

OilMixProp 1.0: Package for Thermophysical Properties of Oils, Common Fluids, and Their Mixtures

Xiaoxian Yang* and Markus Richter

Chemnitz University of Technology, Applied Thermodynamics, 09107 Chemnitz, Germany

*E-mail: xiaoxian.yang@mb.tu-chemnitz.de

Abstract. This work presents the first version of OilMixProp (short for oil mixture properties), which could be the first software package capable of calculating all the core thermophysical properties of fluids involving user-defined oils. Properties needed in thermodynamic cycle analysis are all included: density, phase equilibria, heat capacity, entropy, enthalpy, speed of sound, viscosity, and thermal conductivity. Approximately 640 pure fluids are available, and users can define their oil by determining its fluid constants using embedded fitting tools (see the fundamentals developed in our previous work: Ind. Eng. Chem. Res. 2023, 62, 18736–18749). Analogously, but in some aspects better than *refpropm* (Matlab interface of REFPROP 10.0), a function *OilPropm* is developed as the only interface for calculating thermophysical properties. Various input combinations (e.g., temperature and pressure, pressure and enthalpy, temperature and vapor fraction, etc.) are available to cater to the needs of thermodynamic cycle analysis. Sophisticated outputs are designed to deliver either complete information (phases and properties in each phase) or selected information about a fluid at the given condition. The package is written in Matlab and is free for academic institutions; please contact the authors to acquire it.

1. Introduction

Oils are widely used in everyday life, e.g. for cooking (sunflower oil, olive oil, etc.) and for health care (fish oil, etc.) and are essential components of various industrial applications, e.g. as lubricants in automobiles, screw machines and in refrigeration technology. The oil products available on the market are usually composed of a base oil and a variety of additives and impurities. This makes modeling all essential thermophysical properties (including density, phase behavior, heat capacity, entropy, enthalpy, viscosity, and thermal conductivity) of an oil product very difficult, since conventional models for thermophysical properties, such as empirical multiparameter equations of state (EoS), can only be developed for pure fluids or mixtures with known components and composition. To tackle this challenge, a novel modeling approach was developed in our previous work [1] by treating an oil product as a quasi-pure fluid, setting up a simple set of equations for all the essential properties, and proposing a parameter-fitting procedure using a minimal set of experimental data (fewer than 20 and at least 12 data points). This approach can be easily extended for calculations of fluid mixtures, such as oils + refrigerants. For most properties, calculations using this approach generally agree with the experimental data within the same level of experimental uncertainty.

The basis of this modeling approach [1] is complicated as it involves the Patel-Teja-Valderrama (PTV) EoS [2,3] (more complicated than the famous Soave-Redlich-Kwong (SRK) [4] and Peng-Robinson (PR) [5] EoS) and the latest residual entropy scaling (RES) approaches developed by Yang et al. [6–9]. In addition, a robust flash algorithm for vapor-liquid equilibrium (VLE) is required for the calculation of gas solubility, which is an important property of mixtures containing oils. To facilitate the implementation and application of our modeling approach and to make the thermophysical properties of oils and their mixtures easily accessible to users, in this work, a software package, OilMixProp (short for oil mixture properties) was developed. Part of this effort is funded within subproject 3 of the KETEC (Research Platform Refrigeration and Energy Technology) project [10]. The package is written in Matlab and will be converted to other languages (e.g., Python, C++ and Julia) in the future.

In this paper, we present the first version of OilMixProp. The main functions of OilMixProp 1.0 include: (1) parameter fitting package to obtain fluid constants of an oil from experimental data, (2) calculations of thermophysical properties of oils, common fluids, and their mixtures with various input combinations (e.g. temperature and pressure T - p , pressure and enthalpy p - h , pressure and vapor fraction p - Q , etc), and (3) plot various phase diagrams in VLE conditions, like pressure vs. composition, p - x . The primary feature of OilMixProp 1.0 is that all the core thermophysical properties of oils and their mixtures with other fluids can be calculated, which is not available in many other existing thermophysical property software packages, even in those commonly used reference ones (such as REFPROP 10.0 [11], TREND 5.0 [12] and CoolProp 6.4.1 [13]). Furthermore, when oils are involved, the fluid mixture is typically in a two-phase region: an oil-rich liquid phase dissolved with other components and a gas phase with a trace amount of oil. To study such systems well, a robust VLE flash algorithm was developed as the core of OilMixProp. Last but not least, a sophisticated design of inputs and outputs is implemented. Various input combinations are available, which are analogous to the function *refpropm*, the Matlab interface of REFPROP 10.0 [11]. The output is either a class (a datatype in Matlab) of a fluid with all information available or a specified property at each phase. The output information could include compositions and all the core thermophysical properties in each phase, as well as the phase fraction. Such an output design is superior to *refpropm*, which might only output a single density value (when density is the requested output), while according to the inputs, the fluid is obviously in the two-phase region (see more details in Section 4.1).

Finally, it is important to mention that there are about 640 pure fluids, unlimited user-defined oils, and their blends available in OilMixProp 1.0. OilMixProp does not aim to become a reference software package for the calculation of thermophysical properties, such as REFPROP 10.0 [11], which is focused on high accuracy. OilMixProp aims to be a reliable solution when the fluids to be analyzed (e.g. fluids containing oils) are not available in reference software packages. Compared to REFPROP 10.0, OilMixProp 1.0 has a higher uncertainty for pure fluids and in single phases. However, when it comes to the VLE calculation of mixtures, OilMixProp 1.0 could be as accurate as REFPROP 10.0 and even provide more reliable results in some situations (see Section 4.1).

2. Fundamentals

2.1 Models and fluid constants

Thermophysical properties calculated with OilMixProp mainly include (but not only include): density, phase behavior, heat capacity, entropy, enthalpy, speed of sound, viscosity and thermal conductivity. These properties are typically required in the analysis of thermodynamic cycles. An

overview of the properties and the used calculation models are given in Figure 1. In total seven Cubic EoS [PTV [2,3], SRK [4], PR [5], Yang-Frotscher-Richter (YFR) [14], Peng-Robinson-Stryjek-Vera (PRSV) [15], Wilson-Redlich-Kwong (WRK) [16] and Redlich-Kwong (RK)[17]] are available for calculations of density, phase behavior and the residual properties. Here YFR EoS [14] is a new cubic EoS recently developed within our group aided with symbolic regression [18]; it has better accuracy in liquid phase property calculation. Due to the lack of parameters, PRSV is not recommended, and due to the lack of accuracy, WRK and RK should not be used. With an additional equation for the ideal gas isobaric heat capacity as a linear function of temperature [linear- $c_p^o(T)$], heat capacities, entropy, enthalpy and speed of sound can be obtained. Adopting the latest RES approach developed by Yang et al. [6–9], viscosity and thermal conductivity can be calculated. The fundamentals of these properties' calculation from this model set [cubic EoS + linear $c_p^o(T)$ + RES] has already been given in the supporting information of our previous paper [1]. For mixtures, the van der Waals mixing rule is used in Cubic EoS and the mixing rules developed by Yang et al.[8,9] are used for both viscosity and thermal conductivity.

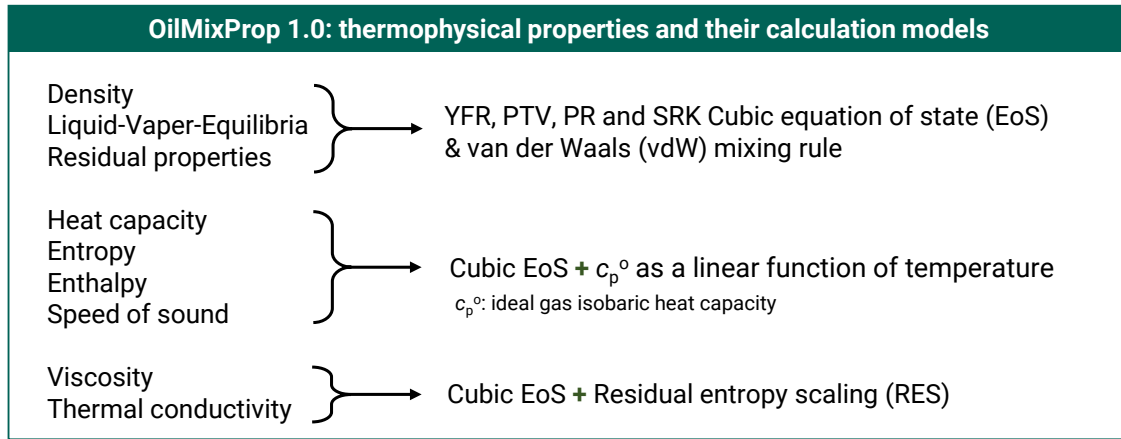


Figure 1. Available key thermophysical properties in OilMixProp 1.0 and the calculation models.

With the model set of cubic EoS + linear- $c_p^o(T)$ + RES, fluid characteristic constants of a pure fluid or a quasi-pure oil that are needed include: molar mass M , critical temperature T_c , critical density ρ_c , critical pressure p_c , acentric factor ω , two parameters k_0, k_1 in the linear- $c_p^o(T)$, Lennard–Jones (L-J) pair potential parameters ε/k_B and σ , RES fitted parameters for viscosity $n_{\mu k}$ ($k = 1,2,3,4$) and for thermal conductivity $n_{\lambda k}$ ($k = 1,2,3,4$), and three parameters for the critical enhancement of thermal conductivity (φ_0, Γ, q_D). For the 151 pure fluids (see Classes/Fluid_Constants.txt in the package) available in REFPROP 10.0 [11], the fluid characteristic constants are obtained by fitting to calculations of REFPROP (k_0 and k_1), from Yang et al's RES approach[6–9] ($n_{\mu k}$ and $n_{\lambda k}$), or from REFPROP directly (all other constants). For remaining pure fluids, the constants M, T_c, ρ_c, p_c , and ω are obtained from various sources mainly from DWSIM, $n_{\mu k}$ are from Yang et al's RES approach[6–9], and the remaining properties are constants being the average of those 151 fluids in REFPROP. Therefore, one should expect a lower accuracy when calculating properties of these fluids. For an oil product, the determination or estimation methods for these fluid characteristic constants have been given in our previous work [1]. In brief, Raoult's law of boiling point elevation (RBPE) is used for determining the average M , the modified Rackett equation (mRE) for critical point information (T_c, p_c and ρ_c), PTV or YFR EoS for ω , linear- $c_p^o(T)$ for k_0 and k_1 , RES approaches of Yang et al.[6–

9] for $n_{\mu k}$ and $n_{\lambda k}$ ($k = 1, 2, 3, 4$). Parameters ε/k_B , σ , φ_0 , Γ and q_D have negligible impact on the result and are set as constants. An overview of these constants and their relations with the models are given in Figure 2.

OilMixProp 1.0: fluid constants and the fitting models	
<u>Raoult's law of boiling point elevation (RBPE)</u> <ul style="list-style-type: none"> Molar mass M 	<u>Estimations</u> <ul style="list-style-type: none"> Lennard-Jones parameters ε/k_B Lennard-Jones parameters σ $\varepsilon/k_B = T_c/1.2593 \quad \sigma^3 = 0.3189/\rho_c$ for viscosity and thermal conductivity at dilute gas limit.
<u>Modified Rackett equation (mRE)</u> <ul style="list-style-type: none"> compressibility factor Z_c temperature T_c density ρ_c pressure p_c 	<u>Residual entropy scaling (RES) of Yang et al</u> <ul style="list-style-type: none"> viscosity $n_{\mu k}$ ($k = 1, 2, 3, 4$) thermal conductivity $n_{\lambda k}$ ($k = 1, 2, 3, 4$) for residual viscosity and thermal conductivity
<u>PTV or YR cubic Equation of State (EoS)</u> <ul style="list-style-type: none"> Acentric factor ω 	<u>Constant values</u> <ul style="list-style-type: none"> φ_0, Γ, q_D for the critical enhancement of thermal conductivity
<u>c_p° as linear function of T</u> <ul style="list-style-type: none"> k_0 and k_1 in the linear-$c_p^\circ(T)$ <small>c_p°: ideal gas isobaric heat capacity</small> 	

Figure 2. Fluid constants of a quasi-pure oil in OilMixProp 1.0, and their determination methods.

2.2 Vapor-liquid equilibria

Although the fundamentals of VLE calculation are available in literature [19], to explain the self-developed VLE flash algorithm as clearly as possible in Section 3.1, the generalized fundamentals of phase equilibria are described here briefly. Assuming a mixture with n components stays in equilibrium in m phases, the relations among temperature T , pressure p , overall composition $\mathbf{Z} = [z_1, z_2, \dots, z_n]$ where z_i is mole fraction of component i , compositions $\mathbf{Z}_j = [z_{j1}, z_{j2}, \dots, z_{jn}]$ in the j^{th} phase, phase fraction $\mathbf{Q} = [q_1, q_2, \dots, q_m]$, fugacity coefficients φ (an $n \times m$ matrix, with φ_{ji} being that of the i^{th} component in the j^{th} phase) are:

$$\sum_{i=1}^n z_i = 1 \quad (1)$$

$$\sum_{i=1}^n z_{ji} = 1, (j = 1, 2, \dots, m) \quad (2)$$

$$\sum_{j=1}^m q_j = 1 \quad (3)$$

$$\sum_{j=1}^m z_{ji} \cdot q_j = z_i, (i = 1, 2, \dots, n) \quad (4)$$

$$\varphi_{ji} = \text{func}(T, p, \mathbf{Z}_j), (i = 1, 2, \dots, n, \text{ and } j = 1, 2, \dots, m) \quad (5)$$

$$\varphi_{ji} \cdot z_{ji} = \varphi_{ki} \cdot z_{ki} (i = 1, 2, \dots, n; k \text{ and } j = 1, 2, \dots, m, \text{ but } k \neq j) \quad (6)$$

Here, Eq. (1) is not an independent equation as it can be derived from Eqs. (2) to (4). Eqs. (5) require additional fluid constants which are typically known. Therefore, there are in total $2+n+m+m \times n \times 2$ variables with $1+n+m+m \times n + (m-1) \times n$ equations; and with $n+1$ variables given, equilibrium conditions could be solved. The most typical scenario is that, the known variables are temperature T , pressure p and overall composition \mathbf{Z} , which happen to be $n+1$ variables (only $n-1$ independent variables in \mathbf{Z}). This can be solved by the VLE flash algorithm to be discussed in Section 3.1. Another typical scenario is to calculate the phase transition

temperature (or pressure), with overall composition \mathbf{Z} , pressure (or temperature) known and one of the phase fractions q_j equals 0. This corresponds to the bubble-point and dew-point calculations in VLE.

3. Algorithms

The calculations of most of the thermophysical properties are pretty straightforward according to models described in section 2.1. Here, in this section, we only describe important calculations that require iterations: the self-developed flash algorithm for VLE calculation, the phase transition point (bubble-point and dew-point) determination, and the calculation with one input existing in the two-phase region.

3.1 Flash algorithm

For VLE conditions, there exist only 2 phases ($m = 2$), therefore, the generalized phase equilibria described in section 2.2 can be specified and simplified as follows. Some variables can be specified as: $\mathbf{X} = [x_1, x_2, \dots, x_n]$ and $\mathbf{Y} = [y_1, y_2, \dots, y_n]$ are molar fractions of components in the liquid and vapor phases, respectively; $\boldsymbol{\varphi}_L = [\varphi_{L1}, \varphi_{L2}, \dots, \varphi_{Ln}]$ and $\boldsymbol{\varphi}_V = [\varphi_{V1}, \varphi_{V2}, \dots, \varphi_{Vn}]$ are fugacity coefficients of components in the liquid and vapor phases, respectively; and Q is the vapor fraction. New variables $\mathbf{K} = [K_1, K_2, \dots, K_n]$ are defined:

$$K_i = y_i/x_i, (i = 1, 2, \dots, n) \quad (7)$$

as typically seen in literature [19]. The Rachford-Rice equation [20]:

$$\sum_{i=1}^n x_i - y_i = 0 \quad (8)$$

is adopted to ensure that Eqs. (2) are all satisfied. Then Eq. (7) is applied to Eqs. (4) and to Eqs. (6) respectively to yield

$$x_i = \frac{z_i}{1 + Q(K_i - 1)}, (i = 1, 2, \dots, n) \quad (9)$$

$$y_i = K_i x_i, (i = 1, 2, \dots, n) \quad (10)$$

$$K_i = \varphi_{Li}/\varphi_{Vi}, (i = 1, 2, \dots, n) \quad (11)$$

With these new variables and equations, and with T, p and \mathbf{Z} known, the VLE condition (\mathbf{X}, \mathbf{Y} and Q) can be solved using the self-developed flash algorithm, which is illustrated in Figure 3 and described as follows. (I) At given T, p and \mathbf{Z} , with an initial guess of \mathbf{K} according to Wilson's approximation

$$K_i = \frac{p_{c,i}}{p} \exp[5.373(1 + \omega_i)(1 - \frac{T_{c,i}}{T})], (i = 1, 2, \dots, n) \quad (12)$$

Q can be solved according to Eqs. (8) to (10) using the Newton-Raphson (N-R) method and meanwhile \mathbf{X} and \mathbf{Y} are calculated. (II) With \mathbf{X} and \mathbf{Y} known and according to Eq. (5), $\boldsymbol{\varphi}_L$ and $\boldsymbol{\varphi}_V$ can be obtained, and new \mathbf{K} is calculated using Eq. (12). (III) iterating the Q and \mathbf{K} calculations until the converge condition is met.

To improve the robustness of the algorithm, a trick was implemented in the flash algorithm: five different initial guesses (0.001, 0.1, 0.5, 0.9 and 0.999) of Q needed for the N-R method are used. This trick, when convergence fails with all initial guesses of Q , also provides useful and reliable information to judge if the fluid is in a homogeneous liquid or vapor phase: If more iterated Q (the values Q_e in Figure 3) satisfies $Q \geq 1$ than $Q \leq 0$, the fluid is in the vapor phase, otherwise in the liquid phase. There are also tricks in improving calculation efficiency: e.g., once

the looping Q deviates further away from the VLE range ($0 \leq Q \leq 1$), the Q (Q_0 in Figure 3) jumps to the next initial guess.

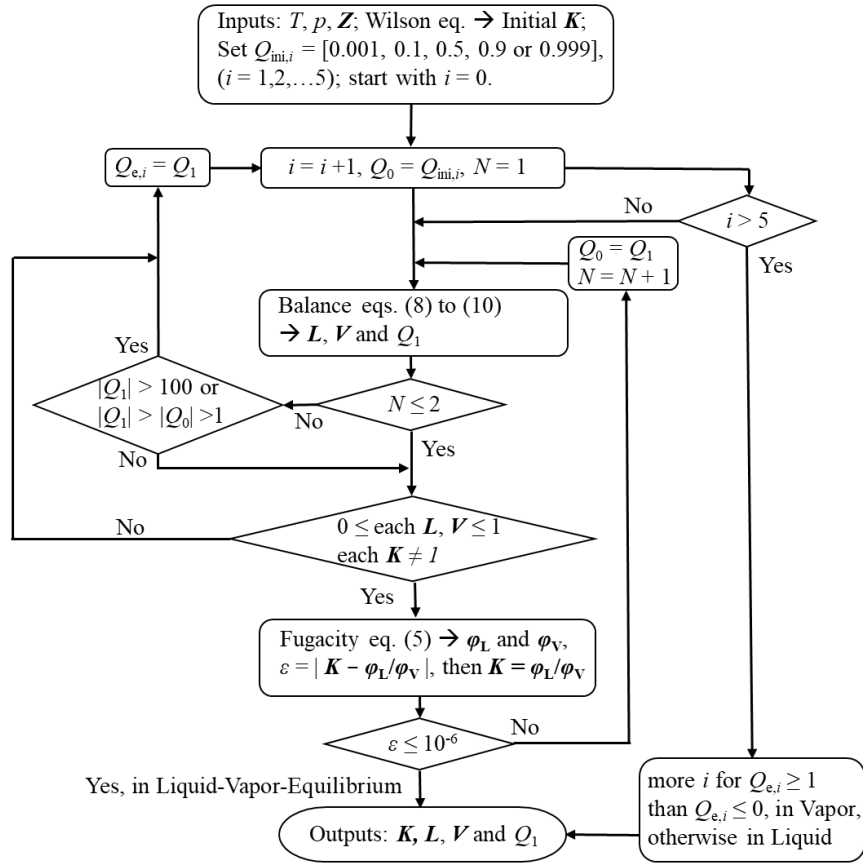


Figure 3. The vapor- liquid-equilibrium flash algorithm implemented in OilMixProp 1.0.

As a result, this algorithm is reliable and robust for almost all the pure fluids and mixtures tested. One unresolved issue lies in the phase identification when the mixture is in the supercritical region: so far the output is the liquid phase in the denser region and the vapor phase vice versa. This issue can be solved by adding the function of calculating mixture critical point, which will be part of the future OilMixProp 2.0.

3.2 Other algorithms

Except for the flash algorithm, there are also other calculations that require iterations, i.e., phase transition point (bubble-point and dew-point) determination, and VLE calculation with one input only exists in the two-phase region.

In phase transition point calculation, Q equals 1 or 0, which makes K the only values that need to be solved iteratively (compared to the flash algorithm, see Figure 3). It seems easier but since one of the T and p is unknown, an initial guess of T or p is always needed and should be reliable otherwise the iteration will fail or converge to a wrong value. Except for suggesting users try to provide an initial guess as accurate as possible, an algorithm is implemented in OilMixProp 1.0 to guarantee convergence to the right value successfully. For example, to calculate the dew-point temperature at a known pressure, the algorithm will first determine the phase condition (V for vapor, L for liquid, and VLE for vapor-liquid equilibrium) using the flash algorithm at the guessed temperature given by the user. If, for example, the fluid is in VLE

condition, 5 K will be added to the guessed temperature and the phase condition is determined again; this step will go on until reaching the vapor phase. Then an initial guess of temperature being the average of the last two temperatures is used for the dew point temperature calculation according to Eqs. (5) and (8) to (11). If no initial guess is given by the users, valid initial guesses, such as the molar average of the transition point of each pure component, will be used.

There are various input combinations available in OilMixProp 1.0 (see all in section 4.2). Sometimes it is necessary to calculate fluid mixture properties with one of the input values only existing in a two-phase region. For example, in heat pump or refrigerant cycle analysis, at the outlet of the valve, the known variables are pressure and enthalpy and the fluid is in a two-phase region. This is a scenario of calculating temperature from pressure and enthalpy. The algorithm will solve this function

$$h_{\text{input}} = Q * h_V + (1 - Q) * h_L \quad (13)$$

using the N-R method with an initial guess of temperature either provided by the user or being those implemented (e.g., a molar average of pure component solutions) in the algorithm. In Eq. (13), h_{input} is the molar enthalpy being the input value, h_V and h_L are molar enthalpies in the vapor and liquid phase, respectively. The values of h_V , h_L and Q are obtained using the flash algorithm discussed in section 3.1. If the fluid is in a single phase, $Q = 1$ or 0 .

4. Functions and examples

The core of OilMixProp 1.0 is to calculate the thermophysical properties of fluids, and this will be described in section 4.1. To determine fluid constants of a custom-defined oil so that this oil and its mixture can also be calculated, fitting tools are developed, see section 4.2. Last but not least, visualization tools are implemented so that important phase diagrams can be plotted, see section 4.3.

4.1 Property calculation function OilPropm

Analogical to the function *refpropm*, which is the Matlab interface of REFPROP 10.0 [11], a function *OilPropm* in OilMixProp 1.0 was developed as the ‘only’ interface that users need to call. Examples of using this function are given in FluidCalc.m in the package. An overview of the function *OilPropm* is given in Figure 4.

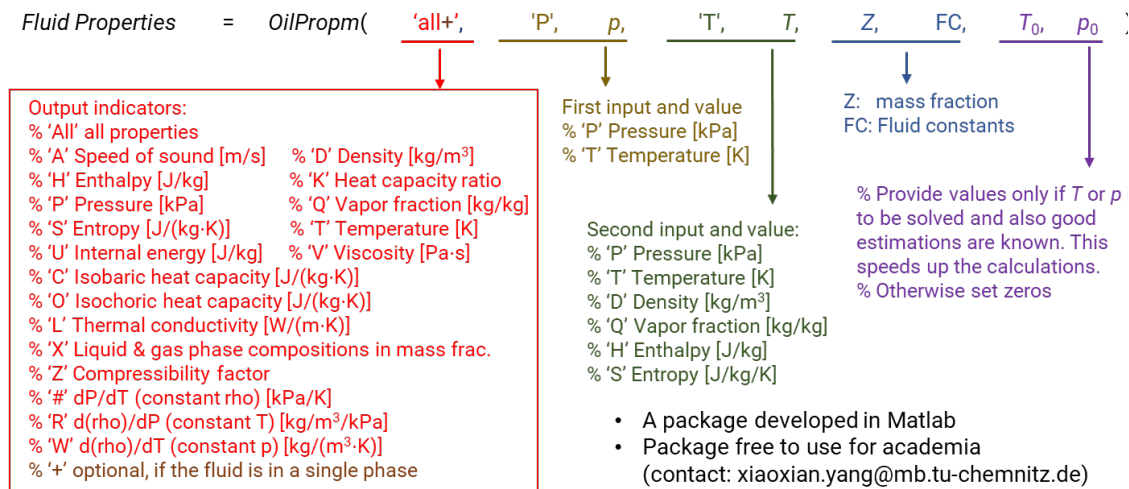


Figure 4. Overview of the function *OilPropm* in OilMixProp 1.0.

As shown in Figure 4, in total eight input values are needed to call *OilPropm*:

- (1) The output indicator. Except for specifying a property to calculate, *OilPropm* could also deliver a class (a datatype in Matlab, using output indicator 'all') that contains all information, compositions and all the core thermophysical properties in each phase as well as the phase fraction, of the fluid at the defined condition.
- (2) First input property: indicator and its value. So far only 'P' for pressure in kPa and 'T' for temperature in K are available. For any calculations, at least one of these two variables has to be known.
- (3) Second input property: indicator and its value. Six indicators are available: 'P', 'T', 'D' for density in $\text{kg}\cdot\text{m}^{-3}$, 'Q' for vapor fraction in mass basis, 'H' for enthalpy in $\text{J}\cdot\text{kg}^{-1}$ and 'S' for entropy in $\text{J}\cdot\text{kg}^{-1}\cdot\text{K}^{-1}$. Together with the first indicator, these combinations should cover all the needs in typical thermodynamic cycle analysis.
- (4) Fluid and model information: Z and FC. Here Z is the overall composition (all phases included) in mass fraction and is an array in column rather than in line, e.g., $Z = [0.3, 0.7]'$ rather than $[0.3, 0.7]$. FC is a class (a datatype in Matlab) containing all fluid constants and model information. A typical example of generating FC of the R1233zde + Emkarate RL32 mixture using PTV EoS is given as follows

```
Fluids = {'R1233zde','Emkarate RL32'};  
ALLEOS = {'PR','SRK','PTV','YFR'}; CubicEOS = ALLEOS{3};  
FC = GetGlobals(CubicEOS, Fluids);
```

This design is different from *refpropm* whose inputs are names of fluids. Every time *refpropm* is called, fluid constants have to be read once; this wastes time if the same fluids are calculated several times or in loops.

- (5) Initial guesses of temperature T_0 or pressure p_0 if they are unknown. They could be set to zero; however, we strongly recommend users provide guesses as accurate as possible, because this will significantly speed up the calculation, and avoid failed convergence or convergence at a wrong value. Here is the reason. If one of the T or p is unknown, the *OilMixProp* will first determine this unknown T or p by adopting the user's guess as the initial value for the equation solving using the N-R method. If the equation-solving fails or yields a wrong value, various initial guesses (e.g., molar average of pure component solutions) for the N-R method will be applied until the right solution is found. In the thermodynamic cycle analysis, such good guesses are always available, e.g., based on the properties of the previous state.

For the output, *refpropm* (Matlab interface of REFPROP 10.0) indicates it specifically, e.g., 'D' for density and the output is usually a single value. This might mislead users who have no sophisticated knowledge about thermophysics and the mixtures under study. For example call *refpropm* at an VLE condition:

```
refpropm('D','T',303.15,'P',1e3,'water','nitrogen',[0.5 0.5])
```

but the output is a single value of $21.924 \text{ kg}\cdot\text{m}^{-3}$ without indicating this value is in the liquid phase, vapor phase or overall phases. With *OilMixProp* 1.0, such a problem could be avoided. For the above mixture calculation (water and nitrogen mixture with an equal mass fraction), the output (with indicator 'all') of *OilPropm* using the YFR EoS is illustrated in Figure 5. From this comprehensive output, users will see that the liquid phase is almost pure water with a trace amount of nitrogen dissolved, and the gas phase is mainly nitrogen with little amount of water. As for the result and accuracy, the vapor phase density ($11.113 \text{ kg}\cdot\text{m}^{-3}$, see Figure 5) is very close to that ($11.129 \text{ kg}\cdot\text{m}^{-3}$) of pure nitrogen at the same T and p condition calculated with *refpropm*, and the liquid phase density ($979.698 \text{ kg}\cdot\text{m}^{-3}$, see Figure 5) agree within 2 % of pure water at

the same T and p condition calculated with *refpropm*. The density output ($21.924 \text{ kg}\cdot\text{m}^{-3}$) of *refpropm* for this mixture is likely the value of the mixture considering all phases.

```

0.50 Water + 0.50 Nitrogen in mass fraction with YFR EoS
T:          303.150000    K
P:          1000.000000   kPa
MM:         21.929       g/mol
Phase:      LV

Phase behavior in mass fraction. Vapor Fraction: 0.501418
      Water      Nitrogen
Liquid:  0.999999  0.000001
Vapor:   0.002829  0.997171

---- Properties in each phase ----
      Liquid      Vapor
Z:       0.007296  0.998578
MM:      18.015    27.970   g/mol
rho:     979.698   11.113   kg/m3
Entropy: -3.165    1.516   kJ/K/kg
Enthalpy: -2393.055 28.871   kJ/kg
cp:      4.166     1.061   kJ/K/kg
cv:      2.961     0.749   kJ/K/kg
SoS:     4513.070  356.844   m/s
vis:     0.785165  0.017813 mPa s
TC:      0.687892  0.027020 W/m/K
Kappa    1.407187   1.416654
d(rho)/dP: 0.000094 0.011125 kg/m3/kPa
d(rho)/dT: -0.771048 -0.037613 kg/m3/K
dp/dT_rho: 6366.504562 3.380864 kPa/K

-- Properties with all phases combined --
rho_all: 21.915      kg/m3
Entropy_all: -0.818   kJ/K/kg
Enthalpy_all: -1178.658 kJ/kg
cp_all: 2.609        kJ/K/kg
cv_all: 1.852        kJ/K/kg
Kappa_all 1.409107

```

Figure 5. An example result of *OilPropm* using output indicator ‘all’.

Such output design of *OilPropm* also significantly reduces the computational time if multiple properties at a defined condition are needed (typically scenarios in thermodynamic cycle analysis) as *OilPropm* only needs to be called once. The order of properties to be calculated is: (1) phase and volumetric information (density, volume, or compression factor) in each phase using the VLE flash algorithm, (2) caloric properties (heat capacities, entropy, enthalpy and speed of sound) in each phase, and (3) transport properties (viscosity and thermal conductivity) in each phase. If one of the T and p is unknown, T or p should be determined in the very beginning. Here, the time-consuming part is the first step (VLE flash calculation), while the last two steps are very quick as no iterations are needed. If users know for sure the fluid is in a single phase, adding the optional output indicator ‘+’ will skip the VLE flash calculation and thus significantly reduce the calculation time. Also, if only a single property is of interest, the user could specify this output property, e.g., ‘S’ for entropy, to speed up calculation; this is the same output design as *refpropm*.

4.2 Fitting tools for oil constants

There are thousands of oil products available in the market. OilMixProp 1.0 offers fitting tools to determine the fluid constants of any oil, so that the thermophysical properties of this oil and its mixtures with other fluids can be calculated in OilMixProp 1.0. The required experiment and experimental data are described in our previous work [1] and summarized in Figure 6.

OilMixProp 1.0: fluid constants and the needed experiments
<p>Less than 20 (at least 12) experimental points are needed → all constants of an oil</p> <p>4 (0) → two masses and two boiling point temperatures → M</p> <p>0 (0) → Z_c fixed at a reasonable value (e.g., 0.2563) → Z_c</p> <p>4 (2) → four, could be down to two, points of $(p = 1 \text{ atm}, T, \rho)$ → T_c, p_c, and ω</p> <p>4 (2) → four, could be down to two, points of $(p = 1 \text{ atm}, T, c_p)$ → k_0 and k_1</p> <p>4 (4) → four points of $(p = 1 \text{ atm}, T, \mu)$ → $n_{\mu k} (k = 1,2,3,4)$</p> <p>4 (4) → four points of $(p = 1 \text{ atm}, T, \lambda)$ → $n_{\lambda k} (k = 1,2,3,4)$</p> <p>Note: Temperature suggested to be (278.15 to 368.15) K or smaller Pressure can be another value rather than 1 atm and does not have to be constant.</p>

Figure 6. Fluid constants in OilMixProp 1.0 and the needed experiments to fit them of oils.

Aided with Figure 6, the needed experimental points for fitting the fluid constants of an oil in OilMixProp 1.0 can be described as follows.

- (1) Optionally, two mass and two boiling point temperature measurements are needed to determine the average M . If data are not available, M is a defaulted value of $200 \text{ g}\cdot\text{mol}^{-1}$, or users can provide a better estimation.
- (2) Z_c is fixed at 0.2563, or users can provide a better estimation.
- (3) Mandatorily, at least two sets of temperature, pressure and density (T, p, ρ) data are needed to fit T_c, p_c , and ω .
- (4) Optionally, at least two heat capacity (T, p, c_p) data sets are needed to fit k_0 and k_1 ; if data are not available, k_0 and k_1 are defaulted values being average of those of the 151 fluids in REFPROP 10.0.
- (5) Optionally, at least four viscosity (T, p, μ) data sets are needed to fit $n_{\mu k} (k = 1,2,3,4)$; if data are not available, $n_{\mu k}$ are defaulted values being global parameters of group 6 according to Yang et al. [8,21].
- (6) Optionally, at least four thermal conductivity (T, p, λ) are needed to fit $n_{\lambda k} (k = 1,2,3,4)$; if data are not available, $n_{\lambda k}$ are defaulted values being global parameters of group 6 according to Li et al. [22].
- (7) Optionally, for a binary system, one data point of mixture bubble-point pressure is needed to determine the binary interaction parameter (BIP) in the vdW mixing rule; otherwise, $\text{BIP} = 0$.

As a result, with a minimal amount of two (T, p, ρ) data sets, an oil could be fitted and implemented into OilMixProp 1.0. This is of course not recommended as the accuracy of other properties will be very low. We recommend collecting data (either from experiment or literature) as many as possible for each needed property (see Figure 6) and carrying out the fit. In OilMixProp 1.0, users can choose a few data for fitting and the remaining for confirming the fitting. An example result is given in Figure 7, where only data in full symbols are used for fitting.

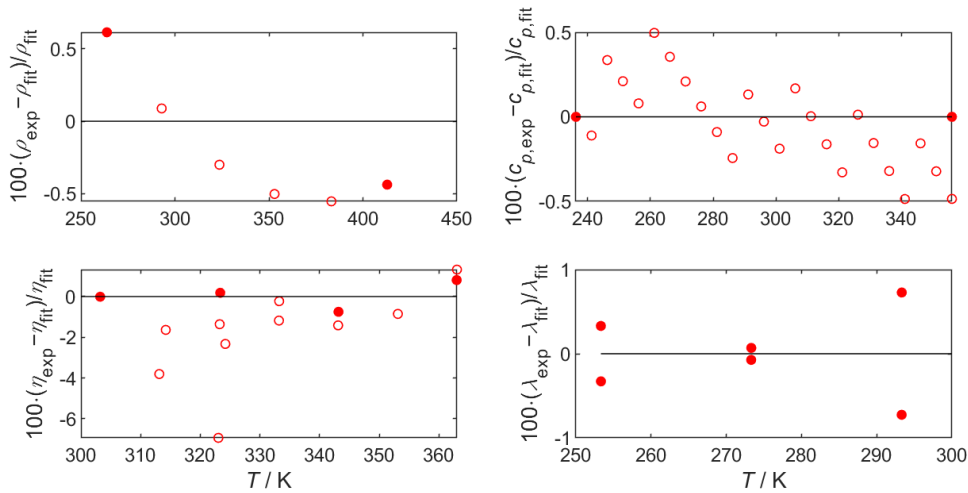


Figure 7. Example of the fluid constants fitting of PAG68: relative deviation of the experimental data ¹ from the model predictions. Only data in full symbol are used in fitting fluid constants.

OilMixProp 1.0 has included many examples to fit oil constants and the BIP from experimental data. The code for fitting oil constants is `Fitter_PureOil.m` and that for BIP is `Fitter_CKij.m`. The experimental data of these examples are given in the folder 'ExpData'. It is important to follow the examples to prepare the experimental data files, which are of course very easy to understand and prepare.

4.3 Phase diagrams

When studying a fluid mixture with unfamiliar components, such as oils, it is important to visualize its phase behavior by plotting phase diagrams of pressure vs. composition (p - x), temperature vs. composition (T - x), or pressure vs. temperature (p - T), etc. In OilMixProp 1.0, the p - x diagram at constant temperature is calculated by running a flash calculation at each pressure point, and the T - x diagram is calculated similarly. For potentially azeotrope mixture, calculating dew-point and bubble-point pressures (or temperatures) at each composition are also implemented to complete the whole phase diagram. Examples of a p - x diagram of a zeotrope mixture of propane + dodecane at 419.15 K and a T - x diagram of an azeotrope mixture of carbon dioxide + ethane mixture at 3.0 MPa are given in Figure 8 and Figure 9, respectively. Many tricks, e.g., well-defined pressure (or temperature) points, are adopted to optimize the calculation speed while providing the completed phase diagram. More details can be seen from the code `PhaseDiag_px.m`, and users are encouraged to modify and improve the plotting code by themselves.

As for p - T phase diagrams, OilMixProp 1.0 handles generally well for oil + other fluid mixtures. For example, see Figure 10, even the extreme asymmetric mixture of nitrogen + Emkarate RL32 at a nitrogen mole fraction of 0.3 can be plotted well. However, it is noted that, oils generally decompose at higher temperatures, therefore, the dew point line in Figure 10 in the high-temperature range might not exist. Similarly, in the low-temperature range, where components might solidify, the low-temperature part of the phase boundary might not be true. Solid phase will be considered in the future OilMixProp version 2.0.

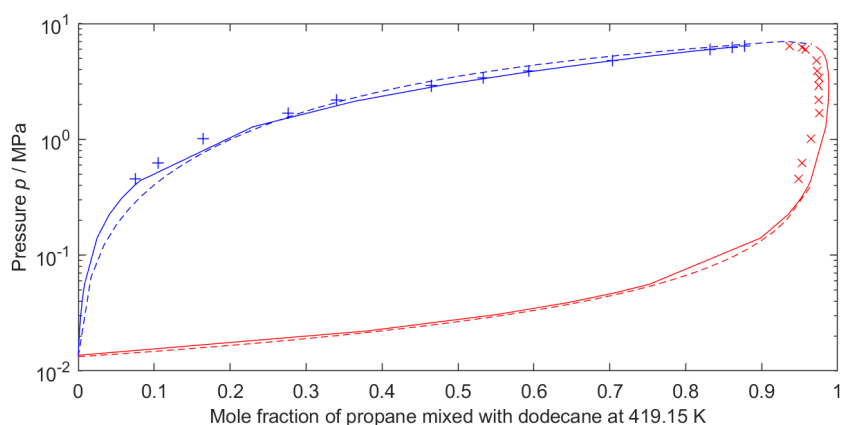


Figure 8. Pressure vs. composition diagram for the propane + dodecane mixture at 419.15 K. Symbols: experimental data ²⁴; solid curves: OilPropMix 1.0; dashed curves: REFPROP 10.0.

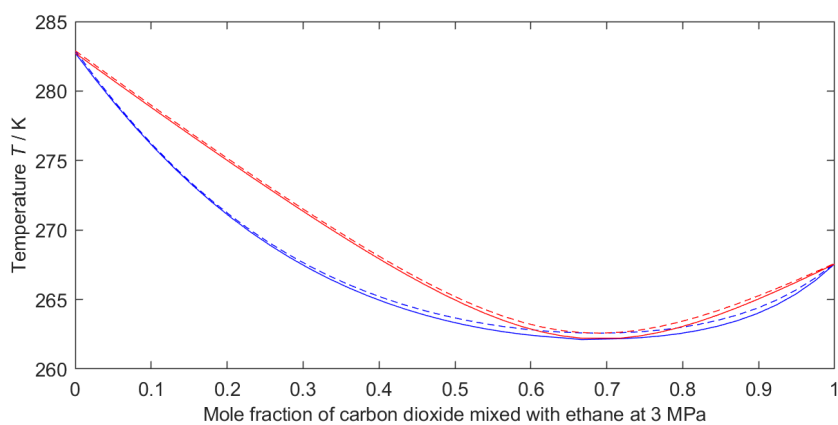


Figure 9. Temperature vs. composition diagram for the carbon dioxide + ethane mixture at 3 MPa. Solid curves: OilPropMix 1.0; dashed curves: REFPROP 10.0.

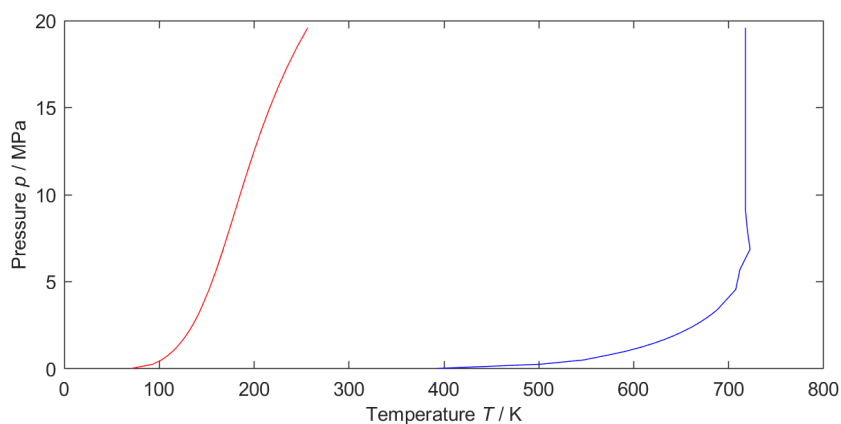


Figure 10. Pressure vs. temperature diagram for the nitrogen + Emkarate RL32 mixture at a nitrogen mole fraction of 0.3.

5. Conclusion and Outlook

In summary, we present here a software package OilMixProp 1.0, which calculates all the core thermophysical properties of oils, common fluids, and their mixtures. Properties that are needed in thermodynamic cycle analysis are all included: density, phase behavior, heat capacity, entropy, enthalpy, speed of sound, viscosity, and thermal conductivity. This package could be the first tool for comprehensive thermophysical property calculations of fluids involving user-defined oils. Using a minimal set of experimental data (fewer than 20 and at least 12 data points), fluid constants of an oil and its mixture with other fluids can be fitted with OilMixProp 1.0, and thus predictions can be achieved at larger temperature and pressure ranges. The models for property calculations in OilMixProp 1.0 include the cubic equation of state (EoS) for density, phase behavior, and residual properties, an additional equation for the ideal gas isobaric heat capacity $c_{p,0}$ as a linear function of temperature T for caloric properties, and the residual entropy scaling developed by Yang et al. [6–8] for transport properties. Here, the latest developed Yang, Frotscher, and Richter (YFR) EoS with improved liquid phase accuracy is first implemented in a software package.

Based on a self-developed robust flash algorithm, reliable calculation of mixtures at vapor-liquid equilibrium (VLE) can be achieved in OilMixProp 1.0. The function *OilPropm* is the interface for property calculations. It was developed analogously to *refpropm*, the Matlab interface of REFPROP 10.0, but with improved input and output designs. The input improvement includes: (1) using fluid constants instead of fluid names as input, which avoids repeating reading fluid constants, and (2) allowing users to provide initial guesses of temperature and pressure (if they are unknown) as optional inputs, which could potentially significantly reduce computational time. The output of *OilPropm* could be a class (a data type in Matlab) containing all information (phase fraction, as well as compositions and properties in each phase) of the fluid at the defined condition. This avoids the ambiguous output as delivered in some cases by *refpropm*, and in most cases reduces the calculation time.

The package is written in Matlab, and is free for academic institutions; please contact the authors to acquire it. Future developments of OilMixProp include: (1) adding functions to calculate critical points of mixtures; (2) updating constants of some fluids needed for transport property calculations based on the RES approach; (3) enabling more phase diagram plots; (4) applying solid constants to more fluids so that solids are considered and the VLE phase diagram could be updated to SVLE phase diagram; (5) development of a graphical user interface (GUI) to facilitate the use of the package for none professional Matlab programmer; (6) converting the package to other languages, e.g., python, so that more it could be accessed by more users. Some of these developments have already been carried out, and version 2.0 is estimated to be released in the middle of 2025.

Acknowledgments

The realization of the project and the scientific work was supported by the German Federal Ministry of Education and Research on the basis of a decision by the German Bundestag (funding code 03SF0623A). The authors gratefully acknowledge this support and carry the full responsibility for the content of this paper.

References

- [1] Yang X, Hanzelmann C, Feja S, Trusler J M and Richter M 2023 Thermophysical Property Modeling of Lubricant Oils and Their Mixtures with Refrigerants Using a Minimal Set of Experimental Data *Ind. Eng. Chem. Res.* **62** 18736–49
- [2] Patel N C and Teja A S 1982 A new cubic equation of state for fluids and fluid mixtures *Chem. Eng. Sci.* **37** 463–73
- [3] Valderrama J O 1990 A Generalized Patel-Teja Equation of State for Polar and Nonpolar Fluids and Their Mixtures *J. chem. eng. Jpn.* **23** 87–91
- [4] Soave G 1972 Equilibrium constants from a modified Redlich-Kwong equation of state *Chem. Eng. Sci.* **27** 1197–203
- [5] Peng D-Y and Robinson D B 1976 A new two-constant equation of state *Ind. Eng. Chem. Fundam.* **15** 59–64
- [6] Yang X, Xiao X, May E F and Bell I H 2021 Entropy Scaling of Viscosity—III: Application to Refrigerants and Their Mixtures *J. Chem. Eng. Data* **66** 1385–98
- [7] Yang X, Kim D, May E F and Bell I H 2021 Entropy Scaling of Thermal Conductivity: Application to Refrigerants and Their Mixtures *Ind. Eng. Chem. Res.* **60** 13052–70
- [8] Yang X, Xiao X, Thol M, Richter M and Bell I H 2022 Linking viscosity to equations of state using residual entropy scaling theory *Int. J. Thermophys.* **43** 183
- [9] Yang X, Xiao X, Thol M, Richter M and Bell I 2023 A Residual Entropy Scaling Approach for Viscosity of Refrigerants, other Fluids and their Mixtures *NIST*
- [10] Urbaneck T, Matthes M, Richter M, Hempel O, Safarik M, Franzke U, Noack R, Honke M, Müller M, Henninger S, Schossig P, Henning H-M, de Oliveira H T, Morgenstern A, Nienborg B, Gschwander S, Engelmann P and Benndorf G 2022 Research Platform Refrigeration and Energy Technology (KETEC)
- [11] Lemmon E W, Bell I H, Huber M L and McLinden M O 2018 *NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 10.0*, National Institute of Standards and Technology
- [12] Span R, Beckmüller R, Hielscher S, Jäger A, Mickoleit E, Neumann T and Pohl S 2020 *TREND. Thermodynamic Reference and Engineering Data 5.0*. (Bochum, Germany: Lehrstuhl für Thermodynamik, Ruhr-Universität Bochum)
- [13] Bell I H, Wronski J, Quoilin S and Lemort V 2014 Pure and Pseudo-pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp *Ind. Eng. Chem. Res.* **53** 2498–508
- [14] Yang X, Ophelia Frotscher and Richter M 2024 A new cubic equation of state developed with symbolic regression to improve liquid density calculation accuracy *Ind. Eng. Chem. Res.* (to be submitted)
- [15] Stryjek R and Vera J H 1986 PRSV: An improved Peng—Robinson equation of state for pure compounds and mixtures *Can. J. Chem. Eng.* **64** 323–33
- [16] Wilson G M 1964 Vapor-liquid equilibria, correlation by means of a modified Redlich-Kwong equation of state *Advances in Cryogenic Engineering* (Springer) pp 168–76
- [17] Redlich O and Kwong J N S 1949 On the thermodynamics of solutions. V. An equation of state. Fugacities of gaseous solutions *Chem. Rev.* **44** 233–44
- [18] Martinek V, Frotscher O, Richter M and Herzog R 2023 Introducing Thermodynamics-Informed Symbolic Regression -- A Tool for Thermodynamic Equations of State Development
- [19] Ritschel T K S and Jørgensen J B 2017 *Computation of Phase Equilibrium and Phase Envelopes* (DTU Compute)
- [20] Rachford H H Jr and Rice J D 1952 Procedure for Use of Electronic Digital Computers in Calculating Flash Vaporization Hydrocarbon Equilibrium *J. Petrol. Tech.* **4** 19–3
- [21] Martinek V, Yang X, Bell I H, Herzog R and Richter M 2024 Entropy scaling of viscosity IV - application to 124 industrially important fluids *J. Chem. Eng. Data* (to be submitted)
- [22] Li Z, Yang X and Duan Y 2024 Linking Thermal Conductivity to Equations of State Using Residual Entropy Scaling Theory *Int. J. Thermophys.* (to be submitted)