Chemical Process Modeling in Modelica

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

Abstract

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 issues encountered during modeling may be valuable to the Modelica language designers.

Keywords: multiple steady states, tearing methods, homotopy continuation, bifurcation

1 Introduction

The object-oriented component-based modeling methodology in Modelica is well-suited for chemical processes modeling. Software with graphical user interface, such as the OpenModelica Connection Editor (OMEdit), can be used to build the chemical process models once the component library is available. **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, it is how it is implemented in commercial process simulators such as Aspen Plus, Aspen HYSYS or CHEMCAD.

`TODO: ref aspen`

The current chemical engineering literature is hardly accessible to mathematicians, partly due to the engineering jargon and unwritten traditions. 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**. We created a prototype Modelica implementation of basic chemical engineering processes. Currently, only steady-state models are supported.

The search for stationary states of chemical processes leads to optimization problems. **AMPL** (FOURER et al. [4]) **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. If one can use the Modelica toolchain to create the models conveniently on a GUI then, 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**. We are currently working on such an interface.

`TODO: ref exisiting`

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 [3], 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 sequentially univariate equations only, in a proper elimination order (BAHAREV & NEUMAIER [2]). 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 model instead. In 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. **The structural information is readily available and easily accessible in Modelica before flattening.**

2 Component-based modeling of chemical processes

Chemical processes are very 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 **unit operations**. The unit operations are connected by **process streams**.

Often, unit operations only exist on the level of abstraction. For example the equipment in YI & LUYBEN [8] referred to as “reactor” cannot be decomposed further into smaller, functioning pieces. However, it can be modelled by connecting 7 unit operations appropriately. None of these operations is a reactor.

In the chemical engineering literature none of the unit operations and equipments are not referred to as components. Instead, the word **“component” refers to a particular chemical substance** in the process. We are not following this chemical engineering tradition in this paper in order to adhere to the terminology of Modelica.

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.

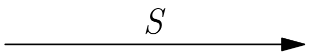

Figure 1: The graphical representation of stream S.

The components are connected by streams. The streams entering the component are called **inlets**, while the streams leaving it are called **outlets**.

2.2 Types of equations

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 and the pressure is trivial to eliminate.
Thermal equilibrium: The outlets’ enthalpy correspond 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 component well-defined. They usually correspond to closed loop control systems. The difficulty of these equations show large variation: they can be trivial equations as well as complicated nonlinear equations.

2.3 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 streams variables.

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

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

2.4 The smallest components: unit operations

The smallest subcomponents of chemical processes are called unit operations. They implement the `UnitOp` interface, that is all the equations listed in subsection 2.2 apply. The unit operations are the followings.

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 unit operations have a single inlet and can have either one or two outlets. Some code snippets are shown below. The simplicity of the implementation is the consequence of the proper decomposition of the original system.

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

class PressureChanger
  extends UnitOp(nInlet=1, nOutlet=1);
  redeclare class ChangeInPressure=pChange;
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;
```

2.5 Notes on the process stream definition

The definition of the process stream is the most innovative step in the design. Traditionally one uses the mole fractions of the chemical substances, the total molar flowrate, the pressure and the temperature. In addition, the molar enthalpy is needed to distinguish, for example, between boiling water and saturated steam, as they both have a temperature of 100°C at atmospheric pressure. See Table 2.

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

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

Having the temperature in the stream definition makes inherently linear unit operation models nonlinear. In particular, the temperature of the mixer's outlet can only be determined by solving a nonlinear system of equations. The mixer is the only unit operation having multiple inlets. Thus, a nonlinear mixer has a domino effect: many of the composite operations are no longer worth decomposing.

However, the temperature of the stream is redundant: it is uniquely determined by the other variables. Indeed, the temperature is never needed outside the unit operations. If, for some reason, the temperature of a stream is needed, one can always calculate it (by running a flash calculation). Therefore the temperature can be safely dropped from the stream definition. This step is difficult to recognize. A process stream definition without the temperature looks at first sight strange to the engineer.

Another undesirable feature of this stream definition is the fact that the process stream definition involves an equality constraint (the mole fractions sum up to 1), which is not allowed in Modelica. Eliminating one of the mole fractions would break the symmetry arbitrarily. This can be resolved if the molar flowrates of the individual substances and the total enthalpy flowrate are used in place of mole fractions. This has the beneficial side-effect that the material and heat balance equations which with the above choice are bilinear, become linear. With these changes to the stream definition, we arrive at the stream definition presented in subsection 2.1.

2.6 Hierarchical modeling: composite components

The set of unit operations listed in subsection 2.4 was found by recursively decomposing a variaty of high complexity chemical processes. As a result, the unit operations in subsection 2.4 are sufficient for general-purpose chemical processes modeling.

Figure 2 shows and 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 abstraction. The stage does not have a mixer or a flash unit iniside. Nevertheless, this decomposition is valid for modeling.

2.7 Issues encountered

The unit models are valid only if the involved molar flowrates are non-negative. This is due to the internal physical structure of the corresponding unit. The natural way to impose these non-negativity constraints is to impose it on the streams' molar flowrates and pressure,

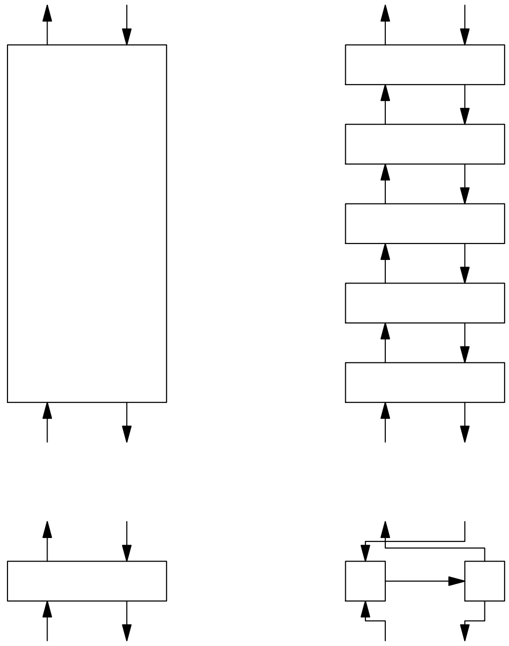

Figure 2: Hierarchical decomposition of the vapor-liquid equilibrium cascade into a cascade of stages, then the decomposition of a stage into a mixer and a flash unit.

that is, in the connector class. The need for specifying inequality constraints is the most significant difficulty when modeling chemical processes in Modelica. Currently, only the Optimica language extension supports inequalities.

Another difficulty is that Modelica cannot handle arrays of components that have optional connections. This makes the creation of cascades somewhat awkward as missing feeds have to be simulated by dummy streams.

`TODO a stage with optional feed`

3 Separation operations

The Modelica implementation discussed in the previous section is tested on the benchmark of BAHAREV et al. [1]. First, the background of the application is briefly presented. Then, numerical results are given for the particular benchmark.

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. 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 by-products. Usually a significant fraction of the reactants remain unreacted. These unreacted reactants, the products and the waste by-products 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 for separation operation equipments (PRAUSNITZ et al. [7]).

Many of the practically relevant equipments used in separation operations (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 an unusual and counterintuitive way.

3.1 Internal physical structure of distillation columns.

The body of a multistage distillation column is a cascades of stages. The output of one stage is the input of its two neighbors and vice versa, see Figure 2.

This internal physical structure is reflected in the mathematical model of the columns. The equations can be evaluated in a sequential manner after guessing a few variables at one end of the cascade. This is well-known for chemical engineers since the 1930's, this method is called the stage-by-stage calculation (LEWIS & MATHESON [6]). The essential dimension of the problem is given by the number of variables that has to be guessed to start the stage-by-stage computations.

3.2 Multiple steady-states in ideal two-product distillation

Perhaps the simplest distillation columns are the single feed two-product columns with ideal vapor-liquid equilibrium. Even these columns can have multiple stead-states. 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 JACOBSEN & SKOGESTAD [5]. 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. The model equations have five distinct solutions in a certain range of the relux flow rate. One of the solutions is infeasible in practice because it would result in negative flow rates. The infeasible steady-state is represented by a dashed line in Figure 3.

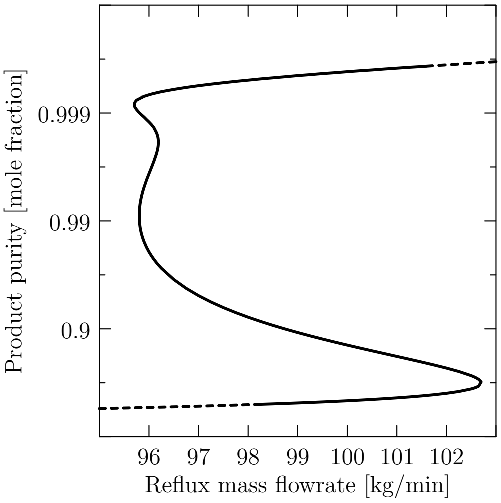

Figure 3: Bifurcation diagram, multiple steady-states in ideal two-product distillation.

4 Future work

4.1 Extensions: optimization and dynamic simulation

The ultimate goal of the this project is to develop structure-driven optimization methods for solving nonlinear programming problems (NLP). At the moment, only the steady-state model equations of the components are implemented. However, it is fairly straighforward to extend the library to support optimization and dynamic simulation.

4.2 Recovering structural information

The structural information (connections of the components) can be exploited to solve the underlying process model efficiently, as already mentioned in the introduction. The core equations of the column in Subsection 3.2 are copied from the implementation and are shown below. The Modelica code is favorable when it comes to structural information, since it speaks about connections as clearly as possible.

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];
```

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

TODO: We have no Modelica reference...

References

[1] 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.

[2] Ali Baharev and Arnold Neumaier. Steady-state multiplicities in reactive distillation: stage-by-stage calculation revisited. *AIChE J.*, 2012.

[3] Amy R. Ciric and Peizhi Miao. Steady state multiplicities in an ethylene glycol reactive distillation column. *Ind. Eng. Chem. Res.*, 33:2738–2748, 1994.

[4] Robert Fourer, David M. Gay, and Brian Wilson Kernighan. *AMPL: A Modeling Language for Mathematical Programming*. Brooks/Cole USA, 2003.

[5] E.W. Jacobsen and S. Skogestad. Multiple steady states in ideal two-product distillation. *AIChE Journal*, 37:499–511, 1991.

[6] W. K. Lewis and G. L. Matheson. Studies in distillation. *Industrial and Engineering Chemistry*, 24: 494–498, 1932.

[7] 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.

[8] 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.