

Interfacing Models for Thermal Separation Processes with Fluid Property Data from External Sources

Kai Wellner^{*1}, Carsten Trapp², Gerhard Schmitz¹ and Francesco Casella³

¹*Hamburg University of Technology, Institute of Thermo-Fluid Dynamics
Denickestraße 17, 21075 Hamburg, Germany*

²*Delft University of Technology, Propulsion and Power, Kluyverweg 1, 2629 HS Delft, The Netherlands*

³*Politecnico di Milano, Dipartimento di Elettronica, Informazione e Bioingegneria
Via Ponzio 34/5, 20133 Milano, Italy*

Abstract

So far, when modelling processes that demand for multi-phase and multi-component fluid property data, the user has to implement the required media models in the Modelica language as these types of fluids are not supported by Modelica.Media. This paper presents a first approach on how to implement fluid property data in process models of an existing library from external sources and highlights which problems have to be overcome. Furthermore, it provides recommendations for the design of an efficient and user-friendly interface to external media packages.

Keywords: thermal separation; two-phase; multi-component mixture; external fluid property data; transient simulation

1 Introduction and motivation

Using existing model libraries for the modelling of complex chemical processes is especially challenging because it can be hard to comprehend the structure of the models. Nevertheless, when using these models one might want to implement different media models from what is already available. One convenient way is accessing external property packages instead of developing and implementing required fluid property models in the Modelica language. In this context it might be necessary to provide, next to the primary thermodynamic property data, also property derivatives.

The main objective is to develop an interface between existing Modelica libraries such as the `ThermalSeparation` library and fluid property packages, which successfully deal with situations where

the total time derivative of fluid properties is required. This can be the case for several reasons:

- The formulation of the model equations results in a higher index problem.
- The system states do not equal the variables in the derivative operator.
- There are more variables in the derivative operator than there are differential equations.

The simulation tool handles all three situations the same way which is by applying Pantelides' algorithm and deriving the necessary equations in this process [1]. In contrast to the first bullet point, the last bullet points do not necessarily result in a high index problem. But in both cases additional equations have to be derived by time (which can include fluid property relations) to solve the whole set of equations. In the following this technique is referred to as automated index reduction, regardless of the origin of the problem.

In comparison to developing models from scratch where it is possible to maintain an index-1 form of the system, reducing the index of existing models is in general not feasible. Further information on the differential index and index reduction is provided by Fritzson [2].

An earlier attempt to couple the `ThermalSeparation` library to an external property package still lacked a proper interface [3]. Later on a coupling to external fluid properties was successful although the balance equations had to be modified for this purpose because no derivatives were available and the automated index reduction failed [4].

Some general advantages and reasons for the need to interface external thermodynamic property data

^{*}kai.wellner@tuhh.de, schmitz@tuhh.de

with Modelica models are summarized in the following:

- The Modelica language is increasingly used for process modelling in the field of chemical engineering which entails the modelling of multi-phase, multi-component media. Currently, this type of fluids are not supported by Modelica.Media. In case the user needs such media models, an implementation from scratch is required. This also involves coding of flash calculations in order to determine the equilibrium composition for example for a given p , T , x_{total} . Depending on the complexity of the fluid mixture and the used equation of state (EoS), the implementation can be quite cumbersome, time-consuming and error-prone.
- Using existing property packages provides a higher flexibility in terms of available components and EoS. For a Modelica model library developed for a wide range of different applications it is typically not practical or even possible to implement a large number of media models covering any application case. Therefore, providing a user-friendly interface to external property packages is an important feature.
- Calculating property data in an external environment can significantly increase the simulation speed [5]. However this is not a general feature because in some cases it can have the exact opposite effect and decrease the simulation speed.

2 The ThermalSeparation library

In process engineering the purification of a product plays an important role. Usually the intermediate products are not of the desired purity and have to be treated in a downstream process. Such processes often involve thermal separation like absorption in gas-liquid separation columns. As the operation of post-treatments like separation can be highly dynamic, computer simulations are indispensable for a better understanding of the process and its optimization targeting for example energy consumption or control strategy design.

The ThermalSeparation library has been developed for the simulation of absorption and rectification processes. The challenge when interfacing external fluid property data is that in the current implementation the balance equations are written in an way that

forces the simulation tool to apply automated symbolic index reduction. If that happens, implementing fluid property data from external sources demands special requirements. Further information on how the formulation of balance equations can affect the use of external fluid property packages and how these problems can be overcome is given in section 3.

2.1 Library Structure

Compared to earlier versions of the ThermalSeparation library [4] a new feature that has been implemented in the framework of this paper is the replaceability of the balance equations which easily allows to switch between different formulations of balance equations.

For this work a modified set of balance equations that does not require automated symbolic index reduction is added to illustrate the differences to the original equations in terms of computational performance. A UML class diagram of the library structure is shown in figure 1.

Vapour and liquid medium models that invoke external fluid property calls are added to the respective classes. Moreover, a film model for the external call of the flash calculations is implemented. The classes that are adapted to use external media data are indicated with a bold frame in the UML diagram.

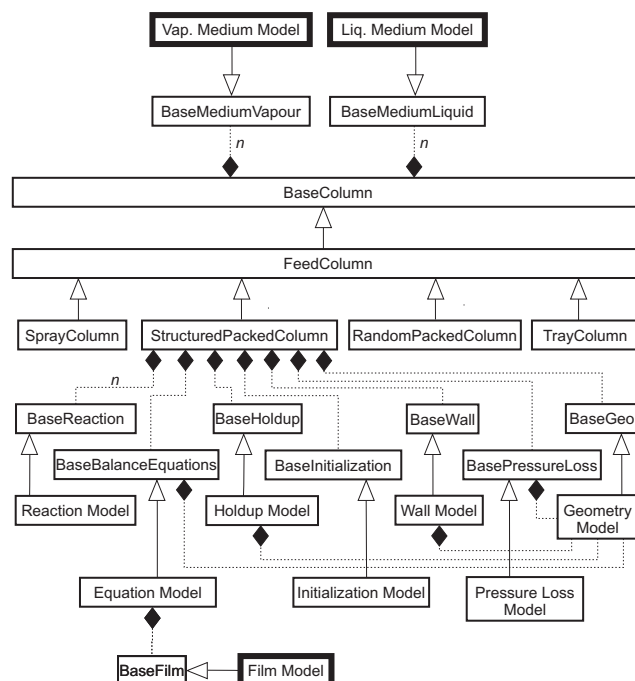


Figure 1: UML diagram of the column model structure.

2.2 Interfacing the property package

The use of external sources for the computation of thermodynamic properties is in this paper exemplary demonstrated with the FluidProp package, which is interfaced via the ModelicaFluidProp library [6, 7]. For calculating the actual properties the "PC-SAFT" (perturbed chain statistical associating fluid theory) EoS is employed due to its success in predicting vapour/gas-liquid equilibria of complex fluids and mixtures for a broad range of conditions and due to fact that it also provides the partial derivatives of specific molar enthalpy and specific molar volume with respect to pressure, temperature and composition. These derivatives are necessary to compute the total time derivative (cf. section 3.2). The derivatives are obtained by separate function calls. To enable the use of the total time derivatives in the models of the ThermalSeparation library, the implemented Modelica property functions are modified such that these derivatives are specified by using the derivative annotation. This way the simulation environment can perform automated symbolic index reduction using the specified total time derivatives of the fluid property functions which enables the modeller to perform simulations with variable system states or of higher order index, which would otherwise fail when using external property packages. An example on how the FluidProp calls are implemented in the ThermalSeparation library media models is illustrated in listing 1. The medium definition is programmed in a separate class and mainly consists of string operators that define the fluid constituents and the medium model (EoS) used for the property calculations.

The implemented property functions require as input a string with the thermodynamic state definition (e.g. "PT" for pressure and temperature), the actual values for these states and the overall molar composition. Partial derivatives are currently available with respect to temperature, pressure and composition.

```

model BaseProperties
  ThermalSeparation.Units.MolarEnthalpy h
    "Specific enthalpy of medium";
  ThermalSeparation.Media.Types.Density d
    "Density of medium";
  FluidPropMedium.AllPropsOut prop_mol
    "external property record";
  SI.MoleFraction conc[nSubstance]
    "total composition";
  SI.Pressure p "system pressure";
  SI.Temperature T "system temperature";
equation

```

```

(prop_mol) = Medium.AllPropsMole(
  "PT",
  p,
  T,
  Conc);
h=prop_mol.h;
d=prop_mol.d;
annotation(derivative=
  Medium.AllPropsMole_der);
end BaseProperties;

```

Listing 1: Fluid property call.

The computation of the equilibrium compositions for the vapour-liquid phase boundary is treated in a separate class. The respective function call requires the same inputs as the property calls and returns the equilibrium composition for each phase:

```

model ThermodynamicEquilibrium
  SI.MoleFraction conc[nSubstance]
    "total composition";
  SI.Pressure p "system pressure";
  SI.Temperature T "system temperature";
  SI.MoleFraction x_eq[nSubstance]
    "liq. phase equilibrium composition";
  SI.MoleFraction y_eq[nSubstance]
    "vap. phase equilibrium composition";
equation
  (x_eq, y_eq) =
    Medium.AllProps2phMole(
      "PT",
      p,
      T,
      Conc);
end BaseProperties;

```

Listing 2: Flash calculation call.

3 Modelling approach

There are two different approaches when using external fluid property data concerning the modelling structure of the Modelica process models.

In the first approach an automated index reduction is avoided by writing (or rewriting in case of existing models) the balance equations in an explicit manner with respect to the state variables. The external property data can easily be integrated in general without the necessity to provide any partial derivatives.

In the second approach, the balance equations are written in the most "natural" way which might foster numerical robustness and simulation speed or ease of initialization in case of new models. In case of existing models the balance equations are maintained as coded.

Typically this leads to an implicit formulation of the balance equations.

Both approaches are explained in the following:

3.1 Explicit modelling of balancing equations

For a separation column with rate-based modelling approach the balance equations for one stage can be written as:

$$\frac{dN_{i,j}^L}{dt} = \dot{N}_{in,j+1}^L \cdot x_{in,i,j+1}^L - \dot{N}_{out,j}^L \cdot x_{out,i,j}^L + \dot{N}_{PB,j}^L \quad (1)$$

$$\frac{dN_{i,j}^V}{dt} = \dot{N}_{in,j-1}^V \cdot x_{in,i,j-1}^V - \dot{N}_{out,j}^V \cdot x_{out,i,j}^V + \dot{N}_{PB,j}^V \quad (2)$$

$$\frac{dU_j^L}{dt} = \dot{N}_{in,j+1}^L \cdot h_{in,j+1}^L - \dot{N}_{out,j}^L \cdot h_{out,j}^L + \dot{Q}_{PB,j}^L \quad (3)$$

$$\frac{dU_j^V}{dt} = \dot{N}_{in,j-1}^V \cdot h_{in,j-1}^V - \dot{N}_{out,j}^V \cdot h_{out,j}^V + \dot{Q}_{PB,j}^V \quad (4)$$

Where $N_{i,j}$ refers to the molar content of component i on stage j for the liquid and vapour phase respectively. \dot{N}_{in} and \dot{N}_{out} are the molar flow rates into and out of the stage. Furthermore, x is the molar fraction, h is the specific molar enthalpy, \dot{N}_{PB} and \dot{Q}_{PB} are the mole and energy flow rates across the phase boundary and are calculated using constitutive equations. Figure 2 shows a schematic diagram of such column stage.

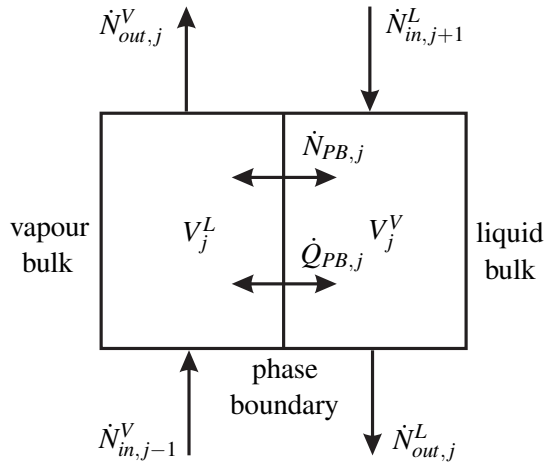


Figure 2: Schematic diagram of a column stage.

In order to avoid the differentiation of additional equations and therefore the necessity to provide total time derivatives of the fluid properties, the system states have to be the same as the differentiated variables, in this case N_i^L , N_i^V , U^L and U^V . To further increase simulation speed, the system states should equal the thermodynamic states, which are the inputs to the property calls. In this way, the fluid properties

can be directly computed from the state variables and therefore no iterations are required. For this purpose the left hand sides of the differential equations have to be rewritten as a function of the thermodynamic state variables. A recommendation how this can be implemented is given in section 5.

Now the model is designed work without fluid property derivatives. However, as soon as the user chooses a different state variable as the ones named above, additional derivatives are required again. Hence, these types of models are not flexible because variable states can be an advantage in model flexibility as they provide additional robustness.

3.2 Implicit modelling of balance equations

Using existing model libraries with external fluid properties can be somewhat more complicated because the balance equations may not always be modelled in an explicit manner as explained in equations (1)-(4). For example in the ThermalSeparation library the balance equations are formulated like this:

$$\frac{d(V_j^L \cdot c_{i,j}^L)}{dt} = \dot{N}_{in,j+1}^L \cdot x_{in,i,j+1}^L - \dot{N}_{out,j}^L \cdot x_{out,i,j}^L + \dot{N}_{PB,j}^L \quad (5)$$

$$\frac{d(V_j^V \cdot c_{i,j}^V)}{dt} = \dot{N}_{in,j-1}^V \cdot x_{in,i,j-1}^V - \dot{N}_{out,j}^V \cdot x_{out,i,j}^V + \dot{N}_{PB,j}^V \quad (6)$$

$$\frac{d[V_j^L \cdot \sum_{i=1}^n (u_{i,j}^L \cdot c_{i,j}^L)]}{dt} = \dot{N}_{in,j+1}^L \cdot h_{in,j+1}^L - \dot{N}_{out,j}^L \cdot h_{out,j}^L + \dot{Q}_{PB,j}^L \quad (7)$$

$$\frac{d[V_j^V \cdot \sum_{i=1}^n (u_{i,j}^V \cdot c_{i,j}^V)]}{dt} = \dot{N}_{in,j-1}^V \cdot h_{in,j-1}^V - \dot{N}_{out,j}^V \cdot h_{out,j}^V + \dot{Q}_{PB,j}^V \quad (8)$$

Where V is the volume, c is the concentration and u is the specific internal energy.

Experience has shown that this way of modelling balance equations in the ThermalSeparation library provides the best performance in terms of numerical robustness and robust initialization. Moreover, to provide the full flexibility it is necessary to be able to choose the system states freely. Further information on this is given in Dietl [4].

It is obvious that the number of differentiated variables (V_j^L , V_j^V , $c_{i,j}^L$, $c_{i,j}^V$, $u_{i,j}^L$ and $u_{i,j}^V$) is higher than the number of differential equations (5)-(8). This results in a system that can only be solved if the missing

derivatives are provided by deriving additional equations. In the present case this also includes the differentiation of fluid properties, which subsequently causes the compilation to fail as the external property functions cannot be differentiated by the simulation environment. One way to circumvent the compilation failure without changing the governing equations is to provide the total time derivatives for the property data according to equations (9) and (10). Therefore, the partial derivatives of the properties with respect to the system states are required. Equations (9) and (10) are contained in the function that is invoked by the derivative annotation in listing 1. In the column model the specific molar volume v is used to convert the molar fraction x to the concentration c .

$$\frac{dh}{dt} = \left(\frac{dh}{dp}\right)_{T,x_i} \cdot \left(\frac{dp}{dt}\right) + \left(\frac{dh}{dT}\right)_{p,x_i} \cdot \left(\frac{dT}{dt}\right) \quad (9)$$

$$+ \left(\frac{dh}{dx_i}\right)_{p,T} \cdot \left(\frac{dx_i}{dt}\right)$$

$$\frac{dv}{dt} = \left(\frac{dv}{dp}\right)_{T,x_i} \cdot \left(\frac{dp}{dt}\right) + \left(\frac{dv}{dT}\right)_{p,x_i} \cdot \left(\frac{dT}{dt}\right) \quad (10)$$

$$+ \left(\frac{dv}{dx_i}\right)_{p,T} \cdot \left(\frac{dx_i}{dt}\right)$$

In contrast to this a higher index problem in chemical process models can arise for example because of the volume constraint when assuming incompressibility of the fluid or when chemical reactions are modelled in an equilibrium fashion [8].

4 Example of use

In the following the implementation of both approaches (use of explicit balance equations and use of implicit balance equations) is demonstrated and compared for the simulation of a rate-based column model employing a fluid mixture of n-propane and n-pentane using external property data. Both approaches of modelling balance equations are successfully implemented, which has been proven by the fact that with both models the same solution has been obtained.

Inside the medium models the specific volume, the specific enthalpy and the specific heat capacity are used from FluidProp for both vapour and liquid phase. The specific internal energy can be directly calculated from the specific enthalpy. Additionally, the equilibrium composition is used in the film model. The property calls are invoked using pressure and temperature

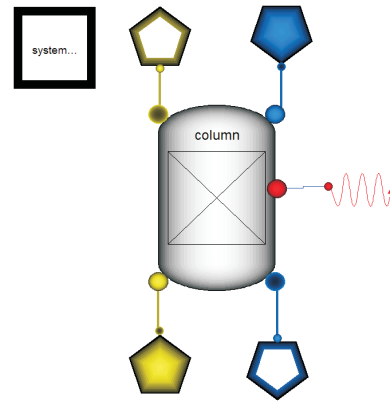


Figure 3: Diagram of the Modelica column model.

as thermodynamic states. The system states of the column model are pressure, temperature and molar composition of liquid and vapour phase.

Figure 4 depicts the comparison in simulation time of both models, implicit and explicit balance equations, for a step change in the feed flow rate at 500 seconds when the system has reached a steady-state after initialization.

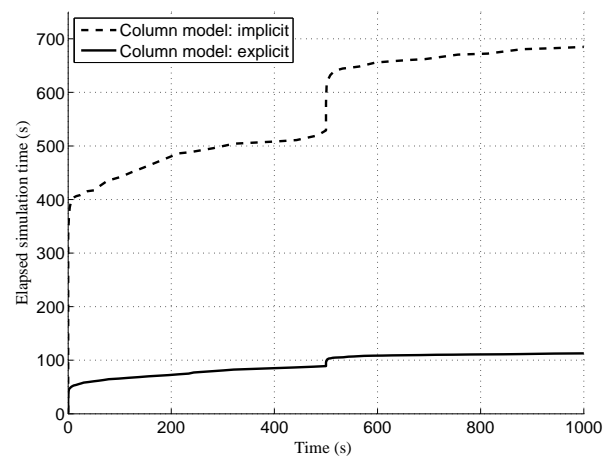


Figure 4: Comparison of simulation times.

When comparing both models it can be observed that the simulation time for the implicit model is much higher. This is explained by the fact that the calculation of partial derivatives is computationally very expensive which can be seen in sections with high gradients like initialization or change of boundary conditions. There the simulation of the implicitly modelled system slows down significantly in comparison to the model that does not use external derivatives. Thus it can be concluded that from a performance point of view it is advantageous to model the system in an explicit manner in order to avoid the necessity to com-

pute partial derivatives.

While the calculation of the external fluid properties requires the biggest part of the computational effort (especially when partial derivatives are needed), it also possesses a large room for improvement and thus for the decrease of the overall simulation time.

Aside from the external calculation of the partial derivatives there are other possible sources for the slowdown of the simulation in this specific case. First of all this modelling approach combined with the use of external property data introduces numerical jacobians that have to be calculated in order to obtain the solution of the system. These jacobians can both destabilize and slow down the simulation. Furthermore, implicit modelling of the balancing equations in combination with the use of external derivatives can lead to larger blocks of non-linear system of equations that have to be solved iteratively as tearing cannot be applied in an efficient way. Table 1 shows a comparison for the sizes of non-linear system of equations between explicit and implicit modelling approaches for the example of use.

	Implicit model	Explicit model
(1)	{58, 52, 68, 1, 1, 1, 1}	{58, 50, 54, 1, 1}
(2)	{6, 2, 56, 0, 0, 0, 0}	{8, 2, 6, 0, 0}

Table 1: Sizes of non-linear system of equations before (1) and after manipulation (2).

The presented simulations are using rate-based column models from the `ThermalSeparation` library, which means that in terms of external property computations one call is required for each phase and one additional call for the film model in order to determine the thermodynamic equilibrium. If the partial derivatives are needed they are invoked via the derivative annotation additionally for each property call. Having three property calls per column stage is obviously computationally expensive, especially when derivatives are calculated, in comparison to e.g. an equilibrium stage model where only one property call per column stage would be sufficient.

5 Recommendations for improvement of the interface and balance equation modelling

The current interface does not provide the possibility to choose between function calls with or without the

output of the total time derivatives via the derivative annotation. If they are needed, they have to be calculated from partial derivatives which have to be invoked with a separate function call. For a future interface it should be possible to opt between functions that provide derivatives and such that do not. In principle it would be sufficient to have one function and the derivatives in the annotation are only calculated when needed. In practice however this does not work because for complex models (like in the example of use) the derivatives are also computed when they are not needed. So on the one hand calculating the derivatives all the time consumes too much CPU time if it is not required but on the other hand sometimes the derivatives are absolutely necessary.

Furthermore it would be convenient if the user can specify the needed derivatives to be able to adapt them to the system states of the model that the derivatives are used in (e.g. the column model). Listing 3 shows a suggestion how such an interface can look like for both options, with and without output of derivatives. The design is inspired by the current implementation and expanded with a `NumericalState` record in case the derivatives are required. The derivatives in the `prop_der.dh[:]` array are arranged in the same order as the numerical states inside the `NumericalState` record. Although these recommendations are related to the current interface of the `ModelicaFluidProp` library they can be taken as general recommendations for the design of future interfaces.

```
(prop) = Medium.AllProps(
    ThermodynamicState,
    Value1,
    Value2,
    Conc);
h=prop.h;

(prop_der) = Medium.AllPropsDer(
    ThermodynamicState,
    Value1,
    Value2,
    Conc,
    NumericalState);
h=prop_der.h;
dhdp=prop_der.dh[1];
dhdT=prop_der.dh[2];
dhdx[1]=prop_der.dh[3];
dhdx[2]=prop_der.dh[4];
```

Listing 3: Design recommendation for fluid property interface

In addition, the general way of modelling balance equations for processes involving multi-phase and multi-component fluids should be reconsidered. The

modelling of balance equations with mass, internal energy and mass fractions originates from the modelling of energy systems. This approach might not be optimal for problems in the field of chemical engineering. For chemical processes the balance equations should rather be set up with volume related properties like for example density or the molar specific volume as explained in Pantelides [8]. For an equilibrium stage model the left hand side of the mole balance can be rewritten according to equation (11).

$$\frac{dN_i}{dt} = V \cdot \frac{d(v \cdot x_i)}{dt} \quad (11)$$

The advantage of this structure is that fluid property calls can directly be invoked with the system states "u, ρ " or "u,v" (depending on whether the model is on mass or molar basis) and still an automated index reduction due to the need of additional derivatives is avoided.

6 Summary and conclusions

This paper demonstrates and compares two approaches on the use of external property packages for the computation of fluid data interfaced to Modelica process models. With the exemplary column model using an explicit formulation of the balance equations the use of partial derivatives is avoided. This model is in terms of computational speed favourable in comparison to the model with an implicit formulation of the balance equations. However, especially when utilizing existing model libraries rewriting balance equations is not convenient or feasible. The paper shows that in this case the external property package has to provide partial derivatives of the fluid properties with respect to the system states. This enables the simulation environment to perform automated symbolic index reduction and maintains the flexibility of choosing the state variables freely.

7 Acknowledgements

This research project is supported by the Federal Ministry of Economics and Technology (project number 03ET2009).



Nomenclature

\dot{N}	molar flow rate (mol/s)
\dot{Q}	heat flow rate (W)
c	concentration (mol/m ³)
h	specific molar enthalpy (J/mol)
N	amount of substance (mol)
p	pressure (Pa)
T	temperature (K)
U	internal energy (J)
u	specific internal energy (J/mol)
V	volume (m ³)
v	specific molar volume (mol/m ³)
x	mole fraction (mol/mol _{tot})

Superscripts

L	liquid phase
V	vapour phase

Subscripts

i	component i
in	inflow stream/attribute
j	column stage j
out	outflow stream/attribute
PB	phase boundary

References

- [1] C. Pantelides. The Consistent Initialization of Differential-Algebraic Systems. *SIAM journal of scientific computing*, 9:213 – 231, 1988.
- [2] P. Fritzson. *Principles of Object-Oriented Modeling and Simulation with Modelica 2.1*. ISBN 0-471-47163-1. Wiley, June 2000.
- [3] A. Joos, K. Dietl, and G. Schmitz. Thermal Separation: An Approach for a Modelica Library for Absorption,

- Adsorption and Rectification. In *Proceedings 7th Modelica Conference, Como, Italy, September 20-22, 2009*.
- [4] K. Dietl. *Equation-Based Object-Oriented Modelling of Dynamic Absorption and Rectification Processes*. Dissertation, Hamburg University of Technology, Institute of Thermo-Fluid Dynamics, Hamburg, 2012. ISBN 978-3-8439-0778-1.
- [5] J. Brunnemann, F. Gottelt, K. Wellner, A. Renz, A. Thüring, V. Roeder, C. Hasenbein, C. Schulze, G. Schmitz, and J. Eiden. Status of ClaRaCCS: Modelling and Simulation of Coal-Fired Power Plants with CO₂ Capture. In *Proceedings 9th Modelica Conference, Munich, Germany, September 3-5, 2012*.
- [6] P. Colonna, T. van der Stelt, and A. Guardone. FluidProp: a program for the estimation of thermophysical properties of fluids. Version 2.4, software (2004-2012). <http://www.FluidProp.com>.
- [7] C. Trapp, F. Casella, and P. Colonna. Use of External Fluid Property Code in Modelica for Modelling of a Pre-combustion CO₂ Capture Process Involving Multi-Component, Two-Phase Fluids. Submitted to the 10th Modelica Conference, Lund, Sweden, March 10-12, 2014.
- [8] C. Pantelides. The Mathematical Modelling of the Dynamic Behaviour of Process System, October 2000.