

Xiaoxian Yang*, Markus Richter†

¹ Chemnitz University of Technology, Applied Thermodynamics, 09107 Chemnitz, Germany

To be submitted to *Computer Physics Communications*

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 behavior, heat capacity, entropy, enthalpy, speed of sound, viscosity, and thermal conductivity. Approximately 632 pure fluids are available, and users can define their oil by determining its constants using the 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. The sophisticated output is designed to deliver the complete information (phase and properties in each phase) of a fluid at the given condition. This is guaranteed by a self-developed robust flash algorithm for liquid-vapor-equilibrium calculation in OilMixProp 1.0. The package is available in the developer's repository link: <https://github.com/runnizone/OilMixProp>. It is written in Matlab and will be converted to other languages in the future.

keywords: OilMixProp, thermophysical property, oil, mixture, flash algorithm

* Corresponding author. Tel.: +49 371 531-36921. Email address: xiaoxian.yang@mb.tu-chmenitz.de.
ORCID ID: <https://orcid.org/0000-0001-8120-5646>.

† Corresponding author. Tel.: +49 371 531-38050. Email address: m.richter@mb.tu-chmenitz.de.
ORCID ID: <https://orcid.org/0000-0001-8120-5646>.

1. Introduction

Oils are commonly used in daily life, for example, for cooking (sunflower oil, olive oil, etc.) and health care (fish oil, etc.), and are essential components in various industrial applications, for example, as lubricants in automobiles and refrigeration. The oil products available in the market are generally composed of one base oil and a variety of additives. This makes modeling all the essential thermophysical properties (including density, phase behavior, heat capacity, entropy, enthalpy, viscosity and thermal conductivity) of an oil product very difficult, because traditional accurate thermophysical property models, such as 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 of the properties, calculations using this approach generally agree with experimental data within the same level of experimental uncertainty.

The fundamental of this modelling approach ¹ is complicated, as it involves the Patel-Teja-Valderrama (PTV) EoS ^{2,3}, which is more complicated than the famous Soave-Redlich-Kwong (SRK) ^{4,5}, Peng-Robinson (PR) ⁶ EoS, and the state-of-the-art residual entropy scaling (RES) approaches developed by Yang et al. ⁷⁻¹⁰. Furthermore, for calculation of gas solubility, which is an important property of mixtures involving oils, a robust flash algorithm for liquid-vapor-equilibrium (LVE) calculation is also needed. To facilitate the implementation and application of our modelling approach, and to make thermophysical properties of oils and their mixtures available to everyone easily, a free software package OilMixProp (refers to oil mixture properties) implemented with a self-developed robust LVE flash algorithm was developed in this work. It is available in the developer's repository link: <https://github.com/runnizone/OilMixProp>. This endeavor is part of subproject 3 within the KETEC (Research Platform Refrigeration and Energy Technology) project ¹¹. The package is written in Matlab and will be converted to other languages 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 (literature) experimental data, (1) 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 LVE 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 are not available in any other existing thermophysical property software packages, even in those commonly-used reference ones (such as REFPROP 10.0 ¹², TREND 5.0 ¹³ and CoolProp 6.4.1 ¹⁴). Furthermore, when oils are involved, the fluid mixture is typically in two-phase region: an oil-rich liquid phase dissolved with other components, and a gas phase with trace amount of oil. To study such systems well, an robust LVE flash algorithm was developed as the core of OilMixProp, and a sophisticated design of inputs and outputs are implemented. Various input combinations are available, which are analogy to the function *refpropm*, the Matlab interface of REFPROP 10.0 ¹². While the output is either a class (a datatype in Matlab) of a fluid at the defined condition 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 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 obvious in two phase region (see more details in section 4.1).

At last, we want to mention that, approximately 632 pure fluids, unlimited custom-defined oils and their mixtures are available in OilMixProp. OilMixProp 1.0 is not aimed at becoming a reference thermophysical property software packages, such as REFPROP 10.0 ¹² which focuses a lot in accuracy. OilMixProp 1.0 is aimed at becoming a reliable solution when the study fluids (e.g., fluids involving oils) are not available in those reference packages. Compared to REFPROP 10.0, OilMixProp 1.0 has higher uncertainty in pure fluids and in single phases. However, when it comes to LVE calculation of mixtures, OilMixProp 1.0 could be as accurate as REFPROP, while, in some situations, deliver more reliable results (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 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^{4,5}, PR⁶, Yang and Richter (YR)¹⁵, Peng-Robinson-Stryjek-Vera (PRSV)¹⁶, Wilson-Redlich-Kwong (WRK)¹⁷ and Redlich-Kwong (RK)¹⁸] are available for calculations of density, phase behavior and the residual properties. Here YR EoS is a new cubic EoS recently developed within our group aided with symbolic regression¹⁹; 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 addition 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 state-of-the-art RES approach developed by Yang et al.^{7–10}, viscosity and thermal conductivity can be calculated. The fundamentals of these properties' calculation from the model set cubic EoS + linear $c_p^o(T)$ + RES has already given in the supporting information of our previous paper¹. For mixtures, the van der Waals mixing rule is used in Cubic EoS and the mixing rules developed by Yang et al.^{9,10} are used for both viscosity and thermal conductivity.

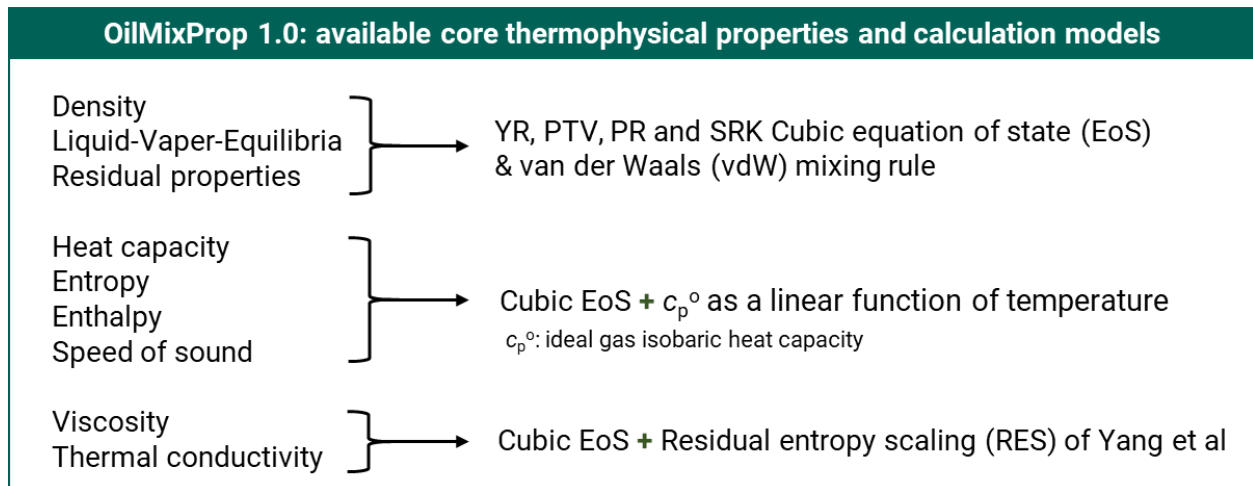


Figure 1. Available core thermophysical properties in OilMixProp 1.0 and the calculation models.

With the model set of cubic EoS + linear- $c_p^\circ(T)$ + RES, fluid constants of a pure fluid or a quasi-pure oil that are needed to achieve calculation all the properties 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- $c_p^\circ(T)$, 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 10.0¹², the constants are obtained by fitting to calculations of REFPROP (k_0 and k_1), from Yang et al's RES approach^{7–10} ($n_{\mu k}$ and $n_{\lambda k}$), or from REFPROP directly (all other constants). For remaining 481 (=632–151) 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^{7–10}, and the remain properties are constants being the average of those 151 fluids in REFPROP. Therefore, one should expect a lower accuracy when calculation properties of these fluids. For an oil product, the determination or estimation methods for these properties has been given in our previous work¹. 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 YR EoS for ω , linear- $c_p^\circ(T)$ for k_0 and k_1 , RES approaches of Yang et al.^{7–10} 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>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>PTV or YR cubic Equation of State (EoS)</u> <ul style="list-style-type: none"> Acentric factor ω 	<u>Estimations</u> <ul style="list-style-type: none"> Lennard–Jones parameters ε/k_B Lennard–Jones parameters σ $\varepsilon/k_B = T_c/1.2593 \quad \sigma^3 = 0.3189/\rho_c$ <p>for viscosity and thermal conductivity at dilute gas limit.</p>
<u>c_p° as linear function of T</u> <ul style="list-style-type: none"> k_0 and k_1 in the linear-$c_p^\circ(T)$ <p><small>c_p°: ideal gas isobaric heat capacity</small></p>	<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 $n_{\lambda k}$ ($k = 1,2,3,4$) <p>for residual viscosity and thermal conductivity</p>
	<u>Constant values</u> <ul style="list-style-type: none"> φ_0, Γ, q_D <p>for the critical enhancement of thermal conductivity</p>

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

2.2 Liquid vapor equilibria

Although the fundamental of LVE calculation is available in literature ²⁰, to explain the self-developed LVE flash algorithm as clear 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 $\boldsymbol{\varphi}$ (an $n \times m$ matrix, with φ_{ji} being that of i^{th} component in j^{th} phase) are:

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

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

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

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

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

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

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 typically 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 LVE flash algorithm to be discussed in Section 3.1. Another typically scenario is to calculate the phase transition temperature (or pressure), with overall composition \mathbf{Z} , pressure (or temperature) known and one of the phase fraction q_j equals 0. This corresponds to the bubble-point and dew-point calculations in LVE.

3. Algorithms

The calculations of most of the thermophysical properties are pretty straight forward 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 LVE calculation, the phase transition point (bubble-point and dew-point) determination, and calculation with one input existing in the two-phase region.

3.1 The flash algorithm

For LVE 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: $\mathbf{X} = [x_1, x_2, \dots, x_n]$ and $\mathbf{Y} = [y_1, y_2, \dots, y_n]$ are molar fractions of components in the liquid and vapor phases, respectively; $\boldsymbol{\phi}_L = [\phi_{L1}, \phi_{L2}, \dots, \phi_{Ln}]$ and $\boldsymbol{\phi}_V = [\phi_{V1}, \phi_{V2}, \dots, \phi_{Vn}]$ 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 []. The Rachford-Rice equation ²¹:

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

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

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

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

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

With these new variables and equations, and with T, p and \mathbf{Z} known, the LVE 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 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), $\boldsymbol{\phi}_L$ and $\boldsymbol{\phi}_V$ can be obtained, and new \mathbf{K} is calculated using Eq. (12). (III) iterating the Q and \mathbf{K} calculations until converge condition is met.

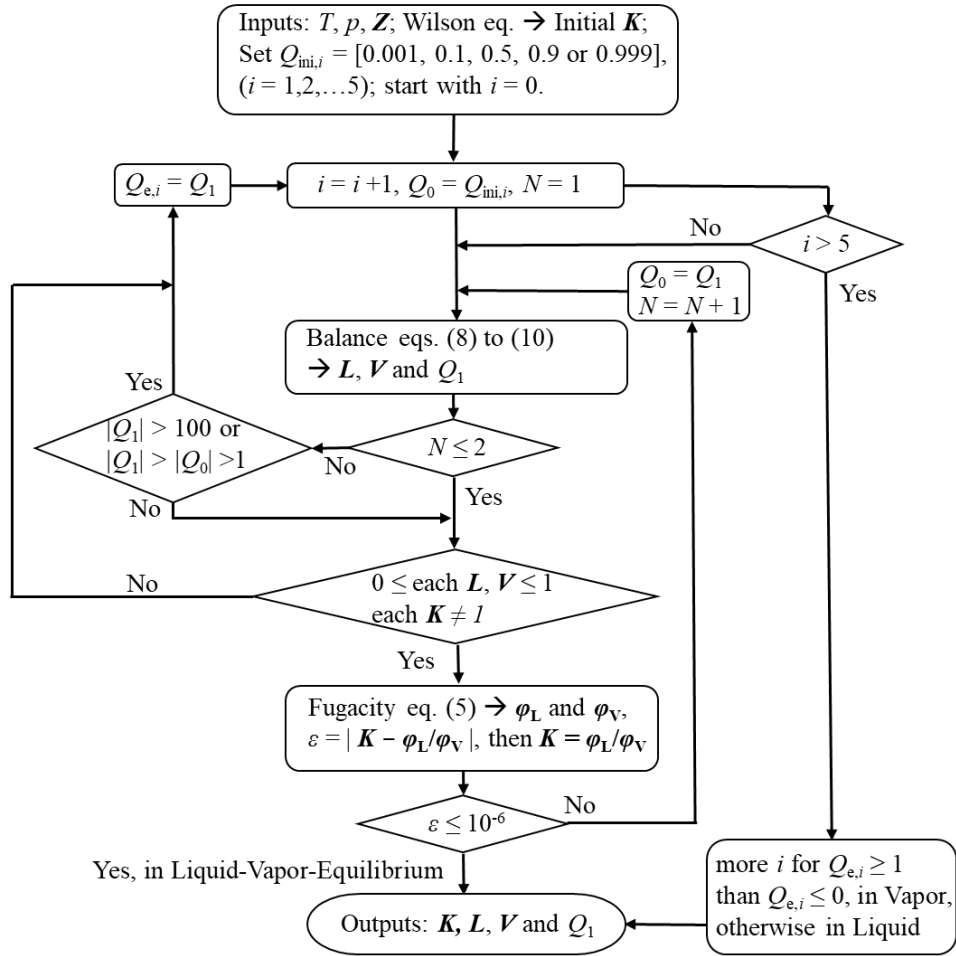


Figure 3. The liquid-vapor-equilibrium flash algorithm implemented in OilMixProp 1.0.

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 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 vapor phase, otherwise in liquid phase. There are also tricks in improving calculation efficiency: e.g., once the looping Q deviate further away from the LVE range ($0 \leq Q \leq 1$), the Q (Q_0 in Figure 3) jumps to the next initial guess.

As a result, this algorithm is correct and robust for almost all the pure fluids and mixtures tested. One unresolved issue lies in the phase identification when the mixture is in supercritical region: so far the output is liquid phase in the denser region and vapor phase vice versa. This issue can be

solved by adding 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 require iterations, i.e., phase transition point (bubble-point and dew-point) determination, and LVE calculation with one input only exist in two-phase region.

In phase transition point calculation, Q equals 1 or 0, which makes K the only values 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 good 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 (L for liquid, V for vapor and LV for liquid-vapor equilibrium) using the flash algorithm at the guessed temperature given by the user. If, for example, the fluid is in LVE condition, 5 K will be added to the guessed temperature and the phase condition is determined again; this step will go on until reaching vapor phase. Then an initial guess of temperature being the average of the last two temperatures are 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 molar average of that 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 exists in 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 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., 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 single phase, $Q = 1$ or 0 .

4. Functions and examples

The core of OilMixProp 1.0 is to calculate 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 The property calculation function *OilPropm*

Analogy to function *refpropm*, which is the Matlab interface of REFPROP 10.0¹², a function *OilPropm* in OilMixProp 1.0 was developed as the ‘only’ interface that users need to call. Examples of using this function is given in FluidCalc.m in the package. An overview of function *OilPropm* is given in Figure 4.

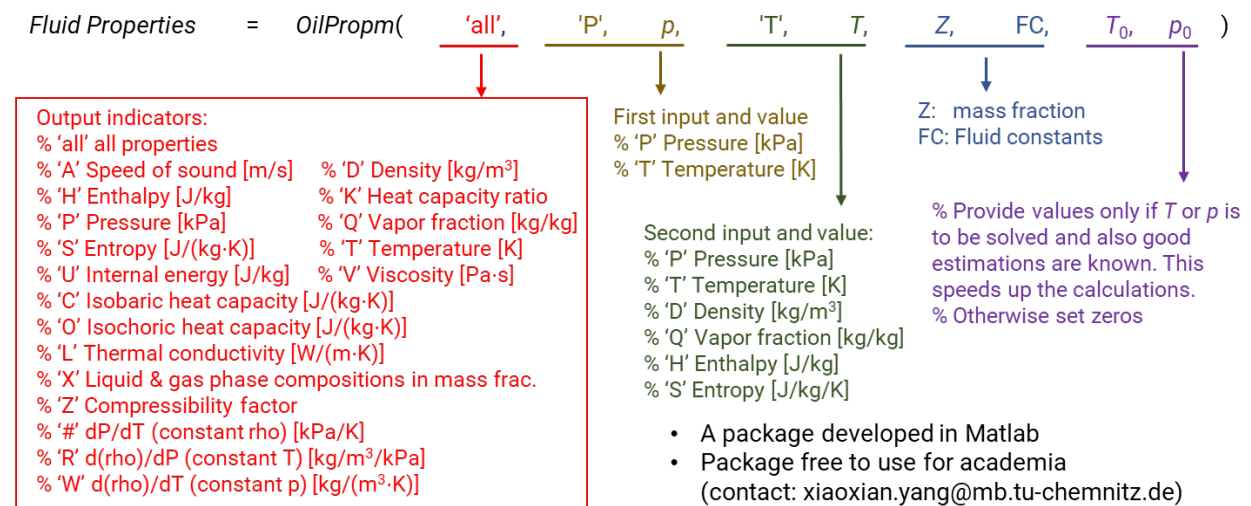


Figure 4. Overview of the function *OilPropm* in OilMixProp 1.0.

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') which 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 $\text{mol}\cdot\text{m}^{-3}$, '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', 'YR'}; CubicEOS = AllEOS{3};  
FC = GetGlobals(CubicEOS, Fluids);
```

This design is different with *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 zero; however, we strongly recommend users provide guesses as good as possible, because this will significantly speed up the calculation, and avoid fail 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 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 guess (e.g., molar average of pure component solutions) for the N-R method will be applied until the right solution is found. In thermodynamic cycle analysis, such good guesses are always available, e.g., based on the properties of the previous state.

For the output, *refpropm* 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 a LVE condition:

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

but the output is a single value $21.924 \text{ kg}\cdot\text{m}^{-3}$ without indicating this value is in liquid phase, vapor phase or the overall phases. With OilMixProp 1.0, such 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 YR EoS is illustrated in Figure 5. From this comprehensive output, users will see that the liquid phase is almost pure water with 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 frac. with YR EoS
T:      303.150000    K
p:      1.000000     MPa
MM:      21.929      g/mol
Phase:      LV

Phase behavior in mass fraction. Vapor Frac: 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: -3.266   1.516   kJ/K/kg
Enthalpy: -2423.887 28.870  kJ/kg
cp:      4.539    1.061   kJ/K/kg
cv:      2.545    0.749   kJ/K/kg
SoS:     4352.140 356.844  m/s
vis:     1.755484 0.017798 mPa s
TC:      1.027107 0.028174 W/m/K
Kappa    1.783642 1.416655
d(rho)/dP: 0.000094 0.011125 kg/m3/kPa
d(rho)/dT: -0.771048 -0.037613 kg/m3/K
dp/dT_rho: 8187.997498 3.380883 kPa/K

-- Properties with all phases combined --
rho_all: 21.915      kg/m3
Entropy_all: -0.868   kJ/K/kg
Enthalpy_all: -1194.031 kJ/kg
cp_all: 2.795        kJ/K/kg
cv_all: 1.644        kJ/K/kg
Kappa_all 1.699797
```

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 one 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 LVE 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 (LVE flash calculation), which is mandatory for all kind of calculations, while the last two steps are very quick as no iterations needed. Here, only when a single property is of interest, user could specify this output property, e.g, ‘S’ for entropy, this is the same 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 fluid constants of any oil, so that 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 ($p = 1$ atm, T, ρ) → T_c, p_c, and ω</p> <p>4 (2) → four, could be down to two, points of ($p = 1$ atm, T, c_p) → k_0 and k_1</p> <p>4 (4) → four points of ($p = 1$ atm, T, μ) → $n_{\mu k}$ ($k = 1,2,3,4$)</p> <p>4 (4) → four points of ($p = 1$ atm, T, λ) → $n_{\lambda k}$ ($k = 1,2,3,4$)</p> <p>Note: Temperature suggested to be (278.15 to 368.15) K or smaller</p> <p>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 M . If data 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 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 not available, $n_{\mu k}$ are defaulted values being global parameters of group 6 according to Yang et al.

(6) Optionally, at least four thermal conductivity (T, p, λ) are needed to fit $n_{\lambda k}$ ($k = 1, 2, 3, 4$); if data not available, $n_{\lambda k}$ are defaulted values being global parameters of group 6 according to XXX et al.

(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 collect data (either from experiment or literature) up to about 10 sets for each needed properties (see Figure 6) and carry out the fit. In OilMixProp 1.0, users can choose a few data for fitting and the remaining for confirming the fitting. An example results are given in Figure 7, where only data in full symbols are used for fitting.

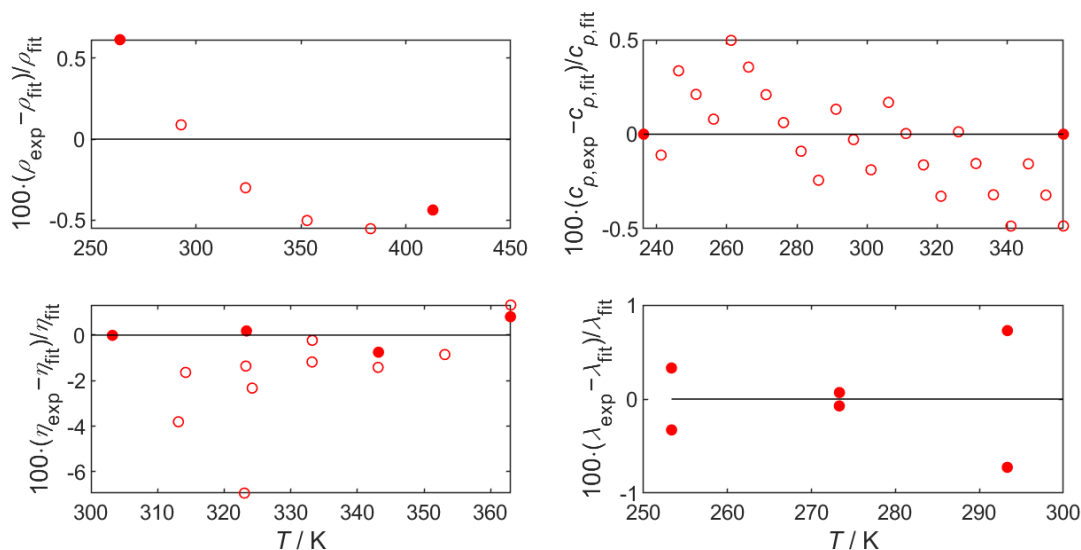


Figure 7. Example of the fluid constants fitting of PAG68: Relative deviation of the experimental data ¹ from the model predictions. Only Data in full symbol are used in fitting fluid constants.

In the developer's repository link: <https://github.com/runnizone/OilMixProp>, OilMixProp 1.0 has includes 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 The 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), pressure vs. temperature (p - T). 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 in a similar way. 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 for a p - x diagram of a zeotrope mixture propane + dodecane at 419.15 K is given in Figure 8 and for a T - x diagram of an azeotrope mixture of carbon dioxide + ethane mixture at 3.0 MPa is given in Figure 9. Many tricks, e.g., well-defined pressure (or temperature) points, are adopted to optimize the calculation speed while provide 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.

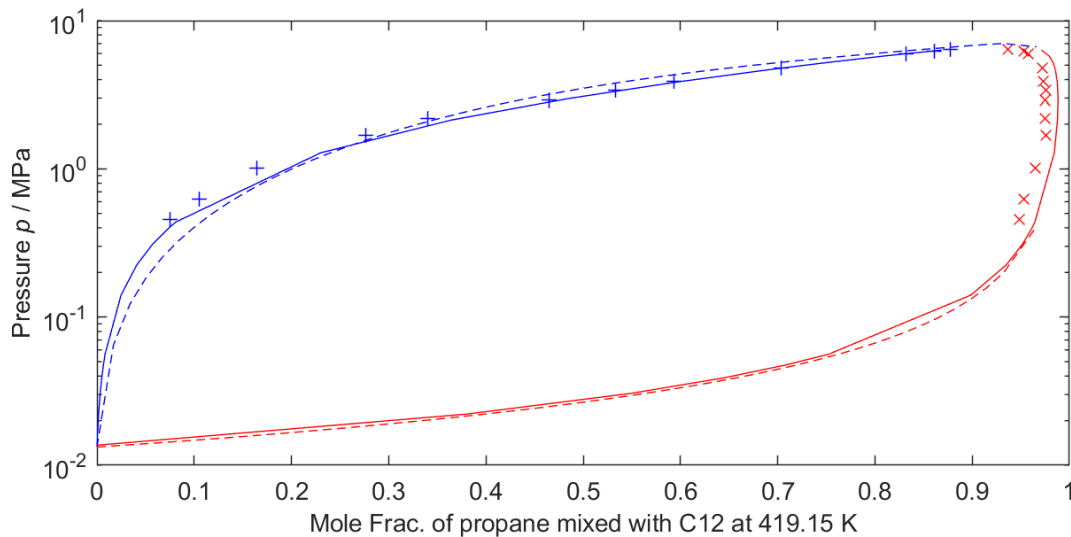


Figure 8. Pressure vs. composition diagram for the propane + dodecane mixture at 419.15 K. Symbols: experimental data²²; solid curves: OilPropMix 1.0; dashed curves: REFPROP 10.0.

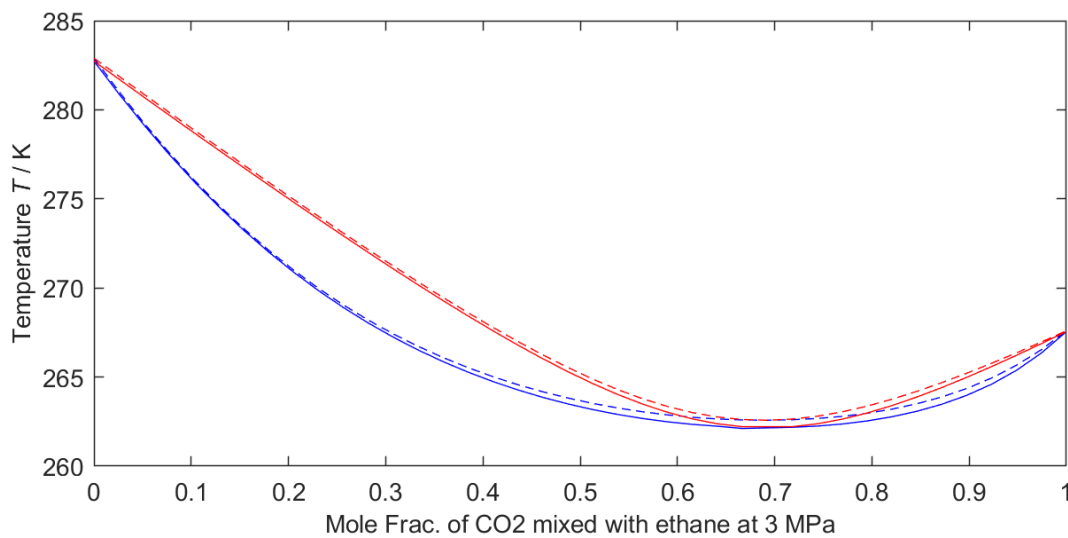


Figure 9. Temperature vs. composition diagram for the carbon dioxide + ethane mixture at 3 MPa. Solid curves: OilPropMix 1.0; dashed curves: REFPROP 10.0.

As for p - T phase diagrams, OilMixProp 1.0 handle 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 be noted that, oils generally decompose at higher temperature, 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. For example, see Figure 11, which is plotted for CO₂ + methane mixtures at by an solid-liquid-vapor equilibrium (SLVE) code. Only those in black are the LVE phase boundaries, while its extensions are ‘erased’ by the existence of solid phase. This SLVE code is not robust yet and thus will only be included in the future OilMixProp version 2.0.

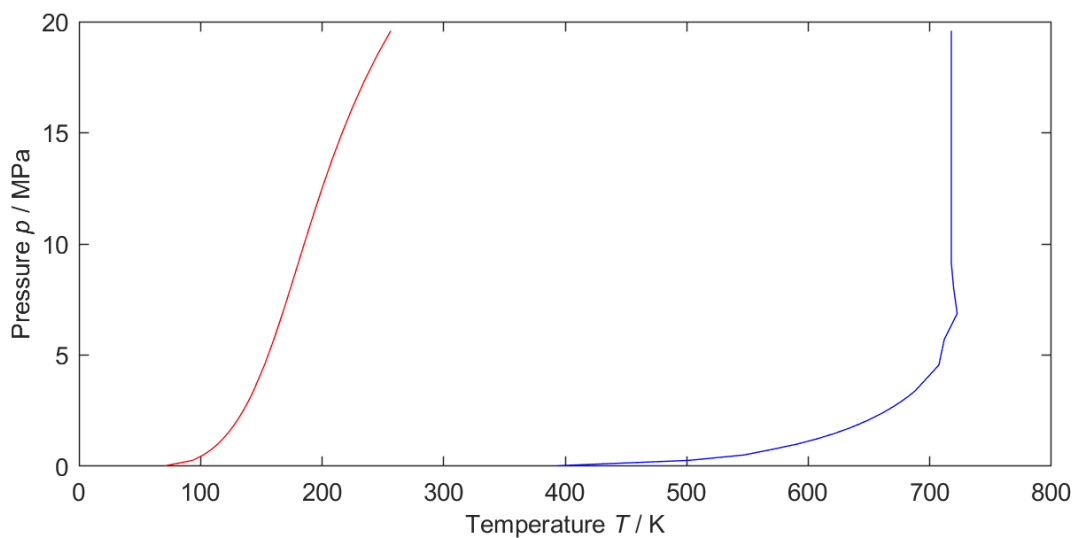


Figure 10. Pressure vs. temperature diagram for the nitrogen + Emkarate RL32 mixture at a nitrogen mole fraction of 0.3.

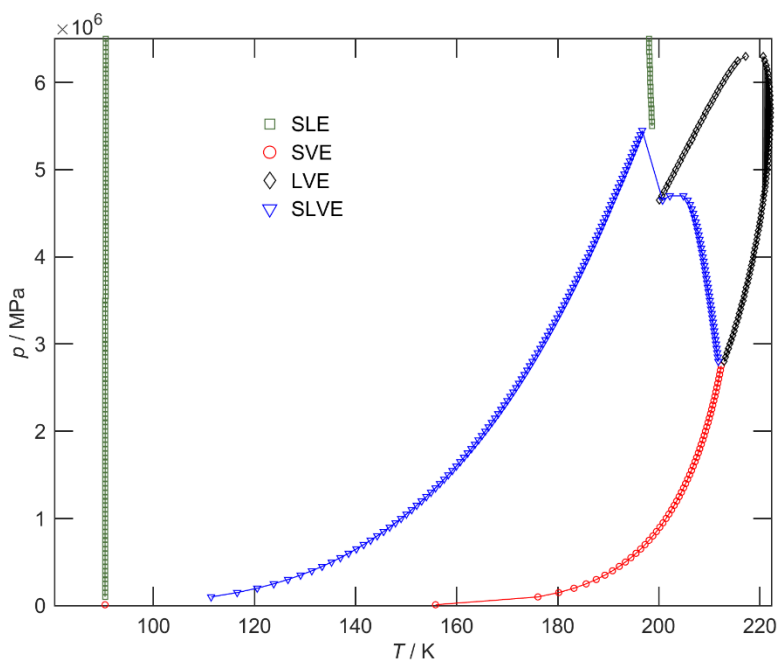


Figure 11. Solid-liquid-vapor equilibrium of a CO₂ + methane mixture. This function will only be available in the future OilMixProp 2.0.

5. Conclusions and Outlooks

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 (see the fundamental developed in our previous work ¹), fluid constants of an oil and its mixture with other fluids can be fitted with OilMixProp 1.0, and thus predictions at larger temperature and pressure ranges can be achieved. The models for the property calculations in OilMixProp 1.0 includes: cubic equation of state (EoS) for density, phase behavior and the residual properties, addition equation for the ideal gas isobaric heat capacity c_p^o as a linear function of temperature T for caloric properties, and residual entropy scaling developed by Yang et al. ⁷⁻⁹ for transport properties. Here the latest developed Yang and Richter (YR) 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 liquid-vapor equilibrium (LVE) can be achieved in OilMixProp 1.0. The function *oilpropm* is the single interface for property calculations. It was developed in analogy 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 optionally inputs, which could potentially significantly reduce computational time. The output of *oilpropm* could be a class (a datatype 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 available in the link: <https://github.com/runnizone/OilMixProp>. Future developments of OilMixProp includes: (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) enable more phase diagram plots; (4) apply solid constants to more fluids so that solids are considered and the LVE phase diagram could be updated to SLVE phase diagram; (5) development of a graphical user interface (GUI) to facility 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.

Reference

- (1) Yang, X.; Hanzelmann, C.; Feja, S.; Trusler, J. P. M.; Richter, M. Thermophysical Property Modeling of Lubricant Oils and Their Mixtures with Refrigerants Using a Minimal Set of Experimental Data. *Ind. Eng. Chem. Res.* **2023**, 62 (44), 18736–18749. <https://doi.org/10.1021/acs.iecr.3c02474>.
- (2) Patel, N. C.; Teja, A. S. A New Cubic Equation of State for Fluids and Fluid Mixtures. *Chem. Eng. Sci.* **1982**, 37 (3), 463–473. [https://doi.org/10.1016/0009-2509\(82\)80099-7](https://doi.org/10.1016/0009-2509(82)80099-7).
- (3) Valderrama, J. O. A Generalized Patel-Teja Equation of State for Polar and Nonpolar Fluids and Their Mixtures. *Journal of Chemical Engineering of Japan* **1990**, 23 (1), 87–91. <https://doi.org/10.1252/jcej.23.87>.
- (4) Soave, G. Equilibrium Constants from a Modified Redlich-Kwong Equation of State. *Chemical Engineering Science* **1972**, 27 (6), 1197–1203. [https://doi.org/10.1016/0009-2509\(72\)80096-4](https://doi.org/10.1016/0009-2509(72)80096-4).
- (5) Yang, F.; Liu, Q.; Duan, Y.; Yang, Z. On the Temperature Dependence of the A Function in the Cubic Equation of State. *Chemical Engineering Science* **2018**, 192, 565–575. <https://doi.org/10.1016/j.ces.2018.08.014>.
- (6) Peng, D.-Y.; Robinson, D. B. A New Two-Constant Equation of State. *Ind. Eng. Chem. Fundam.* **1976**, 15 (1), 59–64. <https://doi.org/10.1021/i160057a011>.
- (7) Yang, X.; Xiao, X.; May, E. F.; Bell, I. H. Entropy Scaling of Viscosity—III: Application to Refrigerants and Their Mixtures. *J. Chem. Eng. Data* **2021**, 66 (3), 1385–1398. <https://doi.org/10.1021/acs.jced.0c01009>.

- (8) Yang, X.; Kim, D.; May, E. F.; Bell, I. H. Entropy Scaling of Thermal Conductivity: Application to Refrigerants and Their Mixtures. *Ind. Eng. Chem. Res.* **2021**, *60* (35), 13052–13070. <https://doi.org/10.1021/acs.iecr.1c02154>.
- (9) Yang, X.; Xiao, X.; Thol, M.; Richter, M.; Bell, I. H. Linking Viscosity to Equations of State Using Residual Entropy Scaling Theory. *Int. J. Thermophys.* **2022**, *43* (12), 183. <https://doi.org/10.1007/s10765-022-03096-9>.
- (10) Yang, X.; Xiao, X.; Thol, M.; Richter, M.; Bell, I. A Residual Entropy Scaling Approach for Viscosity of Refrigerants, Other Fluids and Their Mixtures. *NIST* **2023**.
- (11) 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.; Benndorf, G. *Research Platform Refrigeration and Energy Technology (KETEC)*. www.ketec.online (accessed 2023-04-25).
- (12) Lemmon, E. W.; Bell, I. H.; Huber, M. L.; McLinden, M. O. *NIST Standard Reference Database 23: Reference Fluid Thermodynamic and Transport Properties-REFPROP, Version 10.0*, National Institute of Standards and Technology; 2018. <https://doi.org/10.18434/T4/1502528>.
- (13) Span, R.; Beckmüller, R.; Hielscher, S.; Jäger, A.; Mickoleit, E.; Neumann, T.; Pohl, S. *TREND. Thermodynamic Reference and Engineering Data 5.0*; Bochum, Germany: Lehrstuhl für Thermodynamik, Ruhr-Universität Bochum, 2020.
- (14) Bell, I. H.; Wronski, J.; Quoilin, S.; Lemort, V. Pure and Pseudo-Pure Fluid Thermophysical Property Evaluation and the Open-Source Thermophysical Property Library CoolProp. *Ind. Eng. Chem. Res.* **2014**, *53* (6), 2498–2508. <https://doi.org/10.1021/ie4033999>.
- (15) Yang, X.; Richter, M. A New Cubic Equation of State Developed with Symbolic Regression to Improve Liquid Density Calculation Accuracy. *J. Chem. Eng. Data* (to be submitted).
- (16) Stryjek, R.; Vera, J. H. PRSV: An Improved Peng—Robinson Equation of State for Pure Compounds and Mixtures. *The canadian journal of chemical engineering* **1986**, *64* (2), 323–333.
- (17) Wilson, G. M. Vapor-Liquid Equilibria, Correlation by Means of a Modified Redlich-Kwong Equation of State. In *Advances in Cryogenic Engineering*; Springer, 1964; pp 168–176.
- (18) Redlich, O.; Kwong, J. N. S. On the Thermodynamics of Solutions. V. An Equation of State. Fugacities of Gaseous Solutions. *Chem. Rev.* **1949**, *44* (1), 233–244.
- (19) Martinek, V.; Frotscher, O.; Richter, M.; Herzog, R. Introducing Thermodynamics-Informed Symbolic Regression -- A Tool for Thermodynamic Equations of State Development. arXiv September 6, 2023. <https://doi.org/10.48550/arXiv.2309.02805>.
- (20) Ritschel, T. K. S.; Jørgensen, J. B. *Computation of Phase Equilibrium and Phase Envelopes*; Report; DTU Compute, 2017.
- (21) Rachford, H. H., Jr.; Rice, J. D. Procedure for Use of Electronic Digital Computers in Calculating Flash Vaporization Hydrocarbon Equilibrium. *Journal of Petroleum Technology* **1952**, *4* (10), 19–3. <https://doi.org/10.2118/952327-G>.
- (22) Gardeler, H.; Fischer, K.; Gmehling, J. Experimental Determination of Vapor–Liquid Equilibrium Data for Asymmetric Systems. *Ind. Eng. Chem. Res.* **2002**, *41* (5), 1051–1056. <https://doi.org/10.1021/ie0103456>.

