Reduced distillation models via stage aggregation

Andreas Linhart*, Sigurd Skogestad
Department of Chemical Engineering,
Norwegian University of Science and Technology,
N-7491 Trondheim, Norway

February 1, 2010

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. 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 for a realistic case study 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

*Corresponding author: andreas.linhart@chemeng.ntnu.no; phone +47 73550346

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 the 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). The approach of Lévine and Rouchon 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. The "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. The assumptions of constant molar flows, constant relative volatility and constant holdup used in the original method by Lévine and Rouchon (1991) are not 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. In the original derivation (Lévine and Rouchon, 1991), a singular perturbation argument (Kokotovic et al., 1986) that relies on the different time-scales of the slow compartment dynamics and fast stage dynamics is employed. In the proposed method, the spatial transport through the column is approximated by fast transport through steady-state stages, which is slowed down by assigning large "aggregated holdup factors" to the aggregation stages that connect the blocks of steady-state stages. Although related, the method is not a singular perturbation method (Linhart and Skogestad, 2009).

This paper is organized as follows. In section 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 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, 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 5.

2 Full model

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 of i-butanol and n-butanol, 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.

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.

2.2.1 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

$$\dot{M}_i^{tot} = L_{i-1} + V_{i+1} - L_i - V_i, \tag{1}$$

$$\dot{\mathbf{M}}_i = L_{i-1}\mathbf{x}_{i-1} + V_{i+1}\mathbf{y}_{i+1} - L_i\mathbf{x}_i - V_i\mathbf{y}_i, \tag{2}$$

$$\dot{U}_i^{tot} = L_{i-1}h_{i-1}^L + V_{i+1}h_{i+1}^V - L_i h_i^L - V_i h_i^V - Q_i^{hl}. \tag{3}$$

The balance equations for the feed stage i_F read

$$\dot{M}_{i_F}^{tot} = L_{i_F-1} + V_{i_F+1} - L_{i_F} - V_{i_F} + F, \tag{4}$$

$$\dot{\mathbf{M}}_{i_F} = L_{i_F-1}\mathbf{x}_{i_F-1} + V_{i_F+1}\mathbf{y}_{i_F+1} - L_{i_F}\mathbf{x}_{i_F} - V_{i_F}\mathbf{y}_{i_F} + F\mathbf{z}_F, \tag{5}$$

$$\dot{U}_{i_F}^{tot} = L_{i_F-1}h_{i_F-1}^L + V_{i_F+1}h_{i_F+1}^V - L_{i_F}h_{i_F}^L - V_{i_F}h_{i_F}^V - Q_{i_F}^{hl} + Fh_F. \tag{6}$$

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

$$\dot{M}_1^{tot} = V_{top} - (R+D), \tag{7}$$

$$\dot{\mathbf{M}}_1 = V_{top}\mathbf{y}_2 - (R+D)\mathbf{x}_1, \tag{8}$$

$$\dot{U}_1^{tot} = V_{top}h_2^V - (R+D)h_1^L + Q_c. \tag{9}$$

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

$$\dot{M}_N^{tot} = L_{N-1} - B - V_N, \tag{10}$$

$$\dot{\mathbf{M}}_N = L_{N-1}\mathbf{x}_{N-1} - B\mathbf{x}_N - V_N\mathbf{y}_N, \tag{11}$$

$$\dot{U}_N^{tot} = L_{N-1}h_{N-1}^L - Bh_N^L - V_Nh_N^V + Q_r. \tag{12}$$

The variables used in the above equations are explained in table 1. 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.

2.2.2 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 make 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

$$M_i^{tot} = M_i^L + M_i^V, \tag{13}$$

$$\mathbf{M}_i = M_i^L\mathbf{x}_i + M_i^V\mathbf{y}_i, \tag{14}$$

$$U_i^{tot} = M_i^Lh_i^L + M_i^Vh_i^V - p_iV_i, \tag{15}$$

$$V_i = M_i^Lv_i^L + M_i^Vv_i^V, \tag{16}$$

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, equation (13) and one of equations (14)-(16) can be combined to obtain explicit expressions for M_i^L and M_i^V. The simplest choice is to combine equations (13) and (16) to

get

$$M_i^L = (V_i - M_i^{tot} v_i^V)/(v_i^L - v_i^V), \tag{17}$$

$$M_i^V = M_i^{tot} - M_i^L, \tag{18}$$

where v_i^L and v_i^V are the specific volumes of liquid and vapor phase, respectively. The remaining N_c algebraic equations are then (14) and (15).

2.2.3 Algebraic thermodynamic relationships

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

$$f_i^L(\mathbf{x}_i, p_i, T_i) = f_i^V(\mathbf{y}_i, p_i, T_i). \tag{19}$$

In this study, the thermodynamic quantities f_i^L, f_i^V, 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).

2.2.4 Algebraic hydraulic relationships

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

$$L_i = \gamma \rho_i^L |l_i/\beta - h_{W,i}|^{1.5}, \tag{20}$$

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

$$V_i = \gamma \sqrt{|p_i - p_{i-1} - \rho_{i-1}^L g l_{i-1}| \rho_i^V}, \tag{21}$$

where g is the standard gravity.

2.2.5 Algebraic equations for heat loss

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

$$Q_i^{hl} = \alpha_i (T_i - T_{environment}), \tag{22}$$

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.

2.2.6 Algebraic equations for condenser cooling

The cooling of the condenser is modeled as

$$Q_c = -V_{top}(h_2^V + \beta), \tag{23}$$

where β an adjustable parameter.

2.2.7 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 2.

2.3 Alternative strategies for solution of the algebraic equations (full model)

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 for the thermodynamic quantities 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. They can be applied to all or just a part of the algebraic equations:

- **Approach 1.** The algebraic equations are solved separately at each evaluation of the right hand sides of the dynamic balance equations using a nonlinear equation solver.

- **Approach 2.** The differential and algebraic equations (DAEs) are solved simultaneously using a DAE solver.

- **Approach 3.** The algebraic equations are solved beforehand and the solutions are expressed as functions of suitable variables.

Generally, one tries to minimize the number of algebraic equations and associated algebraic state variables to be solved 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 (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 (p, T, $\mathbf{x}$, $\mathbf{y}$). Thus, on each stage the associated $2N_c$ algebraic equations given in (14), (15) and (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}).
If applied exclusively to all algebraic equations, approach 1 is in general numerically not efficient due to the nested solver structure. Approach 2 is numerically efficient, but requires the computation of relatively complex thermodynamic expressions. Approach 3 is numerically the most efficient, but requires the representation of the relevant solutions (the thermodynamic quantities) as function of the $N_c + 1$ dynamic state variables (M_i^{tot}, $\mathbf{M}_i$, U_i^{tot}). Since usually a numerical solution is necessary, these functions can become very complex and relatively inaccurate.
In this study, a combination of approaches 2 and 3 is used to obtain a numerically efficient model. From the Gibbs phase rule, it is sufficient to specify N_c intensive variables for a system in vapor-liquid equilibrium to determine all remaining thermodynamic quantities. For the binary mixture in this study

($N_c = 2$), pressure p and temperature T are chosen as independent variables. All remaining thermodynamic quantities are expressed as precomputed functions of these independent variables. The associated N_c algebraic equations are solved within the DAE solver, yielding a DAE system of $2N_c + 1$ variables.

2.4 Final DAE equation set for full model

In this study, a binary mixture is considered with T and p (on each stage) as algebraic state variables. On each of the N stages, the DAE set includes three differential equations (see equations (1)-(12)), plus one algebraic equation for the sum of phases holdup of component 1 (14) and one algebraic equation for the sum of phases internal energy (15). The 5 associated state variables X_i on each stage are

$$X_i = \{M_i^{tot}, M_i, U_i^{tot}, p_i, T_i\}. \tag{24}$$

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

$$M\frac{d\mathbf{X}}{dt} = \mathbf{F}(\mathbf{X}, \mathbf{u}), \tag{25}$$

where $\mathbf{X}$ are the $5N + 4$ state variables used by the DAE solver, $\mathbf{u}$ is a 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 (17) and (18), and also the flash equations which are represented by tables, are explicit in $\mathbf{X}$ and are solved at each evaluation of the right hand side $\mathbf{F}(\mathbf{X}, \mathbf{u})$.

2.5 Jacobian structure

The DAE set for the full model described by (25) in section 2.4 is highly structured as can be seen from figure 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 2.6.4.

2.6 Implementation of full model

2.6.1 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 $\mathbf{X}$.

2.6.2 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.

2.6.3 Tabulation of thermodynamic properties

The thermodynamic VLE relations and property relations were programmed as two-dimensional look-up tables. From these 8 tables, 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.

2.6.4 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 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.

2.6.5 Model code

Implementations of the full and reduced models in C and FORTRAN 77 code to be used in combination with DASPK 3.0 (Li and Petzold, 2000) are available on the homepage of Sigurd Skogestad (Skogestad, 2009).

3 Reduced model

3.1 Summary of reduction method

The model reduction method used in this study is based on the aggregated modeling method of Lévine and Rouchon (1991). It was shown in a previous study by Linhart and Skogestad (2009) that the method can actually be derived without the notion of compartments and use of average concentrations. 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.

However, as found previously by Linhart and Skogestad (2009), applying the reduction procedure as described above does not necessarily improve the computation speed, since the number of differential and algebraic 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 by replacing them with precomputed functions where the block-wise structure of the reduced model is exploited. Note that these functions are different from the precomputed thermodynamic functions used to obtain the thermodynamic quantities on each stage. These functions are still used on the aggregation stages of the reduced model and remain the same as in the full model.

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 3.2. Step 2a is necessary to produce a reduced-order model that increases the simulation speed. The basic procedure is described in section 3.3.1. 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 3.3.2.

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 $\bar{U}_i^{tot}$. In the full model (1)-(12), 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 again denoted as in the full model M_j^{tot}, $\mathbf{M}_j$ and U_j^{tot}. However, the numbering of the variables (j) is now different from the full model (i), 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 $\bar{()}$. The final form of the reduced model is shown in table 3.

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

Figure 2 illustrates the reduction method: A number of n stages on original stages s_j, $j = 1...n$ are selected as dynamic aggregation stages, see figure 3a. For example, $s_3 = 10$ means that aggregation stage $j = 3$ corresponds to original stage $i = 10$.

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 $H_j >> 1$:

$$H_j \dot{\bar{M}}_{s_j}^{tot} = \bar{L}_{s_j-1} + \bar{V}_{s_j+1} - \bar{L}_{s_j} - \bar{V}_{s_j}, \tag{26}$$

$$H_j \dot{\bar{\mathbf{M}}}_{s_j} = \bar{L}_{s_j-1}\bar{\mathbf{x}}_{s_j-1} + \bar{V}_{s_j+1}\bar{\mathbf{y}}_{s_j+1} - \bar{L}_{s_j}\bar{\mathbf{x}}_{s_j} - \bar{V}_{s_j}\bar{\mathbf{y}}_{s_j}, \tag{27}$$

$$H_j \dot{\bar{U}}_{s_j}^{tot} = \bar{L}_{s_j-1}\bar{h}_{s_j-1}^L + \bar{V}_{s_j+1}\bar{h}_{s_j+1}^V - \bar{L}_{s_j}\bar{h}_{s_j}^L - \bar{V}_{s_j}\bar{h}_{s_j}^V - \bar{Q}_{s_j}^{hl}. \tag{28}$$

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:

$$0 = \bar{L}_{i-1} + \bar{V}_{i+1} - \bar{L}_i - \bar{V}_i, \tag{29}$$

$$0 = \bar{L}_{i-1}\bar{\mathbf{x}}_{i-1} + \bar{V}_{i+1}\bar{\mathbf{y}}_{i+1} - \bar{L}_i\bar{\mathbf{x}}_i - \bar{V}_i\bar{\mathbf{y}}_i, \tag{30}$$

$$0 = \bar{L}_{i-1}\bar{h}_{i-1}^L + \bar{V}_{i+1}\bar{h}_{i+1}^V - \bar{L}_i\bar{h}_i^L - \bar{V}_i\bar{h}_i^V - \bar{Q}_i^{hl}. \tag{31}$$

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.

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

Figure 3 illustrates the principle. To avoid complicated notation, aggregation stages j =2 and j =3 are used for demonstration. Table 3 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 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 (29)-(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

$$\bar{\mathbf{y}}_{s_2+1} = \mathbf{y}_3^{()}(\mathbf{z}_3), \tag{32}$$

$$\bar{h}^V_{s_2+1} = h_3^{V()}(\mathbf{z}_3), \tag{33}$$

$$\bar{V}_{s_2+1} = V_3^{()}(\mathbf{z}_3), \tag{34}$$

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

$$\bar{\mathbf{x}}_{s_3-1} = \mathbf{x}_3^{()}(\mathbf{z}_3), \tag{35}$$

$$\bar{h}^L_{s_3-1} = h_3^{L()}(\mathbf{z}_3), \tag{36}$$

$$\bar{L}_{s_3-1} = L_3^{()}(\mathbf{z}_3), \tag{37}$$

$$\bar{V}_{s_3} \quad = \quad V_3^{b()}(\mathbf{z}_3). \tag{38}$$

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 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 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 j =3 is used for illustration of the the dynamic equations of the reduced model after the substitution:

$$H_3\dot{M}_3^{tot} \quad = \quad L_3^{()} - V_3^{b()} + V_4^{()} - L_3, \tag{39}$$

$$H_3\dot{\mathbf{M}}_3 \quad = \quad L_3^{()}\mathbf{x}_3^{()} - V_3^{b()}\mathbf{y}_3 + V_4^{()}\mathbf{y}_4^{()} - L_3\mathbf{x}_3, \tag{40}$$

$$H_3\dot{U}_3^{tot} \quad = \quad L_3^{()}h_3^{L()} - V_3^{b()}h_3^V + V_4^{()}h_4^{V()} - L_3h_3^L - Q_3^{hl}. \tag{41}$$

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 (32)-(34) and (35)-(38) is the set

$$\mathbf{z}_3 = \{M_2^{tot}, \mathbf{M}_2, U_2^{tot}, M_3^{tot}, \mathbf{M}_3, U_3^{tot}\}, \tag{42}$$

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.

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

The functions (32)-(34) and (35)-(38) are $2N_c + 3$ functions, while the variable set (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 (29)-(31) for each of the steady-state stages, depend on the variables of the aggregation stages 2 and 3:

$$\bar{L}_{s_2}, \bar{\mathbf{x}}_{s_2}, \bar{h}^L_{s_2} \text{ and } \bar{V}_{s_2+1} \tag{43}$$

depend on variables of aggregation stage 2, and

$$\bar{V}_{s_3}, \bar{\mathbf{y}}_{s_3} \text{ and } \bar{h}^V_{s_3} \tag{44}$$

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,

$$\mathbf{z}_3 = \{\mathbf{x}_2, T_2, L_2, \mathbf{y}_3, T_3\}. \tag{45}$$

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

$$\mathbf{z}_3 = \{T_2, p_2, L_2, T_3, p_3\}. \tag{46}$$

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 3b),

$$0 = L_2 - V_3^{()} - L_3^{()} + V_3^{b()}, \tag{47}$$

$$0 = L_2\mathbf{x}_2 - V_3^{()}\mathbf{y}_3^{()} - L_3^{()}\mathbf{x}_3^{()} + V_3^{b()}\mathbf{y}_3, \tag{48}$$

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 (39)-(41). Energy is, however, only conserved if the heat loss occurring at each stage is neglected:

$$0 = L_2 h_2^L - V_3^{()} h_3^{V()} - L_3^{()} h_3^{L()} + V_3^{b()} h_3^V - Q_3^{hl()}. \tag{49}$$

Here, $Q_3^{hl()}$ is the accumulated heat loss of steady-state block 3.

Equations (47)-(49) can be rearranged to

$$L_3^{()} - V_3^{b()} = L_2 - V_3^{()}, \tag{50}$$

$$L_3^{()}\mathbf{x}_3^{()} - V_3^{b()}\mathbf{y}_3 = L_2\mathbf{x}_2 - V_3^{()}\mathbf{y}_3^{()}, \tag{51}$$

$$L_3^{()} h_3^{L()} - V_3^{b()} h_3^V = L_2 h_2^L - V_3^{()} h_3^{V()} - Q_3^{hl()}, \tag{52}$$

and can then be used to eliminate the corresponding terms in the dynamic equations of aggregation stages. The equations for aggregation stage 3 (39)-(41) then read

$$H_3\dot{M}_3^{tot} = L_2 - L_3 + V_4^{()} - V_3^{()}, \tag{53}$$

$$H_3\dot{\mathbf{M}}_3 = L_2\mathbf{x}_2 - L_3\mathbf{x}_3 + V_4^{()}\mathbf{y}_4^{()} - V_3^{()}\mathbf{y}_3^{()}, \tag{54}$$

$$H_3\dot{U}_3^{tot} = L_2h_2^L - L_3h_3^L + V_4^{()}h_4^{V()} - V_3^{()}h_3^{V()} - Q_3^{hl()} - Q_3^{hl}, \tag{55}$$

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 3c). Note that equation (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 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

$$Q_3^{hl()} = Q_3^{hl()}(\mathbf{z}_3) \tag{56}$$

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

$$T_3^{()}(\mathbf{z}_3),\ p_3^{()}(\mathbf{z}_3),\ Q_3^{hl()}(\mathbf{z}_3), \tag{57}$$

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 2.6.3.

3.4 Jacobian structure

The Jacobian of the reduced model as given in table 3 has exactly the same structure as the Jacobian of the full model, but the reduced model has fewer stages (see figure 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.

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 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.
- Reflux drum ($j = 1$) and reboiler ($j = n$) 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 top-most 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 between the temperature stage and the bottom (reboiler).

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 **fitting** the top concentration trajectory of the reduced model on the full model trajectory using the input signal described in section 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 3.2. 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.

3.6 Implementation of steady-state functions by table interpolation

In the reduced model, the steady-state functions (57) are used to calculate the vapor flow variables in the dynamic equations of the aggregation stages (53)-(55). These functions are the solutions of the steady-state blocks described by equations (29)-(31) between the aggregation stages that depend on the set of independent variables (46) as described in section 3.3.2. The solutions can only be obtained numerically due to the nonlinear nature of the equations. The continuous functions (57) have therefore to be generated from discrete numerical solutions on the domain of the independent variables.

In this study, a five-dimensional look-up 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 look-up 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 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. 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.

(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 5.

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 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.

4.1 Test input trajectories

Figure 5 shows the six sequential input trajectories used for the performance assessment. The inputs F, z_F and h_F describe the feed into the column, and are disturbance variables as be seen from a control perspective. The model

includes some basic control loops and the inputs p^s, T^s and 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 $15 \cdot 10^4$s, allowing the system to approach steady state again.

4.2 Accuracy of reduced model

Figures 6-11 show snapshots of the responses of the top and bottom concentrations of the full (94 stages including reflux drum and reboiler and 474 states) and the reduced models (9 aggregation stages and 49 states) 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 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), indicating that the locations of the aggregation stages have a considerable influence on the approximation accuracy.

4.3 Computational performance of reduced model

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

$$\theta^{abs} = \theta^{rel} = 10^{i/2},\ i = 2, ..., 8, \tag{58}$$

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.

4.3.1 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:

$$\varepsilon = \frac{1}{t_{end}} \int_0^{t_{end}} \left| x_1^{exact}(t) - x_1^{model}(t) \right| dt, \tag{59}$$

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. 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

error ε is called the average error in the following. It is a practical measure for the overall error averaged over time.

The overall error 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 overall error 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 implementation error is small compared to the reduction error. In the following, the implementation error is therefore neglected.

Figure 12 shows the simulation time of the full and the optimized reduced model versus the average 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 $\theta = 10^{-2.5}$.

4.3.2 Computational complexity of model and solver

Table 6 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 12).

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). However, a doubling of execution speed here would still 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 look-up 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.

5 Discussion

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 dis-

cussed previously by Linhart and Skogestad (2009). As shown in section 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 computation time 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 where the steady-state blocks are replaced by precomputed functions 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 look-up 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.

5.2 Implementation of steady-state functions

The implementation of the steady-state functions as described in section 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.

5.2.1 Large number of independent variables

In this study, the steady-state functions were implemented using five-dimensional look-up tables with multi-linear interpolation. However, as shown in section 4.3.2, the table look-up 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 exam-

ple, 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 look-up 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 look-up 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.

5.2.2 Large number of components

For systems with many components the dimension of the tables may get large and unmanageable. An alternative is to use individual tables for the low-concentration components. The justification is that in each column section there is usually only a few dominant components that affect the overall behavior. A low-concentration component will affect itself but not the other components.

5.2.3 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:

$$f(x_1, x_2, x_3) \approx f_1(x_1, x_2) + f_2(x_1, x_2)x_3. \tag{60}$$

  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 5 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 look-up operations per dimension. If one table dimension is interpolated with cubic splines instead of linear interpolation, the computational complexity of the interpolation will therefore double. However, since the interpolation error is of higher order, the size 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).

5.2.4 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.

5.2.5 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.

5.3 Selection of reduced model parameters

In section 3.5, some guidelines on how to select the reduced model parameters, namely the number and locations of the aggregation stages and the aggregated holdup factors, were given. The guidelines are not sufficient to determine all reduced model parameters. While an "equal distribution" of the free aggregation stages and the aggregated holdup factors yields models with satisfactory approximation quality, it was shown that better reduced models can be obtained by selecting the parameters using a more specialized procedure. In the present study, the free reduced model parameters were determined by fitting

the top concentration trajectories of the full and the reduced models. Clearly, the reduced model will be optimized for the input sequence used to generate the fitting trajectory, but this does not guarantee that the approximation quality is close to optimal when a significantly different input is applied.

It would be desirable to have an extended set of rules at hand to derive near-optimal reduced model parameters without the need of simulation data and optimization. An analytical derivation of such rules is probably difficult due to the nonlinear and multiple-input multiple-output character of the system. However, since the reduced model is structurally similar to the original model, it should be possible in most cases to choose the number and locations of the aggregation stages using some physical insight into the dynamics of the column. To determine any remaining parameters, especially the aggregated holdup factors, it seems most practical to fit the model to available process data or some carefully designed reference trajectory. When fitting, care should be taken that the optimization criterion is selected with the intended application of the reduced model in mind. For example, when the accuracy of the reduced model after a certain time after excitation (long-term or short-term response) is important, the error should be weighted accordingly in the optimization criterion.

An alternative to determining the parameters "off-line" could be some adaptive procedure during simulation or application of the model. Similar to adaptive mesh methods for partial differential equations (Baker, 1997), the number of aggregation stages could be adapted locally to the dynamic activity in the different parts of the column. Here, the "density" of aggregation stages should locally match the dynamic activity in the different parts of the column. The residuals of the dynamic aggregation stage equations could be taken as a measure for the dynamic activity of the corresponding part of the

column, and aggregation stages could be added or removed to change their density. To simplify implementation, a limited number of possible aggregation stages should be fixed beforehand. This procedure requires the calculation of additional steady-state functions for the different possiblities of active aggregation stages.

A different adaptive approach can be used when the model is used repeatedly with similar inputs during dynamic optimization. Prior to the optimization iterations, a suitable model could be selected from a bank of precalculated reduced models. The selection can be based on the optimal inputs that have been calculated from the previous optimization, and are now used as starting guess for the new optimization. One reduced model with a large number of aggregation stages can be used to generate a reference trajectory. From the remaining models, the model which has the lowest number of aggregation stages (lowest order) at an acceptable deviation from the reference trajectory is selected for the optimization. As a refinement, the model could be changed repeatedly during one simulation. This procedure is rather generic, and not restricted to models derived with the method proposed in this paper.

5.4 Application of reduced model in real-time optimizing control

It was shown in section 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.

5.5 Alternative model reduction methods for distillation models

There exist several alternative model reduction methods for distillation models. Collocation methods (Cho and Joseph, 1983, Dalaouti and Seferlis, 2006**, Stewart et al., 1984**) are probably the most similar methods in terms 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).

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 over an efficient implementation of the full model. The elimination of the algebraic equations is conceptually straightforward, but requires the approximation of functions of five independent variables. In this study, look-up 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 as well as Heinz Preisig and Stefan de Graaf 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 look-up 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.

References

[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] **Baker, T.J., 1997. Mesh adaptation strategies for problems in fluid dynamics. Finite Elements in Analysis and Design 25, 243-273.**

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

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

[7] 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.

[8] 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.

[9] 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.

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

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

[12] 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.

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

[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. Proceedings of the 2004 American Control Conference 5, 4237-4242.

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

[16] 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.

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

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

[19] 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.

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

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

[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] 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.

[25] 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.

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

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

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

[29] Skogestad, S., 2009. Homepage.
http://www.nt.ntnu.no/users/skoge/publications/thesis/2009_linhart/software/

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

[31] **Stewart, W.E., Levien, K.L., 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.

List of figures

List of tables

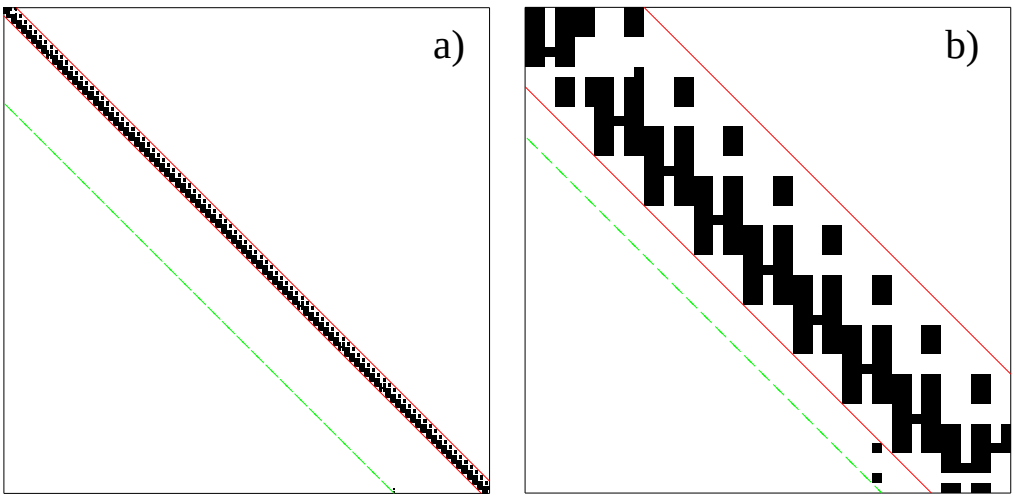

Figure 1:

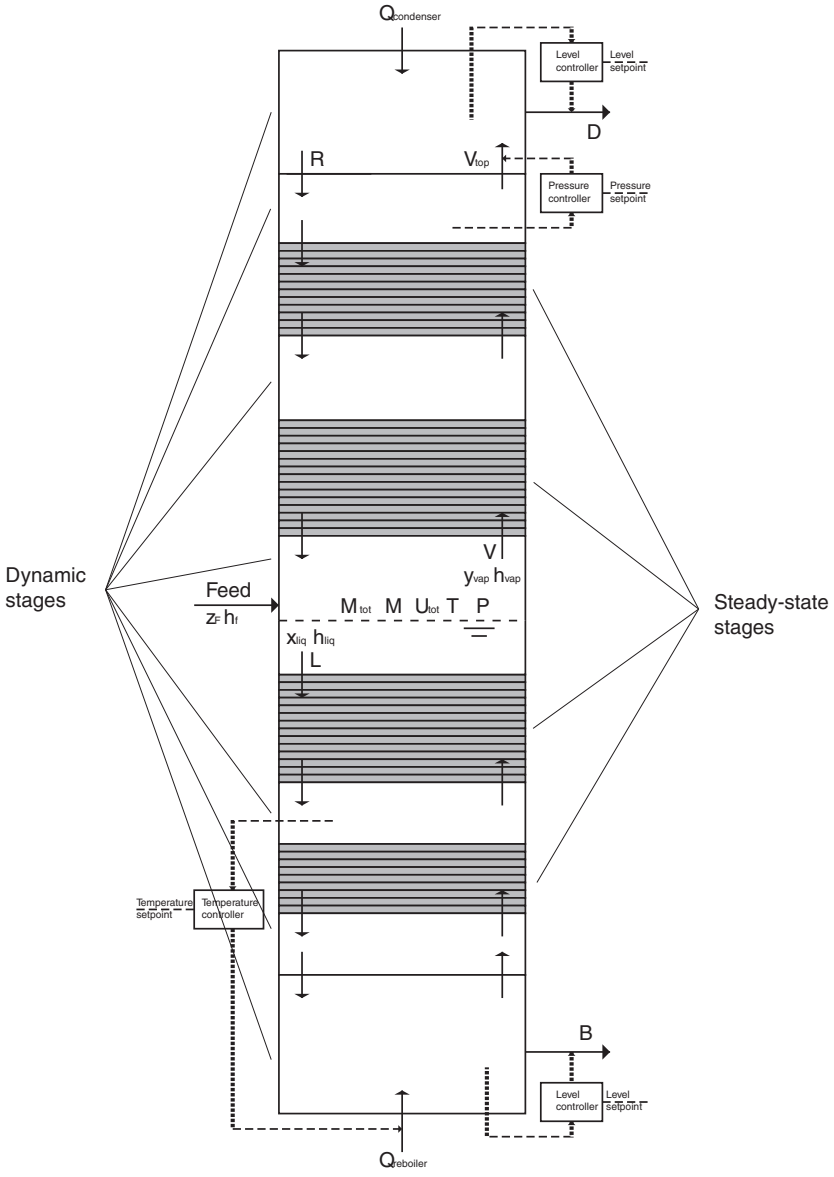
Q condenser
Level controller
Level setpoint
D
R
V top
Pressure controller
Pressure setpoint
V
y vap h vap
Dynamic stages
Feed
z F h f
M tot M U tot T P
x liq h liq
L
Steady-state stages
Temperature setpoint
Temperature controller
B
Level controller
Level setpoint
Q reboiler

Figure 2:

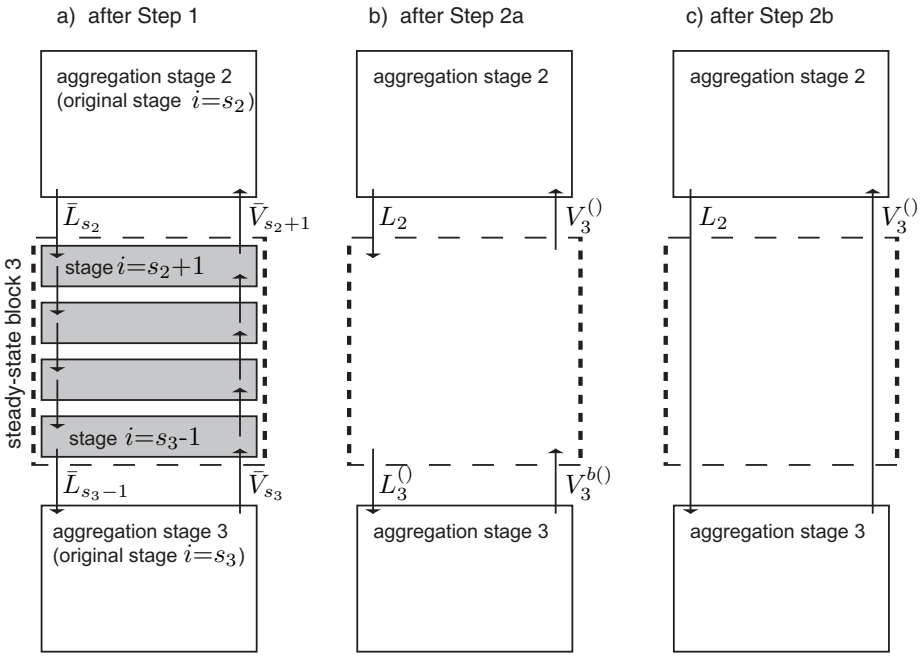

Figure 3:

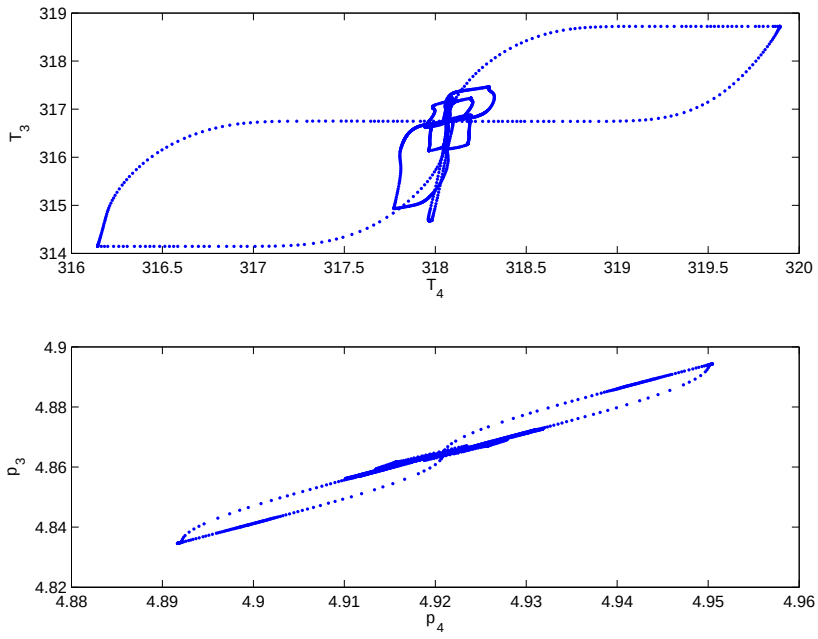

Figure 4:

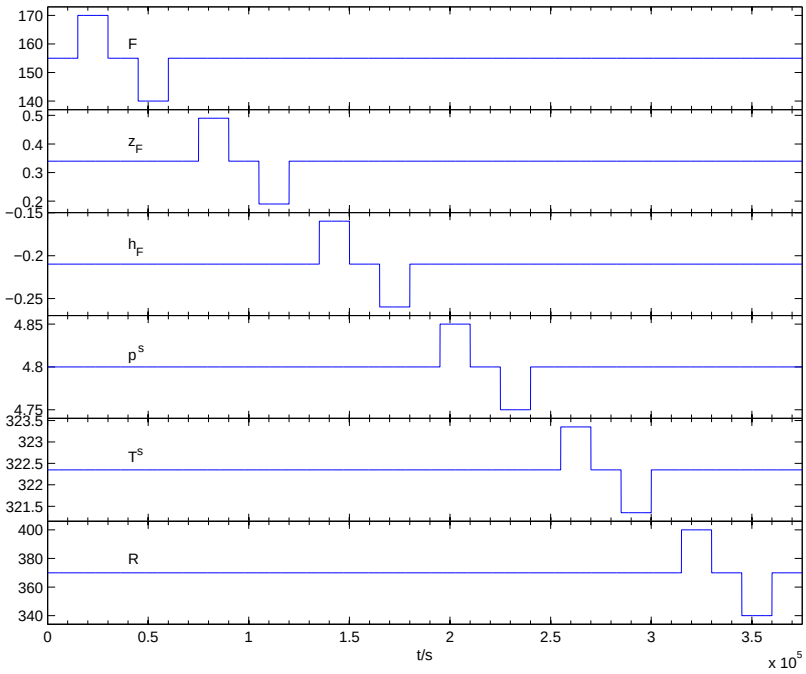

Figure 5:

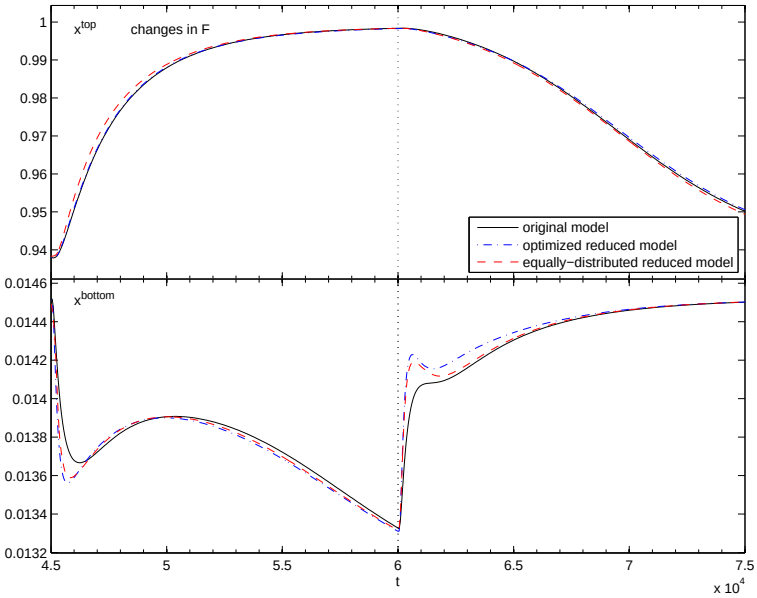

Figure 6:

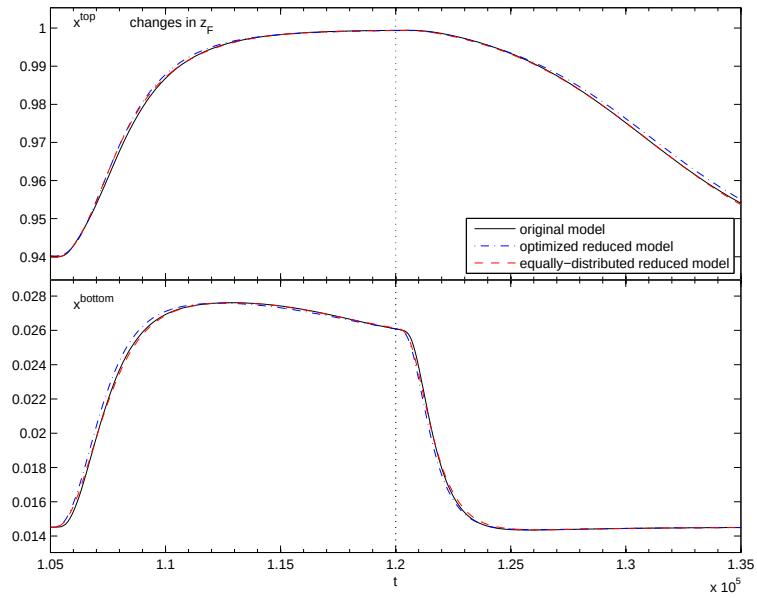

Figure 7:

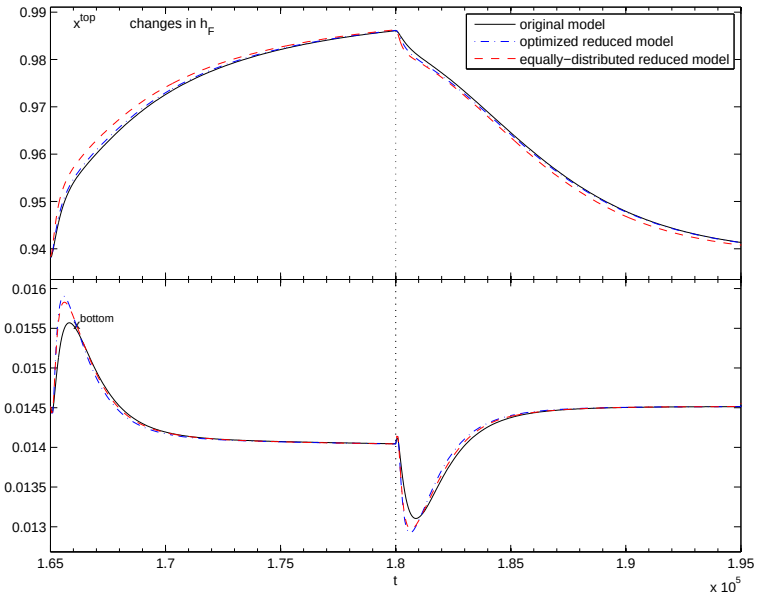

Figure 8:

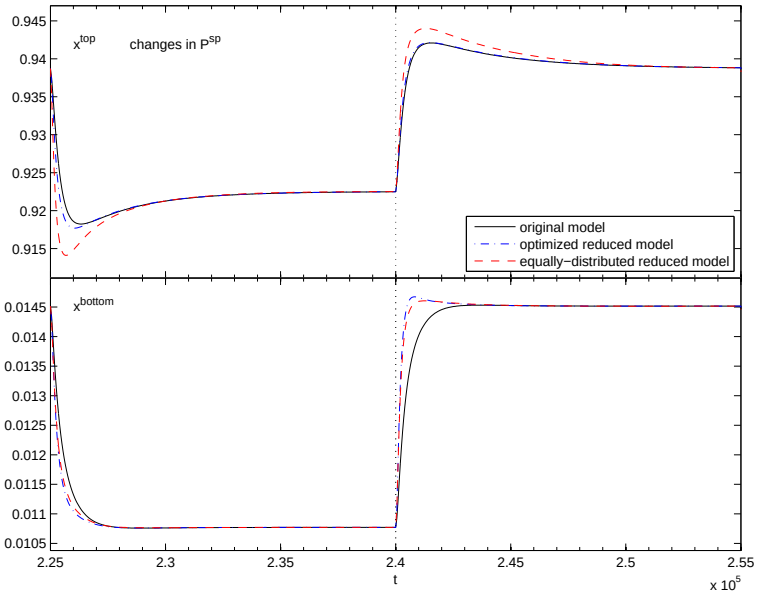

Figure 9:

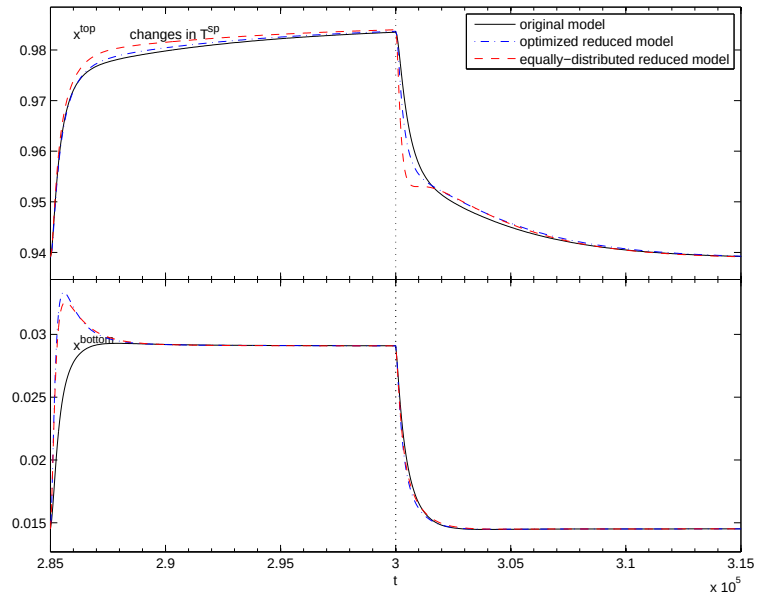

Figure 10:

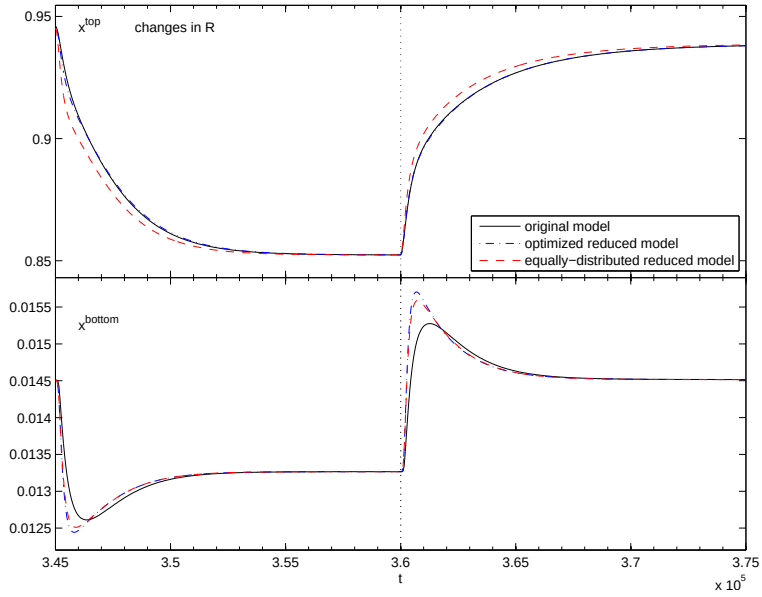

Figure 11:

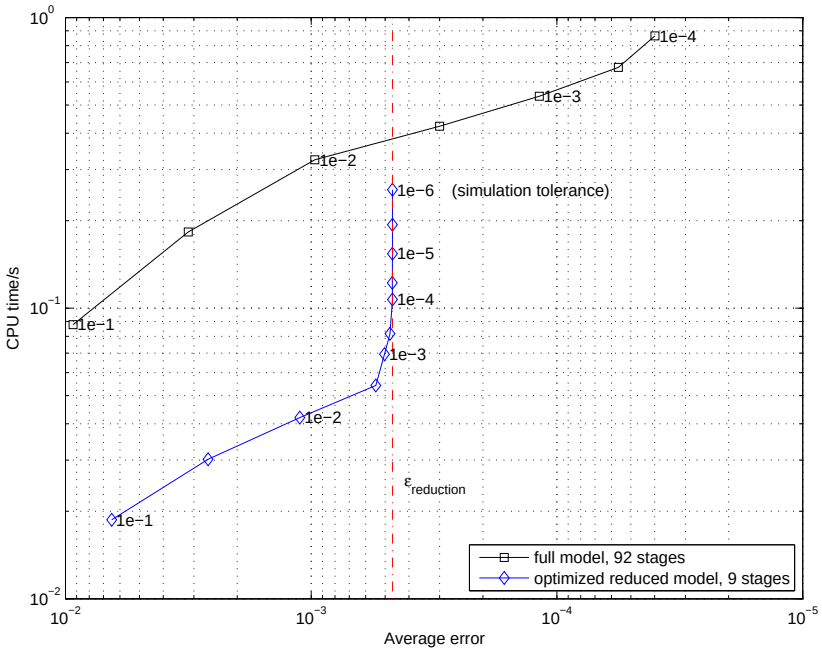

Figure 12:

Table 1:

Variable	Description	SI-Unit	typical values		
			stage	reflux drum	reboiler
i	stage index				
i_F	index of feed stage				
M_i^{tot}	total mole number	mol	3500	49000	100000
M_i	total mole number of component 1	mol	3300	45500	1500
U_i^{tot}	total internal energy	kJ	$-5.7 \cdot 10^4$	$-1.0 \cdot 10^6$	$-1.7 \cdot 10^6$
x_i	liquid concentration of component 1		0.34	0.94	0.015
y_i	vapor concentration of component 1		0.40	0.96	0.019
h_i^L	liquid enthalpy	kJ/mol	-18	-22	-18
h_i^V	vapor enthalpy	kJ/mol	1.2	1.6	1.8
L_i	liquid outflow	mol/s	500		
V_i	vapor outflow	mol/s	500		500
V_{top}	vapor flow from top stage into reflux drum	mol/s		425	
D	liquid distillate outflow of reflux drum	mol/s		55	
B	liquid bottom product outflow of reboiler	mol/s			100
R	reflux flow out of reflux drum	mol/s		370	
F	feed flow into feed stage	mol/s	155		
z_F	concentration of component 1 in feed		0.34		
h_F	feed enthalpy	kJ/mol	-22		
Q_c	heat flow into condenser	kW		-9500	
Q_r	heat flow into reboiler	kW			10000
Q_i^{hl}	heat loss from stage to environment	kW	2.4		

Table 2:

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

Table 3:

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:

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

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

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

reboiler (aggregation stage n):

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

Table 4:

	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

Table 5:

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

Table 6:

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