

# Chemical Process Modeling in Modelica

Ali Baharev    Arnold Neumaier  
Fakultät für Mathematik, Universität Wien  
Nordbergstraße 15, A-1090 Wien, Austria

## Abstract

Chemical process models are highly structured. Information on how the hierarchical components are connected helps to solve the model efficiently. Our ultimate goal is to develop structure-driven optimization methods for solving nonlinear programming problems (NLP). The structural information retrieved from the JModelica environment will play an important role in the development of our novel optimization methods. Foundations of a Modelica library for general-purpose chemical process modeling have been built. Multiple steady-states in ideal two-product distillation were computed as a proof of concept. The Modelica source code is available at the project homepage. The issues encountered during modeling may be valuable to the Modelica language designers.

*Keywords: separation, distillation column, tearing methods, homotopy continuation, bifurcation*

## 1 Introduction

The object-oriented component-based modeling methodology in Modelica (FRITZSON [13]) is well-suited for chemical processes modeling. Accordingly, Modelica has received attention in the chemical engineering literature (SANDROCK & DE VAAL [19]). Creating a component-based framework for chemical process modeling is one of the goals of our project. This framework then serves as a common language between mathematicians and chemical engineers. The current chemical engineering literature is hardly accessible to mathematicians, partly due to the engineering jargon and unwritten traditions.

We created a prototype Modelica implementation of basic chemical engineering processes. Currently, only steady-state models are supported. Once this component library is finished, software with a graphical user interface, such as the [OpenModelica Connection Editor](#) (OMEdit), can be used to build chemical process models. The process

model creation involves only high-level operations on a GUI; low-level coding is not required. This is the desired way of input. Not surprisingly, this is also how it is implemented in commercial chemical process simulators such as [Aspen Plus](#)<sup>®</sup>, [Aspen HYSYS](#)<sup>®</sup> or [CHEMCAD](#)<sup>®</sup>.

Nonlinear system of equations are generally solved using optimization techniques. AMPL (FOURER et al. [12]) is the de facto standard for model representation and exchange in the optimization community. Many solvers for solving nonlinear programming (NLP) problems are interfaced with the AMPL environment. We are aiming to create a ‘Modelica to AMPL’ converter. One could use the Modelica toolchain to create the models conveniently on a GUI. After exporting the Modelica model in AMPL format, the already existing software environments (solvers with AMPL interface, AMPL scripts) can be used. Thus an AMPL export facility builds a bridge between Modelica users and the optimization community. Such an implementation exists (ÅKESSON [3]) but it is no longer supported, and not publicly available.

Our ultimate goal is the development of structure-driven optimization methods for solving nonlinear programming problems (NLP). The structural information (hierarchical components and the connections between them) can be exploited to solve the underlying process model efficiently. For example the process model of the reactive distillation column in CIRIC & MIAO [8], producing ethylene glycol from ethylene oxide and water, has 70 variables and 70 equations. However, the steady-state process model can be solved by solving univariate equations only, in a proper elimination order (BAHAREV & NEUMAIER [5]). In other words, the problem is essentially 1-dimensional. Typically, chemical process models are essentially low-dimensional even if their steady-state model is large-scale.

The structural information is difficult to get from an AMPL source directly, one would rather try to extract it from the flattened AMPL file instead. In prin-

principle, one could recover the required structural information from the flattened model, at least to some extent. This means that the flattening step throws away the structural information first, then one must try to recover it inside a solver. In contrast, the structural information is programmatically accessible in JModelica (ÅKESSON et al. [2]) before flattening, and we intend to utilize this.

## 2 Component-based modeling of chemical processes

Chemical processes are well-suited for component-based modeling since they are networks of equipments. In turn, it is natural to model the equipments hierarchically, as a composite of smaller components. The smallest subcomponents are called **atomic units**. The atomic units are connected by **process streams**.

### 2.1 Connector class: process streams

A process stream  $S$  consisting of  $C$  substances has  $C + 2$  independent variables. It is characterized by the list of variables

$$S = \{S.f, S.p, S.H\},$$

where  $S.f$  is an array of size  $C$ . See also Table 1.

variable	physical meaning	SI unit
$f[i] \geq 0$	molar flow rate of substance $i = 1 : C$	mol/s
$p \geq 0$	pressure	Pa
$H$	enthalpy flowrate	J/s

Table 1: The  $C + 2$  variables characterizing a process stream.

The graphical representation of process streams is by arrows, as shown in Figure 1.

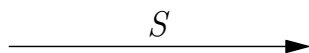


Figure 1: The graphical representation of stream  $S$ .

The units are connected by streams. The streams entering the unit are called **inlets**, while the streams leaving it are called **outlets**. The causal flows reflect the fact that the chemical process streams are directed, the material can only flow into the direction specified.

### 2.2 Sources and sinks

Given their simplicity, the easiest way to describe these components is by their implementation, see below. The only equations that sources and sinks can be

involved in are the connecting equations and specifications on their stream variables.

```
class Source
  output Stream outlet;
end Source;

class Sink
  input Stream inlet;
end Sink;
```

### 2.3 Types of equations

These equations apply to all atomic units in subsection 2.4. Only flows of chemicals are considered. Heat flows allowing thermal coupling or multidomain models would need an extension.

**Material balances:** A system of  $C$  linear equations, reflecting the conservation of mass.

**Heat balance:** A linear equation reflecting the conservation of energy.

**Mechanical equilibrium:** The outlets have the same pressure as the unit. With the exception of the mixer and the pressure changer, the pressure of the unit equals the pressure of its only inlet.

**Thermal equilibrium:** The enthalpy of the outlets corresponds to the temperature of the unit. This relation is expressed by nonlinear equations (equation of state). If the temperature is not an internal variable of the unit then these nonlinear equations are missing.

**Characterizing equations:** These equations characterize how the unit works and cannot be changed.

**Connections with other units:** These equations describe how the units are connected by equating the corresponding variables of the involved streams.

**Specifications:** These equations make the steady state model of the unit well-defined. They usually correspond to closed loop control systems. The form of these equations shows large variation: they can be trivial equations as well as complicated nonlinear equations.

### 2.4 Atomic units

As the name suggests, these units cannot be decomposed further to smaller, connected Modelica components. Atomic units implement the `UnitOp` interface, that is all the equations listed in Subsection 2.3 apply. These units are the followings.

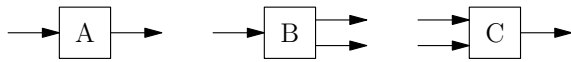


Figure 2: Structural types of the atomic units: (A) heat exchanger, pressure changer, reactor; (B) divider, flash; (C) mixer.

1. Mixer
2. Heat exchanger
3. Pressure changer
4. Reactor
5. Divider
6. Flash

The mixer has multiple inlets and a single outlet. All other atomic units have a single inlet and can have either one or two outlets. See Figure 2. Some code snippets are shown below. The simplicity of the implementation is a consequence of proper decomposition.

```
class Mixer
  extends UnitOp(nInlet=2, nOutlet=1);
end Mixer;

class PressureChanger
  extends UnitOp(nInlet=1, nOutlet=1);
  redeclare class ChangeInPressure=DeltaP;
end PressureChanger;

class Divider
  extends UnitOp(nInlet=1, nOutlet=2);
  Real zeta;
  equation
    outlet[1].f = zeta*inlet[1].f;
    outlet[1].H = zeta*inlet[1].H;
end Divider;
```

The Divider has one so-called unit parameter,  $\zeta$ , its value typically comes from specification.

The atomic units or the equipments are not referred to as components in the chemical engineering literature. Unlike Modelica, the word “component” refers to a particular chemical substance in the process. We call the smallest Modelica components atomic units and the composite Modelica components composite units.

## 2.5 Notes on the process stream definition

Traditionally, one uses the total molar flowrate, the mole fractions of the chemical substances, the pressure and the temperature to characterize a process stream. In addition, the specific enthalpy is needed to distinguish, for example, between boiling water and saturated steam, as they both have a temperature of 100°C

variable	physical meaning	SI unit
$F \geq 0$	total molar flow rate	[mol/s]
$x[i] \geq 0$	mole fraction of substance $i = 1 : C$	[-]
	$\sum x[i] = 1$	
$p \geq 0$	pressure	[Pa]
$T \geq 0$	temperature	[K]
$h$	specific enthalpy flowrate	[J/s mol]

Table 2: Traditional choice of variables to characterize a process stream.

at atmospheric pressure. The traditional representation is shown in Table 2.

There are three problems with this representation. (1) The temperature is uniquely determined by the other variables and this relation is nonlinear (equation of state). (2) The material and heat balance equations are nonlinear because mole fractions are used to describe the stream composition. (3) The process stream definition involves an equality constraint (the mole fractions must sum up to 1).

The first two issues make linear atomic unit models nonlinear. In particular, the mixer becomes nonlinear. (The thermodynamically consistent model of the mixer is nonlinear. However, it is practically always made linear in the chemical engineering literature by ignoring the so-called heat of mixing.) The mixer is the only atomic unit having multiple inlets. Thus, a nonlinear mixer has a domino effect: many of the composite units are no longer worth decomposing.

The temperature can be safely dropped from the stream definition. It is uniquely determined by the other variables and it is never needed outside the units. If, for some reason, the temperature of a stream is needed, one can always calculate it by running a flash calculation.

At first sight, it looks strange to the engineer to drop the temperature from the stream definition. Traditionally, the temperature is included in the stream variables (e.g. the EMSO model library, DE P. SOARES & SECCHI [9]) as it is easily measured in real life with a thermometer. Nevertheless, it can be safely excluded.

To make the balance equations linear we use the molar flowrates of the individual substances and the total enthalpy flowrate in place of the total molar flow rate, the mole fractions and the specific enthalpy flowrate. This has the beneficial side-effect that the equality constraint disappears since the mole fractions are not present. With these changes to the stream definition given in Table 2, we arrive at the stream definition presented in Table 1.

## 2.6 Why not Modelica.Fluid?

The Modelica.Fluid library superficially resembles our library. However, according to the documentation: “*The Modelica.Fluid library provides basic interfaces and components to model 1-dimensional thermo-fluid flow in networks of pipes. [...] there is the restriction that only media models are supported that have  $T$ ,  $(p,T)$ ,  $(p,h)$ ,  $(T,X)$ ,  $(p,T,X)$  or  $(p,h,X)$  as independent variables. [...] (Note,  $T$  is temperature,  $p$  is pressure,  $d$  is density,  $h$  is specific enthalpy, and  $X$  is a mass fraction vector).*”

The Modelica.Fluid library does not aim at supporting chemical process models. Chemical process models are different from flows in networks of pipes.

We want to preserve the linearity of the material and heat balances because it plays an important role in our novel methods. Since the presence of the temperature, the mass / mole fractions or the specific enthalpy would make the balance equations nonlinear, none of them should not appear in the connector class. As already discussed in subsection 2.5, only the *molar* flow rates of the substances, the pressure and the enthalpy flowrate together guarantee linearity. Unfortunately, the Modelica.Fluid library does not allow this choice of the independent variables.

## 2.7 Hierarchical modeling: composite units

We call the smallest Modelica components atomic units and the composite Modelica components composite units. Often, atomic units only exist on the level of abstraction. For example the equipment in YI & LUYBEN [20] referred to as reactor cannot be decomposed further into smaller, functioning pieces. However, it can be modeled by connecting 7 atomic units and a sink appropriately. None of these units is a reactor. See Figure 3.

The set of atomic units listed in Subsection 2.4 was determined by recursively decomposing a variety of chemical processes. As a result, this set of atomic units is sufficient for general-purpose chemical process modeling.

Figure 4 shows an example of hierarchical decomposition. The vapor-liquid equilibrium cascade is a cascade of stages. A stage is a mixer and a flash unit connected appropriately. In real life, the stages are the smallest, still functioning pieces. The decomposition of the stage into a mixer and a flash unit is an abstraction, as the stage does not have a mixer or a flash unit inside. Nevertheless, this decomposition is valid for modeling.

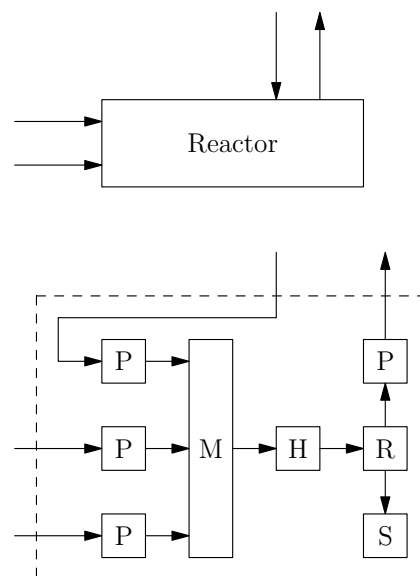


Figure 3: The reactor of Yi & Luyben and its abstract decomposition into atomic units. P: pressure changer, M: mixer, H: heat exchanger, R: reactive flash, S: sink.

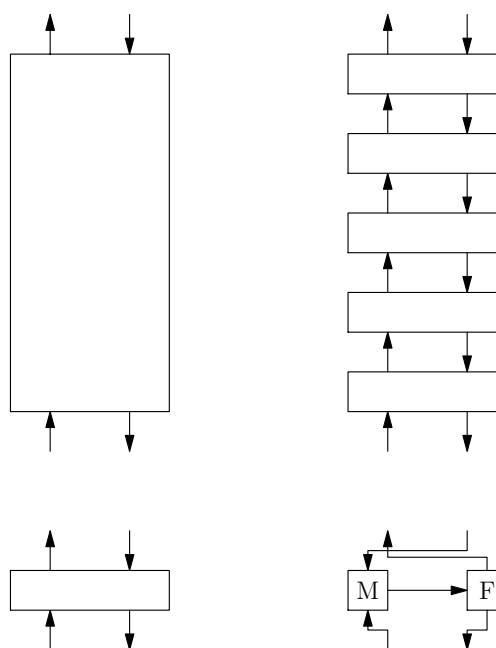


Figure 4: Hierarchical decomposition of the vapor-liquid equilibrium cascade into a cascade of stages, then the decomposition of a stage into a mixer M and a flash unit F.

## 2.8 Modelica issues encountered

The unit models are valid only if the molar flowrates are nonnegative. This is due to the internal physical structure of the corresponding unit. The natural way to impose these nonnegativity constraints is to impose it on the molar flowrates and the pressure of

the stream, that is, in the connector class. Inequality constraints can be represented within the Modelica language but only by introducing slack variables and setting the min/max on these variables accordingly. This approach is rather inconvenient. The Optimica language extension (ÅKESSON et al. [1]) supports inequalities, it is our preferred way of defining inequality constraints.

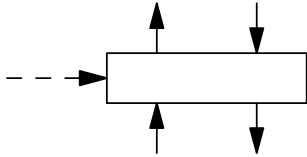


Figure 5: A stage with an optional connection (dashed arrow).

Another difficulty is that Modelica cannot handle arrays of components that have optional connections. All stages have an optional inlet, see Figure 5. This makes the creation of cascades somewhat awkward as missing inlets have to be simulated by dummy streams. The details are difficult to explain in text but easy to understand from the source code. The reader is referred to the source code of the VLEcascade, available from the project homepage at NEUMAIER [17].

### 3 Application: separation operations

The Modelica implementation discussed in the previous section is tested on a separation operation model. The background of the application is briefly presented. Then numerical results are given for the particular benchmark in subsection 3.2.

A chemical plant takes raw materials as input and produces products as output. Roughly speaking, three steps can be distinguished in a chemical plant: preparation, reaction and purification. See Figure 6. Unwanted chemical substances are separated from the raw input materials in the first step. The unwanted substances may interfere with the reaction in the second step. The reaction produces the desired products and byproducts. Usually a significant fraction of the reactants remain unreacted. These unreacted reactants, the products and the waste byproducts are separated in the third step, called the purification step. The unreacted reactants are recycled, that is, they are fed back to the first step.

Both the first and the third step involves separation operations. In a typical chemical plant, 40–80% of the investment is spent on separation operation equip-

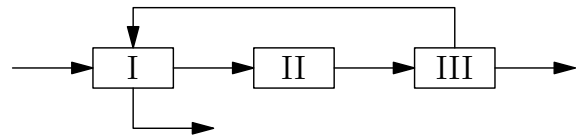


Figure 6: Schematic figure of a chemical plant. Input: raw materials, output: products and byproducts. The steps are (I) preparation, (II) reaction and (III) purification.

ments (PRAUSNITZ et al. [18], p. 2).

Many of the practically relevant equipments used in separation operations (multistage extraction, absorption, desorption, stripping and distillation) are internally a cascade. Not surprisingly, their mathematical model can be solved in a sequential manner.

Identifying multiple steady states is critical to proper design, simulation, control, and operation of these equipments. Unfortunately, professional simulators return only one solution at a time, without indicating the possible existence of other solutions. Usually, only one of the steady-states is desired, the so-called high purity branch. The other steady states are undesirable and potentially harmful as they can lead to unexpected behavior, meaning that the equipment may respond to perturbation in a counterintuitive way.

Given the importance of separation operations, they have already been modeled in Modelica by several authors, for example DURO & MORILLA [11], JOOS et al. [15] and CHANG et al. [7]. Our implementation is based on our Modelica component library for general-purpose chemical process modeling. This distinguishes our implementation from the previous ones.

#### 3.1 Internal physical structure of distillation columns

Distillation columns are used in separation operations. The body of a multistage distillation column is a cascade of stages. In the cascade, the output of one stage is the input of its two neighbors and vice versa, see Figure 4. This structural information can be exploited to solve the underlying process model efficiently.

The internal physical structure is reflected in the mathematical model of the columns. The equations can be evaluated in a sequential manner after guessing just a few variables at one end of the cascade. The essential dimension of the problem is given by the number of variables that have to be guessed to start the stage-by-stage computations. The steady-state model of distillation columns are essentially low-dimensional even if their steady-state model is large-scale.

This approach, reducing the large-scale model to a low-dimensional one, is called the stage-by-stage calculation (LEWIS & MATHESON [16]). Unfortunately, solving the low-dimensional model is very difficult if not impossible with this method, as it shows an extreme sensitivity to the initial estimates. Thus, currently only high-dimensional techniques are in use (DOHERTY et al. [10], 13–33). But a proof-of-concept method remedies the numerical difficulties of the stage-by-stage calculation, see BAHAREV & NEU-MAIER [5].

### 3.2 Example: multiple steady-states in ideal two-product distillation

The Modelica implementation discussed in Section 2 is tested on the distillation column presented in JACOBSEN & SKOGESTAD [14]. Its main structure corresponds to the linear structure presented in Figure 4, and detailed in subsection 3.1.

Perhaps the simplest distillation columns are the single feed two-product columns with ideal vapor-liquid equilibrium. Even these columns can have multiple steady-states (JACOBSEN & SKOGESTAD [14]). One type of multiplicity can occur when the column has its input specified on a mass or volume basis (e.g., mass reflux and molar boilup). This is of high practical relevance as industrial columns usually have their inputs specified in this way.

The model equations are taken from BAHAREV et al. [4]. Specifications are: methanol-propanol feed composition, mass reflux flow rate and vapor molar flow rate of the boilup. Heat balances are included in the model.

In many studies, one is interested in the dependence of the characteristics on a design parameter (the bifurcation parameter) that can be varied, resulting in bifurcation diagrams. In this case, the design parameter is the reflux flowrate specified on mass basis, and the observed parameter is the product purity. The bifurcation diagram is given in Figure 7. The model equations have five distinct solutions in a certain range of the reflux flow rate. One of the solutions is infeasible in practice because it would result in negative flow rates. The fact that the Modelica implementation gives the expected steady-states suggests that the implementation of the involved atomic and composite units is correct.

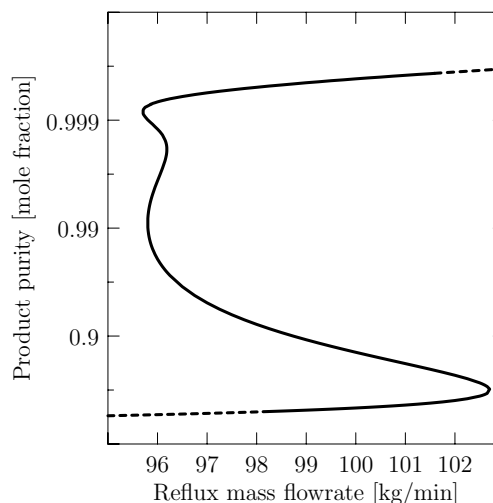


Figure 7: Bifurcation diagram, multiple steady-states in ideal two-product distillation. The infeasible steady-states are represented by dashed lines.

## 4 Future work

### 4.1 Recovering structural information

The structural information (connections of the units) can help to solve the underlying process model efficiently, as already mentioned in the introduction and in Subsection 3.1. The core equations of the column in Subsection 3.2 are shown below.

Modelica source:

```
connect(cascade.outVapor,    condenser.inlet);
connect(condenser.distillate,distillate.inlet);
connect(condenser.reflux,   cascade.reflux);
connect(feed.outlet,        cascade.feed);
connect(cascade.boilup,     reboiler.vapor);
connect(cascade.outLiquid,  reboiler.inLiquid);
connect(reboiler.bulk,      bulk.inlet);
```

AMPL source:

```
M_eq{j in 1..N-1}:
  sum{k in 1..j} F[k]*z[k] + V[j+1]*y[j+1]
  = D*y[1] + (sum{k in 1..j} F[k]+V[j+1]-D)*x[j];

M_tot:
  F[N_F]*z[N_F] = D*y[1] + (F[N_F]-D)*x[N];

H_eq{j in 1..N-1}:
  sum{k in 1..j} qF[k] + V[j+1]*HV[j+1]
  = V[1]*(HV[1]-HL[0]) + D*HL[0]
  + (sum{k in 1..j} F[k]+V[j+1]-D)*HL[j];
```

The Modelica code is favorable when it comes to structural information, since it speaks about connec-

tions as clearly as possible. The JModelica environment (ÅKESSON et al. [2]) supports programmatic access to the connectivity information. JModelica will play an important role in the further development of our novel methods.

## 4.2 Optimization

Optimization methods are used in almost all areas of engineering. Typical problems in chemical engineering arise in process design, process control, model development, process identification and real-time optimization, see BIEGLER [6]. Our ultimate goal is to develop structure-driven optimization methods for solving nonlinear programming problems (NLP). This requires an objective function (e.g. minimize cost, maximize yield or profit) to be included in the model. Inequality constraints often required too. Unfortunately, Modelica does not support cost function and inequalities, only the Optimica language extension (ÅKESSON et al. [1]) does.

## 4.3 Dynamic simulation

At the moment, only the steady-state model equations of the units are implemented in Modelica. It is possible to extend the library to support dynamic simulation, but it is not easy in practice. Often, the model equations are not accurately known and the dynamic calculations may involve additional pitfalls.

**Acknowledgements.** The research was funded by the Austrian Science Fund (FWF): P23554.

## References

- [1] Johan Åkesson, Tove Bergdahl, Magnus Gäfvert, and Hubertus Tummescheit. Modeling and Optimization with Modelica and Optimica Using the JModelica.org Open Source Platform. In *Proceedings of the 7th International Modelica Conference, Como, Italy, 20-22 September 2009*, pp. 29–38. Linköping University Electronic Press, Linköpings universitet, 2009.
- [2] Johan Åkesson, Torbjörn Ekman, and Görel Hedin. Implementation of a Modelica compiler using JastAdd attribute grammars. *Science of Computer Programming*, 75:21–38, 2010.
- [3] Johan Åkesson. *Languages and Tools for Optimization of Large-Scale Systems*. PhD thesis, Regler, nov 2007.
- [4] Ali Baharev, Lubomir Kolev, and Endre Rév. Computing multiple steady states in homogeneous azeotropic and ideal two-product distillation. *AIChE Journal*, 57:1485–1495, 2011.
- [5] Ali Baharev and Arnold Neumaier. Steady-state multiplicities in reactive distillation: stage-by-stage calculation revisited. *AIChE J.*, 2012.
- [6] Lorenz T. Biegler. *Nonlinear programming: concepts, algorithms, and applications to chemical processes*. SIAM, 2010.
- [7] Chen Chang, Ding Jianwan, Chen Liping, and Wu Yizhong. Media Modeling for Distillation Process with Modelica/Mworks. In *Proceedings of the 8th International Modelica Conference, March 20th-22nd, Technical Univeristy, Dresden, Germany*, pp. 239–245. Linköping University Electronic Press, Linköpings universitet, 2011.
- [8] Amy R. Ciric and Peizhi Miao. Steady state multiplicities in an ethylene glycol reactive distillation column. *Ind. Eng. Chem. Res.*, 33:2738–2748, 1994.
- [9] R. de P. Soares and A. R. Secchi. EMSO: A new environment for modelling, simulation and optimisation. In *Computer Aided Chemical Engineering*, vol. 14, pp. 947–952. Elsevier, 2003.
- [10] M. F. Doherty, Z. T. Fidkowski, M. F. Malone, and R. Taylor. *Perry's Chemical Engineers' Handbook*. McGraw-Hill Professional, 8th ed., 2007.
- [11] N. Duro and F. Morilla. A Modelling Methodology for Distillation Columns using Dymola and Simulink. In *Applied Simulation and Modelling*. ACTA Press, 2003.
- [12] Robert Fourer, David M. Gay, and Brian Wilson Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Brooks/Cole USA, 2003.
- [13] Peter Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. Wiley-IEEE Press, 2004.
- [14] E.W. Jacobsen and S. Skogestad. Multiple steady states in ideal two-product distillation. *AIChE Journal*, 37:499–511, 1991.

- [15] Andreas Joos, Karin Dietl, and Gerhard Schmitz. Thermal separation: An approach for a modelica library for absorption, adsorption and rectification. In *Proceedings of the 7th International Modelica Conference, Como, Italy, 20-22 September 2009*, pp. 804–813. Linköping University Electronic Press, Linköpings universitet, 2009.
- [16] W. K. Lewis and G. L. Matheson. Studies in distillation. *Industrial and Engineering Chemistry*, 24:494–498, 1932.
- [17] Arnold Neumaier. Structure-driven methods for large-scale optimization, 2012. URL <http://www.mat.univie.ac.at/~neum/structure.html>.
- [18] John M. Prausnitz, Rüdiger N. Lichtenthaler, and Edmundo Gomes de Azevedo. *Molecular Thermodynamics of Fluid-Phase Equilibria*. Prentice Hall PTR, Upper Saddle River, NJ, third ed., 1999.
- [19] Carl Sandrock and Philip L. de Vaal. Dynamic simulation of chemical engineering systems using OpenModelica and CAPE-OPEN. In *19th European Symposium on Computer Aided Process Engineering*, vol. 26 of *Computer Aided Chemical Engineering*, pp. 859–864. Elsevier, 2009.
- [20] Chang K. Yi and William L. Luyben. Design and control of coupled reactor/column systems—Part 1. A binary coupled reactor/rectifier system. *Computers & Chemical Engineering*, 21(1):25–46, 1996.