

HelmholtzMedia — A Fluid Properties Library

Matthis Thorade, Ali Saadat

Helmholtz Centre Potsdam GFZ German Research Centre for Geosciences

Abstract

HelmholtzMedia is a library for the calculation of fluid properties. It is implemented in Modelica and published under the Modelica license. All thermodynamic state properties and their partial derivatives are calculated from a Helmholtz energy equation of state. Further properties that can be calculated include surface tension, viscosity and thermal conductivity.

Keywords: thermodynamic properties, Helmholtz energy, surface tension, viscosity, thermal conductivity

1 Introduction

For the simulation and design of power or refrigeration cycles, accurate properties of the working fluid are indispensable. The most accurate equations of state (EoS) available today for a variety of working fluids are fundamental EoS in terms of Helmholtz energy. From such EoS all thermodynamic state properties, like pressure p or specific entropy s , as well as all partial derivatives of thermodynamic state variables can be calculated.

Further properties of interest are surface tension, viscosity and thermal conductivity. For each of these properties an independent correlation is necessary.

Both the Helmholtz energy EoS as well as correlations for additional properties have been implemented in the HelmholtzMedia library. Details of the implementation are given in the following text.

2 Helmholtz energy fundamental equation of state

A historical overview over the development of fundamental EoS in general is given by [2](in German), an overview over the functional form used today by almost all Helmholtz EoS is given by [8]. The independent variables of the Helmholtz EoS are temperature T and specific volume v or density ρ . Both are non-dimensionalised by their critical values. The

Helmholtz energy f is non-dimensionalised by the specific gas constant R and the temperature T and split up into an ideal gas part α^0 and a residual part α^r . This allows for developing a functional form for the two parts independently.

$$\tau = \frac{T_c}{T}, \quad \delta = \frac{v_c}{v} = \frac{\rho}{\rho_c}, \quad \alpha = \frac{f}{RT} = \alpha^0 + \alpha^r$$

The functional form for the description of the ideal part of the Helmholtz energy results from the thermal equation of state of the ideal gas and a two-fold integration of the heat capacity of the ideal gas. The heat capacity of the ideal gas can be described by polynomial terms, by so-called Planck-Einstein terms or by a combination of the two. Alternatively, hyperbolic functions can be used, but these have not been implemented so far.

$$\begin{aligned} \alpha^0(\delta, \tau) = & \log(\delta) \\ & + \sum_{i=1}^{i=nL} l_{[i,1]} \log[\tau^{l_{[i,2]}}] \\ & + \sum_{i=1}^{i=nP} p_{[i,1]} \cdot \tau^{p_{[i,2]}} \\ & + \sum_{i=1}^{i=nE} e_{[i,1]} \cdot \log[1 - \exp(e_{[i,2]} \cdot \tau)] \end{aligned}$$

The functional form for the description of the residual part of the Helmholtz energy as implemented uses three groups of terms: polynomial terms, so-called Benedict-Webb-Rubin terms and Gaussian bell-shaped terms. For some fluids (e. g. CO₂ or water) the functional form contains additional non-analytical terms

Table 1: Thermodynamic state properties [7]

Property	Algorithm
pressure p	$\rho TR [1 + \delta\alpha_\delta^r]$
entropy s	$R [\tau(\alpha_\tau^0 + \alpha_\tau^r) - (\alpha^0 + \alpha^r)]$
internal energy u	$TR [\tau(\alpha_\tau^0 + \alpha_\tau^r)]$
enthalpy h	$TR [(1 + \delta\alpha_\delta^r) + \tau(\alpha_\tau^0 + \alpha_\tau^r)]$
Gibbs-energy g	$TR [1 + (\alpha^0 + \alpha^r) + \delta\alpha_\delta^r]$

that have not been implemented so far.

$$\alpha^r(\delta, \tau) = \sum_{i=1}^{i=nP} p_{[i,1]} \cdot \delta^{p_{[i,3]}} \cdot \tau^{p_{[i,2]}} + \sum_{i=1}^{i=nB} b_{[i,1]} \cdot \delta^{b_{[i,3]}} \cdot \tau^{b_{[i,2]}} \cdot \exp[-\delta^{b_{[i,4]}}] + \sum_{i=1}^{i=nG} g_{[i,1]} \cdot \delta^{g_{[i,3]}} \cdot \tau^{g_{[i,2]}} \cdot \exp[g_{[i,6]} \cdot (\delta - g_{[i,9]})^2 + g_{[i,7]} \cdot (\tau - g_{[i,8]})^2]$$

A short discussion of all terms is given in [13, Section 5], a very comprehensive discussion is given in [7]. The parameters of the two contributions to the Helmholtz energy are then fitted to experimental data for each fluid. Details on the fitting procedure can be found in [7].

Once the functional form and values for the parameters are known, all state properties can be calculated as simple combinations of the partial derivatives of the Helmholtz energy¹. Algorithms for the calculation of the state properties are given in [7], an extract is repeated in Table 1.

In addition to the state properties, the partial derivatives of state properties are often needed in engineering applications, for example specific heat capaci-

¹The partial derivatives of the Helmholtz energy are abbreviated as follows:

$$\alpha_\tau^0 = \left(\frac{\partial \alpha^0}{\partial \tau} \right)_\delta, \quad \alpha_{\tau\tau}^0 = \left(\frac{\partial^2 \alpha^0}{\partial \tau^2} \right)_\delta, \quad \alpha_{\tau\delta}^0 = \left(\frac{\partial^2 \alpha^0}{\partial \tau \partial \delta} \right)_\delta$$

$$\alpha_\delta^0 = \left(\frac{\partial \alpha^0}{\partial \delta} \right)_\tau, \quad \alpha_{\delta\delta}^0 = \left(\frac{\partial^2 \alpha^0}{\partial \delta^2} \right)_\tau$$

$$\alpha_\tau^r = \left(\frac{\partial \alpha^r}{\partial \tau} \right)_\delta, \quad \alpha_{\tau\tau}^r = \left(\frac{\partial^2 \alpha^r}{\partial \tau^2} \right)_\delta, \quad \alpha_{\tau\delta}^r = \left(\frac{\partial^2 \alpha^r}{\partial \tau \partial \delta} \right)_\delta$$

$$\alpha_\delta^r = \left(\frac{\partial \alpha^r}{\partial \delta} \right)_\tau, \quad \alpha_{\delta\delta}^r = \left(\frac{\partial^2 \alpha^r}{\partial \delta^2} \right)_\tau$$

Table 2: Partial derivatives of state properties [10]

Property	Algorithm
$\left(\frac{\partial p}{\partial \rho} \right)_T$	$TR [1 + 2\delta\alpha_\delta^r + \delta^2\alpha_{\delta\delta}^r]$
$\left(\frac{\partial p}{\partial T} \right)_\rho$	$\rho R [1 + \delta\alpha_\delta^r - \delta\tau\alpha_{\tau\delta}^r]$
$\left(\frac{\partial s}{\partial \rho} \right)_T$	$\frac{R}{\rho} [-(1 + \delta\alpha_\delta^r) + \tau\delta\alpha_{\tau\delta}^r]$
$\left(\frac{\partial s}{\partial T} \right)_\rho$	$\frac{R}{T} [-\tau^2(\alpha_{\tau\tau}^0 + \alpha_{\tau\tau}^r)]$
$\left(\frac{\partial u}{\partial \rho} \right)_T$	$\frac{TR}{\rho} [\tau\delta\alpha_{\tau\delta}^r]$
$\left(\frac{\partial u}{\partial T} \right)_\rho$	$R [-\tau^2(\alpha_{\tau\tau}^0 + \alpha_{\tau\tau}^r)]$
$\left(\frac{\partial h}{\partial \rho} \right)_T$	$\frac{TR}{\rho} [\tau\delta\alpha_{\tau\delta}^r + \delta\alpha_\delta^r + \delta^2\alpha_{\delta\delta}^r]$
$\left(\frac{\partial h}{\partial T} \right)_\rho$	$R [1 - \tau^2(\alpha_{\tau\tau}^0 + \alpha_{\tau\tau}^r) + \delta\alpha_\delta^r - \tau\delta\alpha_{\tau\delta}^r]$
$\left(\frac{\partial g}{\partial \rho} \right)_T$	$\frac{TR}{\rho} [1 + 2\delta\alpha_\delta^r + \delta^2\alpha_{\delta\delta}^r]$
$\left(\frac{\partial g}{\partial T} \right)_\rho$	$R [(\alpha^0 + \alpha^r) + (1 + \delta\alpha_\delta^r) - \tau(\alpha_\tau^0 + \alpha_\tau^r) - \tau\delta\alpha_{\tau\delta}^r]$

ties, the thermal expansion coefficient β , or the isothermal compressibility κ . Any partial derivative can be calculated in a two-step procedure: First, the partial derivatives with respect to *temperature and density*, the independent variables of the EoS, are formed. These are given in [10] and repeated in Table 2. Second, all further derivatives with respect to *arbitrary* state properties can then be transformed into simple combinations of the partial derivatives with respect to temperature and density, using the rules for Jacobian matrix transformations.

For example, the partial derivatives of density with respect to pressure and enthalpy, which are helpful for transient simulation of power cycles, can be expressed as

$$\left(\frac{\partial \rho}{\partial p} \right)_h = \left[\left(\frac{\partial p}{\partial \rho} \right)_T - \left(\frac{\partial p}{\partial T} \right)_\rho \left(\frac{\partial h}{\partial \rho} \right)_T \left(\frac{\partial h}{\partial T} \right)_\rho^{-1} \right]^{-1}$$

and

$$\left(\frac{\partial \rho}{\partial h}\right)_p = \left[\left(\frac{\partial h}{\partial \rho}\right)_T - \left(\frac{\partial h}{\partial T}\right)_\rho \left(\frac{\partial p}{\partial \rho}\right)_T \left(\frac{\partial p}{\partial T}\right)_\rho^{-1} \right]^{-1}$$

More examples are given in [10].

3 Vapor-liquid equilibrium and two-phase state

The vapour-liquid equilibrium (VLE) of a pure fluid is characterized by three conditions:

$$\begin{aligned} \text{thermal equilibrium: } \Delta T &= (T' - T'') = 0 \\ \text{mechanical equilibrium: } \Delta p &= (p' - p'') = 0 \\ \text{chemical equilibrium: } \Delta g &= (g' - g'') = 0. \end{aligned}$$

For a given temperature T the equilibrium state can be determined by simultaneously solving the equation for mechanical and chemical equilibrium. Using the relations from Table 1 the mechanical equilibrium can be rewritten as

$$\begin{aligned} \Delta p &= \rho' R [1 + \delta' \alpha_\delta^r(\delta', \tau)] \\ &- \rho'' R [1 + \delta'' \alpha_\delta^r(\delta'', \tau)] = 0 \end{aligned}$$

and the chemical equilibrium as

$$\begin{aligned} \Delta g &= TR [1 + \alpha^0(\delta', \tau) + \alpha^r(\delta', \tau) + \delta' \alpha_\delta^r(\delta', \tau)] \\ &- TR [1 + \alpha^0(\delta'', \tau) + \alpha^r(\delta'', \tau) + \delta'' \alpha_\delta^r(\delta'', \tau)] = 0 \end{aligned}$$

resulting in two equations with ρ' and ρ'' as two unknowns. These two equations can be simplified by canceling out the constant and purely temperature-dependent parts and then be solved simultaneously using a Newton-Raphson algorithm as described in [1]. A simplified flowchart for this algorithm is shown in Figure 1. The actual implementation uses dimensionless, scaled variables and gradients.

Once the VLE and the respective saturation states are known, all state properties can be calculated using the vapour mass fraction x . It is defined as

$$x = \frac{m''}{m' + m''} = \frac{\text{mass of vapour}}{\text{mass of liquid} + \text{mass of vapour}}$$

Using $m = m' + m''$ and $v = V/m$ this can be re-written as

$$x = \frac{v - v'}{v'' - v'} = \frac{1/\rho - 1/\rho'}{1/\rho'' - 1/\rho'}$$

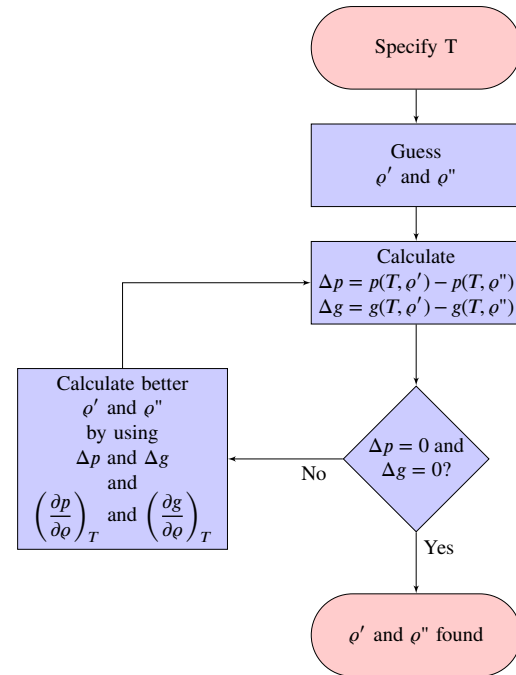


Figure 1: Simplified flowchart for finding the vapour-liquid-equilibrium iteratively, adapted from [1]

Solving for v yields

$$v = xv'' + (1 - x)v' = v' + x(v'' - v')$$

All other state properties can be calculated in the same manner.

In order to calculate the partial derivatives of state properties within the two-phase region, the derivatives along the saturation line are needed. The derivatives of saturation pressure and temperature along the saturation line are given by the Clausius-Clapeyron equation:

$$\begin{aligned} \left(\frac{dp_\sigma}{dT}\right) &= \frac{s'' - s'}{v'' - v'} = \frac{1}{T} \frac{h'' - h'}{v'' - v'} \\ \left(\frac{dT_\sigma}{dp}\right) &= \frac{v'' - v'}{s'' - s'} = T \frac{v'' - v'}{h'' - h'} \end{aligned}$$

These derivatives can then be used to calculate arbitrary derivatives along the saturation line, and, in a second step, partial derivatives within the two-phase state [10].

4 Iterative procedures

So far, it was assumed that temperature T and density d are known. But the thermodynamic state can as well be defined by specifying any other combination of two independent state variables. In engineering applications,

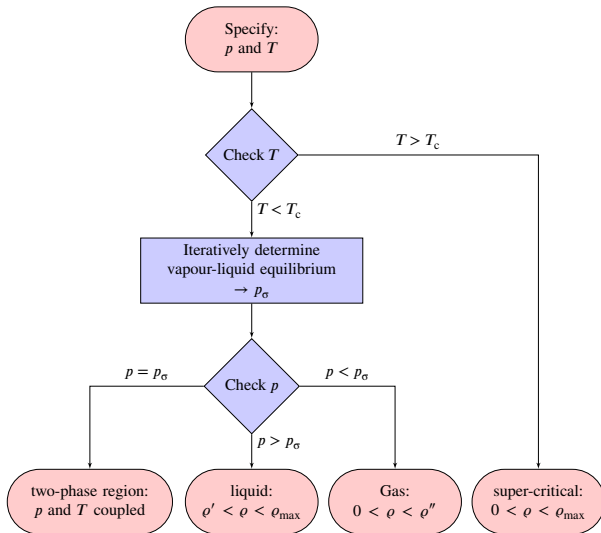


Figure 2: Simplified flowchart for determination of density iteration bounds when pressure and temperature are specified

the known variable combinations often are (p, T) , (p, h) or (p, s) . When any of these combinations is given, the corresponding (T, d) have to be determined iteratively. Two examples of such iterative procedures are given below.

4.1 Density as a function of temperature and pressure

By specifying pressure and temperature, only single-phase states can be described, because in the two-phase region pressure and temperature are not independent. In order to find the density corresponding to the given pressure in the single-phase region, a residual function is defined as

$$RES(\rho) = p - p_{\text{calc}}(\rho, T) \quad .$$

As $p = p(\rho)$ is strictly monotonic for a given temperature, the residual function is as well strictly monotonic and has one single root. Finding the root of the residual function is then equal to finding the density corresponding to the specified pressure. In literature many algorithms for root finding are known, this library uses the algorithm by Brent [3]. It is implemented in the Modelica Standard Library as `Modelica.Math.Nonlinear.solveOneNonlinearEquation`. The mandatory input for this algorithm is a residual function and a lower and upper bound. A flowchart for finding the upper and lower bounds of density is shown in Figure 2.

Once the density is known, all state properties can be calculated using the relations given in Table 1 with

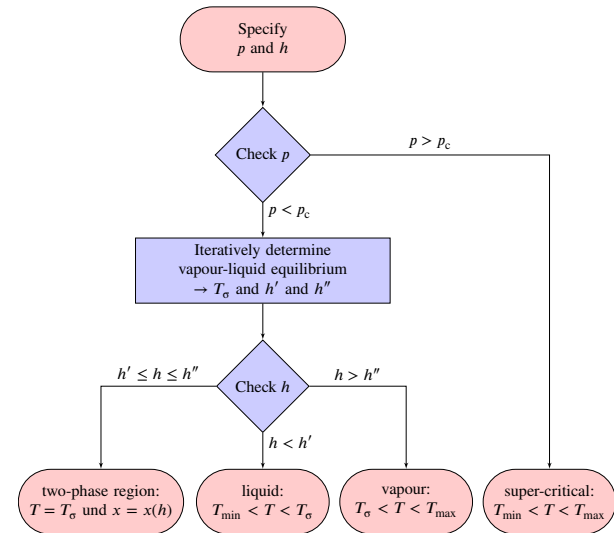


Figure 3: Simplified flowchart for determination of temperature iteration bounds when pressure and enthalpy are specified

density and temperature as input. In the following section, enthalpy and entropy are needed as a function of pressure and temperature. These are calculated by first calculating the density iteratively and then calculating enthalpy and entropy using temperature and density as input variables.

4.2 Density and temperature as a function of pressure and enthalpy

By specifying pressure and specific enthalpy, it is possible to describe single-phase as well as two-phase states. If the pressure is below critical pressure, the first step thus is to determine the vapour-liquid equilibrium corresponding to the specified pressure. The algorithm for VLE determination as described in section 3 uses temperature as input. When the VLE is to be determined from a specified pressure, the residual function

$$RES(T) = p - p_{\sigma, \text{calc}}(T) \quad .$$

is used. The lower bound and upper bound for the temperature are the triple temperature and the critical temperature. The VLE information is then used to determine the region and temperature iteration bounds as shown in Figure 3.

Density and temperature can then be determined using the Brent algorithm and the residual function

$$RES(T) = h - h_{\text{calc}}(p, T) \quad ,$$

where $h_{\text{calc}}(p, T)$ already is an iterative function, as described earlier.

5 Ancillary equations

For the determination of the region during the iterative procedures the vapour pressure and the saturation states have to be evaluated. In order to minimize the computational effort, three ancillary equations are given that are sufficiently precise for a first region check. Only if the thermodynamic state is very close to or within the two-phase region the VLE has to be determined from the EoS for best consistency.

Additionally, the results from the ancillary equations are used as start values for the iterative determination of the VLE from the EoS.

5.1 Vapour pressure

The vapour pressure increases sharply with increasing temperature, as shown in Figure 4. The HelmholtzMedia library uses the vapour pressure equation suggested by [12]:

$$\ln \left(\frac{p_\sigma}{p_c} \right) = \frac{T_c}{T} \cdot \sum a_i \left(1 - \frac{T}{T_c} \right)^{n_i} .$$

This vapour pressure equation can be solved for temperature numerically only.

5.2 Density of saturated liquid and saturated vapour

Six models are implemented for the saturated density. These are similar to the models implemented in RefProp [5]. As before the reduced density δ and the reduced inverse temperature τ are defined as

$$\delta = \frac{\rho}{\rho_c} \quad \tau = \frac{T_c}{T} .$$

The reduced density δ at saturation is calculated in a two-step procedure:

$$\Theta = \begin{cases} \left(1 - \frac{T}{T_c} \right) & \text{model 1,3 or 5} \\ \left(1 - \frac{T}{T_c} \right)^{1/3} & \text{model 2,4 or 6} \end{cases}$$

and

$$\delta = \begin{cases} 1 + \sum a_i \Theta^{n_i} & \text{model 1 or 2} \\ \exp \left(\sum a_i \Theta^{n_i} \right) & \text{model 3 or 4} \\ \exp \left(\tau \sum a_i \Theta^{n_i} \right) & \text{model 5 or 6.} \end{cases}$$

Multiplying the reduced density δ with the critical density ρ_c then yields the density ρ .

6 Further properties

6.1 Surface Tension

The surface tension σ between liquid and vapour phase decreases with saturation temperature approaching the critical temperature. This is modeled according to [6]:

$$\sigma = \sum a_i \left(\frac{T_c - T_\sigma}{T_c} \right)^{n_i} .$$

6.2 Viscosity

In this library two viscosity models are implemented that are similar to the models implemented in RefProp [5]. In both models, the viscosity is split into three contributions: the dilute gas viscosity η_0 , the initial density viscosity η_1 and the residual viscosity η_r . This allows for an individual model for each contribution.

$$\eta = \eta_0(T) + \eta_1(\rho, T) + \eta_r(\rho, T) .$$

6.3 Thermal conductivity

One thermal conductivity model has been implemented that is similar to the model implemented in RefProp [5]. The thermal conductivity is split into three contributions: the dilute gas thermal conductivity λ_0 , the residual thermal conductivity λ_r and the critical enhancement contribution λ_c . Each contribution is then individually modeled.

$$\lambda = \lambda_0(T) + \lambda_r(\rho, T) + \lambda_c(\rho, T) .$$

7 Modelica implementation

This library is compatible to and based on Modelica.Media [4]. HelmholtzMedia defines a partial package PartialHelmholtzMedium which extends from Modelica.Media.Interfaces.PartialTwoPhaseMedium. All functions available in the base class are either inherited without modification or they are modified by implementing a new algorithm.

The Record ThermodynamicState contains density, temperature, pressure, specific enthalpy, specific internal energy and specific entropy. Compared to the base class, specific entropy was added. The Record SaturationProperties was modified by adding the states liq and vap.

Where possible, annotation(inverse=...); and annotation(derivative=...); were used.

For fluids that can be modeled by the implemented algorithms, adding a new fluid is done by extending

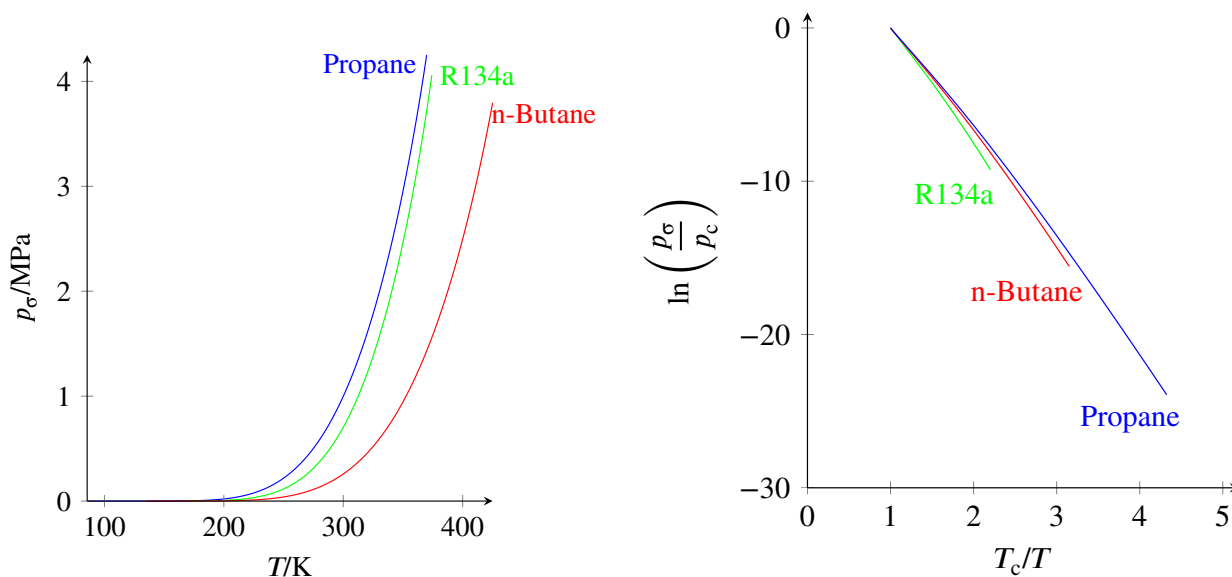


Figure 4: Vapour pressure as a function of temperature

from `PartialHelmholtzMedium` and modifying the parameters for the algorithms. The parameters need to be copied from the respective publications and saved in the format used by `HelmholtzMedia`. `RefProp` [5] comes with a comprehensive compilation of these parameters, so that `RefProp` licencees could alternatively copy them from the `RefProp` fluid files. So far, six fluids have been implemented: n-Butane, Isobutane, Isopentane Propane, R134a and Ethanol. The parameters for these six fluids have been copied from `RefProp`.

8 Summary and Outlook

The most accurate equations of state (EoS) available today for a variety of working fluids are fundamental EoS in terms of Helmholtz energy. The *HelmholtzMedia* library implements the Helmholtz energy EoS in a generalized form that makes adding more fluids very easy. In addition to the equation of state, algorithms for the calculation of viscosity, thermal conductivity and surface tension are given, as well as ancillary equations for saturation properties that speed up iterative procedures. Apart from these ancillary equations, the library is not optimized for speed.

Possible extensions for future versions include the addition of non-analytic terms for the residual part of the Helmholtz energy and hyperbolic terms for the ideal part of the Helmholtz energy. For viscosity and thermal conductivity two more models could be added, an extended corresponding states model and a model based on the generalized friction theory.

In order to add accurate EoS for mixtures like the

GERG-2008 model, a template for multi-component multi-phase media would be necessary. The structure of `Modelica.Media` might change in a future version of the `Modelica Standard Library` [11].

The library is completely written in `Modelica` and released as open-source under the terms of the `Modelica` license. Anybody interested in the library is invited to contribute; the source code and an issue tracker are available at [9].

Acknowledgment

The authors would like to thank Eric W. Lemmon for answering an abundance of questions and for providing a modified version of `RefProp` capable of outputting additional intermediate results.

This work was performed in the framework of the `GeoEn` project and was funded by the Federal Ministry of Education and Research of Germany (Grant 03G0767A).

References

- [1] R. Akasaka. “A Reliable and Useful Method to Determine the Saturation State from Helmholtz Energy Equations of State”. In: *Journal of Thermal Science and Technology* 3.3 (2008), pp. 442–451. DOI: 10.1299/jtst.3.442.

- [2] H. D. Baehr. “Thermodynamische Fundamentalgleichungen und charakteristische Funktionen”. In: *Forschung im Ingenieurwesen* 64.1 (1998), pp. 35–43. DOI: 10.1007/PL00010764.
- [3] R. Brent. *Algorithms for minimization without derivatives*. Prentice-Hall, 1973.
- [4] H. Elmqvist, H. Tummescheit, and M. Otter. “Object-oriented modeling of thermo-fluid systems”. In: *Proceedings of the 3rd International Modelica Conference*. 2003, pp. 269–286.
- [5] E. W. Lemmon, M. L. Huber, and M. O. McLinden. *NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties - REFPROP*. 9.0. National Institute of Standards and Technology, Standard Reference Data Program. Gaithersburg, 2010.
- [6] G. R. Somayajulu. “A generalized equation for surface tension from the triple point to the critical point”. In: *International Journal of Thermophysics* 9.4 (1988), pp. 559–566. DOI: 10.1007/BF00503154.
- [7] R. Span. *Multiparameter equations of state: an accurate source of thermodynamic property data*. Springer Verlag, 2000.
- [8] R. Span, W. Wagner, E. W. Lemmon, and R. T. Jacobsen. “Multiparameter equations of state — recent trends and future challenges”. In: *Fluid Phase Equilibria* 183-184.1-2 (2001), pp. 1–20. DOI: 10.1016/S0378-3812(01)00416-2.
- [9] M. Thorade. *HelmholtzMedia*. 2012. URL: <https://github.com/thorade/HelmholtzMedia/>.
- [10] M. Thorade and A. Saadat. “Partial derivatives of thermodynamic state properties for dynamic simulation”. In: *will be submitted to: Environmental Earth Sciences* (2012).
- [11] H. Tummescheit. *Ticket 85: Re-design and simplification of Modelica.Media*. 2008. URL: <https://trac.modelica.org/Modelica/ticket/85>.
- [12] W. Wagner. *Eine mathematisch statistische Methode zum Aufstellen thermodynamischer Gleichungen — gezeigt am Beispiel der Dampfdruckkurve reiner fluider Stoffe*. Vol. 3. Fortschrittberichte der VDI Zeitschriften 39. VDI Verlag, 1974.
- [13] W. Wagner and A. Pruß. “The IAPWS Formulation 1995 for the Thermodynamic Properties of Ordinary Water Substance for General and Scientific Use”. In: *Journal of Physical and Chemical Reference Data* 31.2 (2002), pp. 387–535. DOI: 10.1063/1.1461829.

