Memo

Implementing MBWR and SPUNG equations of state in ThermoPack

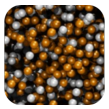
Thermotools
Location:
Trondheim
NORWAY

https://thermotools.github.io/thermopack/index.html

AUTHOR
Ailo Aasen

DATE
2024-12-05

Contents



1 Introduction

This memo documents the theory behind and ThermoPack-implementation of the MBWR-19 and the MBWR-32 equations of state for pure fluids, as well as their extension to mixtures via the SPUNG equation of state. ThermoPack is a SINTEF Energy Research in-house thermodynamic library, and is documented in Skaugen et al. [9].

MBWR-19 and MBWR-32 are single-component equations of state, having respectively 19 and 32 parameters fitted to a specific substance. They are both examples of so-called *multiparameter equations of state*, and generally outperform cubic equations of state when it comes to accuracy, but lose to them in speed. MBWR-19 was first used by Bender (1970), and is also referred to as the Bender equation. MBWR-32 is due to Jacobsen and Stewart (1973), and is sometimes called the Jacobsen-Stewart equation, or simply *the MBWR equation*. The MBWR-19 and MBWR-32 have been fitted to a range of components.

MBWR equations are often used as reference equations for so-called *Corresponding States* equations. The previously implemented Lee-Kesler equation of state, documented in Aarnes [1], is one example. Another Corresponding States equation is the SPUNG equation. SPUNG stands for *State Research and Development Program for Utilization of Natural Gas*, after the Norwegian state program which partly financed the equation's development, and is documented in Jørstad [2].

1.1 How to implement equations of state in ThermoPack

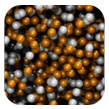
ThermoPack is a thermodynamics library which is documented in Skaugen et al. [9]. The library has the convention of using $(T, P, \mathbf{n})$ as independent variables. For an equation of state to be considered implemented in ThermoPack, the following is required: For each of the thermodynamic functions

- the compressibility z
- the residual entropy S^R
- the residual enthalpy H^R
- the logarithmic fugacity coefficients $\ln \phi_i$

there should be routines which takes in temperature T, pressure P, composition $\mathbf{n}$ and phase as input, and returns the values of these functions, as well as the values of their first order partial derivatives when the independent variables are $(T, P, \mathbf{n})$.

In addition, a routine for calculating the residual Gibbs energy G^r and its partial derivatives is usually desired.

When it comes to units, ThermoPack mostly uses base SI-units, e.g K for temperature and Pa for pressure. An important exception is molar volume and density, which are measured in $\mathrm{dm}^3/\mathrm{mol}$ and $\mathrm{mol}/\mathrm{dm}^3$.



2 Expressing thermodynamic functions using the reduced Helmholtz energy

2.1 Residual properties

An arbitrary temperature-volume-composition state $(T, V, \mathbf{n})$ can be reached by mixing the pure fluids at temperature T_0 and approximately zero density and pressure, heating the mixture to the temperature T and then compressing it to the volume V. It follows that the calculation of an arbitrary thermodynamic property $M(T, V, \mathbf{n})$ can be split up according to

$$M = (\Delta M^\star)_{\text{mixing}} + \int_{T_0}^{T} \left(\frac{\partial M^\star}{\partial T}\right)_{V=\infty,\mathbf{n}} dT + \int_{\infty}^{V} \left(\frac{\partial M}{\partial V}\right)_{T,\mathbf{n}} dV \tag{2.1}$$

where $M^\star$ denotes the property at zero density and pressure, and $(\Delta M^\star)_{\text{mixing}}$ denotes the property due to mixing at zero density and pressure.

If instead $(T, P, \mathbf{n})$ are used as independent variables, the equivalent of (2.1) is

$$M = \Delta M^\star + \int_{T_0}^{T} \left(\frac{\partial M^\star}{\partial T}\right)_{P=0,\mathbf{n}} dT + \int_{0}^{P} \left(\frac{\partial M}{\partial P}\right)_{T,\mathbf{n}} dP. \tag{2.2}$$

The right-hand sides of (2.1) and (2.2) can be rearranged to comprise two terms, one which is the property $M^\star$ of the hypothetical perfect gas at the state $(T, V, \mathbf{n})$ or $(T, P, \mathbf{n})$ over some fixed, chosen zero, and a second term $M^R = M - M^\star$, called the residual. From its definition $M^R = M - M^\star$, we see that the residual M^R must be defined as

$$\begin{aligned} M^R(T, V, \mathbf{n}) &= \int_{\infty}^{V} \left[\left(\frac{\partial M}{\partial V_1}\right)_{T,\mathbf{n}} - \left(\frac{\partial M^\star}{\partial V_1}\right)_{T,\mathbf{n}}\right] dV_1, \\ M^R(T, P, \mathbf{n}) &= \int_{0}^{P} \left[\left(\frac{\partial M}{\partial P_1}\right)_{T,\mathbf{n}} - \left(\frac{\partial M^\star}{\partial P_1}\right)_{T,\mathbf{n}}\right] dP_1. \end{aligned} \tag{2.3}$$

Beware that in general we have $M^R(T, V, \mathbf{n}) \neq M^R(T, P, \mathbf{n})$. This may seem strange since the value of a thermodynamic property shouldn't depend on which coordinates one happens to use; for example it is of course always true that $M(T, P, \mathbf{n}) = M(T, V, \mathbf{n})$. The reason this equality does not hold for residual properties is that having temperature, volume and composition $(T, V, \mathbf{n})$ as a perfect gas state, is not the same the same perfect gas state as having temperature, pressure and composition $(T, P, \mathbf{n})$, because in general $PV \neq nRT$, and thus $M^\star(T, V, \mathbf{n}) \neq M^\star(T, P, \mathbf{n})$

2.2 Relating thermodynamic functions functions to F

Let

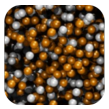
$$A = -PV + \sum \mu_i n_i$$

denote the Helmholtz energy of an arbitrary fluid. We have $dA = -SdT - PdV + \sum_i \mu_i dn_i$, and the natural variables for A are $T, V, \mathbf{n}$, all of which are accessible variables (in contrast to e.g. entropy). Let as above A^R denote the residual Helmholtz energy. Then from (2.3) we see that

$$A^R(T, V, \mathbf{n}) = \int_{\infty}^{V} \left(-P(T, V_1, \mathbf{n}) + nRT/V_1\right) dV_1. \tag{2.4}$$

A quantity of prime importance is the *reduced* residual Helmholtz energy, defined as

$$F(T, V, \mathbf{n}) = \frac{A^R(T, V, \mathbf{n})}{RT} \tag{2.5}$$



In modern thermodynamic engineering, equations of state are usually formulated as an expression for F as a function of temperature and volume. Note that F is written as a function of $(T, V, \mathbf{n})$, and indeed explicit functional expressions for F are usually only obtainable in these variables. Thus if one wants to compute F from knowledge only of $(T, P, \mathbf{n})$, one first has to compute V. For all but the simplest equations of state (like e.g. cubic equations), this has to be done using an iterative solver, and can be a time-consuming part of the program. See also section 5.

When we are dealing with pure fluids, which after all is the setting of the MBWR equations, we have $\mathbf{n} = n$, where n is the total number of moles of the pure fluid, and it is more convenient to consider extensive properties such as F on a per mole (molar) basis. In this setting it is natural to use density $\rho = n/V$ as a variable (or alternatively, molar volume $v = 1/\rho$). Indeed, the MBWR equations are usually formulated as equations for the pressure $P = P(T, \rho)$. They are easily converted – using (2.5) – to an equation for the molar Helmholtz energy α as a function of (T, ρ), where α is defined as:

$$\alpha(T, \rho) = \frac{A^R(T, n/\rho, n)}{nRT} \tag{2.6}$$

$$= \frac{F(T, n/\rho, n)}{n}, \tag{2.7}$$

where the functions on the right hand sides are the ones in equation (2.5). Note that F has units mol, while α is adimensional.

If one has an equation for F, it turns out that all thermodynamic properties can be computed from F and its partial derivatives. Below, the thermodynamic functions needed in ThermoPack, together with their first order partials, are given in terms of F and its first and second order partial derivatives.

Compressibility factor

The compressibility factor z is a dimensionless number which can be written in several equivalent forms:

$$z = \frac{PV}{nRT} = \frac{Pv}{RT} = \frac{P}{\rho RT} = \frac{v}{v_{ig}} \tag{2.8}$$

In the equation (2.8) $v = V/n$ is the specific molar volume, $\rho = 1/v$ is the molar density, v_{ig} is the specific volume of an ideal gas, R is the universal gas constant, and P, V, T and n are pressure, volume, temperature and total number of moles of the fluid, respectively. For ideal gases $z = 1$ for all T and P and n, whereas for real gases z is a non-constant function of T and P and n.

Residual entropy

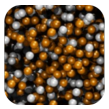
$$S^R(T, P, n) = nR \ln z - RF - RT \left(\frac{\partial F}{\partial T} \right)_{V,n}. \tag{2.9}$$

Residual enthalpy

$$H^R(T, P, n) = -RT^2 \left(\frac{\partial F}{\partial T} \right)_{V,n} + PV - nRT, \tag{2.10}$$

Logarithmic fugacity coefficient

$$\ln \phi_i(T, P, n) = \left(\frac{\partial F}{\partial n_i} \right)_{T,V} - \ln z. \tag{2.11}$$



2.3 Partial derivatives of the thermodynamic properties

The following formulas are mostly taken directly from Aarnes [1].

Pressure

To simplify the notation in the following derivatives, a relation between the pressure and its derivatives, and the reduced residual Helmholtz function is introduced.

$$P(T,V,\mathbf{n}) = -RT\left(\frac{\partial F}{\partial V}\right)_{T,\mathbf{n}} + \frac{nRT}{V} \tag{2.12}$$

The partial derivatives of the pressure, with respect to temperature, volume and composition, respectively, are given by:

$$\left(\frac{\partial P}{\partial T}\right)_{V,\mathbf{n}} = \frac{P}{T} - RT\left(\frac{\partial^2 F}{\partial T \partial V}\right)_{n_i} \tag{2.13}$$

$$\left(\frac{\partial P}{\partial V}\right)_{T,\mathbf{n}} = -RT\left(\frac{\partial^2 F}{\partial V^2}\right)_{T,\mathbf{n}} - \frac{nRT}{V^2} \tag{2.14}$$

$$\left(\frac{\partial P}{\partial n_i}\right)_{T,V} = -RT\left(\frac{\partial^2 F}{\partial n_i \partial V}\right)_{T} + \frac{RT}{V} \tag{2.15}$$

Furthermore, the partial derivatives of the volume with respect to composition and temperature are defined, by the use of the triple product rule and the derivatives of the pressure:

$$\bar{V}_i \equiv \left(\frac{\partial V}{\partial n_i}\right)_{T,P} = -\frac{\left(\frac{\partial P}{\partial n_i}\right)_{T,V}}{\left(\frac{\partial P}{\partial V}\right)_{T,\mathbf{n}}} \tag{2.16}$$

$$\bar{V}_T \equiv \left(\frac{\partial V}{\partial T}\right)_{P,\mathbf{n}} = -\frac{\left(\frac{\partial P}{\partial T}\right)_{V,\mathbf{n}}}{\left(\frac{\partial P}{\partial V}\right)_{T,\mathbf{n}}} \tag{2.17}$$

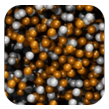
The following derivatives are all carried out for functions that are $(T,P,\mathbf{n})$-states, that is $z = z(T,P,\mathbf{n})$, $S^R = S^R(T,P,\mathbf{n})$, $H^R = H^R(T,P,\mathbf{n})$ and $\ln\phi_i = \ln\phi_i(T,P,\mathbf{n})$. Several of these calculations get rather involved, and only the resulting expressions are presented here.

Compressibility

$$\left(\frac{\partial z}{\partial T}\right)_{P,\mathbf{n}} = -z\left[\frac{1}{T} - \frac{\bar{V}_T}{V}\right] \tag{2.18}$$

$$\left(\frac{\partial z}{\partial P}\right)_{T,\mathbf{n}} = z\left[\frac{1}{P} + \frac{1}{V\left(\frac{\partial P}{\partial V}\right)_{T,\mathbf{n}}}\right] \tag{2.19}$$

$$\left(\frac{\partial z}{\partial n_i}\right)_{T,P} = -z\left[\frac{1}{n} - \frac{\bar{V}_i}{V}\right] \tag{2.20}$$



Entropy

$$\left(\frac{\partial S^R(T,P,\mathbf{n})}{\partial T}\right)_{P,\mathbf{n}} = \bar{V}_T\left(\frac{\partial P}{\partial T}\right)_{V,\mathbf{n}} - R\left[2\left(\frac{\partial F}{\partial T}\right)_{V,\mathbf{n}} + T\left(\frac{\partial^2 F}{\partial T^2}\right)_{V,\mathbf{n}} + \frac{n}{T}\right] \tag{2.21}$$

$$\left(\frac{\partial S^R(T,P,\mathbf{n})}{\partial P}\right)_{T,\mathbf{n}} = \frac{nR}{P} - \bar{V}_T \tag{2.22}$$

$$\left(\frac{\partial S^R(T,P,\mathbf{n})}{\partial n_i}\right)_{T,P} = \bar{V}_i\left(\frac{\partial P}{\partial T}\right)_{V,\mathbf{n}} - R\left[\left(\frac{\partial F}{\partial n_i}\right)_{T,V} + T\left(\frac{\partial^2 F}{\partial T \partial n_i}\right)_{V} + 1 - \ln z\right] \tag{2.23}$$

Enthalpy

$$\left(\frac{\partial H^R}{\partial T}\right)_{P,\mathbf{n}} = \bar{V}_T T\left(\frac{\partial P}{\partial T}\right)_{V,\mathbf{n}} - RT\left[2\left(\frac{\partial F}{\partial T}\right)_{V,\mathbf{n}} + T\left(\frac{\partial^2 F}{\partial T^2}\right)_{V,\mathbf{n}} + \frac{n}{T}\right] \tag{2.24}$$

$$\left(\frac{\partial H^R}{\partial P}\right)_{T,\mathbf{n}} = V - T\bar{V}_T \tag{2.25}$$

$$\left(\frac{\partial H^R}{\partial n_i}\right)_{T,P} = \bar{V}_i T\left(\frac{\partial P}{\partial T}\right)_{V,\mathbf{n}} - RT^2\left(\frac{\partial^2 F}{\partial T \partial n_i}\right)_{V} - RT \tag{2.26}$$

Fugacity coefficients

$$\left(\frac{\partial \ln \phi_i}{\partial T}\right)_{P,\mathbf{n}} = \left(\frac{\partial^2 F}{\partial T \partial n_i}\right)_{V} + \frac{1}{T} - \frac{\bar{V}_i}{RT}\left(\frac{\partial P}{\partial T}\right)_{V,\mathbf{n}} \tag{2.27}$$

$$\left(\frac{\partial \ln \phi_i}{\partial P}\right)_{T,\mathbf{n}} = \frac{\bar{V}_i}{RT} - \frac{1}{P} \tag{2.28}$$

$$\left(\frac{\partial \ln \phi_i}{\partial n_j}\right)_{T,P} = \left(\frac{\partial^2 F}{\partial n_j \partial n_i}\right)_{T,V} + \frac{1}{n} + \frac{\left(\frac{\partial P}{\partial V}\right)_{T,\mathbf{n}}}{RT}\bar{V}_j\bar{V}_i \tag{2.29}$$

Gibbs energy

Residual Gibbs energy is not documented in Aarnes [1] and is therefore documented here. From Michelsen [3] we have

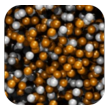
$$G^R(T,P,\mathbf{n}) = RTF(T,V,\mathbf{n}) + PV - nRT(1+\ln z),$$

whence

$$\left(\frac{\partial G^R}{\partial T}\right)_{P,\mathbf{n}} = R\left(F + T\left(\frac{\partial F}{\partial T}\right)_{V,\mathbf{n}} - n\ln z\right) + \left(P - P/z + RT\left(\frac{\partial F}{\partial V}\right)_{T,\mathbf{n}}\right)\bar{V}_T \tag{2.30}$$

$$\left(\frac{\partial G^R}{\partial P}\right)_{T,\mathbf{n}} = V - \frac{nRT}{P} \tag{2.31}$$

$$\left(\frac{\partial G^R}{\partial n_i}\right)_{T,P} = RT\left(\left(\frac{\partial F}{\partial n_i}\right)_{T,P} - \ln z\right) \tag{2.32}$$



Internal energy

Residual internal energy is not documented in Aarnes [1] and is therefore documented here. We have

$$U^R(T,P,\mathbf{n}) = nRT\ln z - RT^2\left(\frac{\partial F}{\partial T}\right)_{V,n}$$

and thus

$$\left(\frac{\partial U^R}{\partial T}\right)_{P,\mathbf{n}} = nR\left[\ln z + \frac{T\bar{V}_T}{V} - 1\right] - RT\left(2\left(\frac{\partial F}{\partial T}\right)_{V,\mathbf{n}} + T\left(\frac{\partial^2 F}{\partial T^2}\right)_{V,\mathbf{n}}\right) \tag{2.33}$$

$$\left(\frac{\partial U^R}{\partial P}\right)_{T,\mathbf{n}} = nTR\left[\frac{1}{P} + \frac{1}{V\left(\frac{\partial P}{\partial V}\right)_{T,\mathbf{n}}}\right] - \frac{RT^2\left(\frac{\partial^2 F}{\partial T\partial V}\right)_{\mathbf{n}}}{\left(\frac{\partial P}{\partial V}\right)_{T,\mathbf{n}}} \tag{2.34}$$

$$\left(\frac{\partial U^R}{\partial n_i}\right)_{T,P} = TR\left[\ln z + \frac{n\bar{V}_i}{V} - 1\right] - RT^2\left(\frac{\partial^2 F}{\partial T\partial n_i}\right)_{V} \tag{2.35}$$

2.4 Relating F-derivatives to α-derivatives

In the case of pure fluids, it is as mentioned more convenient to use the function α defined by

$$F(T,V,n) = n\alpha(T,\rho). \tag{2.36}$$

This implies the following relations between the derivatives of F and the derivatives of α.

T-derivatives

$$\left(\frac{\partial F}{\partial T}\right)_{V,n} = n\left(\frac{\partial \alpha}{\partial T}\right) \tag{2.37}$$

$$\left(\frac{\partial^2 F}{\partial T^2}\right)_{V,n} = n\left(\frac{\partial^2 \alpha}{\partial T^2}\right) \tag{2.38}$$

V-derivatives

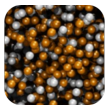
$$\left(\frac{\partial F}{\partial V}\right)_{T,n} = -\frac{n^2}{V^2}\left(\frac{\partial \alpha}{\partial \rho}\right) \tag{2.39}$$

$$\left(\frac{\partial^2 F}{\partial V^2}\right)_{T,n} = \frac{2n^2}{V^3}\left(\frac{\partial \alpha}{\partial \rho}\right) + \frac{n^3}{V^4}\left(\frac{\partial^2 \alpha}{\partial \rho^2}\right) \tag{2.40}$$

n-derivatives

$$\left(\frac{\partial F}{\partial n}\right)_{V,n} = \alpha + \frac{n}{V}\left(\frac{\partial \alpha}{\partial \rho}\right) \tag{2.41}$$

$$\left(\frac{\partial^2 F}{\partial n^2}\right)_{V,n} = \frac{2}{V}\left(\frac{\partial \alpha}{\partial \rho}\right) + \frac{n}{V^2}\left(\frac{\partial^2 \alpha}{\partial \rho^2}\right) \tag{2.42}$$



Cross-derivatives

$$\left(\frac{\partial^2 F}{\partial T \partial V}\right)_V = -\frac{n^2}{V^2}\left(\frac{\partial^2 \alpha}{\partial T \partial \rho}\right) \tag{2.43}$$

$$\left(\frac{\partial^2 F}{\partial T \partial n}\right)_{V,n} = \left(\frac{\partial \alpha}{\partial T}\right) + \frac{n}{V}\left(\frac{\partial^2 \alpha}{\partial T \partial \rho}\right) \tag{2.44}$$

$$\left(\frac{\partial^2 F}{\partial V \partial n}\right)_{V,n} = -\frac{2n}{V^2}\left(\frac{\partial \alpha}{\partial \rho}\right) - \frac{n^2}{V^3}\left(\frac{\partial^2 \alpha}{\partial \rho^2}\right) \tag{2.45}$$

3 The MBWR equations of state

Both the MBWR-19 and the MBWR-32 equations of state take the general form

$$P(T,\rho) = \rho RT + \sum_{k=1}^{I_{\text{pol}}} a_k T^{t_k} \rho^{d_k} + \sum_{k=I_{\text{pol}}+1}^{I_{\text{tot}}} a_k T^{t_k} \rho^{d_k} \exp\left(-\gamma \rho^2\right). \tag{3.1}$$

The last sum on the right hand side of (3.1) side is referred to as the exponential part, and the rest is called the polynomial part[1]. The d_k are positive integers and the t_k are rational numbers, both inherent to the equation of state. Their values are given in Table 1 for MBWR-19, and in Table 2 for MBWR-32. The parameters $a_1, \ldots, a_{I_{\text{tot}}}$ are not inherent to the equation of state, but are substance-specific and thus have to be fitted to experimental measurements. For MBWR-19 and MBWR-32 we have I_{tot} equal to 19 and 32, respectively. The parameter γ usually equals $1/\rho_c^2$, but may have been fitted with something else. In any case, even though γ is component dependent, it is not optimized in the fitting process, which is the reason it is not counted as a bona fide parameter. MBWR-19 is however only *our* choice of name, and other sources (e.g. Jørstad [2]) instead calls it MBWR-20.

Note that, for both MBWR-19 and MBWR-32, all the density exponents d_k are positive. This is not coincidental, but a necessity to make it have the desired asymptotic behavior

$$\lim_{\rho \to 0} P(T,\rho) = 0.$$

The MBWR equations are sometimes written with terms grouped according to powers of ρ:

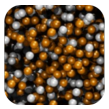
$$P(T,\rho) = \sum_{i=1}^{BP_{\text{len}}} BP_i(T) \cdot \rho^i + \exp(-\gamma\rho^2) \sum_{i=1}^{BE_{\text{len}}} BE_i(T) \cdot \rho^{2i+1}. \tag{3.2}$$

This aggregation of the temperature dependents terms is sometimes convenient, because one often wants to evaluate $P(T,\rho)$ for the same T but several different ρ, the prime example being the iterative procedure in the density solver, see section 5. In this case it is computationally efficient to pre-calculate the coefficients $BP_i(T)$ and $BE_i(T)$.

4 Database for the MBWR substance-specific parameters

The correlated values of a_k for various substances are stored in the file *tpmbwrdata.f90*, and have been retrieved from the old TPlib thermodynamics library. In the current database 9 substances have correlations for MBWR-32, while 54 substances have correlations for MBWR-19.

[1] This is somewhat misleading, since not all the t_k are positive integers.

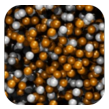


k	d_k	t_k	type	k	d_k	t_k	type
1	2	1	pol	11	5	1	pol
2	2	0	pol	12	5	0	pol
3	2	−1	pol	13	6	0	pol
4	2	−2	pol	14	3	0	exp
5	2	−3	pol	15	3	−1	exp
6	3	1	pol	16	3	−2	exp
7	3	0	pol	17	5	0	exp
8	3	−1	pol	18	5	−1	exp
9	4	1	pol	19	5	−2	exp
10	4	0	pol				

Table 1: Overview of inherent parameters in the MBWR-19 model. $I_{\text{pol}} = 13$, $I_{\text{tot}} = 32$.

k	d_k	t_k	type	k	d_k	t_k	type
1	2	1	pol	17	8	−1	pol
2	2	0.5	pol	18	8	−2	pol
3	2	0	pol	19	9	−2	pol
4	2	−1	pol	20	3	−2	exp
5	2	−2	pol	21	3	−3	exp
6	3	1	pol	22	5	−2	exp
7	3	0	pol	23	5	−4	exp
8	3	−1	pol	24	7	−2	exp
9	3	−2	pol	25	7	−3	exp
10	4	1	pol	26	9	−2	exp
11	4	0	pol	27	9	−4	exp
12	4	−1	pol	28	11	−2	exp
13	5	0	pol	29	11	−3	exp
14	6	−1	pol	30	13	−2	exp
15	6	−2	pol	31	13	−3	exp
16	7	−1	pol	32	13	−4	exp

Table 2: Overview of inherent parameters in the MBWR-32 model. $I_{\text{pol}} = 19$, $I_{\text{tot}} = 32$.



It is important to understand exactly what these coefficients in the database mean. Let us first consider the MBWR-32 equation. For MBWR-32, each correlation in the database consists of 33 coefficients $(a_1, \ldots, a_{32}, \gamma)$, which fit into the equation as follows (see Jørstad [2]):

$$P(T,\rho) = \sum_{i=1}^{19} BP_i \cdot \rho^i + e^{-\gamma\rho^2} \sum_{i=1}^{6} BE_i \cdot \rho^{2i+1}.$$

where

$$\begin{aligned}
BP_1 &= RT \\
BP_2 &= a_1 T + a_2 T^{1/2} + a_3 + a_4/T + a_5/T^2 \\
BP_3 &= a_6 T + a_7 + a_8/T + a_9/T^2 \\
BP_4 &= a_{10} T + a_{11} + a_{12}/T \\
BP_5 &= a_{13} \\
BP_6 &= a_{14}/T + a_{15}/T^2 \\
BP_7 &= a_{16}/T \\
BP_8 &= a_{17}/T + a_{18}/T^2 \\
BP_9 &= a_{19}/T^2 \\
BE_1 &= a_{20}/T^2 + a_{21}/T^3 \\
BE_2 &= a_{22}/T^2 + a_{23}/T^4 \\
BE_3 &= a_{24}/T^2 + a_{25}/T^3 \\
BE_4 &= a_{26}/T^2 + a_{27}/T^4 \\
BE_5 &= a_{28}/T^2 + a_{29}/T^3 \\
BE_6 &= a_{30}/T^2 + a_{31}/T^3 + a_{32}/T^4.
\end{aligned} \tag{4.1}$$

Here P is measured in Pascal, T is measured in Kelvin and ρ is measured in moles per litre – the convention in ThermoPack.

Let us next consider the MBWR-19 equation. In the database each substance has associated with it an array of 20 parameters $(a_1, \ldots, a_{19}, \gamma)$ with the property that (see Polt [5])

$$P(T,\rho) = BP_1\rho + \left[BP_2\rho^2 + BP_3\rho^3 + BP_4\rho^4 + BP_5\rho^5 + BP_6\rho^6 + (BE_1\rho^3 + BE_2\rho^5)e^{-\gamma\rho^2}\right] \cdot 10^3,$$

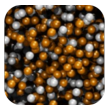
where

$$\begin{aligned}
BP_1 &= RT \\
BP_2 &= a_1 T - a_2 - a_3/T - a_4/T^2 - a_5/T^3 \\
BP_3 &= a_6 T + a_7 + a_8/T \\
BP_4 &= a_9 T + a_{10} \\
BP_5 &= a_{11} T + a_{12} \\
BP_6 &= a_{13} \\
BE_1 &= a_{14}/T^2 + a_{15}/T^3 + a_{16}/T^4 \\
BE_2 &= a_{17}/T^2 + a_{18}/T^3 + a_{19}/T^4.
\end{aligned} \tag{4.2}$$

Take note of the factor 10^3, as well as the sign inversions for terms a_2, a_3, a_4, a_5. Also for MBWR-19, P is measured in Pascal, T is measured in Kelvin and ρ is measured in moles per litre.

No MBWR data set for CO_2

Although there was a data set for CO_2 for MBWR-19 in the old TPlib library, it is wrong. According to this data set, CO_2 is still subcritical at 310 K, which is about 5 K over its measured critical



temperature. VLE calculations using SPUNG with CO_2 as reference component also gives absurd results. Although the dissertation by Polt [5] has been checked, he surprisingly doesn't give a correlation for CO_2. The correlation in the TPlib database has no reference.

5 The MBWR density solver

When the MBWR equations of state are called from ThermoPack, the temperature, pressure and the phase is usually what is given as input. However, the independent variables in MBWR equations are density and temperature, not temperature and pressure. Therefore, an algorithm which takes in T and P and solves for ρ is needed.

The density solver in the old thermodynamics library, TPlib, was not as robust as desired. Jørstad writes the following in his thesis [2]: *Calculation of the density results sometimes in an incorrect solution or in a few circumstances breaks down (...) Approximately 1 out of 500 calculations fails.* A better density solver is therefore required. This section discusses the theory behind the implementation of a new, more robust density solver.

In this section T_0 and P_0 will denote arbitrary but fixed temperatures and pressures.

5.1 Phases and the correct density root

A typical plot of the function $\rho \mapsto P(T_0, \rho)$ is given in Figure 1. As illustrated by this graph, there will from a purely mathematical standpoint be several choices of ρ which satisfies $P(\rho, T_0) = P_0$. However, not all of these densities correspond to a physically realistic state. A first criterion for the density being a physical solution is that belongs to some interval over which $P(T_0, \rho)$ is positive and increasing. This is due to the fact that, at a given temperature, higher pressure will always correspond to a higher density. Applying this first criterion, we see that there are two candidates for the physically correct density. To choose the correct one of these two, we need to know what phase, gas or liquid, the fluid is in: the gas phase corresponds to the lowest density, the liquid phase corresponds to the highest density. Let us call the part of the graph corresponding to valid vapor densities the "vapor hill", and similarly call the part of the graph corresponding to liquid solutions the "liquid hill". Thus, given the phase, one has to choose the solution lying on the corresponding hill.

The choice of root is even easier for supercritical temperatures, $T_0 > T_c$, as illustrated in Figure 2. In this case there is only one interval where $P(T_0, \rho)$ is increasing, and so the density is independent of phase. Physically, this reflects the fact that for supercritical temperatures, there is no discontinuous phase change.

A case where it is not so straightforward how choose the physical density is shown in Figure 3. If the user inputs liquid phase, the question is whether one should return the solution corresponding to the gas phase, or the density corresponding to the liquid phase having minimum pressure.

Sometimes the user may not want to give the algorithm an input phase, but rather ask that it chooses the phase which corresponds to the physically stable solution. This is possible by letting the algorithm search for both the liquid and gas root, and then returning the density corresponding to minimum Gibb's energy.

5.2 The density solver algorithm

Although we tried many algorithms (e.g. the Illonois algorithm, the Pegasus method, Halley's method), in the end the best method proved to be Newton's method. A major reason for its speed is that it is cheap to calculate the derivative $\partial_\rho P(T_0, \rho)$ by precalculating the temperature dependent coefficients, and that if one evaluates both $P(T_0, \rho)$ and its derivative at the same point, much of the computation overlaps, allowing a considerable speedup compared to computing each of them separately. (Of course, the same overlap occurs when computing the second derivative.)

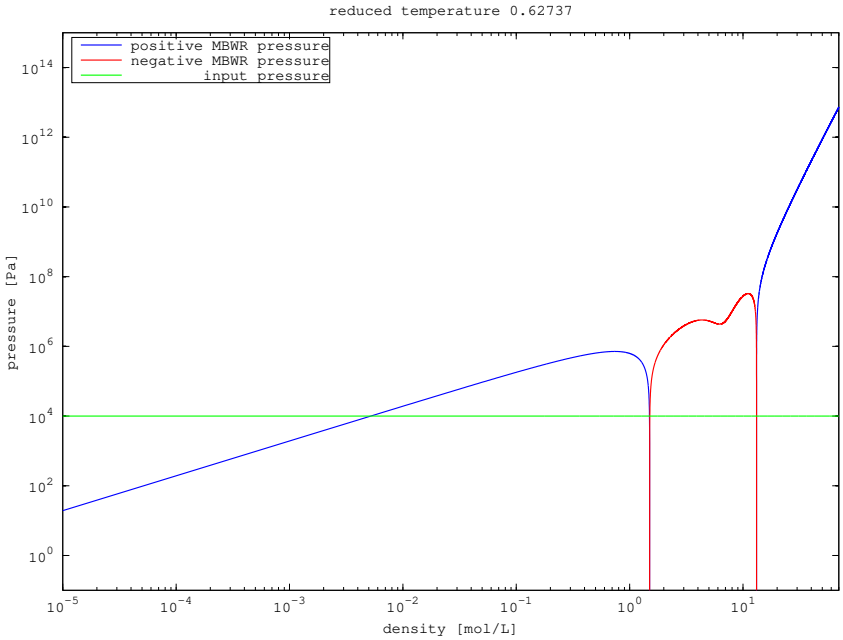


Figure 1: A log-log plot of the MBWR-19 density-pressure curve for CH_4 and a subcritical temperature, and an example input pressure (green) drawn in. Where $P(T_0, \rho)$ becomes negative we have plotted $|P(T_0, \rho)$ in red.

The never-ending worry with Newton's method is that it may shoot off and not converge to the desired root. But there are special cases where Newton's method is guaranteed to converge:

1. If a function is concave and increasing and the initial guess is smaller than the root, then Newton's method is guaranteed to converge, and the rate of convergence is quadratic.

2. If a function is convex and increasing and the initial guess is larger than the root, then Newton's method is guaranteed to converge, and the rate of convergence is quadratic.

This of course requires us to know something about the convexity of $P(T_0, \rho)$. Interestingly, by plotting $\partial^2 P(T_0, \rho)/\partial \rho^2$ and $P(T_0, \rho)$ together, we have found evidence for the "gas hill" being concave and increasing, and the "liquid hill" being convex and increasing[2]. An example is shown in Figure 4. Of course, these convexity properties have not been checked for all substances in the database. It has been verified to hold for the most commonly used components, including. C1, C2, C3, N2, O2, H2O, R152a, R134a, HE. Moreover, for some components (e.g. NC7, benzene), the vapor hill is only concave for pressures below the saturated vapor pressure. If these convexity properties really demonstrate an inherently physical feature of fluids, rather than simply being numerical artefacts, is not known. E.g. Span [8] seems not to mention anything about convexities. In any case, let us from now on call the criterion

$$\partial_\rho^2 P(\rho_{vap}, T_0) < 0, \qquad \partial_\rho^2 P(\rho_{liq}, T_0) > 0 \tag{5.1}$$

the **phase convexity test**.

Now, by 1. above, we know that if one is looking for a gas root, then Newton's method will always converge if given an initial density lower than the true gas density. Similarly, 2. guarantees that Newton's method converges to the liquid density if the starting density is a little higher than

[2] A possible physical interpretation: When density is low, increasing pressure increases density little at first, but then more and more as long-range attraction becomes significant. When density is high, increasing pressure increases density less and less as short-term repulsion becomes dominant.

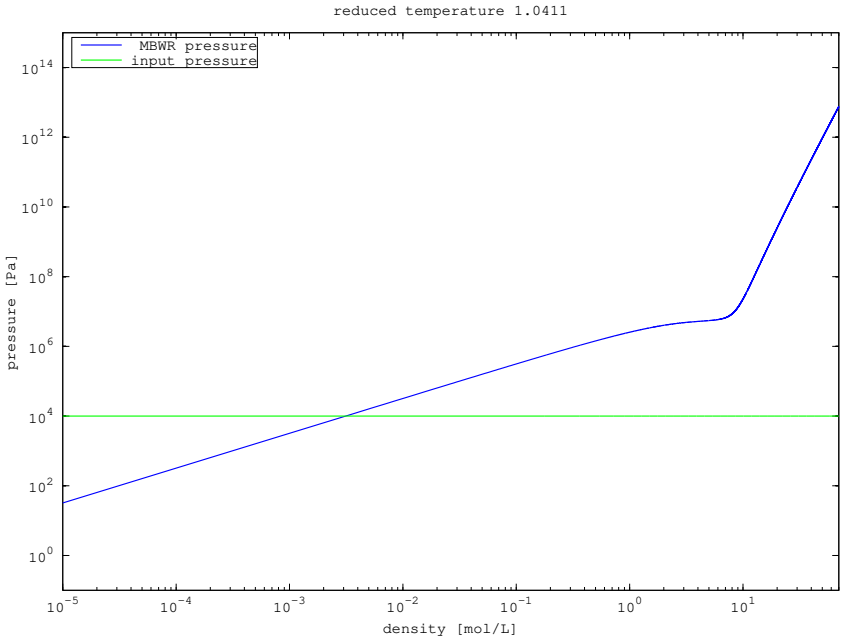


Figure 2: The MBWR-19 density-pressure curve for CH_4 and a supercritical temperature, and an example input pressure drawn in.

the true liquid density. Of course, this is only true if there really exists a density solution in the required phase. If not, Newton's method may shoot off. To prevent this, the density algorithm uses three tests. If the input phase is vapor, they are:

i) $\partial_\rho P(\rho_n) > 0$,

ii) $P(\rho_{n-1}) < P(\rho_n)$,

iii) $\left(P(\rho_n) - P(\rho_{n-1})\right)/(\rho_n - \rho_{n-1}) < \partial_\rho P(\rho_n)$.

If the input phase is liquid, they are the same with the exception of ii, where the direction of the inequality has to be reversed. If the current iterate fails any of these tests, there is no density root with the given input phase. Once again, this assumes that the initial guess is an underestimate in the case of vapor phase, and an overestimate in the case of liquid phase.

If there exists a vapor root, the solver will always find it when given vapor as input phase, simply because it is very easy to find a good underestimate (see section 5.3). Given liquid as input phase, a too large overestimate may result in the density solver diverging. The problem is illustrated in figure 5 for the MBWR-32 equation (this problem does not occur with MBWR-19): if the initial liquid density is too great, we will have negative slope, and Newton's method will diverge. To counter this from happening, a fallback Newton solver is implemented which kicks in if the main density solver fails. The differences between the fallback Newton solver and the main solver are two things: the initial liquid guess is lower, and there are less tests for divergence. In fact, the only situation where the fallback Newton solver terminates (besides from performing more than the maximum number of iterations), is when $\partial_\rho P(\rho_n)$ becomes negative. A second fallback is also implemented as a last resort, namely the old solver in TPlib, which is described in section 5.4.

Finally, some words about the actual code. The density solver is now divided up into three routines. The routine the user actually calls is MBWR_density. MBWR_density generates initial guesses and then calls newton_density, which given an initial density on the correct side of the root, either determines the root, or outputs -1 if no root is found. An optional argument can force the algorithm to choose the metastable extremum if no density exists for the input phase; once

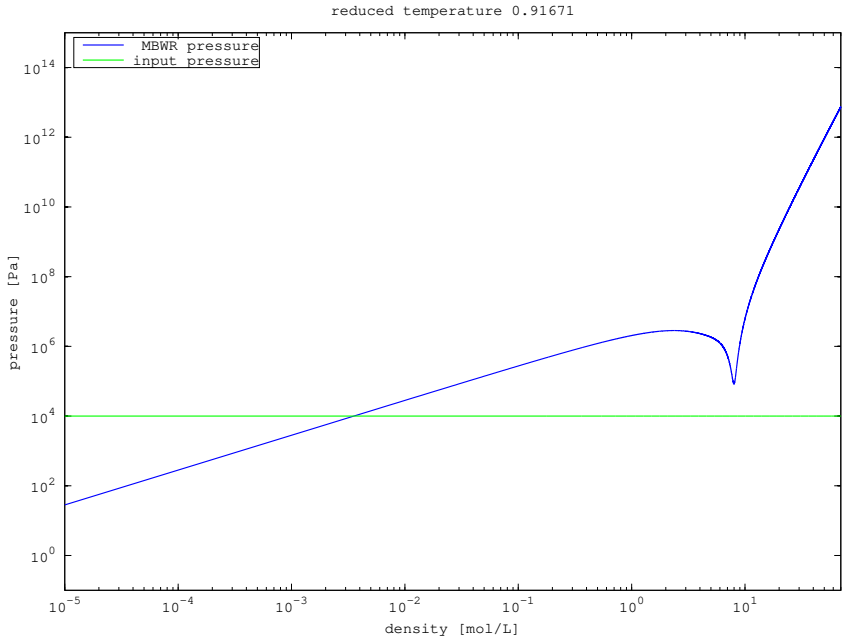


Figure 3: The MBWR-19 density-pressure curve for CH_4 and a temperature below but close to the critical temperature, and an example input pressure drawn in.

again, the routine will only converge if the initial guess is on the correct side of the density root. If the newton_density fails, the function barenewton is called, and if also this fails the TPlib solver is called.

5.3 Choosing the initial density

As already pointed out, choosing a good initial density is of the utmost importance for the algorithm to converge to the correct root. For the purposes of reaching the correct density in a robust and time-efficient manner, experimentation showed that the following choices were favorable.

Choosing the initial vapor density

For the vapor root, we use the initial guess

- $\rho_{0,\text{vapor}} = 10^{-6}$ if $P \geq 100$ Pa,
- $\rho_{0,\text{vapor}} = 10^{-12}$ if $P < 100$ Pa,

This will be lower than the MBWR vapor density (if it exists), just as we desire. Moreover, experiments show that Newton's method converges quickly even though the initial guess is so low. Another advantage of this method of choosing the initial value is that no computation time is spent.

Choosing the initial liquid density

Although one might think that analogously to the choice of initial vapor density, a really large value (e.g. 10^2 mol/L) would be a good choice of initial liquid density, there are two reasons that this is a bad idea. The first is that unlike in the vapor phase, where the MBWR equation is designed to have the correct asymptotic behavior $\lim_{\rho \to 0} P(T, \rho) = 0$, one has no guarantee of physical behavior for large ρ. Indeed, for MBWR-32, for large enough densities, $P(T, \rho)$ becomes negative. Interestingly

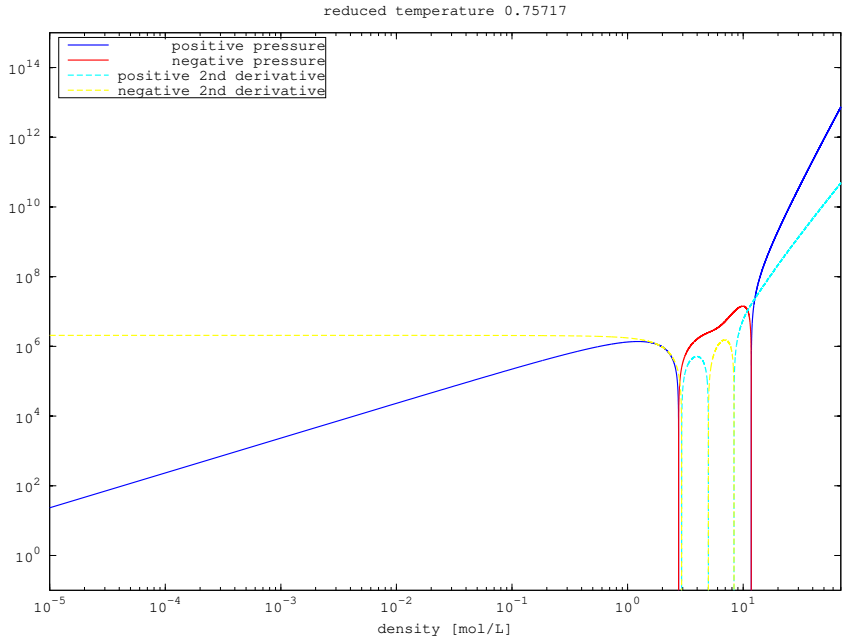


Figure 4: A log-log plot of $P(T_0, \rho)$ and $\partial_\rho^2 P(T_0, \rho)$ for the MBWR-19 equation with component C3.

and luckily, however, the MBWR-19 equation has a predictable behavior for large ρ, namely[3]

$$\lim_{\rho\to\infty} P_{\text{MBWR-19}}(T,\rho) = \infty, \quad \lim_{\rho\to\infty} \partial_\rho P_{\text{MBWR-19}}(T,\rho) = \infty, \quad \lim_{\rho\to\infty} \partial_\rho^2 P_{\text{MBWR-19}}(T,\rho) = \infty$$

No one in the consulted literature seems to mention this, and although it hasn't been checked for all substances in the MBWR-19 database of substances, it holds in all cases we have encountered.

For subcritical temperatures, the density computed by Soave-Redlich-Kwong is used. For supercritical temperatures, the ideal gas equation is used. For temperatures and pressures near the critical point, the critical density is used. In all of these three cases, the initial density is scaled up by up to 50% or more to ensure that the density is on the right side of the MBWR density. The actual scaling factor varies depending on the situation, guided by experimentation.

5.4 The density solver in TPlib

TPlib uses the second order Newton method, also called Halley's method:

$$\begin{aligned}\Delta x_n &= -\frac{f'(x_n)}{f''(x_n)} \pm \frac{1}{f''(x_n)}\sqrt{f'(x_n)^2 - 2f''(x_n)(f(x_n)-P)} \\ &= -\frac{f'(x_n)}{f''(x_n)} \pm \sqrt{\left(\frac{f'(x_n)}{|f''(x_n)|}\right)^2 + 2\frac{P+f(x_n)}{f''(x_n)}},\end{aligned}$$

and if the radicand is negative, one sets $\Delta x_n = -f'(x_n)/f''(x_n)$.

If the reduced pressure is less than 1, a so-called Modified Rackett technique[4] is used to estimate the saturated vapor pressure at the given temperature, and from this the saturated vapor density can be estimated. This estimate is then scaled by a factor greater than 1.

$$\begin{aligned}z_{\text{rackett}} &= (0.29056 - 0.08775\omega)^{1+(1-T_r)^{2/7}} \\ \rho_0 &= \frac{P_c}{RT_c z_{\text{rackett}}}\end{aligned}$$

[3] Of course, the two first limits are implied by the last limit.

[4] See e.g. [4], section 4.11.

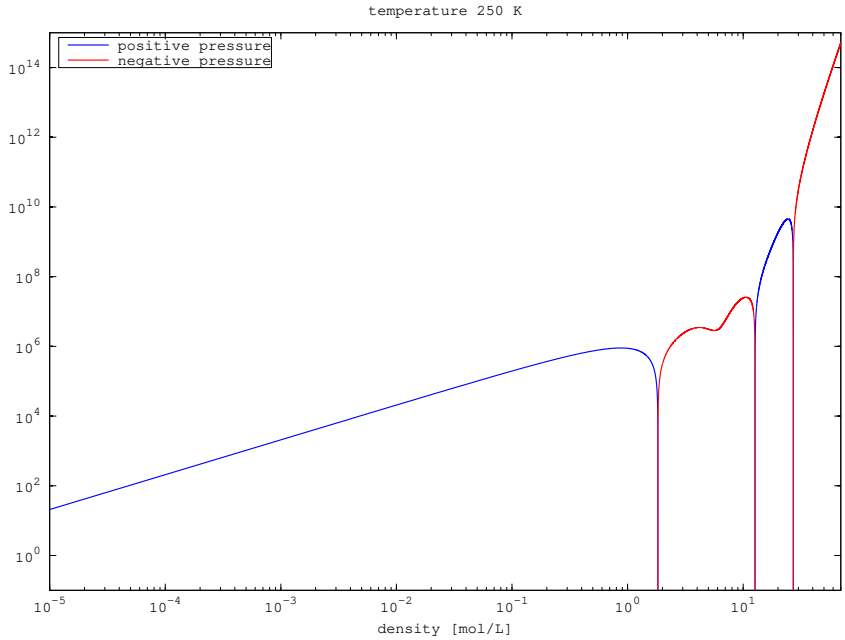


Figure 5: A density-pressure curve for the MBWR-19 equation with component C3. Note how the computed pressure becomes negative for large ρ.

For supercritical pressures, the critical density is used as the initial value.

The density solver from TPlib has been copied over to the MBWR module in ThermoPack, with the only modification being that it has been given the same initial density guess as the main density algorithm. This reimplementation of the TPlib solver is used to compare the robustness and computational time with the new density solver. It is also used as a last fallback routine if the new density algorithm fails.

5.5 Testing the density solver robustness

To optimize and test the density solver, a program which bombards the density solver with test cases was written. The density solver is tested for 1000 equispaced temperatures between the triple point temperature and 1000 K; for each of these the pressure input is 1000 equispaced points from the triple point pressure to 10^7 Pa. This is done for both the vapor and the liquid phase. All in all, the density solver is tested on a grid of $2 \cdot 10^6$ points. Our critieria for convergence are

1. $|P_{MBWR}(\rho) - P_{in}| < 10^{-5}$,
2. $\partial_\rho P_{MBWR}(\rho) > 0$,
3. the phase convexity criterion is fulfilled.

Results from the robustness tests are presented below.

MBWR-32

For the tested components,

- C1,
- C2,
- C3,

- R134a,
- O2,
- N2,

the density solver converges in all cases.

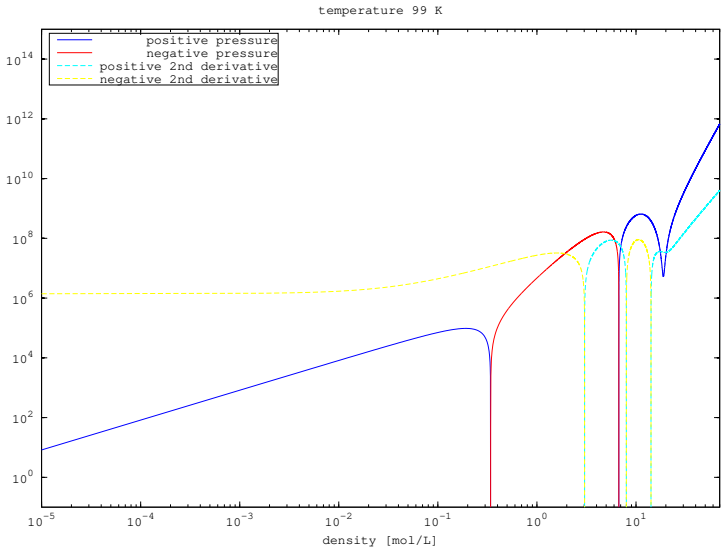


Figure 6: Density versus pressure for the MBWR-19 equation, component C2, temperature 99 K.

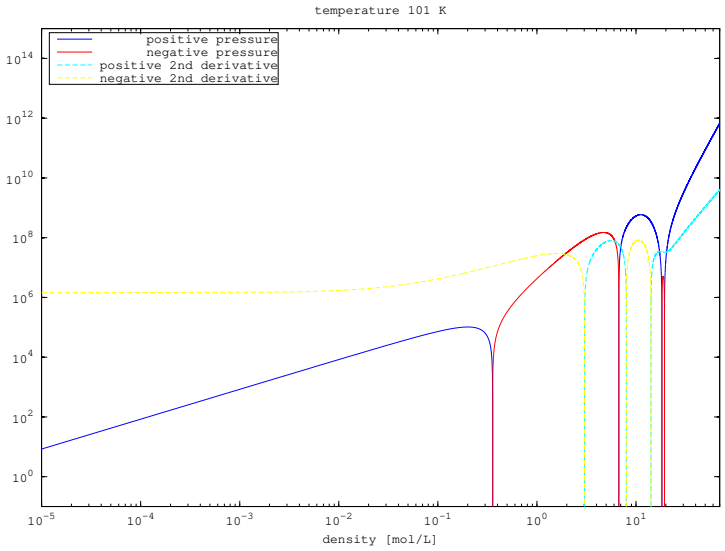
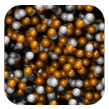


Figure 7: Density versus pressure for the MBWR-19 equation, component C2, temperature 101 K.

MBWR-19

For MBWR-19, we tested 10 components. For the following components the density solver converges in all cases:

- C1,



T [K]	P [Pa]	Comp.	ThermoPack vap. [s]	TPlib vap. [s]	Thermopack liq. [s]	TPlib liq. [s]
254	1.26e6	C3	0.14	0.34	0.11	0.15
143	2.09e6	O2	0.076	0.21	0.12	0.16
129	7.46e4	HE	0.065	0.14	0.059	0.068
172	5.12e5	R152a	0.12	0.27	0.10	0.13
1000	1e8	C1	0.087	0.32	0.087	0.12

Table 3: Performance of the ThermoPack and TPlib density solvers in various circumstances.

- C3,
- O2,
- N2,
- R152a,
- HE,
- H2O,
- NH3.

We also tested NC7 and C2. As mentioned above, the NC7 vapor hill becomes convex for pressures higher than the saturated vapor pressure, and therefore some of the computed vapor densities can fail the phase convexity test. This is indeed what happens – 175 failed cases – when testing the solver and giving "vapor" as input phase (using "liquid" as input phase always yields convergence). If one reruns the tests for NC7 while only using the 1. and 2. convergence criterion, while ignoring the phase convexity tests, we get attain convergence in all cases.

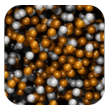
For C2, the liquid hill looks unusual, see figures 6 and 7. These nonstandard features make it hard for any gradient based solver to solve for the density, as it requires a very accurate initial density guess. These problems disappear for temperatures above 100 K, and the solver then chooses as liquid density the root corresponding to the rightmost solution; but in this case it is not clear whether this is really the correct root. In other words, C2 seems to be a difficult component for MBWR-19. Although this can be further investigated, and can probably be remedied by tweaking the solver, this has not been done.

5.6 Testing the density solver which minimizes the Gibbs energy

We have also implemented a density algorithm which finds the density in the phase such that the Gibbs energy is minimized. A quick test using MBWR-19 and water shows that for $P = 101325$ Pa it changes from having $\rho = 53.2$ mol/L to having $\rho = 0.0331$ mol/L at 373.14 K. This indeed coincides with the normal boiling point of water.

5.7 Speed test

When the MBWR main density routine converges, it is always faster than the TPlib solver. The speedup depends on the phase, and to a lesser extent on the given temperature and pressure. Typical situations are shown in Table 3, for various substances and temperature-pressure states. For this table the MBWR-19 equation has been used, and the cpu-time has been measured for 10^5 calls to the density solver (to render the inherent inaccuracy in the measurement of CPU-time insignificant).



5.8 Further improvements

In one sense the density solver converges *too often*; indeed, it often finds a vapor density even though we are far above the saturated vapor pressure at the given temperature. Ideally, a correlation for the saturated vapor pressure should be used in the density routine. This information could stop the density solver from finding nonexistent roots, and could further speed up the density solver by terminating the search when the pressure at the current density iterate is above/below the saturated vapor pressure (depending on which phase is being solved for).

6 Routines for calculating necessary thermodynamic functions

To get a complete program for the MBWR equations, we have also written a module for calculating the residual entropy, the residual enthalpy, the residual Gibbs energy, the z-factor and the logarithmic fugacity coefficients, as well as their first order partial derivatives with respect to T and P. Per now the MBWR equations are only used as a part of the SPUNG framework, so these thermodynamic functions are never called. The one exception is the routine for calculating Gibbs energy, which is used in the in the density routine which chooses the root having minimal Gibbs energy.

7 Testing the MBWR models

To validate the implementation of the MBWR-19 and MBWR-32 models, various tests have been performed.

7.1 Thermodynamic identities

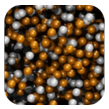
In addition to the numerical test of the derivatives, thermodynamic identities and identities found from Euler's theorem, serve as decent consistency tests for the analytical derivatives. The test supplied here are all found in Michelsen [3]. To test the derivatives of the reduced residual Helmholtz function, the following identities may be applied:

$$F = V\left(\frac{\partial F}{\partial V}\right)_{T,\mathbf{n}} + \sum_i n_i \left(\frac{\partial F}{\partial n_i}\right)_{T,V} \tag{7.1}$$

$$V\left(\frac{\partial^2 F}{\partial V \partial n_i}\right)_{T} + \sum_j n_j \left(\frac{\partial^2 F}{\partial n_j \partial n_i}\right)_{T,V} = 0 \tag{7.2}$$

$$V\left(\frac{\partial^2 F}{\partial V^2}\right)_{T,\mathbf{n}} + \sum_j n_j \left(\frac{\partial^2 F}{\partial n_j \partial V}\right)_{T} = 0 \tag{7.3}$$

When these are all satisfied, the fugacity coefficients and their derivatives may be tested by the



identities:

$$\left(\frac{\partial}{\partial n_j}\sum_i n_i \ln\phi_i\right)_{T,P} = \ln\phi_j \tag{7.4}$$

$$\left(\frac{\partial \ln\phi_i}{\partial n_j}\right)_{T,P} = \left(\frac{\partial \ln\phi_j}{\partial n_i}\right)_{T,P} \tag{7.5}$$

$$\sum_i n_i \left(\frac{\partial \ln\phi_i}{\partial n_j}\right)_{T,P} = 0 \tag{7.6}$$

$$\left(\frac{\partial}{\partial P}\sum_i n_i \ln\phi_i\right)_{T,\mathbf{n}} = \frac{(z-1)n}{P} \tag{7.7}$$

$$\sum_i n_i \left(\frac{\partial \ln\phi_i}{\partial T}\right)_{P,\mathbf{n}} = -\frac{H^R(T,P,\mathbf{n})}{RT^2} \tag{7.8}$$

Of course, since it is an equation for pure substances, the MBWR equations only has one fugacity coefficient. All of these identities have been implemented, and the code seems to fulfill them when tested on a few points.

7.2 Comparing numerical and analytical derivatives

The analytical derivatives for the implemented thermodynamic functions have been compared to their finite-difference counterparts, with consistent results.

7.3 Comparison with previous MBWR implementations

In the code there is an algorithm which calculates the component-specific coefficients for $\alpha(T,P)$ using the component-specific coefficients for $P(T,\rho)$. The coefficients for the Helmholtz energy in the MBWR-32 model with R152a as the substance, checks out with the coefficients computed by an earlier implementation in the NIST thermodynamic library REFPROP.

8 Extension to mixtures: The SPUNG equation of state

8.1 Pure fluid scale factors from a cubic equation of state

Consider a generic cubic equation of state,

$$P = \frac{RT}{v-b} - \frac{a(T)}{(v+\delta_1 b)(v+\delta_2 b)}.$$

From an equation for P, one can find the residual Helmholtz energy from the integral $A^r(T,V,\mathbf{n}) = -\int_\infty^V (P - nRT/V')\,dV'$, and for the generic cubic equation we get

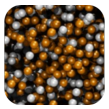
$$\frac{A^r(T,v)}{RT} = -\ln(1-b/v) - \frac{a(T)}{RTb}\frac{1}{\delta_1-\delta_2}\ln\left(\frac{1+\delta_1 b/v}{1+\delta_2 b/v}\right) \tag{8.1}$$

$$= -\ln(1-\beta) - \frac{\Gamma}{\delta_1-\delta_2}\ln\left(\frac{1+\delta_1\beta}{1+\delta_2\beta}\right), \tag{8.2}$$

where we defined the adimensional parameters $\Gamma = a(T)/bRT$ and $\beta = b/v$.

We first develop the SPUNG model for pure fluids. Suppose therefore we have two pure fluids called fluid 1 and fluid 0. We say they are in **corresponding states** when $\Gamma_1 = \Gamma_0$ and $\beta_1 = \beta_0$, i.e. when

$$\frac{a_1(T_1)}{b_1 RT_1} = \frac{a_0(T_0)}{b_0 RT_0}, \quad \text{and} \quad \frac{b_1}{v_1} = \frac{b_0}{v_0}. \tag{8.3}$$



In particular, this implies that the fluids have the same reduced residual Helmholtz energy. Now, for cubic equations of state like PR and SRK, $a(T)$ and b take the special form

$$a(T) = \Omega_a(R^2T_c^2/P_c)\alpha(T), \quad \alpha(T) = \left(1 + m(\omega)(1 - \sqrt{T/T_c})\right)^2 \tag{8.4}$$

and

$$b = \Omega_b \cdot RT_c/P_c, \tag{8.5}$$

where Ω_a and Ω_b are substance-independent constants. From equations (8.3) and (8.4) and (8.5), we will be able to calculate the pure fluid **scale factors**, defined as

$$h = v_1/v_0, \qquad f = T_1/T_0.$$

The scale factor for volume is

$$h = \frac{v_1}{v_0} = \frac{b_1}{b_0} = \frac{T_{c_1}P_{c_0}}{T_{c_0}P_{c_1}}, \tag{8.6}$$

while the scale factor for temperature can be written

$$f = \frac{T_1}{T_0} = \frac{a_1(T_1)b_0}{a_0(T_0)b_1} = \frac{T_{c_1}}{T_{c_0}}\frac{\alpha(T_{r_1})}{\alpha(T_{r_0})}, \tag{8.7}$$

Explicit temperature scale factor when $\alpha = \alpha_{SRK}$ or $\alpha = \alpha_{PR}$

When the ratio of reduced temperatures, $\theta = T_{r_1}/T_{r_0}$, is calculated from Soave's or Peng-Robinson's correlation for α, we get

$$\theta = \left(\frac{1 + m_1 - m_1\sqrt{T_{r_1}}}{1 + m_0 - m_0\sqrt{T_{r_0}}}\right)^2, \tag{8.8}$$

It is possible to solve for θ_1 as an explicit function of T_{r_1}. Indeed, taking the square root on both sides of (8.8) and substituting $T_{r_0} = T_{r_1}/\theta$, we get

$$\sqrt{\theta} = \frac{1 + m_1 - m_1\sqrt{T_{r_1}}}{1 + m_0 - m_0\sqrt{(T_{r_1}/\theta)}},$$

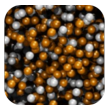
and multiplying both sides with the denominator, we can easily solve for θ:

$$\theta = \left(\frac{1 + m_1}{1 + m_0} + \frac{m_0 - m_1}{1 + m_0}\sqrt{T_{r_1}}\right)^2.$$

The morale is: when obtaining pure fluid scale factors from cubic equations of state, one can get explicit expressions for the scale factors, depending only on the accentric factors and critical parameters, together with the temperature of one of the substances.

8.2 Mixtures

The most interesting application of SPUNG is when one is dealing with a *mixture* of components, to which we now turn our focus. The idea is to map the thermodynamic state (T, v) for the mixture to some corresponding (T_0, v_0) of a reference fluid. The idea is that if one has a very accurate description (using e.g. a multiparameter equation of state) of the thermodynamics of the pure fluid 0, then one can use this mapping to get an accurate description of the mixture.



Using a cubic equation, the expression for the Helmholtz energy for n moles of a mixture is (see e.g. Michelsen [3, p.105–107])

$$\begin{aligned}\frac{A_r(T,V,\mathbf{n})}{nRT} &= -\ln(1-B(\mathbf{n})/V) - \frac{D(T,\mathbf{n})}{nRTb}\frac{1}{\delta_1-\delta_2}\ln\left(\frac{1+\delta_1 B(\mathbf{n})/V}{1+\delta_2 B(\mathbf{n})/V}\right)\\ &= -\ln(1-\beta_{mix}) - \frac{\Gamma_{mix}}{\delta_1-\delta_2}\ln\left(\frac{1+\delta_1\beta_{mix}}{1+\delta_2\beta_{mix}}\right),\end{aligned}$$

where we have defined the adimensional mixture parameters as $\Gamma_{mix} = D(T,\mathbf{n})/bRT$ and $\beta_{mix} = B(\mathbf{n})/V$. The quantities $D(T,\mathbf{n})/n^2$ and $B(\mathbf{n})/n$ are the mixture analogs of the parameters $a(T)$ and b for a pure fluid.

The principle of corresponding states allows us to calculate the reduced residual energy of a mixture from the reduced residual Helmholtz energy of a pure reference fluid 0, by equating $\Gamma_{mix} = \Gamma_0$ and $\beta_{mix} = \beta_0$, i.e.

$$\frac{D(T,\mathbf{n})}{nRTB(\mathbf{n})} = \frac{a_0(T_0)}{RT_0b_0} \qquad \text{and} \qquad \frac{B(\mathbf{n})}{V} = \frac{b_0}{v_0}.$$

We thus get the two mixture scale factors $\hat{H}$ and $\hat{F}$

$$\hat{H} = \frac{V}{v_0} = \frac{B(\mathbf{n})}{b_0}, \qquad \text{and} \qquad \hat{F} = \frac{nT}{T_0} = \frac{D(T,\mathbf{n})}{B(\mathbf{n})}\frac{b_0}{a_0(T_0)}.$$

Note that the mixture scale factors are first order homogeneous functions in the mole numbers. The mixing rules adopted for $B(\mathbf{n})$ and $D(T,\mathbf{n})$ are optional as long as they give a consistent model.

Adopting the conventional mixing rules – also called the van der Waals one-fluid mixing rules, or the quadratic mixing rules – for $B(\mathbf{n})$ and $D(T,\mathbf{n})$, we get

$$\hat{H}nb_0 = nB(\mathbf{n}) = \sum_{i,j} n_i n_j b_{ij}$$

and

$$\hat{F}\hat{H}a_0(T_0) = \hat{F}\hat{H}a_0(nT/\hat{F}) = D(T,\mathbf{n}) = \sum_{i,j} n_i n_j a_{ij}(T).$$

For mixtures involving polar substances, mixture rules based on excess Gibbs energy models may be more appropriate. The Huron-Vidal mixing rule is a prominent example. We mention that the cubic equation of state which is used to calculate the scale factors $\hat{H}$ and $\hat{F}$ is called the **scale factor equation** or the **shape factor equation**. $\hat{H}$ and $\hat{F}$ are often called shape factors.

Although $\hat{H}$ is given explicitly in terms of the mixture composition as $\hat{H} = \frac{B(\mathbf{n})}{b_0}$, it is in the general case impossible to give an explicit expression for $\hat{F}$ in terms of the mixture temperature and composition. In certain cases however, this can be done.

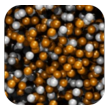
Explicit temperature scale factor when $\alpha = \alpha_{SRK}$ or $\alpha = \alpha_{PR}$

If $a_0(T_0) = a_{0c}(1 + m_0 - m_0\sqrt{T_0/T_{0c}})^2$, it turns out that we can solve for $\hat{F}$ from the implicit expression

$$\hat{F} = \frac{1}{\hat{H}}\frac{D(T,\mathbf{n})}{a_0(nT/\hat{F})}.$$

Indeed, by inserting the form for a_0 we get

$$\hat{F} = \frac{D(T,\mathbf{n})}{\hat{H}a_{0c}\left(1 + m_0 - m_0\sqrt{nT/(\hat{F}T_{0c})}\right)^2}$$



$$\sqrt{\hat{F}}\left(1+m_0-m_0\sqrt{\frac{nT}{\hat{F}T_{0c}}}\right)=\left(\frac{D(T,\mathbf{n})}{\hat{H}a_{0c}}\right)^{1/2}$$

$$\sqrt{\hat{F}}(1+m_0)=m_0\sqrt{\frac{nT}{T_{0c}}}+\left(\frac{D(T,\mathbf{n})}{\hat{H}a_{0c}}\right)^{1/2}$$

$$\hat{F}=\frac{1}{(1+m_0)^2}\left(m_0\sqrt{\frac{nT}{T_{0c}}}+\left(\frac{D(T,\mathbf{n})}{\hat{H}a_{0c}}\right)^{1/2}\right)^2. \tag{8.9}$$

Temperature scale factor when $\alpha=\alpha_{TWU}$

Suppose now that we use the alpha formulation of Twu-Coon-Cunningham:

$$a_0(T_0)=a_{0c}\cdot T_{0r}^{N(M-1)}\exp\left(L-LT_{0r}^{MN}\right),$$

where the L, M and N have been fitted to vapor pressure data for each fluid. This alpha correlation is more tailored to specific components than α_{SRK} , seeing as the only component-specific input to α_{SRK} is the acentric factor and the critical temperature. In the current database the parameters have only been stored for the four substances CO_2, CH_4, H_2S and H_2O.

Let us find the temperature shape factor using this alpha formulation. Using $\hat{F}=\frac{D(T,\mathbf{n})}{\hat{H}a_{0c}\alpha_{TWU}(T_0)}$ and $T_0=nT/\hat{F}$, we get

$$\hat{F}\cdot\left(\frac{nT}{\hat{F}T_{0c}}\right)^{N(M-1)}\exp\left(L-L\left(\frac{nT}{\hat{F}T_{0c}}\right)^{MN}\right)=\frac{D(T,\mathbf{n})}{\hat{H}a_{0c}},$$

or

$$\hat{F}^{1+N(1-M)}\exp\left(L-L\left(\frac{nT}{\hat{F}T_{0c}}\right)^{MN}\right)=\left(\frac{T_{0c}}{nT}\right)^{N(M-1)}\frac{D(T,\mathbf{n})}{\hat{H}a_{0c}}. \tag{8.10}$$

From this last expression, it is clear that it is not possible to solve for $\hat{F}$ using simple functions[5]. The code therefore solves this using Newton's method, with the $\hat{F}$ factor computed with α_{SRK}, (8.9), as starting value.

A drawback is that these coefficients are only valid for subcritical temperatures $T_{0r}<1$. Skaugen [7] gives a more detailed discussion of the Twu correlation, and what can be done for supercritical temperatures.

Finally, we point out that if one wants to implement other α-formulations into the SPUNG code, this is straightforward as one can simply mirror the code for α_{SRK} (if an explicit expression for $\hat{F}$ is available) or α_{TWU} (if $\hat{F}$ has to be solved iteratively).

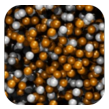
8.3 Partial derivatives

We now calculate partial derivatives in the case where we use a cubic equation to compute the shape factors, and where we use the conventional mixing rules for D and B. Note that they choice of α in the cubic equation can be anything.

Let us sum up the relevant formulas once more. The principle of corresponding states tells us that given a mixture in the state $(T,V,\mathbf{n})$, we have that

$$A^r(T,V,\mathbf{n})=\hat{F}M(T_0,v_0), \tag{8.11}$$

[5]It is solvable by using the Lambert W function, but this transcendental function is not available from the numerical libraries used by ThermoPack.



where (T_0, v_0) is the reference state, defined as

$$v_0 = \frac{V}{\hat{H}}, \qquad T_0 = \frac{nT}{\hat{F}}, \tag{8.12}$$

where the scale factors $\hat{H}$ and $\hat{F}$ are given by

$$\hat{H} = \frac{B}{b_0}, \qquad \hat{F} = \frac{D}{B}\frac{b_0}{a_0(T_0)}. \tag{8.13}$$

Here D and B given by the van der Waals mixing rule

$$nB = \sum_i n_i \sum_j n_j b_{ij} \tag{8.14}$$

$$D = \sum_i n_i \sum_j n_j a_{ij}(T) \tag{8.15}$$

$B(\mathbf{n})$ and D are completely determined from the underlying cubic equation of state.

We again stress the point that $\hat{H} = \hat{H}(\mathbf{n})$ only depends on composition, while $\hat{F} = \hat{F}(T, \mathbf{n})$ only depends on temperature and composition.

First we calculate the first and second order partial derivatives of $A^r(T, V, \mathbf{n})$ with respect to T, V and n_i in terms of the partial derivatives of $\hat{H}$ and M with respect to T, V and n_i.

$$\left(\frac{\partial A^r}{\partial T}\right)_{V,\mathbf{n}} = \hat{F}_T M + \hat{F} M_T \tag{8.16}$$

$$\left(\frac{\partial A^r}{\partial V}\right)_{T,\mathbf{n}} = \hat{F} M_V \tag{8.17}$$

$$\left(\frac{\partial A^r}{\partial n_i}\right)_{T,V} = \hat{F}_i M + \hat{F} M_i \tag{8.18}$$

$$\left(\frac{\partial^2 A^r}{\partial T^2}\right)_{V,\mathbf{n}} = \hat{F}_{TT} M + 2\hat{F}_T M_T + \hat{F} M_{TT} \tag{8.19}$$

$$\left(\frac{\partial^2 A^r}{\partial V^2}\right)_{T,\mathbf{n}} = \hat{F} M_{VV} \tag{8.20}$$

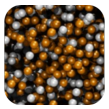
$$\left(\frac{\partial^2 A^r}{\partial n_i \partial n_j}\right)_{T,V} = \hat{F}_{ij} M + \hat{F}_i M_j + \hat{F}_j M_i + \hat{F} M_{ij} \tag{8.21}$$

$$\left(\frac{\partial^2 A^r}{\partial T \partial V}\right)_{\mathbf{n}} = \hat{F}_T M_V + \hat{F} M_{TV} \tag{8.22}$$

$$\left(\frac{\partial^2 A^r}{\partial T \partial n_i}\right)_{V} = \hat{F}_{Ti} M + \hat{F}_i M_T + \hat{F}_T M_i + \hat{F} M_{Ti} \tag{8.23}$$

$$\left(\frac{\partial^2 A^r}{\partial V \partial n_i}\right)_{T} = \hat{F}_i M_V + \hat{F} M_{Vi} \tag{8.24}$$

Next we calculate the first and second order partial derivatives of M with respect to T, V and n_i in terms of the partial derivatives of M with respect to T_0, v_0 and the derivatives of T_0 and v_0 with



respect to T, V and n_i.

$$M_T = M_{T_0} T_{0,T} \tag{8.25}$$
$$M_V = M_{v_0} v_{0,V} \tag{8.26}$$
$$M_i = M_{T_0} T_{0,i} + M_{v_0} v_{0,i} \tag{8.27}$$
$$M_{TT} = M_{T_0 T_0} T_{0,T}^2 + M_{T_0} T_{0,TT} \tag{8.28}$$
$$M_{VV} = M_{v_0 v_0} v_{0,V}^2 + M_{v_0} v_{0,VV} \tag{8.29}$$
$$M_{ij} = M_{T_0 T_0} T_{0,i} T_{0,j} + M_{T_0} T_{0,ij} + M_{v_0 v_0} v_{0,i} v_{0,j} \tag{8.30}$$
$$+ M_{v_0} v_{0,ij} + M_{T_0 v_0} (T_{0,i} v_{0,j} + T_{0,j} v_{0,i}) \tag{8.31}$$
$$M_{TV} = M_{T_0 v_0} T_{0,T} v_{0,V} \tag{8.32}$$
$$M_{Ti} = M_{T_0 T_0} T_{0,T} T_{0,i} + M_{T_0 v_0} T_{0,T} v_{0,i} + M_{T_0} T_{0,Ti} \tag{8.33}$$
$$M_{Vi} = M_{v_0 v_0} v_{0,V} v_{0,i} + M_{T_0 v_0} T_{0,i} v_{0,V} + M_{v_0} v_{0,Vi} \tag{8.34}$$

Partial derivatives of B with respect to n

$$B_i = \frac{\partial}{\partial n_i} \left(\frac{\sum_i \sum_j n_i n_j b_{ij}}{n} \right) = \frac{\left(2 \sum_j n_j b_{ij} \right) n - \sum_i \sum_j n_i n_j b_{ij}}{n^2} = \frac{2 \sum_j n_j b_{ij} - B}{n}$$
$$B_{ij} = \frac{\partial^2}{\partial n_i \partial n_j} \left(\frac{2 \sum_k n_k b_{ik} - B}{n} \right) = \frac{(2 b_{ij} - B_j) n - (2 \sum_k n_k b_{ik} - B)}{n^2} = \frac{2 b_{ij} - B_i - B_j}{n}$$

Partial derivatives of $\hat{H}$ with respect to n

We can get the partial derivatives of $\hat{H}$ can be written in terms of the partial derivatives of B:

$$\hat{H}_i = \frac{B_i}{b_0} = \frac{2 \sum_j n_j b_{ij} - B}{n b_0} \tag{8.35}$$
$$\hat{H}_{ij} = \frac{B_{ij}}{b_0} = \frac{2 b_{ij} - B_i - B_j}{n b_0} \tag{8.36}$$

Partial derivatives of v_0 with respect to V and n

We now find the partial derivatives of v_0. To find the derivatives with respect to composition, we use $\hat{H} v_0 = V$ to get

$$\hat{H}_i v_0 + \hat{H} v_{0,i} = 0, \qquad \hat{H}_{ij} v_0 + \hat{H}_i v_{0,j} + \hat{H}_j v_{0,i} + \hat{H} v_{0,ij} = 0.$$

Thus

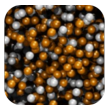
$$\frac{v_{0,i}}{v_0} = -\frac{\hat{H}_i}{\hat{H}} = -\frac{B_i}{B}, \tag{8.37}$$

and

$$\frac{v_{0,ij}}{v_0} = -\frac{\hat{H}_{ij}}{\hat{H}} - \frac{\hat{H}_i}{\hat{H}} \frac{v_{0,j}}{v_0} - \frac{\hat{H}_j}{\hat{H}} \frac{v_{0,i}}{v_0} = -\frac{B_{ij}}{B} + 2 \frac{B_i}{B} \frac{B_j}{B} \tag{8.38}$$

The V-derivatives of v_0 can be found by differentiating $\hat{H} v_0 = V$ with respect to V. This gives

$$v_{0,V} = \frac{1}{\hat{H}} \tag{8.39}$$
$$v_{0,VV} = 0 \tag{8.40}$$
$$v_{0,Vi} = -\frac{\hat{H}_i}{\hat{H}^2} = -\frac{B_i b_0}{B^2} \tag{8.41}$$



Partial derivatives of T_0 with respect to T and n

To find the derivatives with respect to composition, we use $\hat{F}T_0 = nT$ to get

$$\hat{F}_i T_0 + \hat{F} T_{0,i} = T, \qquad \hat{F}_{ij} T_0 + \hat{F}_i T_{0,j} + \hat{F}_j T_{0,i} + \hat{F} T_{0,ij} = 0.$$

Thus

$$\frac{T_{0,i}}{T_0} = \frac{T}{\hat{F} T_0} - \frac{\hat{F}_i}{\hat{F}} = \frac{1}{n} - \frac{\hat{F}_i}{\hat{F}}, \tag{8.42}$$

and similarly we find

$$\frac{T_{0,T}}{T_0} = \frac{1}{T} - \frac{\hat{F}_T}{\hat{F}} \tag{8.43}$$

The second order partials are given by

$$\frac{T_{0,ij}}{T_0} = -\frac{\hat{F}_{ij}}{\hat{F}} - \frac{\hat{F}_i}{\hat{F}}\frac{T_{0,j}}{T_0} - \frac{\hat{F}_j}{\hat{F}}\frac{T_{0,i}}{T_0}, \tag{8.44}$$

$$\frac{T_{0,Ti}}{T_0} = -\frac{\hat{F}_{Ti}}{\hat{F}} - \frac{\hat{F}_i}{\hat{F}}\frac{T_{0,T}}{T_0} - \frac{\hat{F}_T}{\hat{F}}\frac{T_{0,i}}{T_0} + \frac{1}{\hat{F}} \tag{8.45}$$

$$\frac{T_{0,TT}}{T_0} = -\frac{\hat{F}_{TT}}{\hat{F}} - 2\frac{\hat{F}_T}{\hat{F}}\frac{T_{0,T}}{T_0}. \tag{8.46}$$

Note that Michelsen [3] has an error in the expression for $T_{0,Ti}/T_0$, as the last term on the right hand side is missing.

Partial derivatives of D with respect to T and n

$$D_i = 2\sum_j n_j a_{ij} \tag{8.47}$$

$$D_{iT} = 2\sum_j n_j \left(\partial a_{ij}/\partial T\right) \tag{8.48}$$

$$D_{ij} = 2a_{ij} \tag{8.49}$$

$$D_T = \tfrac{1}{2}\sum_i n_i D_{iT} \tag{8.50}$$

$$D_{TT} = \sum_i n_i \sum_j n_j \left(\partial^2 a_{ij}/\partial T^2\right) \tag{8.51}$$

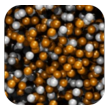
Partial derivatives of $\hat{F}$ with respect to T and n

We now calculate the partial derivatives for $\hat{F}$ with respect to temperature and composition in terms of the partial derivatives of D, $\hat{H}$ and a_0. To do this differentiate $\hat{F}(T,\mathbf{n})\hat{H}(\mathbf{n})a_0(T_0) = D(T,\mathbf{n})$ with respect to composition, giving

$$\hat{F}_i \hat{H} a_0 + \hat{F}\hat{H}_i a_0 + \hat{F}\hat{H} a_{0,T_0} T_{0,i} = D_i = 2\sum_j n_j a_{ij},$$

which when divided by $D = \hat{F}\hat{H}a_0$ becomes

$$\frac{\hat{F}_i}{\hat{F}} + \frac{\hat{H}_i}{\hat{H}} + \frac{a_{0,T_0}}{a_{T_0}} T_{0,i} = \frac{D_i}{D}.$$



By using the expression (8.42) to eliminate $T_{0,i}$, we end up with

$$\frac{\hat{F}_i}{\hat{F}}\left(1-\frac{a_{0,T_0}}{a_0}T_0\right)+\frac{\hat{H}_i}{\hat{H}}+\frac{a_{0,T_0}}{a_{T_0}}\frac{T_0}{n}=\frac{D_i}{D},$$

and thus

$$\frac{\hat{F}_i}{\hat{F}}=\frac{\frac{D_i}{D}-\frac{B_i}{B}-\frac{a_{0,T_0}}{a_{T_0}}\frac{T_0}{n}}{1-\frac{a_{0,T_0}}{a_0}T_0}. \tag{8.52}$$

Similarly, we find

$$\frac{\hat{F}_T}{\hat{F}}=\frac{\frac{D_T}{D}-\frac{a_{0,T_0}}{a_{T_0}}\frac{T_0}{T}}{1-\frac{a_{0,T_0}}{a_0}T_0}. \tag{8.53}$$

To derive the second order partial derivative of $\hat{F}$ with respect to composition we differentiate $\hat{F}(T,\mathbf{n})\hat{H}(\mathbf{n})a_0(T_0)=D(T,\mathbf{n})$ twice. This gives

$$\begin{aligned}
&\hat{F}_{ij}\hat{H}a_0+\hat{F}_i\hat{H}_ja_0+\hat{F}_i\hat{H}a_{0,T_0}T_{0,j}+\\
&\hat{F}_j\hat{H}_ia_0+\hat{F}\hat{H}_{ij}a_0+\hat{F}\hat{H}_ia_{0,T_0}T_{0,j}+\\
&\hat{F}_j\hat{H}a_{0,T_0}T_{0,i}+\hat{F}\hat{H}_ja_{0,T_0}T_{0,i}+\hat{F}\hat{H}(a_{0,T_0T_0}T_{0,i}T_{0,j}+a_{0,T_0}T_{0,ij})=2a_{ij},
\end{aligned}$$

and dividing by $D=\hat{F}\hat{H}a_0$, we get the cleaner expression

$$\begin{aligned}
&\frac{\hat{F}_{ij}}{\hat{F}}+\frac{\hat{F}_i}{\hat{F}}\frac{\hat{H}_j}{\hat{H}}+\frac{\hat{F}_i}{\hat{F}}\frac{a_{0,T_0}}{a_0}T_{0,j}+\\
&\frac{\hat{F}_j}{\hat{F}}\frac{\hat{H}_i}{\hat{H}}+\frac{\hat{H}_{ij}}{\hat{H}}+\frac{\hat{H}_i}{\hat{H}}\frac{a_{0,T_0}}{a_0}T_{0,j}+\\
&\frac{\hat{F}_j}{\hat{F}}\frac{a_{0,T_0}}{a_0}T_{0,i}+\frac{\hat{H}_j}{\hat{H}}\frac{a_{0,T_0}}{a_0}T_{0,i}+\frac{a_{0,T_0T_0}}{a_0}T_{0,i}T_{0,j}+\frac{a_{0,T_0}}{a_0}T_{0,ij}=\frac{D_{ij}}{D}.
\end{aligned} \tag{8.54}$$

We similarly get

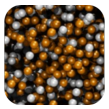
$$\begin{aligned}
&\frac{\hat{F}_{Ti}}{\hat{F}}+\frac{\hat{F}_i}{\hat{F}}\frac{a_{0,T_0}}{a_0}T_{0,T}+\\
&\frac{\hat{F}_T}{\hat{F}}\frac{\hat{H}_i}{\hat{H}}+\frac{\hat{H}_i}{\hat{H}}\frac{a_{0,T_0}}{a_0}T_{0,T}+\\
&\frac{\hat{F}_T}{\hat{F}}\frac{a_{0,T_0}}{a_0}T_{0,i}+\frac{a_{0,T_0T_0}}{a_0}T_{0,i}T_{0,T}+\frac{a_{0,T_0}}{a_0}T_{0,Ti}=\frac{D_{Ti}}{D},
\end{aligned} \tag{8.55}$$

and

$$\frac{\hat{F}_{TT}}{\hat{F}}+\frac{2\hat{F}_T}{\hat{F}}\frac{a_{0,T_0}}{a_0}T_{0,T}+\frac{a_{0,T_0T_0}}{a_0}(T_{0,T})^2+\frac{a_{0,T_0}}{a_0}T_{0,TT}=\frac{D_{TT}}{D}. \tag{8.56}$$

By substituting the expression for $T_{0,ij}$ into (8.54), we get

$$\begin{aligned}
&\frac{\hat{F}_{ij}}{\hat{F}}+\frac{\hat{F}_i}{\hat{F}}\frac{\hat{H}_j}{\hat{H}}+\frac{\hat{F}_i}{\hat{F}}\frac{a_{0,T_0}}{a_0}T_{0,j}+\\
&\frac{\hat{F}_j}{\hat{F}}\frac{\hat{H}_i}{\hat{H}}+\frac{\hat{H}_{ij}}{\hat{H}}+\frac{\hat{H}_i}{\hat{H}}\frac{a_{0,T_0}}{a_0}T_{0,j}+\\
&\frac{\hat{F}_j}{\hat{F}}\frac{a_{0,T_0}}{a_0}T_{0,i}+\frac{\hat{H}_j}{\hat{H}}\frac{a_{0,T_0}}{a_0}T_{0,i}+\frac{a_{0,T_0T_0}}{a_0}T_{0,i}T_{0,j}+\frac{a_{0,T_0}}{a_0}\left(-\frac{\hat{F}_{ij}}{\hat{F}}T_0-\frac{\hat{F}_i}{\hat{F}}T_{0,j}-\frac{\hat{F}_j}{\hat{F}}T_{0,i}\right)=\frac{D_{ij}}{D},
\end{aligned}$$



and thus

$$\begin{aligned}\frac{\hat{F}_{ij}}{\hat{F}} =& \frac{\frac{D_{ij}}{D} - \frac{\hat{F}_i}{\hat{F}}\frac{\hat{H}_j}{\hat{H}} - \frac{\hat{F}_j}{\hat{F}}\frac{\hat{H}_i}{\hat{H}} - \frac{\hat{H}_{ij}}{\hat{H}}}{1 - \frac{a_{0,T_0}}{a_0}T_0} \\ &+ \frac{-\frac{\hat{H}_j}{\hat{H}}\frac{a_{0,T_0}}{a_0}T_{0,i} - \frac{\hat{H}_i}{\hat{H}}\frac{a_{0,T_0}}{a_0}T_{0,j} - \frac{a_{0,T_0T_0}}{a_0}T_{0,i}T_{0,j}}{1 - \frac{a_{0,T_0}}{a_0}T_0}.\end{aligned} \tag{8.57}$$

We also get[6]

$$\frac{\hat{F}_{Ti}}{\hat{F}} = \frac{\frac{D_{iT}}{D} - \frac{\hat{F}_T}{\hat{F}}\frac{\hat{H}_i}{\hat{H}} - \frac{\hat{H}_i}{\hat{H}}\frac{a_{0,T_0}}{a_0}T_{0,T} - \frac{a_{0,T_0T_0}}{a_0}T_{0,i}T_{0,T} - \frac{a_{0,T_0}}{a_0\hat{F}}}{1 - \frac{a_{0,T_0}}{a_0}T_0}. \tag{8.58}$$

and

$$\frac{\hat{F}_{TT}}{\hat{F}} = \frac{\frac{D_{TT}}{D} - \frac{a_{0,T_0T_0}}{a_0}(T_{0,T})^2}{1 - \frac{a_{0,T_0}}{a_0}T_0}. \tag{8.59}$$

Relationship between P_0 and P

In general, we have

$$P = -\left(\frac{\partial A^r}{\partial V}\right)_{T,\mathbf{n}} + \frac{nRT}{V}.$$

Using that $A^r(T, V, \mathbf{n}) = \hat{F}M(T_0, v_0)$ we get

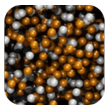
$$\begin{aligned}P &= -\partial_V\left(\hat{F}(T,\mathbf{n})M(T_0,v_0)\right) + \frac{RT}{v} \\ &= -\hat{F}M_{v_0}v_{0,V} + \frac{nR(\hat{F}T_0/n)}{\hat{H}v_0} \\ &= \frac{\hat{F}}{\hat{H}}\left(-M_{v_0} + \frac{RT_0}{v_0}\right) \\ &= \frac{\hat{F}}{\hat{H}}P_0.\end{aligned}$$

Expressing F in terms of M

Often we are more interested in the reduced residual Helmholtz energy $F(T, V, \mathbf{n}) = A^r(T, V, \mathbf{n})/RT$ than in $A^r(T, V, \mathbf{n})$ itself. First note that since $\hat{F} = nT/T_0$ we get that $A^r(T, V, \mathbf{n}) = \hat{F}M(T_0, v_0)$ is equivalent with

$$F(T, V, \mathbf{n}) = \frac{n}{RT_0}M(T_0, v_0).$$

[6] Michelsen [3] has an error in the expression for $\hat{F}_{Ti}/\hat{F}$: he is missing the last term in the numerator.



$$\left(\frac{\partial F}{\partial T}\right)_{V,\mathbf{n}} = -\frac{n}{RT_0^2}T_{0,T}M + \frac{n}{RT_0}M_T \tag{8.60}$$

$$\left(\frac{\partial F}{\partial V}\right)_{T,\mathbf{n}} = \frac{n}{RT_0}M_V \tag{8.61}$$

$$\left(\frac{\partial F}{\partial n_i}\right)_{T,V} = \left(\frac{1}{RT_0} - \frac{n}{RT_0^2}T_{0,i}\right)M + \frac{n}{RT_0}M_i \tag{8.62}$$

$$\left(\frac{\partial^2 F}{\partial V^2}\right)_{T,\mathbf{n}} = \frac{n}{RT_0}M_{VV} \tag{8.63}$$

$$\left(\frac{\partial^2 F}{\partial T \partial V}\right)_{\mathbf{n}} = -\frac{n}{RT_0^2}T_{0,T}M_V + \frac{n}{RT_0}M_{TV} \tag{8.64}$$

$$\left(\frac{\partial^2 F}{\partial V \partial n_i}\right)_{T} = \left(\frac{1}{RT_0} - \frac{n}{RT_0^2}T_{0,i}\right)M_V + \frac{n}{RT_0}M_{Vi} \tag{8.65}$$

$$\left(\frac{\partial^2 F}{\partial T^2}\right)_{V,\mathbf{n}} = \left(\frac{2n}{RT_0^3}T_{0,T}^2 - \frac{n}{RT_0^2}T_{0,TT}\right)M - \frac{2n}{RT_0^2}T_{0,T}M_T + \frac{n}{RT_0}M_{TT} \tag{8.66}$$

$$\begin{aligned}\left(\frac{\partial^2 F}{\partial n_i \partial n_j}\right)_{T,V} &= \left(-\frac{1}{RT_0^2}T_{0,j} - \frac{1}{RT_0^2}T_{0,i} + \frac{2n}{RT_0^3}T_{0,i}T_{0,j} - \frac{n}{RT_0^2}T_{0,ij}\right)M \\ &+ \left(\frac{1}{RT_0} - \frac{n}{RT_0^2}T_{0,i}\right)M_j + \left(\frac{1}{RT_0} - \frac{n}{RT_0^2}T_{0,j}\right)M_i + \frac{n}{RT_0}M_{ij}\end{aligned} \tag{8.67}$$

$$\begin{aligned}\left(\frac{\partial^2 F}{\partial T \partial n_i}\right)_{V} &= \left(-\frac{1}{RT_0^2}T_{0,T} + \frac{2n}{RT_0^3}T_{0,i}T_{0,T} - \frac{n}{RT_0^2}T_{0,Ti}\right)M \\ &+ \left(\frac{1}{RT_0} - \frac{n}{RT_0^2}T_{0,i}\right)M_T - \frac{n}{RT_0^2}T_{0,T}M_i + \frac{n}{RT_0}M_{Ti}\end{aligned} \tag{8.68}$$

9 Testing the SPUNG model

To validate the implementation of the SPUNG model, various tests have been performed.

9.1 Equivalence of cubic equations vs SPUNG with cubic reference equation

Using SPUNG with SRK as both shape factor equation and reference equation should be equivalent to using SRK directly. It is readily checked that this is indeed the for the ThermoPack SPUNG implementation.

9.2 Computing phase envelopes

We compare the performance of the ThermoPack SPUNG model against the TPlib SPUNG model, as well as experimental measurements. We use SRK as the shape equation and MBWR-32 as the reference equation The mixture we consider consists of CO_2 and N_2 at fixed temperature $T = 240\,\mathrm{K}$. The results are shown in Figure 8.

The reason why there are no TPlib points on the top of the graph is that the TPlib SPUNG model is not able to close the envelope. This demonstrates the improved robustness of the ThermoPack library compared to TPlib.

Since the model is the same in TPlib and ThermoPack, the computed points for TPlib should lie exactly on the computed curve for ThermoPack. This is clearly not the case in Figure 8. The reason for this is that in TPlib there is a database of interaction parameters k_{ij} that have been optimized for the SPUNG model. They are given for SRK, SRK-GD and PR, and for a range

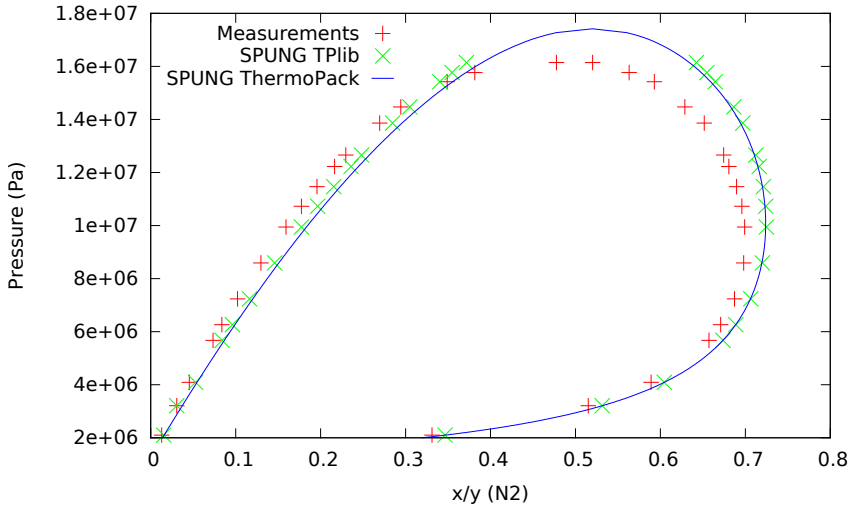


Figure 8: Computed and measured points on the phase envelope for the mixture $CO_2 - N_2$ at $T = 240\,\mathrm{K}$.

of mixtures. In Figure 9 we have plotted the same curve as in Figure 8, but using the optimized interaction parameters in ThermoPack. The TPlib points fit perfectly, except one point (the second

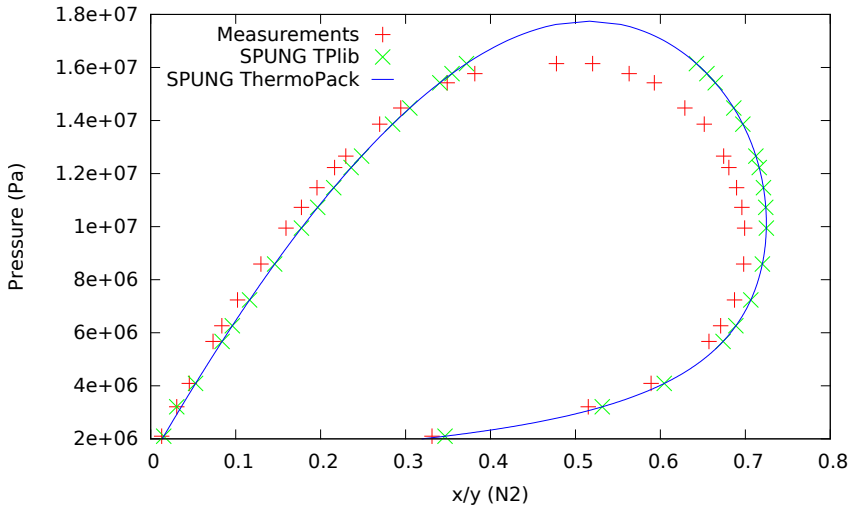


Figure 9: The $CO_2 - N_2$ phase envelope using optimized interaction parameters.

point from the left) which is an almost perfect match with with the experimental measurement. The reason for this one anomaly is unknown. Fortunately, we have experimental data also for CO_2-O_2, and Figures 10 and 11 show that the ThermoPack implementation is consistent with the TPlib implementation. How much the optimized interaction parameters actually improve the fit, is not known.

9.3 Comparing with density measurements

We compare density measurements for a mixture of 98 % CO_2 and 2 % CH_4 with three models: standard SRK, and SPUNG-SRK with CH_4 as reference component, and using respectively MBWR-19 and MBWR-32 as reference equation. The measurements are in the pressure range $2 \cdot 10^6 - -3.5 \cdot 10^7$ Pa, and the temperature range $225 - -350$ K. In Figure 12 we have plotted the deviation from the measurements (the line) and the results using SRK (the points). There are two outliers, which

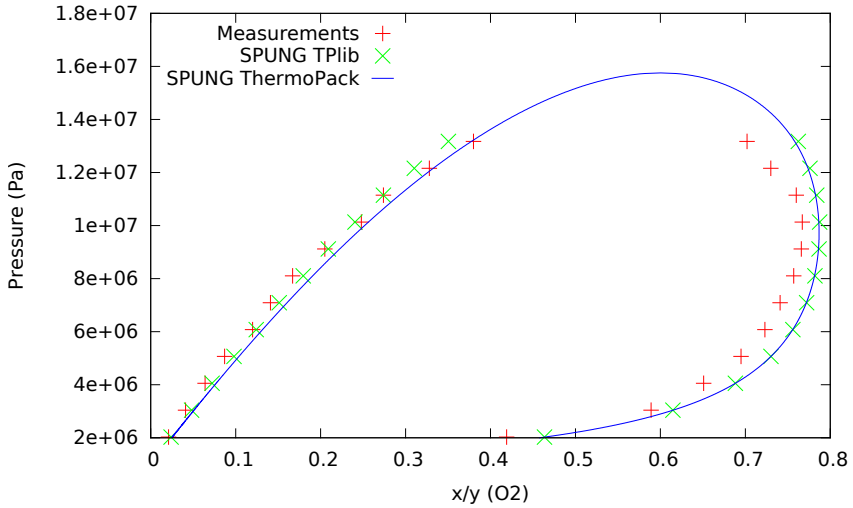


Figure 10: The $CO_2 - O_2$ phase envelope using regular interaction parameters.

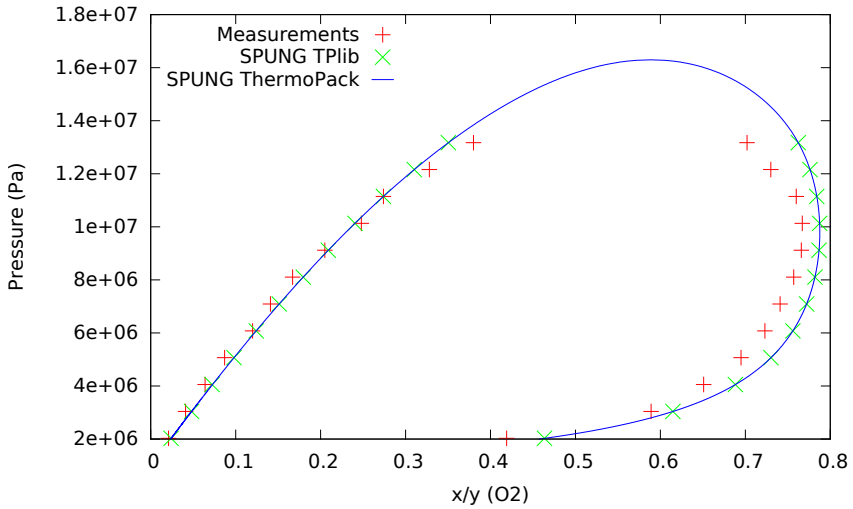


Figure 11: The $CO_2 - O_2$ phase envelope using optimized interaction parameters.

is probably due to the SRK density solver choosing the wrong phase. To avoid this from happening to the SPUNG calculations, the density solver which minimizes Gibbs energy was invoked when the density for the reference component was computed. The results for the SPUNG models are shown in Figure 13 and Figure 14.

It is clear that the SPUNG-MBWR models grossly outperforms SRK, and it seems like SPUNG-MBWR32 is slightly better than SPUNG-MBWR19, as is to be expected. To verify this, the absolute average (relative) deviation was computed for the datasets. The two outliers in the SRK computations were removed before the AAD was computed, as they can probably be remedied by choosing the most stable phase. The results are given in Table 4.

	SRK	SPUNG-MBWR19	SPUNG-MBWR32
AAD (%)	6.693	1.527	0.867

Table 4: Absolute average deviation from experimental density measurements.

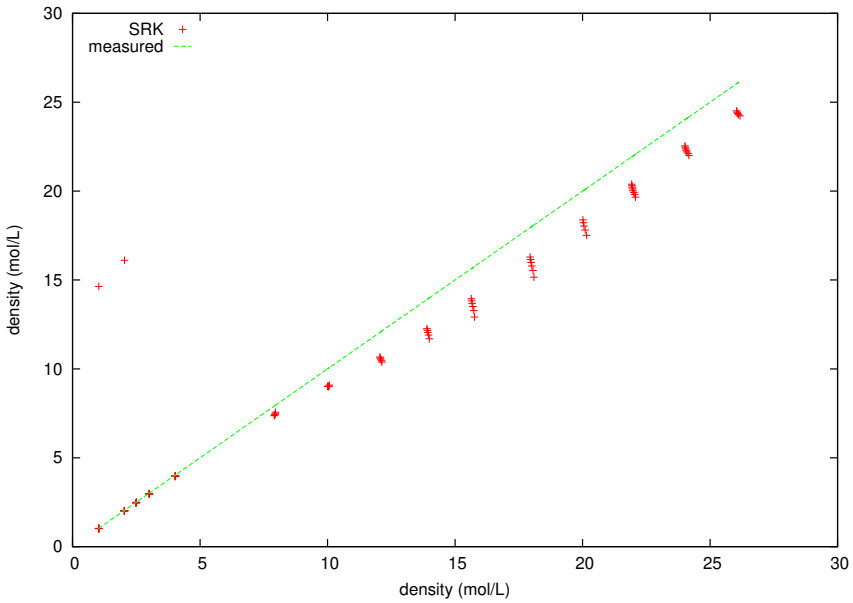


Figure 12: Comparing measured densites with SRK-densities. The vertical distance from point to line is the deviation measured in mol/L.

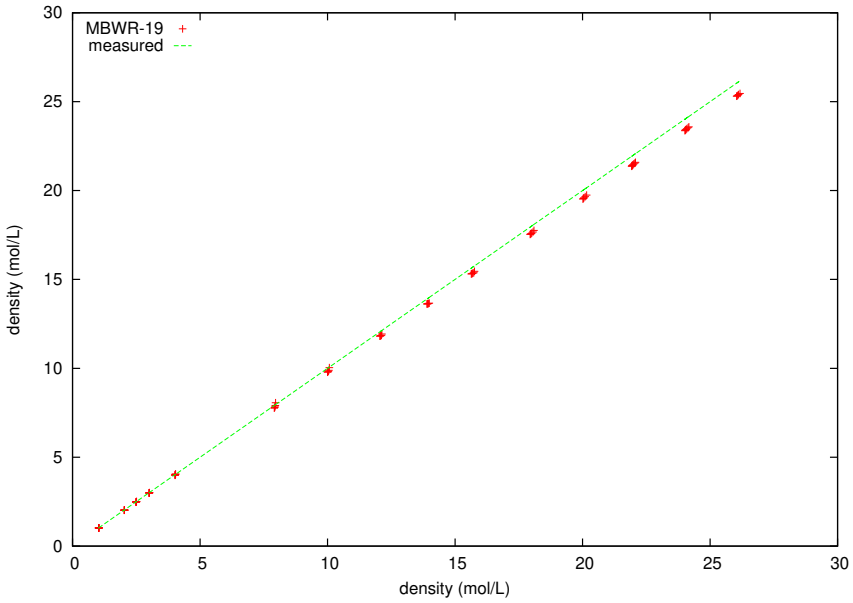


Figure 13: Comparing measured densites with SPUNG-MBWR19-densities. The vertical distance from point to line is the deviation measured in mol/L.

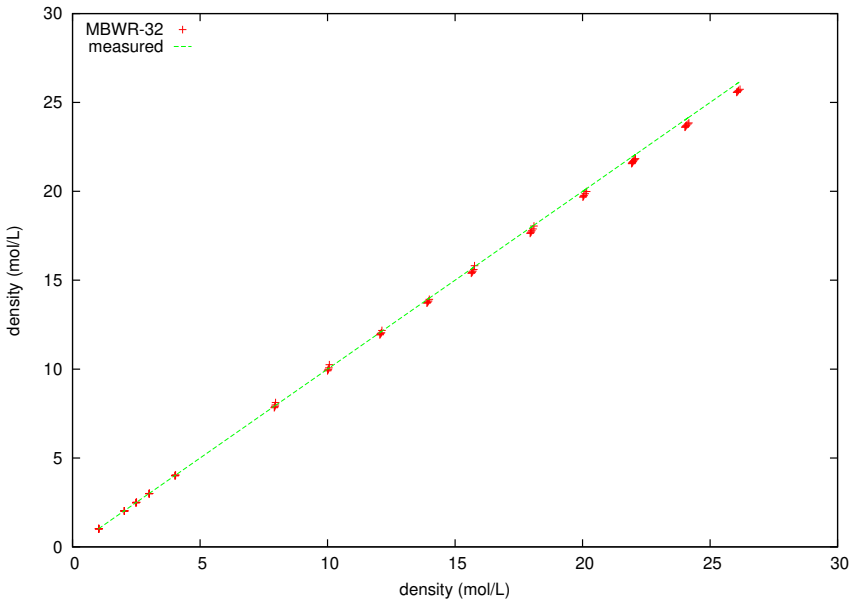
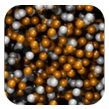


Figure 14: Comparing measured densites with SPUNG-MBWR32-densities. The vertical distance from point to line is the deviation measured in mol/L.



References

[1] Aarnes J.R. Implementation and testing of the Lee-Kesler equation of state in ThermoPack. SINTEF Energy Research internal memo, 2013.

[2] Jørstad, O. "Equation of state for hydrocarbon mixtures." Dr. Ing. dissertation, Trondheim 1993.

[3] Michelsen J.M. and Mollerup M.L. *Thermodynamic Models: Fundamentals and Computational Aspects, 2nd edition.* Tie-Line Publications, 2007.

[4] Poling B.E., Prausnitz J.M. and O'Connell J.P. *The Properties of Gases and Liquids, 5th edition.* McGraw-Hill, 2001.

[5] Polt, Axel. *Zur Beschreibung der thermodynamischen Eigenschaften reiner Fluide mit "Ertwerterten BWR-Gleichungen".* Dr. Ing. Dissertation, Kaiserslautern 1987.

[6] Press W.H, Teukolsky S.A, Vetterling W.T and Flannery B.P. *Numerical Recipes, The Art of Scientific Computing, 3rd edition.* Cambridge University Press, 2007.

[7] Skaugen G. Implementation of Cubic EOS – A note about the alpha-parameter formulation. SINTEF Energy Research internal memo, 2013.

[8] Span, Roland. *Multiparameter equations of state.* Springer, 2003.

[9] Wilhelmsen Ø., Skaugen G. and Hammer M. Flexible thermodynamic workbench for CCS thermodynamics - Update 2013. SINTEF Energy Research internal memo, 2013.