \end{aligned}$$

where H_i is the total liquid molar holdup, x_i and $y_i = k(x_i)$ are the concentrations of the first component in the liquid and vapor phase, respectively, of stage i, N is the number of stages including the condenser and reboiler, i_F is the index of the feed stage, V and L are the liquid and vapor flows in the column, respectively, and F and z_F are the feed flow rate and the feed concentration, respectively. The molar holdups, liquid and vapor flows are assumed to be constant. The energy balance is simplified using the constant relative volatility assumption

$$y_i = k(x_i) = \frac{\alpha x_i}{1 + (\alpha - 1)x_i}. \qquad (5.40)$$

After applying steps 1 to 3 of the model reduction method, the reduced model equations read

$$\begin{aligned} \bar{H}_1 \dot{\bar{x}}_1 &= Vk(\bar{x}_2) - V\bar{x}_1, & (5.41) \\ \bar{H}_j \dot{\bar{x}}_{s_j} &= L\bar{x}_{s_j-1} + Vk(\bar{x}_{s_j+1}) - L\bar{x}_{s_j} - Vk(\bar{x}_{s_j}), & (5.42) \\ & j = 2, ..., n-1, j \neq j_F & \\ \bar{H}_{j_F} \dot{\bar{x}}_{i_F} &= L\bar{x}_{i_F-1} + Vk(\bar{x}_{i_F+1}) - (L+F)\bar{x}_{i_F} - Vk(\bar{x}_{i_F}) & \\ & +Fz_F, & (5.43) \\ 0 &= L\bar{x}_{i-1} + Vk(\bar{x}_{i+1}) - L\bar{x}_i - Vk(\bar{x}_i), & (5.44) \\ & i = 2, ..., N-1, i \neq s_j, j = 1, ..., n & \\ \bar{H}_n \dot{\bar{x}}_N &= (L+F)\bar{x}_{N-1} - (L+F-V)\bar{x}_N - Vk(\bar{x}_N), & (5.45) \end{aligned}$$

where n is the number of aggregation stages, $\bar{H}_j$ and s_j are the aggregated holdup and the index of aggregation stage j, respectively, and j_F is the index of the aggregation stage where the feed is entering. The terms "aggregation stage" and "aggregated holdup" are used here for the more general terms "aggregation element" and "capacity" as used in section 5.2. Steps 4 and 5 imply the solution of the algebraic equations and the substitution of the required solutions Y_j into the dynamic equations.

$$\bar{H}_1 \dot{\tilde{x}}_1 = VY_1(\tilde{x}_1, \tilde{x}_2, V/L) - V\tilde{x}_1, \qquad (5.46)$$

parameter	value
N	74
n_F	36
H_1	20 mol
H_N	20 mol
$H_i, i = 2, ..., N-1$	1 mol
α	1.33
input	nominal value
z_F	0.45
F	0.04 mol/s
L	0.12 mol/s
V	0.14 mol/s

Table 5.1: Parameters of the distillation column model.

$$\begin{aligned}
\bar{H}_j \dot{\tilde{x}}_j &= L\tilde{x}_{j-1} + VY_j(\tilde{x}_j, \tilde{x}_{j+1}, V/L) - L\tilde{x}_j \\
&\quad -VY_{j-1}(\tilde{x}_{j-1}, \tilde{x}_j, V/L), \qquad (5.47) \\
&\quad j = 2, ..., n-1, j \neq j_F \\
\bar{H}_{j_F} \dot{\tilde{x}}_{j_F} &= L\tilde{x}_{j_F-1} + VY_{j_F}(\tilde{x}_{j_F}, \tilde{x}_{j_F+1}, V/L) - L\tilde{x}_{j_F} \\
&\quad -VY_{j_F-1}(\tilde{x}_{j_F-1}, \tilde{x}_{j_F}, V/L) + Fz_F, \qquad (5.48) \\
\bar{H}_n \dot{\tilde{x}}_n &= (L+F)\tilde{x}_{n-1} - (L+F-V)\tilde{x}_n \\
&\quad -VY_{n-1}(\tilde{x}_{n-1}, \tilde{x}_n, V/(L+F)). \qquad (5.49)
\end{aligned}$$

The functions Y_j correspond to the functions ϕ_j in equation (5.9). Due to mass conservation of the steady-state systems (5.44), only the functions ϕ, but not the functions ψ are needed. The model parameters are given in table 5.1. Reduced models of a more complex distillation model with complex hydrodynamic and thermodynamic relationships have been described in Linhart and Skogestad (2009b).

aggregation stage index j	1	2	3	4	5	6	7
s_j	1	36	74				
$\bar{H}_j$	20	72	20				
s_j	1	14	36	60	74		
$\bar{H}_j$	20	21	28	23	20		
s_j	1	8	20	36	53	67	74
$\bar{H}_j$	20	10	15	19	18	10	20

Table 5.2: Positions and holdups of the aggregation stages of the reduced models.

Simulation study

Figure 5.3 shows the responses of the top and bottom concentrations of the full distillation model with 74 stages, and reduced distillation models with 3, 5 and 7 aggregation stages, to a step change in the feed concentration z_F from 0.45 to 0.55. The reduced model parameters, i.e. the position of the aggregation stages and their aggregated holdups, are given in table 5.2. They are taken from Linhart and Skogestad (2009a). The parameter sets for the models with 5 and 7 aggregation stages are “optimized” to minimize the deviation from the original model over a broad range of changes in the feed concentration z_F and liquid and vapor flows L and V as described in Linhart and Skogestad (2009a). However, the optimization is restricted to the position and the aggregated holdups of the aggregation stages except reflux drum and reboiler, and constrained to the requirement that the sum of the aggregation stage capacities equals to the number of stages in the system. Therefore, there is no degree of freedom for the model with 3 aggregation stages. If these restrictions are lifted, better approximation quality, especially for the model with 3 aggregation stages, can be expected. It can be seen that especially the approximation quality of the reduced model with 7 aggregation stages is very good. This model has less than 10% of the states of the full model. The gain in computation time of the models has been shown in Linhart and Skogestad (2009a) to be in the same

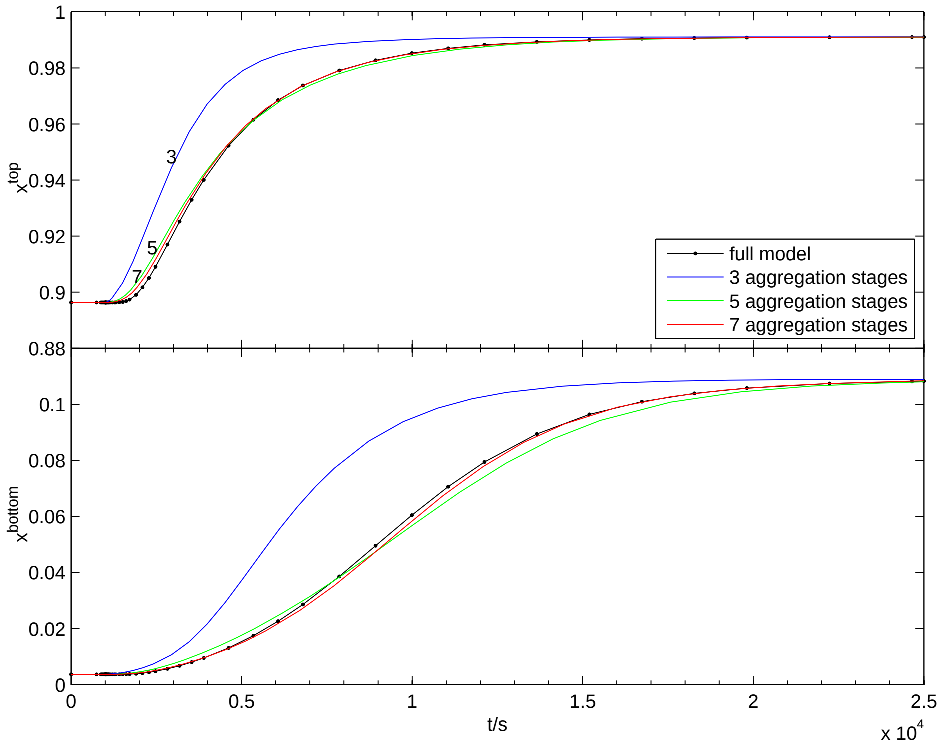

Figure 5.3: Distillation model top and bottom concentration responses to step change in feed concentration z_F from 0.45 to 0.55.

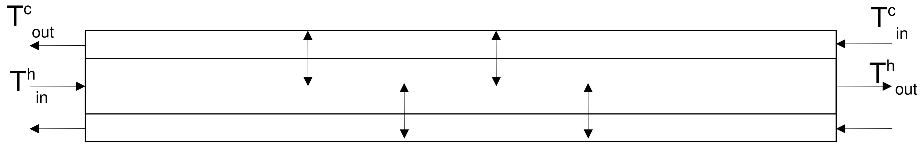

Figure 5.4: Schematic diagram of a tubular heat exchanger.

order of magnitude as the reduction in the number of states.

5.3.2 Heat exchanger

Model

As an example of a first-order continuous system, a tubular counter-current heat exchanger is considered (see figure 5.4). A description of these types of heat exchangers can be found in Skogestad (2008). The partial differential equations for this system are in the form of equation (5.29) and read

$$\begin{aligned} \frac{\partial T^h}{\partial t} &= -v^h \frac{\partial T^h}{\partial z} - \frac{Up}{A^h \rho^h c_p^h}(T^h - T^c), && (5.50)\\ \frac{\partial T^c}{\partial t} &= v^c \frac{\partial T^c}{\partial z} + \frac{Up}{A^c \rho^c c_p^c}(T^h - T^c),\ 0 < z < l, && (5.51)\\ T^h(t,0) &= T^h_{in}, && (5.52)\\ T^c(t,l) &= T^c_{in}, && (5.53) \end{aligned}$$

where T^h, T^c, v^h, v^c, A^h, A^c, ρ^h, ρ^c, c_p^h, and c_p^c are the temperatures, flow velocities, tube cross-sectional areas, densities and heat capacities of the hot and the cold streams, respectively, U and p are the heat transmission coefficient and the perimeter of the surface between the hot and cold stream, respectively, l is the tube length, and T^h_{in} and T^h_{in} are the inlet temperatures of the hot and the cold stream, respectively. The main assumptions in this model are incompressible fluids, temperature-independent fluid properties,

parameter	value
ρ^c	1000 kg/m^3
ρ^h	1000 kg/m^3
A^c	0.0314 m^2
A^h	0.0393 mol
c_p^c	0.0314 kJ/(kgK)
c_p^h	0.0393 kJ/(kgK)
U	0.5 kW/m^2
p	0.6283 m
input	nominal value
m^c	2 kg/s
m^h	1 kg/s
T_{in}^c	320 K
T_{in}^h	360 K

Table 5.3: Parameters of the heat exchanger model.

no diffusive heat transport, and negligible heat capacity of the tube walls. The parameter values are given in table 5.3.

A straightforward choice of n aggregation points according to step 1 of the reduction procedure is an equal-distribution of the aggregation points over the whole domain with the end points placed at the ends of the heat exchanger:

$$s_j = \frac{j-1}{n-1}l, \; j = 1, ..., n. \tag{5.54}$$

The heat exchanger equations are a combination of two counter-current transport equations. The dynamic equations for the aggregation elements can therefore be derived from equation (5.34) to be

$$C_j \frac{d\bar{T}_j^h}{dt} = -\frac{v^h}{l}(\bar{T}_j^h - \psi_j(\bar{T}_{j-1}^h, \bar{T}_j^c)), \tag{5.55}$$

$$Cj \frac{d\bar{T}_j^c}{dt} = -\frac{v^c}{l}(\bar{T}_j^c - \phi_j(\bar{T}_j^h, \bar{T}_{j+1}^c)), \tag{5.56}$$

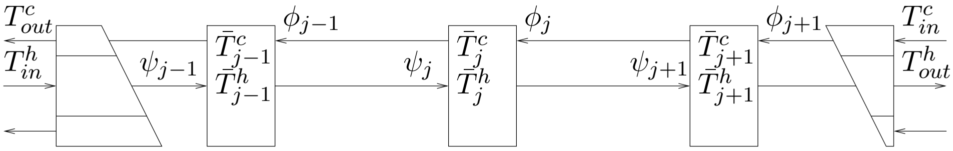

Figure 5.5: Schematic diagram the reduced heat exchanger model.

where C_j is the capacity of aggregation element j, and ψ_j and ϕ_j are the solutions of the steady-state systems on the left and right sides of aggregation element j, respectively. Figure 5.5 shows a schematic diagram of the reduced model.

A straightforward choice for the capacities is $C_j = 1/n$. This way, the continuously distributed heat capacity of the original model is equally distributed over the aggregation elements.

For heat exchangers, analytic steady-state solutions are available (Kern, 1950):

$$\begin{bmatrix} T^h_{out} \\ T^c_{out} \end{bmatrix} = \frac{1}{1-R^c a}\begin{bmatrix} 1-R^c & R^c(1-a) \\ 1-a & a(1-R^c) \end{bmatrix}\begin{bmatrix} T^h_{in} \\ T^c_{in} \end{bmatrix}. \tag{5.57}$$

This expression can be used in step 4 of the reduction procedure to calculate the steady-state functions ϕ and ψ:

$$\begin{bmatrix} \psi_j \\ \phi_{j-1} \end{bmatrix} = \frac{1}{1-R^c a}\begin{bmatrix} 1-R^c & R^c(1-a) \\ 1-a & a(1-R^c) \end{bmatrix}\begin{bmatrix} \bar{T}^h_{j-1} \\ \bar{T}^c_j \end{bmatrix}. \tag{5.58}$$

Here, $\bar{T}^h_{j-1}$ and $\bar{T}^c_j$ are the temperatures of the neighboring aggregation elements $j-1$ and j of the steady-state system (compare figure 5.5).

The parameters R^c and a are defined as follows:

$$R^c = \frac{m^c c^c_p}{m^h c^h_p}, \tag{5.59}$$

where m^h and m^c are the mass flows of the hot and cold stream, respectively, which are connected to the flow velocities v^h and v^c by the respective densities ρ^h and ρ^c and the tube cross-sectional areas:

$$m^h = v^h A^h \rho^h, \qquad (5.60)$$

$$m^c = v^c A^c \rho^c. \qquad (5.61)$$

a is defined as

$$a = exp(-N_{TU}^c(1 - R^c)), \qquad (5.62)$$

where the number of transfer units N_{TU}^c is defined as

$$N_{TU}^c = \frac{Up}{m^c c_p^c}. \qquad (5.63)$$

Simulation study

To demonstrate the approximation quality of the reduced models, figures 5.6 to 5.9 compare the responses of reduced models with 2, 5 and 30 aggregation elements with finite-difference approximations with 100 and 2000 finite-differences. The simulation with 2000 finite-differences is referred to as the exact solution.

The variables that are compared are the outlet temperatures T_{out}^h and T_{out}^c of the hot and the cold stream, respectively. In the given models, they are identical with the corresponding temperatures of the aggregation stages at both ends of the heat exchanger, i.e. $T_{out}^c = T_1^c$ and $T_{out}^h = T_n^h$. Figure 5.6 shows the responses to a step in the hot stream inlet temperature T_{in}^h from 360 K to 370 K. It can be seen that the response of the cold stream outlet temperature T_{out}^c, which is located at the same side as the hot stream inlet, is approximated very well by the reduced models. The response of the model with 30 aggregation elements is almost indistinguishable from the exact solution. All reduced aggregation models perfectly reproduce the steady-state. The finite-difference approximation with 100 elements shows a certain steady-state deviation from the reference solution. For this heat exchanger model, this deviation can be corrected rather easily (Mathiesen, 1994). However, without any modification of the finite-difference models, the aggregated models achieve a certain approximation quality with much

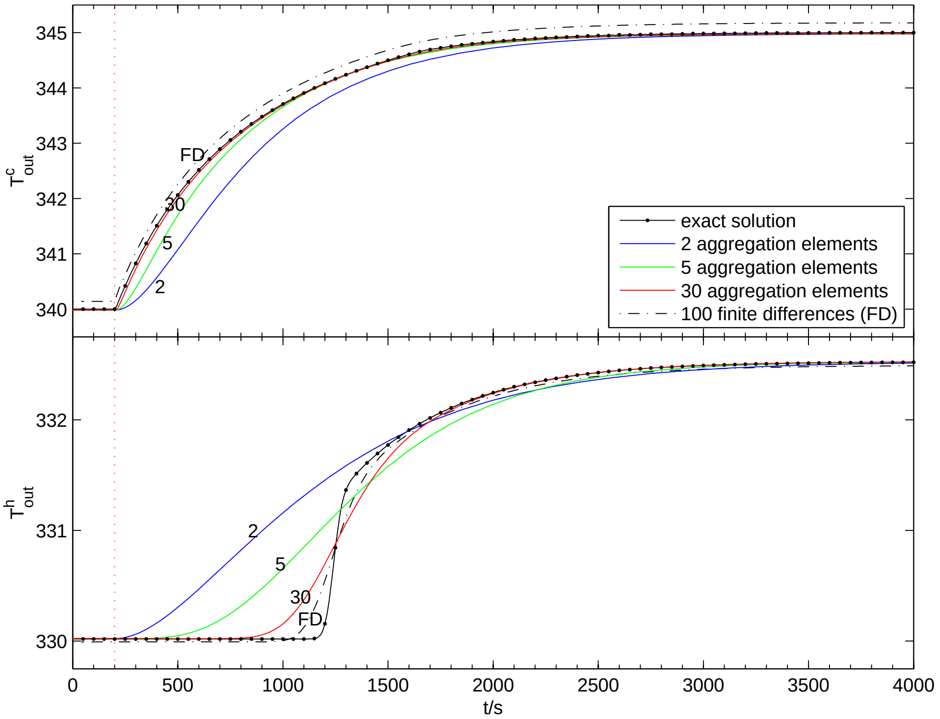

Figure 5.6: Heat exchanger outlet temperature responses of cold (upper plot) and hot (lower plot) streams to a step change in the hot inlet temperature T^h_{in}. The dotted vertical line marks the time when the step change is applied.

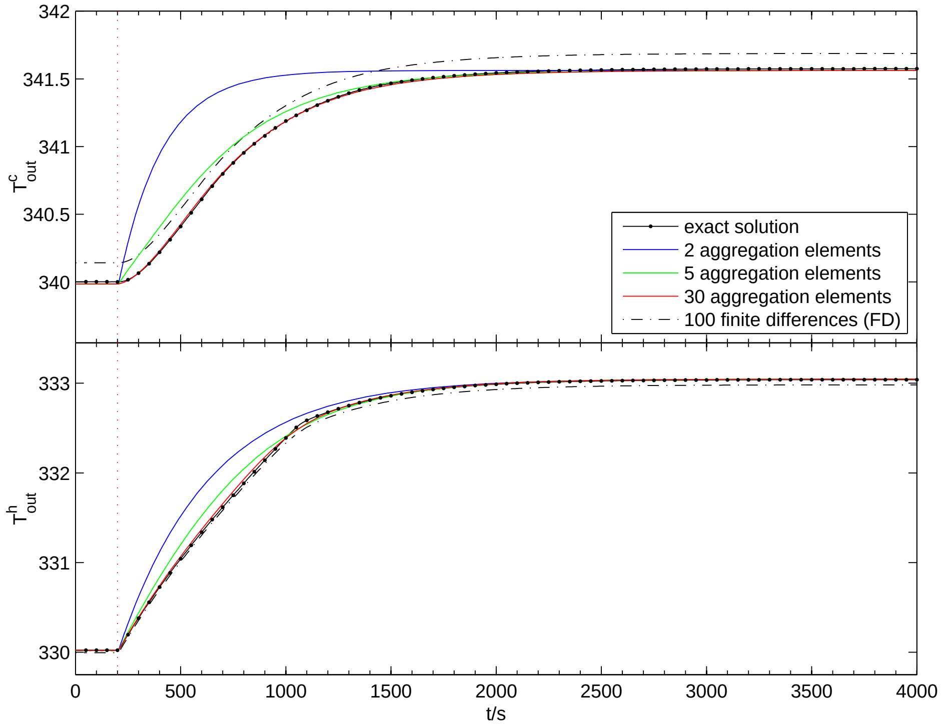

Figure 5.7: Heat exchanger outlet temperature responses of cold (upper plot) and hot (lower plot) streams to a step change in the hot stream flow rate v^h.

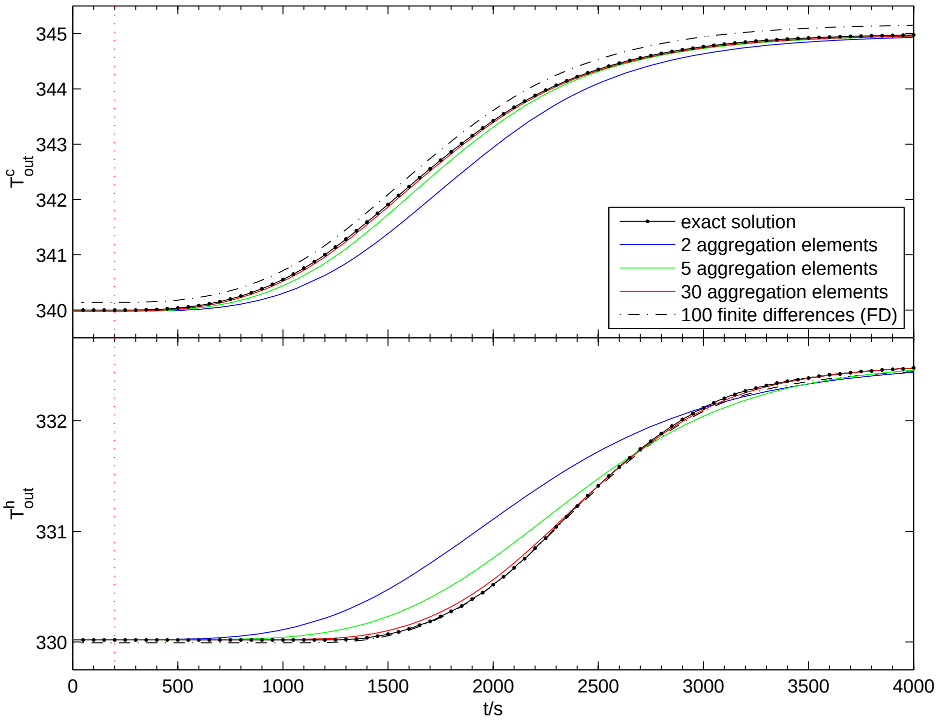

Figure 5.8: Heat exchanger outlet temperature responses of cold (upper plot) and hot (lower plot) streams to a slow change in the hot inlet temperature T^h_{in}.

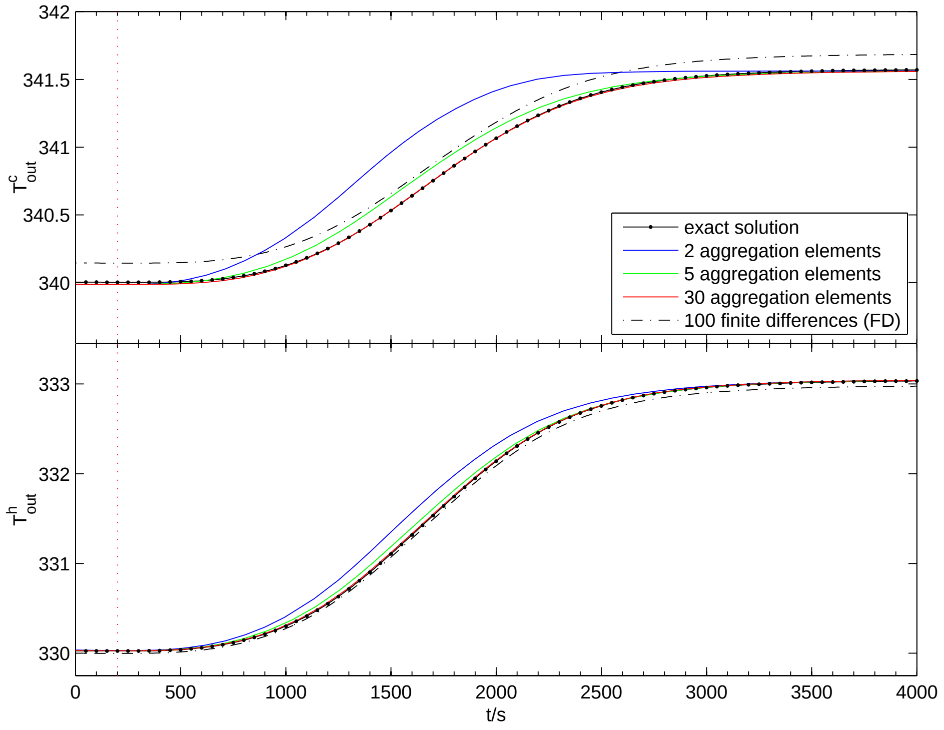

Figure 5.9: Heat exchanger outlet temperature responses of cold (upper plot) and hot (lower plot) streams to a slow change in the hot stream flow rate v^h.

less dynamic states.

The response of the hot stream outlet temperature T^h_{out} (lower part of figure 5.6) shows a dead-time period, which is characteristic for transport systems, since the hot stream outlet is located on the opposite side of the hot stream inlet where the change is applied. The approximation quality of the reduced aggregation models is rather poor here, since a dead-time system requires a model of high dynamic order for good approximation. Therefore, the 100 finite-difference approximation is superior to the aggregated model with 30 aggregation elements. Still, the aggregated models show a better approximation towards the steady-state.

Figure 5.7 shows the responses to a 20% step change in the hot stream flow rate v^h. This is approximated very well by the model with 30 aggregation elements. Since the fluid is assumed incompressible, the flow rate changes simultaneously throughout the whole system. Due to the increased velocity of the hot fluid, both the temperature of the hot and cold outlet streams rise. The transport characteristic of the system is still present in the response of the hot stream outlet temperature T^h_{out}, where the initial slope is flattened for the residual time of the hot fluid in the system.

Figures 5.8 and 5.9 show the responses to slow changes in T^h_{in} and v^h, respectively. Here, the input signal is a cubic spline curve with a transient time of 1000 s. Generally, the approximation quality of the reduced models with 5 and 30 aggregation elements is good. The approximation of the dead-time period of the hot stream outlet temperature T^h_{out} (lower part of figure 5.8) is much better than in case of a step change. This is explicable by the diffusive character of the heat exchange between the counter-current flows, which is more dominant in this case, and is approximated better by the reduced models.

5.3.3 Fixed bed reactor

Model

As an example of a second-order continuous system, an adiabatic fixed-bed reactor model investigated by Liu and Jacobsen (2004) is considered (see

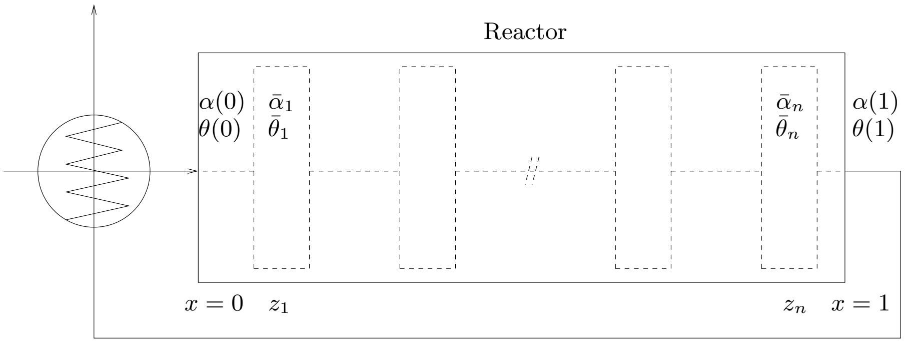

Figure 5.10: Schematic diagram of a fixed bed reactor with heat recycle. The structure of the reduced model is shown schematically using dashed lines.

figure 5.10):

$$\sigma\frac{\partial \alpha}{\partial t} = -\frac{\partial \alpha}{\partial x} + \frac{1}{Pe_m}\frac{\partial^2 \alpha}{\partial x^2} + DaR(\alpha, \theta), \tag{5.64}$$

$$\frac{\partial \theta}{\partial t} = -\frac{\partial \theta}{\partial x} + \frac{1}{Pe_h}\frac{\partial^2 \theta}{\partial x^2} + DaR(\alpha, \theta), \tag{5.65}$$

which is in form of equation (5.14). Here, α is the conversion, θ a dimensionless temperature, and the reaction term is given by

$$R(\alpha, \theta) = (1-\alpha)^r exp\left(\gamma\frac{\beta\theta}{1+\beta\theta}\right). \tag{5.66}$$

The boundary conditions are

$$\alpha(0,t) = \frac{1}{Pe_m}\frac{\partial \alpha}{\partial x}\bigg|_{x=0}, \tag{5.67}$$

$$\theta(0,t) = f\theta(1,t) + \frac{1}{Pe_h}\frac{\partial \theta}{\partial x}\bigg|_{x=0}, \tag{5.68}$$

$$\frac{\partial \alpha}{\partial x}\bigg|_{x=1} = 0, \tag{5.69}$$

$$\frac{\partial \theta}{\partial x}\bigg|_{x=1} = 0. \tag{5.70}$$

The derivation of a reduced model for this system is shown in detail in section 5.2.5. For the purpose of demonstrating the approximation quality of the reduced models, models derived using steps 1 to 3 are sufficient. If a gain in computational performance is desired, the steady-state systems have to be eliminated from the model using step 4 and 5. All aggregation points are chosen at locations z_j inside the domain of the partial differential equation, i.e. $0 < z_j < 1, j = 1, ..., n$. Therefore, the boundary conditions of the original model have to be included in the solutions of the steady-state systems on the boundary of the system. The left boundary condition (5.68) is special in a way that it includes the state $\theta(1,t)$ on the right side of the system. This results in expressions of the form

$$\left.\frac{\partial\alpha}{\partial x}\right|_{x_1}^{-} = \psi_1^{\alpha}(\bar{\alpha}_1, \bar{\theta}_1), \tag{5.71}$$

$$\left.\frac{\partial\theta}{\partial x}\right|_{x_1}^{-} = \psi_1^{\theta}(\bar{\alpha}_1, \bar{\theta}_1, \bar{\alpha}_n, \bar{\theta}_n) \tag{5.72}$$

for the left side, and

$$\left.\frac{\partial\alpha}{\partial x}\right|_{x_n}^{+} = \phi_n^{\alpha}(\bar{\alpha}_n, \bar{\theta}_n), \tag{5.73}$$

$$\left.\frac{\partial\theta}{\partial x}\right|_{x_n}^{+} = \phi_n^{\theta}(\bar{\alpha}_n, \bar{\theta}_n) \tag{5.74}$$

for the right side of the system.

Simulation study

To demonstrate the approximation quality of the reduced models, figures 5.11 and 5.12 compare the responses of reduced models with 5, 15 and 30 aggregation elements with finite-difference approximations with 100 and 2000 finite differences. The simulation with 2000 finite-differences is referred to as the exact solution.

Liu and Jacobsen (2004) show that the system exhibits a complex bifurcation behavior when Da is chosen as bifurcation parameter. At $Da =$

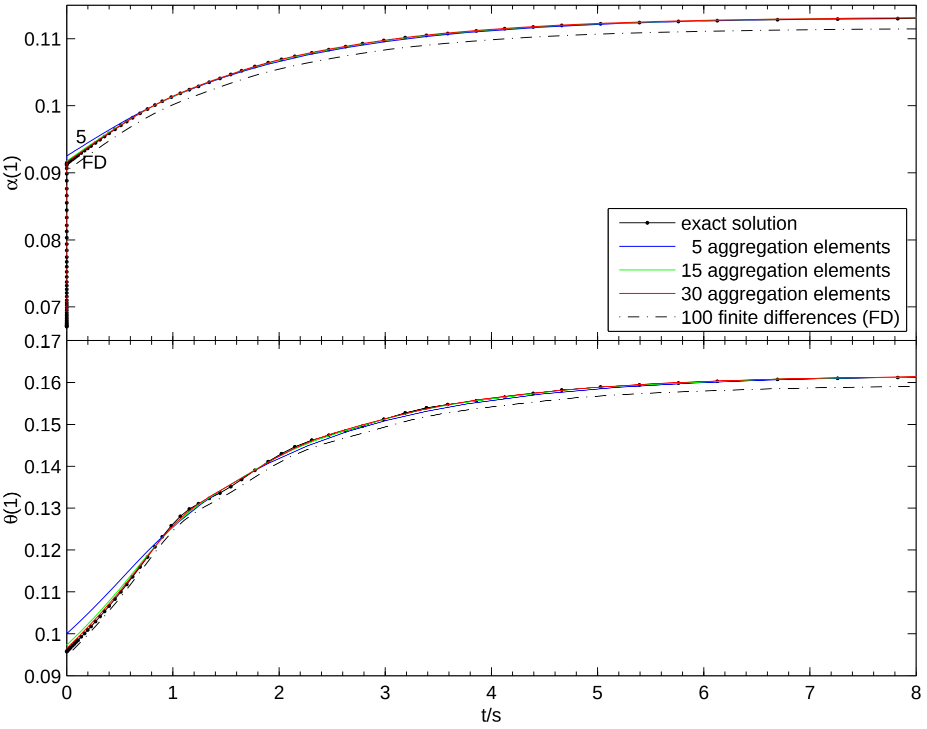

Figure 5.11: Responses of fixed-bed reactor conversion α and temperature θ at the right end to a change of Da from 0.05 to 0.07.

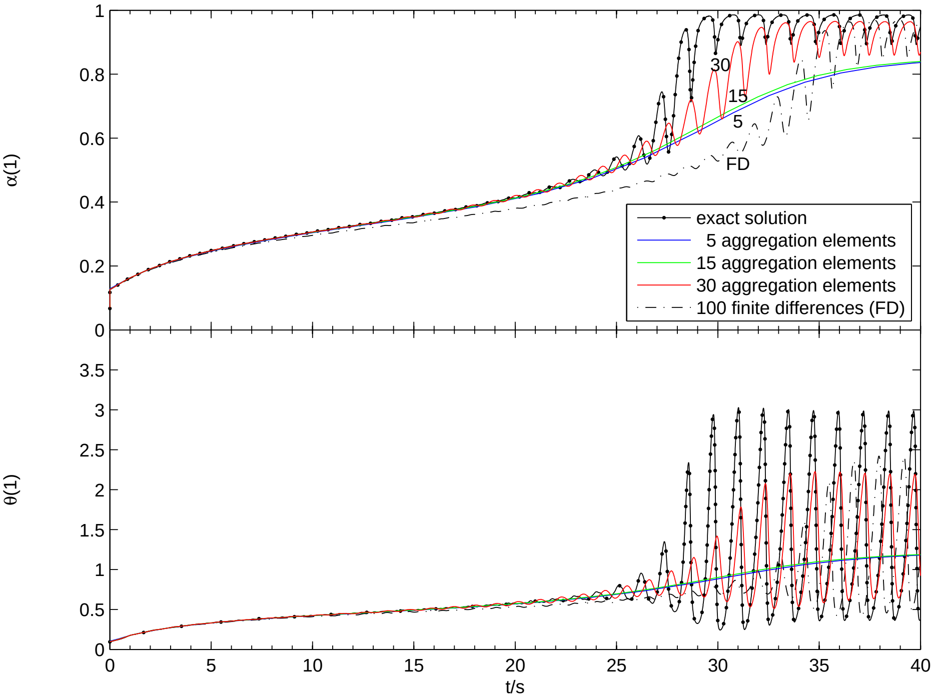

Figure 5.12: Responses of fixed-bed reactor conversion α and temperature θ at the right end simulated to a change of Da from 0.05 to 0.1.

0.05 and $Da = 0.07$, the system has one stable steady-state, whereas at $Da = 0.1$, the steady-state is unstable, and the system performs limit cycle oscillations.

Figure 5.11 shows the trajectories of α and θ at the right end of the reactor, when a step change in Da from 0.05 to 0.07 is applied. The trajectories show a fast initial change in α, which is due to the small parameter σ multiplying the left-hand side of equation (5.64). After that, the system performs a slow transient to a stable steady-state at $Da = 0.07$. It can be seen that the approximation quality of all reduced models is excellent, except for some deviation of the model with 5 aggregation elements in the beginning of the slow transient phase. While the reduced aggregation models perfectly reproduce the steady-state of the original system, the 100 finite-differences approximation shows a certain deviation.

Figure 5.12 shows the trajectories of the same variables, when a larger step change in Da from 0.05 to 0.1 is applied. At $Da = 0.1$, the system exhibits high-frequency limit-cycle oscillations. It can be seen that the approximation quality of all reduced models of the slow motion towards the limit-cycle oscillations is excellent. The reduced model with 30 aggregation elements is also capable to reproduce the fast limit-cycle oscillations. It is remarkable that the reduced model can follow the fast movement despite its slow nature.

5.4 Discussion

5.4.1 Advantages and limitations of the aggregation method

The method presented in this paper is conceptually straightforward. The good approximation quality of the reduced models has been demonstrated in several examples. The approximation quality can even be improved by optimizing the location and capacities of the aggregation elements for the given problem.

The main limitation of the method lies in step 4 and 5 of the method. The problem is the high dimension of the functions that have to be substituted into the dynamic equations if the original system has a large number of spatially distributed state variables. In Linhart and Skogestad (2009b), the method was applied to a complex distillation model containing energy balances and complex thermodynamic and hydraulic relationships. There, substitution was possible by using five-dimensional tables with linear interpolation. If, on the other hand, simple analytic solutions for the steady-state systems as in case for the heat exchanger model are available, the reduction method is easy to apply and yields models of good approximation quality.

5.4.2 Relationship to singular perturbation models

In the following, the reduction procedure of this study is compared to the procedure to derive slow reduced models in singular perturbation theory (Kokotovic et al, 1986, Lin and Segel, 1988). This is done for discrete systems. Since the continuous procedure is derived using the discrete procedure, the argument applies to continuous systems as well.

Singular perturbation procedure

In singular perturbation theory, systems with dynamics on two or more time-scales are analyzed mathematically. For this, a system

$$\frac{d\mathbf{x}}{dt} = \mathbf{f}(\mathbf{x}, \mathbf{u}), \tag{5.75}$$

is transformed into the standard form of singular perturbations

$$\begin{aligned} \frac{d\mathbf{y}}{dt} &= \mathbf{f}(\mathbf{y}, \mathbf{z}, \mathbf{u}), & (5.76) \\ \varepsilon\frac{d\mathbf{z}}{dt} &= \mathbf{g}(\mathbf{y}, \mathbf{z}, \mathbf{u}), & (5.77) \end{aligned}$$

where $\mathbf{y}$ is a vector of “slow” variables, $\mathbf{z}$ is a vector of “fast” variables, and $\varepsilon << 1$ is a small singular perturbation parameter. This is usually achieved by scaling the original equations and by a transformation of the state vector $\mathbf{x}$. In general, there is no unique procedure to choose the scaling of the equations or the state transformation.

If the time-scales of the system are sufficiently separated, and the scaling and state transformation is suitable, then equations (5.76) and (5.77) represent the slow and the fast dynamics in the system, respectively. Then, these equations can be used for further analysis of the system. One common procedure is to apply the quasi-steady-state assumption $\varepsilon \rightarrow 0$ to equation (5.77), thus obtaining the reduced slow model

$$\begin{aligned} \frac{d\mathbf{y}}{dt} &= \mathbf{f}(\mathbf{y}, \mathbf{z}, \mathbf{u}), && (5.78) \\ 0 &= \mathbf{g}(\mathbf{y}, \mathbf{z}, \mathbf{u}). && (5.79) \end{aligned}$$

Here, the dynamic equations (5.77) are converted into the algebraic equations (5.79). This is one reason why ε is called the singular perturbation parameter. Depending on the time-scale separation and the appropriate transformation of the system, this system approximates the original dynamics more or less accurately. Due to the replacement of the fast equations by algebraic relationships, the fast dynamics are approximated by "instantaneous" dynamics. This is significant for changes in the inputs $\mathbf{u}$, where the response of the slow model is actually faster than the response of the original model. The term "slow model" therefore refers to the internal dynamics of the reduced model, and not to its input-output behavior.

If a low-order model is desired, equations (5.79) can be solved for $\mathbf{z}$,

$$\mathbf{z} = \mathbf{h}(\mathbf{y}, \mathbf{u}), \tag{5.80}$$

and used for eliminating the fast variables $\mathbf{z}$ from the slow model

$$\frac{d\mathbf{y}}{dt} = \mathbf{f}(\mathbf{y}, \mathbf{h}(\mathbf{y}, \mathbf{u}), \mathbf{u}). \tag{5.81}$$

This is, however, only possible if equations (5.79) can be solved explicitly for $\mathbf{z}$.

Comparison with aggregation method

To compare the singular perturbation procedure with the aggregation method proposed in this study, it can first be observed that after step 3 of the aggregation method described in this study, the system is basically in the form of equations (5.78) and (5.79). Steps 4 and 5 correspond to the procedure in equations (5.80) and (5.81). The main difference of the procedures lies in the derivation of the form (5.78) and (5.79). In contrast to the singular perturbation procedure, the aggregation method does not use a state transformation and scaling of the equations to arrive at this form. Instead, the left-hand sides of the dynamic equations are manipulated in a way that cannot be achieved by a state transformation and scaling. The method does therefore not rely on the existence of a time-scale separation in the system. Instead, the method is based on approximating the spatial signal transport through the system by instantaneous transport through intervals connected by large capacity elements. This is an artificial construction, which deviates from the treatment of singular perturbation systems.

Lévine and Rouchon (1991) derive their method for staged distillation columns, which ultimately leads to the reduction procedure described in this study, as a singular perturbation method. They partition the column into compartments of consecutive stages, and use a singular perturbation procedure to separate the time-scales created by the ratio of the large compartment holdups and the small stage holdups. This time-scale separation is, however, not present in the original model, since the compartments are introduced completely artificially. The reason that the resulting models still approximate the original model sufficiently well is the simplification of certain terms during the quasi-steady-state approximation due to the incorrect introduction of the singular perturbation parameter ε. As a consequence, the compartment boundaries do not appear anymore in the resulting models. If a reduced model is derived without this simplification, it shows some unphysical inverse response, which is clear evidence of the incorrect introduction of the singular perturbation parameter. This is discussed in detail in Linhart and Skogestad (2009a).

The crucial property for the success of the aggregated models is the perfect reproduction of the steady-state. This property the aggregated models have in common with slow singular perturbation models. Both their derivation and their dynamics can therefore be said to be related.

5.4.3 Comparison with other numerical discretization schemes

As mentioned before, the method introduced in this study can be seen as a discretization method for continuous systems. A good overview of these methods for equations of the type of equation (5.14) is given in Hundsdorfer and Verwer (2007). In the following, some qualitative similarities and differences will be discussed.

Steady-state approximation

One difference between the aggregation method and other methods such as finite-volume and finite-element methods is immediately obvious: the aggregation method perfectly reproduces the steady-state even when the number of dynamic states is zero, while the above mentioned methods achieve this only in the limit case when the number of dynamic states approaches infinity. This is due to the incorporation of steady-state information into the aggregated models, which is not the case in the other methods.

finite-element methods

In finite-element methods, the solution is approximated by weighted sums of basis functions, which are usually polynomials. The weights of the basis functions are determined by inserting the approximation into the original equations and weighting the residual over the spatial domain by certain functions. If these functions are the basis functions themselves, the method is called a Galerkin method. In collocation methods, the residual is required to vanish at certain discrete points, the so-called collocation points. This method is popular in chemical engineering for the reduction of distillation models (Cho and Yoseph, 1983, Stewart et al., 1984). The efficiency of the method is based on the assumption that the solution profiles can be approximated by polynomials. In order to account for solution profiles that are difficult to approximate with polynomials over the whole spatial interval, the latter can be divided into finite-elements, on each of which a polynomial approximation by collocation is used. This procedure is therefore different from the aggregation procedure. Collocation models might be superior in approximating the fast responses of a system, whereas aggregation models will show better approximation of the behavior of systems that are close to steady-state.

5.5 Conclusions

A new approach for deriving reduced models of one-dimensional distributed systems is presented in this paper. The approach extends the aggregated modeling method of Lévine and Rouchon (1991) to general discrete and continuous one-dimensional systems. The main idea is the approximation of the spatial transport of signals through the system by instantaneous transport through intervals of steady-state systems, which are connected by aggregation elements of large capacity, which slow down the system dynamics to match the dynamics of the original system. The most important property of the method is the perfect reproduction of the steady-state of the original system. The method has been demonstrated on three example systems that are typical in process engineering. The method presents an alternative method to established spatial discretization methods such as finite-differences and finite-elements for spatially continuous systems, and to methods such as collocation or wave propagation methods for spatially discrete models. The approximation quality of the reduced models depends on the number, position and capacity of the aggregation elements. Generally, a good approximation quality can be achieved with a relative low number of aggregation elements compared to other discretizations methods. The implementation effort of the reduced models depends on the difficulty to express the solutions of the steady-state systems as functions of the aggregation element variables in a suitable way.

5.6 Acknowledgment

The authors thank Johannes Jäschke for fruitful discussions. This work has been supported by the European Union within the Marie-Curie Training Network PROMATCH under the grant number MRTN-CT-2004-512441.

Bibliography

[1] Cho, Y. S., and Joseph, B., 1983. Reduced-Order Steady-State and Dynamic Models for Separation Processes. Part I. Development of the Model Reduction Procedure. AIChE Journal 29, 261-269.

[2] Hundsdorfer, W., and Verwer, J. G., 2007. Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations. Springer, Berlin.

[3] Kern, D. Q., 1950. Process Heat Transfer. McGraw-Hill, New York.

[4] Kienle, A., 2000. Low-order dynamic models for ideal multicomponent distillation processes using nonlinear wave propagation theory. Chemical Engineering Science 55, 1817-1828.

[5] Kokotovic, P., Khalil, H. K., and O'Reilly, J., 1986. Singular Perturbation Methods in Control: Analysis and Design. SIAM classics in applied mathematics 25, SIAM, London.

[6] Lévine, J., and Rouchon, P., 1991. Quality Control of Binary Distillation Columns via Nonlinear Aggregated Models. Automatica 27, 463-480.

[7] Lin, C. C., and Segel, L. A., 1988. Mathematics Applied to Deterministic Problems in the Natural Sciences, SIAM Classics in Applied Mathematics 1, SIAM, London.

[8] Linhart, A., and Skogestad, S., 2009. Computational performance of aggregated distillation models. Computers & Chemical Engineering 33, 296-308.

[9] Linhart, A., and Skogestad, S., 2009. Reduced distillation models via stage aggregation. In preparation.

[10] Liu, Y., and Jacobsen, E. W., 2004. On the use of reduced order models in bifurcation analysis of distributed parameter systems. Computers & Chemical Engineering 28, 161-169.

[11] Mathisen, K. W., 1994. Integrated design and control of heat exchanger networks, PhD thesis, Norwegian university of science and technology, Trondheim.

[12] Marquardt, W., 1990. Traveling waves in chemical processes. International Chemical Engineering 30, 585-606.

[13] Skogestad, S., 2008. Chemical and energy process engineering. CRC Press, Boca Raton.

[14] Stewart, W. E., Levien, K. L., and Morari, M., 1984. Simulation of fractionation by orthogonal collocation. Chemical Engineering Science 40, 409-421.

Chapter 6

Conclusions and directions for future work

6.1 Conclusions

In this thesis, a model reduction method for one-dimensionally distributed systems has been presented. For staged distillation column models, which are discrete one-dimensional systems, the method is not new, but is based on the aggregated modeling method of Lévine and Rouchon. To extend the method to more complex staged processes with mass and energy balances, the concept of compartments and the interpretation of the system as a singular perturbation system had to be abandoned. Instead, only the notion of aggregation stages remains. By a simple manipulation of the left-hand sides of the differential equations, a reduced model is obtained. This procedure is straightforward to apply to any kind of staged process. However, as was shown in chapter 3, there is no computational benefit to be expected from such a reduced model. Therefore, as a second step, the algebraic equations resulting from the reduction procedure have to be eliminated from the model. Due to the one-dimensional structure of the model, the steady-state stage equations can be solved block-wise, and the solutions can be substituted into the dynamic equations of the aggregation stages. If this solution and substitution can be done successfully, the reduced model can be simulated several times faster than the full model. The factor of speed-up is approximately the ratio of the number of stages of the full model to the number of aggregation stages of the reduced model. However, the evaluation of the right-hand sides and the Jacobian of the reduced model might be computationally more intensive due to the substitution of the steady-state stage solutions.

It was shown in addition that the parameters of the reduced model, namely the position and aggregated holdup factors of the reduced models, can be used as tuning parameters for the performance of the reduced models. While a straightforward choice of the parameters yield models with acceptable approximation quality, the performance of the models can be improved significantly by optimizing the parameters for a certain application.

In chapter 4, the method has been applied to a complex distillation model with mass and energy balances, and complex thermodynamic and hydraulic relationships. It was shown that the approximation quality of a model with 9 aggregation stages could approximate the dynamics of a full model with 94 stages very accurately. In order to eliminate the steady-state equations from the model, the solutions were expressed as functions of the state variables of the aggregation stages on both sides of the system. By considering the flow relationships in the column and mass conservation in the steady-state systems, the complexity of the functions could be reduced to a minimum. Ultimately, the functions were implemented as five-dimensional look-up tables with linear interpolation. By adapting the resolutions of the table dimensions to the curvature of the tabulated function, the interpolation error was minimized. It was shown that the reduced model can be simulated around 8 times faster than the original model. Although the table interpolation using 64 memory accesses is computationally rather intensive, it contributed only about 17% to the total computational complexity. This gives room for using even higher-dimensional tables in more complex models.

In the appendix to chapter 4, it was shown that polynomial approximations can be used as an alternative to look-up tables. This makes the implementation of the reduced models much easier, and results in much more compact and possibly faster models. Care has to be taken that the accuracy of the functional approximations is sufficiently high in the operating range of the model. However, there seems to be some potential in using functional approximations to overcome the main bottleneck of the reduction method.

In chapter 5, the reduction method has been extended to general one-dimensional systems. Both spatially discrete systems, such as staged processes, and spatially continuous processes described by partial differential equations can be considered. For continuous processes, the method can be used as an alternative to other spatial discretization methods such as finite differences, finite volumes etc. It was shown that the approximation quality of the reduced models is superior to the approximation quality using simple finite difference approximations. Furthermore, the reduced models were shown to approximate even high-frequency limit cycle oscillations reasonably well. While the method is conceptually straightforward, the elimination of the steady-state systems and substitution into the dynamic equations remains the part that complicates the implementation of the reduced models. However, for the heat exchanger, it was possible to use analytic steady-state solutions in the reduced models. The resulting models are very compact and easy to implement.

Generally, the success of the reduction method is due to the incorporation of steady-state information into the reduced models. This makes the models very accurate when simulated close to steady-state or with slowly changing input signals. The approximation of fast dynamics can be improved by increasing the dynamic order of the models. The number of aggregation elements, their position and capacities are degrees of freedom, which can be used to adapt the reduced models to the application they are intended for.

6.2 Directions for future work

The following topics remain open for future work:

- **Procedure for determination of reduced model parameters:** The number, position and capacities of the aggregation elements are degrees of freedom that can be used to increase the approximation quality of the reduced models. While good parameter sets can be obtained by fitting to application-specific inputs and trajectories, a more general procedure is still missing. One possibility could be some adaptive procedure which switches between several models during simulation according to dynamics of the system. It seems that a criterion that considers slow and fast dynamics according to the magnitude of the left-hand sides of the differential equations could be employed to decide if locally a finer or more coarse resolution is needed. If the reduced models are used in an optimizing control context, the model is possibly simulated repeatedly with similar input trajectories during the optimization iterations. In this case, a reduced model that has a sufficient accuracy can be selected in the beginning of the iterations by simulating several reduced models with the starting guess for the optimal input trajectory and comparing their deviations from each other.

- **Functional approximation of steady-state system solutions:** The main difficulty of implementing a reduced model with the aggregation method is the functional approximation of the solutions of the steady-state systems. In this thesis, this has been achieved by either tabulation or polynomial approximation for numerical solutions, or by using analytical solutions. In most cases, numerical solutions will have to be used. The main challenges when generating a functional approximation is to balance the complexity of the functions with the approximation accuracy over a sufficiently large operating domain of the independent variables. The higher-dimensional the functions are, the more difficult this tasks gets.

- **Comparison with other methods:** It is interesting to investigate how well the reduction method of this thesis performs in comparison to other methods in terms of approximation quality and computational complexity. Especially for distillation columns, a rigorous comparison to wave propagation methods and orthogonal collocation methods will be interesting.

- **Application in real-time optimizing control:** A reduced model has a certain reduction error that comes in addition to the model-plant mismatch when the model is used to predict the behavior of a real process. Both errors have to be taken into consideration when the model is used in a real-time optimizing control optimization application. It is important to know how this error affects the control quality, for example concerning constraint violations, and what measures have to be taken as a remedy. In case of the reduced models in this study, the reduction error is known to be largest after a step change or during fast changes of the inputs. This knowledge can possibly be used when designing the control application parameters like sampling time, backoffs and input signal shape.