Evaluating the acceptability and accuracy of Phasepy as a Phyton framework to calculate the interfacial properties and phase equilibrium

Javad Tavakolifaradonbe, Sumudu Karunarathne*, Knut Vågsæther

Department of Process, Energy and Environmental Technology, University of South-Eastern Norway
h.javadtf@yahoo.com, sumudu.karunarathne@usn.no, Knut.Vagsather@usn.no

Abstract

Phasepy is a scientifically defined open-source package in python for computational thermodynamics. Phasepy indeed calculates the interfacial properties and fluid phases equilibrium using an equation of state. In addition, Phasepy enables the scientists to optimize the relevant parameters to the equilibrium of multicomponent vapor-liquid, liquid-liquid, or vapor-liquid-liquid mixtures. The Phasepy can model the equilibrium in the continuous approach (combining a cubic equation of state and a mixing rule) or the discontinuous approach (using a virial equation and an activity coefficient model). So, this study is to develop a code in a continuous approach using a combination of Soave-Redlich-Kwong (SRK) or Peng Robinson (PR) as the equation of state and quadratic mixing rule (QMR) and modified-Huron-Vidal mixing rule (MHV) as the mixing Rule. Although the algorithm of the developed model is new, it is tried to utilize the predefined function of Phasepy to calculate fluid phase equilibrium and interfacial properties. In fact, the five well-performed previous experimental studies are modeled using Phasepy, and in the following, the outputs of the developed models are compared with the relevant experimental results. The bubble point features, dew point features, liquid and gas composition, and density of multicomponent mixtures are considered parameters in this extended study to evaluate the accuracy of the Phasepy function based on experimental results.

1. Introduction

To study the homogenous and the inhomogeneous behavior of fluids it is required to know about interfacial properties and fluid phase equilibria. These two physical properties enable scientists to design a process and an operation. From one side, the phase equilibrium discloses the physical limitations of a separation process and reveals the required stages for reaching equilibrium. On the other side, interfacial properties determine the efficiency and the size of equipment [1].

To study the feasibility of a process, firstly the most appropriate thermodynamic model for experimental data should be selected, and even sometimes operations need to optimized. Therefore, it is required study equilibrium thermodynamics [2]. Studies have shown that finding a perfect model and algorithms for computing phase equilibria are the most important concern of industries, to the extent that, industries prefer to not invest time and money in developing a new algorithm for phase equilibrium calculations even when it is required [3]. So, we see oil and gas industries are still using the classical equation of states (EoS). Indeed, developing a new algorithm based on contemporary EoS requires industries to provide a dissimilar set of parameters. In other words, the necessity for developing a more precise molecular-based model would increase computational expenditures [3], [4].

Prausnitz et al. [5] and Michelsen and Mollerup [6] discussed the fundamental computation for calculating phase equilibria in isothermal isobaric two-phase flash, liquid-liquid equilibrium, heteroazeotropic, bubble points, and dew points. Derived solution methods are based on minimizing Gibbs free energy of the system or criteria for isofugacity. They applied a combination procedure of newton methods, successive substitution, and second-order minimizations.

Commercial solutions like Aspen Plus, gPorms, etc. are being developed to simulate the process and calculate phase equilibria. Although these simulators consist some advantages like different thermodynamic models, an extended database on phase equilibria, and stable manners, some cons should be mentioned also, including licenses fee, and the limitation of manipulating prefabricated processes. In addition, these commercial software does not use square gradient theory (SGT) for interfacial descriptions. As a theoretical approach, SGT in connection with a continuous EoS specifies the binodal boundary. Moreover, SGT makes a connection between homogenous phase and transition zone and as a result characterizes interfacial tension, Gibbs energy, and density profile. To deal with this type of problems, scientists are nominating homemade programs as a reliable solution [7]–[10].

This study is to use a Python-based module, namely Phasepy, to calculate interfacial properties and phase equilibrium computation and evaluate how much the defined module is applicable for calculating thermodynamic properties in multicomponent mixtures. Then firstly a short description of Phasepy is provided and then it is tried to evaluate the accuracy of calculations based on the previously performed experimental studies.

2. Theory and Methodology

Although more than 200 EoS were published by 1949, Redlich and Kwong made effort to deal with the limitations and revive the van der Waals EoS for high- and low-density fluids. They proposed following EoS [11], [12]:

$$P = \frac{RT}{V-b} - \frac{a_c\alpha(T)}{V(V+b)} \qquad (1)$$

Where:

$$\alpha(T) = \frac{a}{T^{0.5}} \qquad (2)$$

$$a_c = \frac{\Omega_a R^2 T_c^{2.5}}{P_c} \qquad (3)$$

$$b = \frac{\Omega_b R \mathrm{T}_c}{\mathrm{P}_c} \qquad (4)$$

$$\Omega_a = 0.4278 \quad \text{and} \quad \Omega_b = 0.0867$$

Although RK EoS does not posse a considerable background theoretically, this model provides acceptable results. In the following, Soave presented a new version of RK EoS by keeping the RK volume functionality and redefining α as a function of reduced temperature and acentric factor [12].

$$P = \frac{RT}{V-b} - \frac{a_c\alpha(T_r,\omega)}{V(V+b)} \qquad (5)$$

$$a_c = \frac{0.42747 R^2 T_c^{2.5}}{P_c} \qquad (6)$$

$$b = \frac{0.08664 R \mathrm{T}_c}{\mathrm{P}_c} \qquad (7)$$

$$\alpha(T_r,\omega) = [1 + (0.480 + 1.574\omega - 0.176\omega^2)(1 - T_r^{0.5})]^2 \qquad (8)$$

The SRK now is the most popular EoS in hydrocarbon-related industries. Then scientists tried to define new temperature model $\alpha(T_r,\omega)$ and modify volume dependency of pressure-related terms. Peng and Robinson calculated $\alpha(T_r,\omega)$ again and modified SRK EoS. PR EoS is:

$$P = \frac{RT}{V-b} - \frac{a_c\alpha(T_r,\omega)}{V(V+b)+b(V-b)} \qquad (9)$$

$$a_c = \frac{0.45724 R^2 T_c^{2.5}}{P_c} \qquad (10)$$

$$b = \frac{0.07780 R \mathrm{T}_c}{\mathrm{P}_c} \qquad (11)$$

$$\alpha(T_r,\omega) = [1 + (0.37464 + 1.54226 - 0.26992\omega^2)(1 - T_r^{0.5})]^2 \qquad (12)$$

Theoretically, mixing rules have been developed to connect multicomponent mixture parameters to pure fluid parameters [12]. Mostly, classical van der Waals mixing rules are applied as:

$$a = \sum\sum x_i x_j a_{ij} \qquad (13)$$

$$b = \sum\sum x_i x_j b_{ij} \qquad (14)$$

$$c = \sum\sum x_i x_j c_{ij} \qquad (15)$$

The volume parameters, b_{ij} and c_{ij}, are calculated with arithmetic mean and for force parameter, a_{ij}, the geometric mean is utilized. Therefore, phase equilibrium can be correlated more accurately [12].

$$a_{ij} = \sqrt{a_i a_j}(1 - k_{ij}) \qquad (16)$$

$$b_{ij} = \frac{1}{2}(b_i + b_j)(1 - \beta_{ij}) \qquad (17)$$

$$c_{ij} = \frac{1}{2}(c_i + c_j)(1 - \delta_{ij}) \qquad (18)$$

Although these modifications can keep the concentration-related affinity of the parameters, a better modification is required for complex cases like supercritical fluid processes. k_{ij}, β_{ij}, and δ_{ij} are known as the interaction coefficients between components i and j. moreover, regression analysis of real data or predictive correlations is applied to calculate these interaction coefficients. Some other studies have tried to understand the logic behind the binary interaction coefficient and pure species properties [11].

Phasepy is an open-source, and scientific package based on Python which has been developed to calculate interfacial properties and phase equilibrium. Object-oriented style of Phasepy enables users to apply small codes to calculate thermodynamics' properties. By using Phasepy it is possible to model a pure fluid or a multicomponent mixture fluid based on EoS. Then stability of phase equilibrium computation is evaluated. Finally, selecting continuous approach provides a possibility to investigate interfacial behavior based on SGT [10].

Firstly, the pure components and their properties, including critical temperature [*K*], critical pressure [*bar*], critical compressibility factor, critical volume [*cm3/mol*], and acentric factor should be defined. In this way, the *component* function enables the user to define components separately and then the *mixture* and the *add_component* functions define and develop the fluid mixture. The following code script shows how to define a three-component mixture of Nitrogen, Carbon dioxide, and Methane.

$$N2 = component(name = 'nitrogen', Tc = 126.2, Pc = 34, Zc = 0.289, Vc = 89.2, w = 0.038, GC = \{'H2O': 1\})$$

$$CO2 = component(name =' carbon - dioxide', Tc = 126.2, Pc = 34, Zc = 0.289, Vc = 89.2, w = 0.038, GC = 'H2O': 1)$$

$$CH4 = component(name = 'methan', Tc = 190.6, Pc = 45.99, Zc = 0.286, Vc = 98.6, w = 0.012, GC = \{'CH2': 1, 'CH2' : 1\})$$

$$mix = mixture(N2, CO2)$$

$$mix.add\_component(CH4)$$

Now the mixture is ready, and a model should be selected to compute phase equilibria and interfacial

Table 1: Required information for continuous or discontinuous modeling of a multicomponent mixture using Phasepy [10].

Type of modeling	**Models**	**Component definition**	**Interaction**
Discontinuous model $\phi-\gamma$	ideal gas, Abott Van Ness, NRTL, Wilson, UNIFAC	Critical temperature Critical pressure Critical compressibility	Activity coefficient models need specific interaction parameters
Continuous model $\phi-\phi$	VdW, PR, PR, PRSV, RK, RKS,	Critical temperature Critical pressure Acentric factor	Specific activity coefficient models

properties. Phasepy has provided two approaches to model a mixture. Throughout the $\phi-\gamma$ approach, firstly, a virial expansion is applied to model the vapor phase deviation (ϕ). Then, an activity coefficient model like NRTL, modified-UNIFAC, wilson, or Redlich-Kister, is utilized to explain liquid phase deviation (γ).

$$mix.NRTL(alpha, g, g1)$$

$$model = virialgamma(mix, virialmodel = 'Abbott', actmodel = 'nrtl')$$

The second approach uses a classical cubic EoS and a mixing rule to model the system. EoS functions in Phasepy account for van der Waals (vdW), Redlich-Knowg (RK), Redlich-Knowg-Soave (RKS), Peng-Robinson (PR), Peng-Robinson-Stryjek-Vera EoS (PRSV), and the Pàeloux et al. Moreover, as a mixing rule it is possible to apply quadratic mixing rule (QMR), modified-Huron-Vidal mixing rule (MHV), and Wong-Sandler mixing rule (WS) [10]. In essence, by selecting the preferred EoS and the appropriate mixing rule, it would be possible to calculate interfacial properties and phase equilibrium. For instance, based on Peng Robinson EoS, NRTL for activity coefficient, and MHV mixing rule we can write the following code scripts:

$$eos = preos(mix, \quad 'mhv1\_nrtl')$$

For more information about the EoS, mixing rules, and activity coefficient following references are suggested, [13], [14].

Tab. 1. presents the required information according to the type of modeling, including continuous or discontinuous using Phasepy.

3. Phasepy Two-phase equilibrium calculation

To calculate the isothermal isobaric flash composition of a two-phase mixture, Phasepy needs to revamp the phase compositions and solve Rachford-Rice mass balance continually. Therefore, the developer applied accelerated successive substitution (ASS) [15] and Halley's method to deal with the mentioned difficulties respectively.

Based on Rachford-Rice mass balance:

$$\sum_{i=1}^{c} \frac{z_i(K_i-1)}{1+\psi(K_i-1)} = 0 \qquad (19)$$

$$K_i = \frac{x_i^\alpha}{x_i^\beta} = \frac{\hat{\phi}_i^\alpha}{\hat{\phi}_i^\beta} \qquad (20)$$

α and β represent two phases and ψ depicts the fraction of phase β. In the case of not convergency especially in higher pressures, the algorithm is altered with a second-order procedure to find the minimum Gibbs free energy [10].

$$\sum_{i=1}^{c}(F_i^\alpha ln\hat{f}_i^\alpha + F_i^\beta ln\hat{f}_i^\beta) \qquad (21)$$

Where F and $\hat{f}$ refer to the number of the mole and effective fugacity respectively, and i is the species index. Apart from the flash calculation, Phasepy possesses the ability to calculate saturation points. Computing bubble points and dew points in a two-phase vapor-liquid-equilibrium (VLE) is based on the suggested method by Mollerup [6]. In this way throughout an inner loop phase compositions are updated using ASS, and the outer loop utilizes the quasi-Newton method to recalculate the pressure or the temperature. in the case of slow convergence, SciPy optimization routines are applied to solve equations 4, and 5 based on the iteration factor K, equilibrium constant.

$$f_i = lnK_i + ln\hat{\phi}_i^v(y,T,P) - ln\hat{\phi}_i^l(y,T,P)$$

$$i = 1,2,\dots,c \qquad (22)$$

$$f_{c+1} = \sum_{i=1}^{c}(y_i - x_i) \qquad (23)$$

The author of Phasepy presented Fig. 1 as the algorithm of Phasepy in computing bubble points or dew points. It is worth saying that the model requires the user to guess an initial value for saturation point, though the algorithm has ability enough for tolerating the initial guesses with large errors.

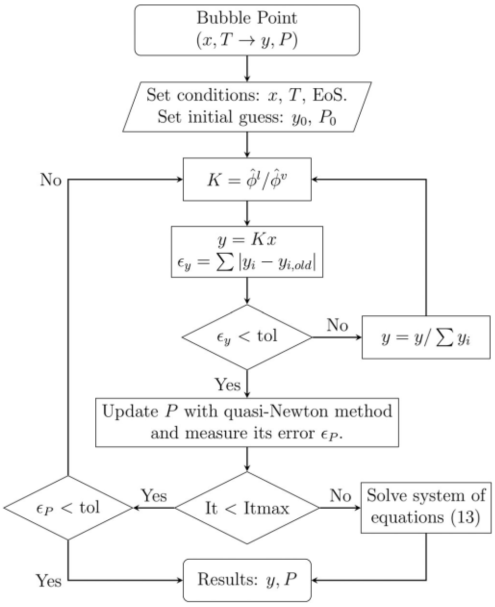
Bubble Point
$(x, T \to y, P)$
Set conditions: x, T, EoS.
Set initial guess: y_0, P_0
No
$K = \hat{\phi}^l/\hat{\phi}^v$
$y = Kx$
$\epsilon_y = \sum |y_i - y_{i,old}|$
$\epsilon_y <$ tol
No
$y = y/\sum y_i$
Yes
Update P with quasi-Newton method and measure its error ϵ_P.
$\epsilon_P <$ tol
Yes
It < Itmax
No
Solve system of equations (13)
Yes
Results: y, P

Figure 1: proposed algorithm for calculating the saturation points in Phasepy [10].

The following code scripts show how the saturation points or flash points can be calculated.

$flash(x\_guess, y\_guess, equilibrium, Z, T, P, model)$
$bubbleTy(y\_guess, T\_guess, X, P, model)$
$bubblePy(y\_guess, P\_guess, X, T, model)$
$dewPx(x\_guess, P\_guess, y, T, model)$
$dewTx(x\_guess, T\_guess, y, P, model)$

Where:

- x_guess(array): mole fraction of phase 1 (initial guess)
- y_guess(array): mole fraction of phase 2 (initial guess)
- T_guess (float): equilibrium temperature [K] (initial guess)
- P_guess (float): equilibrium pressure [bar] (initial guess)
- equilibrium(string): Two-phase system, including ‘LL’ (liquid-liquid) or ‘LV’ (liquid-vapor)
- Z(array): Overall mole fractions of components
- T(float): temperature [K]
- P(float): Pressure [bar]
- model(object): prepared model based on EoS and mixing rule (eos)

4. Results and Discussion

To evaluate how much the developed model is reliable to predict phase equilibrium three two-component mixtures at two pressures and two four-component mixtures are studied and outputs are compared with the experimental results.

In this way, average absolute relative deviation (AARD) and absolute maximum deviation (AMD) are studies.

$$AARD = \frac{1}{n}\sum_{i=i}^{n}\frac{|Exp_i - Cal_i|}{Exp_i} \times 100 \qquad (24)$$

$$AMD = Max(|Exp_i - Cal_i|) \qquad (25)$$

$$i = 1, 2, \ldots, n$$

Knudsen et. al. [16] investigated the most appropriate mixing rule, including the Huron-Vidal rule, the MHV-model, the Schwarzentruber - Galivel-Solasttouk - Renon rule, and the density-dependent local composition rule, for SRK EoS. The authors showed that the Huron-Vidal rule and modified Huron-Vidal rule can be used as the best mixing rule. On the other hand, Pedersen et. al. [17] also suggested that when the system is only containing hydrocarbons and sour gas simple QMR can be adequate. Therefore, for binary mixtures, the Modified Huron Vidal mixing rule and Quadratic Mixing Rule are compared while SRK is used as the EoS, and in four-components mixtures, QMR is utilized as the mixing rule and two EoS, including SRK and PR, are compared.

Marlus et. al. [18] studied binary mixtures of Benzene + Cyclohexane, Benzene + Chlorobenzene, and Cyclohexane + Chlorobenzene at the pressure of 101.5 and 40 kPa. T-x,y diagrams for the mixtures are drawn in Fig. 2, 3, 4, and 5.

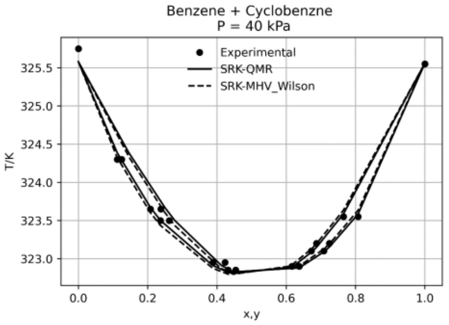

Figure 2: T-x,y diagram for Benzene (1) + Cyclohexane (2) at 40 kPa

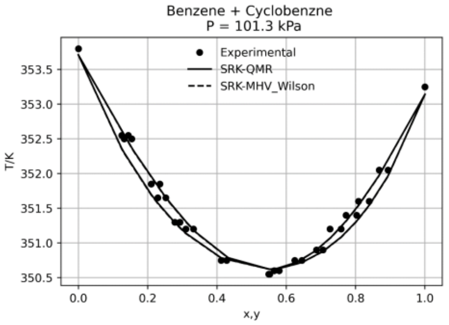

Figure 3: T-x,y diagram for Benzene (1) + Cyclohexane (2) at 101.3 kPa

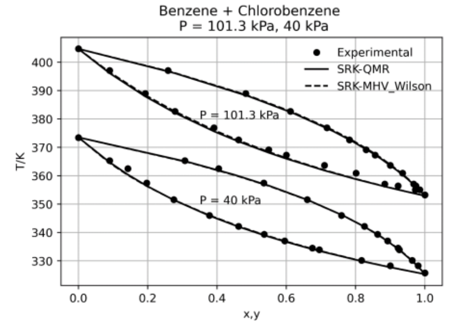

Figure 4: T-x,y diagram for Benzene (1) + Chlorobenzene (2) at P = 40 kPa, and P = 101.3kPa

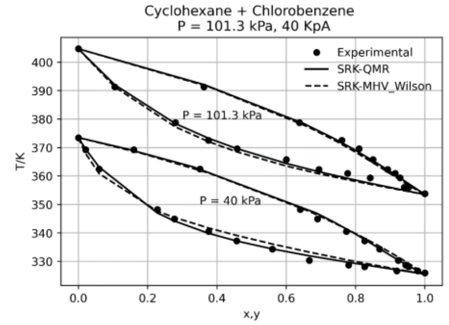

Figure 5: T-x,y diagram for Cyclohexane (1) + Chlorobenzene (2) at P = 40 kPa, and P = 101.3kPa

Thomas et. al. [19] experimentally studies the phase equilibrium of LNG. The authors indeed performed experiments on two different four-component mixtures, and then tried to model the behaviors using PR EoS. This study modeled these two mixtures using the Phasepy package. P-x-y diagrams are shown in Fig. 6 and 7.

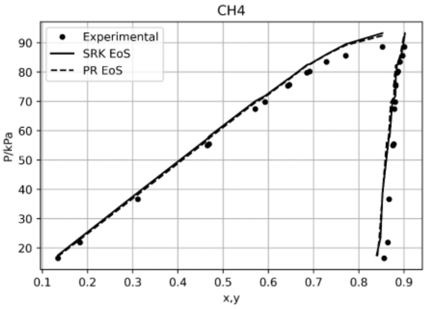

Figure 6: P-x,y diagram for CH4 in CH4, C2H6, C3H8, and n-C4H10 mixture

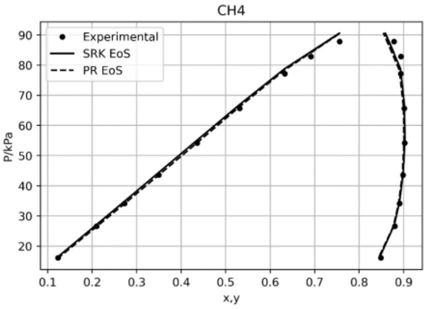

Figure 7: P-x,y diagram for CH4 in CH4, C2H6, C3H8, and i-C4H10 mixture

For all modelled mixture, AARD and AMD are calculated and shown in Table. 2. Therefore, it is possible to compare the consistency of EoS and mixing rules for different mixtures.

Table 2: calculated AARD and AMD for different mixtures

Mixture	P [kPa]	EoS	Mix. rule	AARD (%)	AMD
Benzene + Cyclohexane	40	SRK	QMR	2.87	0.020
			MHV	3.09	0.021
Benzene + Cyclohexane	101.3	SRK	QMR	2.49	0.019
			MHV	2.26	0.019
Benzene + Chlorobenzene	40	SRK	QMR	1.42	0.043
			MHV	1.55	0.046
Benzene + Chlorobenzene	101.3	SRK	QMR	1.73	0.019
			MHV	1.43	0.021
Cyclohexane + Chlorobenzene	40	SRK	QMR	2.93	0.045
			MHV	2.88	0.030
Cyclohexane + Chlorobenzene	101.3	SRK	QMR	1.77	0.045
			MHV	2.06	0.043
CH4, C2H6, C3H8, and n-C4H10		SRK	QMR	0.79	0.029
		PR		0.83	0.032
CH4, C2H6, C3H8, and i-C4H10		SRK	QMR	0.30	0.019
		PR		0.53	0.023

Based on Tab. 2, at the pressure of 40 kPa, SRK-OMR is the better model to predict the equilibrium of the Benzene (1) + Cyclohexane (2) mixture. However, at pressure 101.3 kPa, SRK-MHV-Wilson is the better model.

In the case of the Benzene + Chlorobenzene mixture, Fig. 4, in lower pressure, SRK-QMR is fitted and experimental results are perfectly predicted. However, at Pressure 101.3 kPa SRK-MHV_Wilson is the better model to predict the system.

For Cyclohexane + Chlorobenzene mixture as shown in Fig. 5, in lower pressure, SRK-MHV_Wilson is the better model to be fitted with the experimental results. Moreover, at Pressure 101.3 kPa SRK-QMR and SRK-MHV_Wilson are more closed, but, SRK-QMR is the better model.

Results show that PR and SRK EoS both, are an adequate choice for modeling, but in CH4, C2H6, C3H8, and i-C4H10 mixture, and, for CH4 in CH4, C2H6, C3H8, and n-C4H10 mixture SRK is the better EoS for modeling.

5. Conclusion

Knowing the interfacial properties and fluid phase equilibria are required to study the fluid behavior. In addition, thermodynamic models enable scientists to investigate the feasibility of an operation and process. However, finding an appropriate thermodynamic model and algorithm for computing the phase equilibria and interfacial properties is challenging. Although these days some commercial software has been developed to solve these difficulties, there are still some restrictions, including licenses fee, limitations in manipulating the defined process and lack of square gradient theory (SGT) for interfacial descriptions. Therefore, scientists have been provoked to apply alternative solutions like homemade programs.

Phasepy is a scientifically defined open-source package in python for computational thermodynamics. This package has been developed based on the most popular and reliable theories to calculate the interfacial properties and phase equilibria. The simplicity and accuracy of Phasepy enable studies to compare different EoS, mixing rules, etc.

This study tries to evaluate how much the developed methods based on Phasepy are able to predict the behavior of multicomponent mixtures. In this way, four binary mixtures and two four-components mixtures are modeled. Then the developed models are validated based on the performed experimental results. The results depict that Phasepy could be a solution for thermodynamic modeling, if an appropriate EoS, mixing rule, and activity coefficient model were selected.

References

[1] J. D. Seader and Ernest J. Henley, *Separation-Process-Principles-with-Applications-Using-Process-Simulators*, vol. 1, no. 4. 2006.

[2] P. C. Wankat, *Separation Process Engineering*, vol. 49, no. 11. 2004.

[3] E. Hendriks *et al.*, "Industrial requirements for thermodynamics and transport properties," *Industrial and Engineering Chemistry Research*, vol. 49, no. 22, 2010, doi: 10.1021/ie101231b.

[4] "Applied Thermodynamics of Fluids," *Chemistry International -- Newsmagazine for IUPAC*, vol. 33, no. 2, 2014, doi: 10.1515/ci.2011.33.2.27a.

[5] J. Prausnite, T. Anderson, E. Grens, C. Eckert, R. Hsieh, and J. O'Connell, "Computer calculations for multicomponent vapor-liquid and liquid-liquid equilibrium.," 1980, doi: 10.1016/0378-3812(81)80016-7.

[6] J. Mollerup and M. L. Michelsen, *Thermodynamic Models: Fundamentals & Computational Aspects*. 2004.

[7] A. Mejía, E. A. Müller, and G. Chaparro Maldonado, "SGTPy: A Python Code for Calculating the Interfacial Properties of Fluids Based on the Square Gradient Theory Using the SAFT-VR Mie Equation of State," *Journal of Chemical Information and Modeling*, vol. 61, no. 3, 2021, doi: 10.1021/acs.jcim.0c01324.

[8] J. M. Garrido, A. Mejía, M. M. Piñeiro, F. J. Blas, and E. A. Müller, "Interfacial tensions of industrial fluids from a molecular-based square gradient theory," *AIChE Journal*, vol. 62, no. 5, 2016, doi: 10.1002/aic.15190.

[9] "Statistical mechanics of phases, interfaces, and thin films," *Choice Reviews Online*, vol. 33, no. 09, 1996, doi: 10.5860/choice.33-5162.

[10] G. Chaparro and A. Mejía, "Phasepy: A Python based framework for fluid phase equilibria and interfacial properties computation," *Journal of Computational Chemistry*, vol. 41, no. 29, 2020, doi: 10.1002/jcc.26405.

[11] J. O. Valderrama, "The state of the cubic equations of state," *Industrial and Engineering Chemistry Research*, vol. 42, no. 8. 2003. doi: 10.1021/ie020447b.

[12] J. M. Smith, H. C. van Ness, M. M. Abbot, and M. T. . Swihart, *Introduction to Chemical Engineering Thermodynamics Eight Edition*. 2018.

[13] J. M. Smith, H. C. van Ness, M. M. Abbot, and M. T. . Swihart, *Introduction to Chemical Engineering Thermodynamics Eight Edition*. 2018.

[14] J. O. Valderrama, "The state of the cubic equations of state," *Industrial and Engineering Chemistry Research*, vol. 42, no. 8. 2003. doi: 10.1021/ie020447b.

[15] C. M. Crowe and M. Nishio, "Convergence promotion in the simulation of chemical processes—the general dominant eigenvalue method," *AIChE Journal*, vol. 21, no. 3, 1975, doi: 10.1002/aic.690210314.

[16] K. Knudsen, E. H. Stenby, and A. Fredenslund, "A comprehensive comparison of mixing rules for calculation of phase equilibria in complex systems," *Fluid Phase Equilibria*, vol. 82, no. C, 1993, doi: 10.1016/0378-3812(93)87159-X.

[17] S. Valiollahi, B. Kavianpour, S. Raeissi, and M. Moshfeghian, "A new Peng-Robinson modification to enhance dew point estimations of natural gases," *Journal of Natural Gas Science and Engineering*, vol. 34, 2016, doi: 10.1016/j.jngse.2016.07.049.

[18] M. P. Rolemberg and M. A. Krahenbuhl, "Vapor-liquid equilibria of binary and ternary mixtures of benzene, cyclohexane, and chlorobenzene at 40.0 kPa and 101.3 kPa," *Journal of Chemical and Engineering Data*, vol. 46, no. 2, 2001, doi: 10.1021/je000059l.

[19] T. J. Hughes *et al.*, "High pressure multi-component vapor-liquid equilibrium data and model predictions for the LNG industry," *Journal of Chemical Thermodynamics*, vol. 113, 2017, doi: 10.1016/j.jct.2017.05.023.