2nd Project: Vapor-Liquid Equilibrium Calculations of CH_4/CO_2 Binary Mixture

Kim, Seonggyun

Department of Chemical Engineering, Hanyang University

CHE 3006: Chemical Engineering Thermodynamics 2

Bae, Youngchan

November 2, 2021

Table of Contents

Table of Figures

1 Introduction

Understanding and predicting thermodynamic properties of CH_4/CO_2 binary mixture is crucial in many industrial chemical processes, for example modeling reservoir processes under injection by CO_2 in carbon sequestration process by direct carbon injection [1] and storage of CO_2 in solid gas hydrates by replacing CH_4 from natural accumulations [2].

In this work, methods to calculate vapor-liquid equilibrium (VLE) of the CH_4/CO_2 binary system are reviewed, from simple models such as Henry's law and Lewis/Randall Rule to more complex and accurate models given by cubic equations of state (CEOS). Along with the first-ever CEOS, van der Waals equation of state (EOS), two other two-parameter CEOSs were used to improve the accuracy of the calculation.

Two-parameter CEOSs can be expressed in the generic form in which the parameters are calculated from critical properties of the substance for pure species, and from critical properties of each component and a proper mixing rule for mixtures. From analytic or iterative solution for compressibility factor, useful thermodynamic properties can be derived, such as fugacity and fugacity coefficient. In this project, calculation of fugacity coefficients from compressibility factors obtained by CEOSs are used to calculate VLE of the given system.

To further improve the accuracy of CEOS, a proper mixing rule has to be introduced. In this work, binary interaction parameter (BIP) is used as to improve the calculation by modifying the "attractive term" in the PR EOS. Determining the optimal BIP resulted in significant improvement of VLE calculations for limited application.

2 Simple Models for VLE

2.1 Asymptotes: Henry's Law and Lewis/Randall Rule

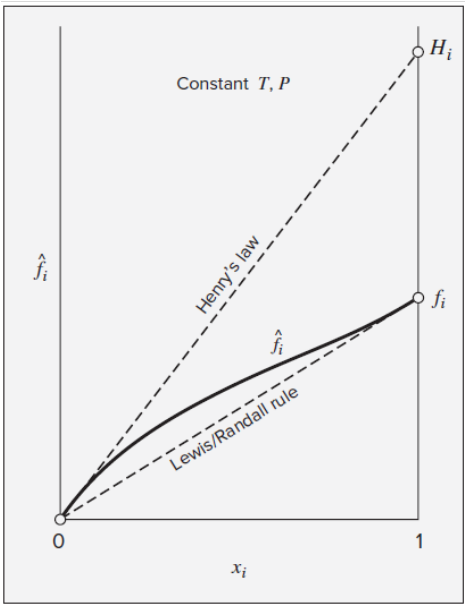

Figure 1. Henry's law and Lewis/Randall Rule

Figure 1 displays two simple models for VLE: Henry's law and Lewis/Randall rule along with experimental data of fugacity of a species i in a mixture. Each model shows a very simple linear correlation between $\hat{f}_i$ and x_i, which agrees with the experimental data at the limits of $x_i \to 0$ and $x_i \to 1$ respectively. This correspondence can be understood as the result of the behavior of actual mixture approaching the two "asymptotes" given by the models at both end limits of fraction of species i in the mixture, i.e., when species i is extremely dilute in the mixture and when the composition approaches pure species i.

The slope of the asymptote drawn by Henry's law, Henry's constant $\mathcal{H}_i$, is calculated by the slope of fugacity curve with respect to mole fraction of species i at $x_i \to 0$. Since both numerator and denominator approach zero at the limit, It can be calculated by applying l'Hôpital's rule:

$$\frac{\hat{f}_i}{x_i} = \mathcal{H}_i = \lim_{x_i \to 0} \frac{\hat{f}_i}{x_i} = \left(\frac{d\hat{f}_i}{dx_i}\right)_{x_i \to 0} \tag{1}$$

On the other hand, Lewis/Randall rule can be derived as the same limit condition for Henry's law for the other species in a binary solution by Gibbs/Duhem equation of chemical potential:

$$x_1 d\mu_1 + x_2 d\mu_2 = x_1\left(RTd\ln\hat{f}_1\right) + x_2\left(RTd\ln\hat{f}_2\right) = 0$$

Diving by dx_1 and taking limit,

$$\lim_{x_1 \to 1}\left(x_1\frac{d\ln\hat{f}_1}{d\,x_1} + x_2\frac{d\ln\hat{f}_2}{-d\,x_2}\right) = \frac{1}{f_1}\left(\frac{d\,ln\,\hat{f}_1}{d\,x_1}\right)_{x_1=1} - \left(\frac{x_2}{f_2}\frac{d\ln\hat{f}_2}{d\,x_2}\right)_{x_2=0} = 0$$

Since $\left(\frac{x_2}{f_2}\frac{d\ln\hat{f}_2}{d\,x_2}\right)_{x_2=0} = 1$ as shown in eq. (1), $\left(\frac{d\,ln\,\hat{f}_1}{d\,x_1}\right)_{x_1=1} = f_1$, which corresponds to the in the Lewis/Randall Rule at $x_1 = 1$ in **Figure 1**.

2.2 Henry's Law and Lewis/Randall Rule in Microscopic Viewpoint

2.2.1 Henry's Law in Microscopic Viewpoint

Henry's Law states that partial pressure in a vapor phase of the dilute component at equilibrium is linearly proportional to the mole fraction of the component dissolved in the ideal-solution liquid. For VLE, Henry's law considers the equilibrium of an ideal gas (low pressure) and very dilute solution to correlate fugacities of a species in each phase.

In microscopic viewpoint, the "very dilute solution" condition can be interpreted as a solution in which the concentration of a "solute" species is extremely low so that a solute molecule is surrounded by "solvent" molecules only, thus having solute-solvent molecular interactions only. Besides, by assuming the ideal gas, Henry's law is only applicable to VLE in low pressure range. Here, the limiting factor of the solubility of the solute in the liquid is the availability of molecular collisions of solute molecules from the vapor to the vapor-liquid interface, i.e., partial pressure of the species in the gas phase. When the pressure itself is very low, only small number of solute molecules in the gas phase end up colliding into the vapor-liquid surface, and once they hit the surface, the resistance of dissolving into the liquid is insignificant due to its low concentration.

2.2.2 Lewis/Randall Rule in Microscopic View

Lewis/Randall rule is built upon Raoult's Law, which considers VLE of an ideal gas and an ideal solution. Ideal solution requires the components to have similar molecular characteristics such as size, shape and polarity [3]. From the perspective of solute, a single molecule of dilute solute is surrounded by the abundant solvent molecules. From the perspective of solvent species, if the solute is very dilute in the solution, the solvent essentially experiences no difference from its pure state since the surroundings of a solvent molecule is mostly those of the same species, thus obeying Raoult's law.

3 Vapor/Liquid Equilibrium

Since van der Waals proposed the first ever cubic equation of state in 1873, rigorous studies and developments have been made to establish CEOS with higher accuracy. In this report, vdW EOS and two other two-parameter CEOSs were employed to predict VLE of CH_4/CO_2 mixture in three temperatures: SRK and PR EOSs, which are the most popular and the most used thermodynamic models in academia and industry as of now [4]. These two EOSs are suitable for this project because of their simplicity that enables relatively fast and simple computing and adequacy for lightweight hydrocarbons and non-polar compounds [5]. To incorporate the three cubic EOSs in one algorithm, a generic CEOS model with EOS-specific parameters is introduced.

3.1 VLE Calculation of pure species from generic CEOS

To accommodate all three CEOSs, a generic cubic equations of state model is used for VLE calculations [3]:

$$P = \frac{RT}{V-b} - \frac{a[T]}{(V+\sigma b)(V+\epsilon b)} \tag{3.1}$$

For a given equation, ϵ and σ are pure numbers and $a[T]$ and b are substance-dependent functions of critical properties, derived from mathematical conditions:

$$\left(\frac{\partial P}{\partial V}\right)_{T;cr} = 0$$

$$\left(\frac{\partial^2 P}{\partial V^2}\right)_{T;cr} = 0$$

The results are:

$$a[T] = \Psi \frac{\alpha(T_r) R^2 T_c^2}{P_c}$$

$$b = \Omega \frac{RT_c}{P_c}$$

Where pure numbers Ψ and Ω, and $a(T_r)$, empirical function of reduced temperature, are specifically set for each EOS. The parameter assignments for vdW, SRK, and PR EOS are summarized are summarized in **Table 1**.

Table 1. Summary of parameters for thee cubic EOSs: vdW, RK, and PR EOS

EOS	$\alpha(T_r)$	σ	ϵ	Ω	Ψ
vdW(1873)	1	0	0	$1/8$	$27/64$
SRK(1972)	$\alpha_{SRK}(T_r;\omega)$	1	0	0.08664	0.42748
PR(1976)	$\alpha_{PR}(T_r;\omega)$	$1+\sqrt{2}$	$1-\sqrt{2}$	0.07780	0.45724

$$\alpha_{SRK}(T_r;\omega) = \left[1 + (0.480 + 1.574\omega - 0.176\omega^2)\left(1 - T_r^{0.5}\right)\right]^2$$

$$\alpha_{PR}(T_r;\omega) = \left[1 + (0.37464 + 1.54226\omega - 0.26992\omega^2)\left(1 - T_r^{0.5}\right)\right]^2$$

Now, the calculation procedure is introduced step-by-step. To acquire a phase diagram, calculating P^{sat} is repeated for each T from T_{min} to T_{max}. The temperature limits are essentially set to cover the temperature range of given experimental data, however, for vdW EOS, the range had to be extended to complete a full diagram although the diagram deviates significantly from the experimental data. For each CEOS and species, the temperature range is divided by n points to repeat the calculation n times in total. In this project, to acquire clear visualization for the entire diagram and allow for precise calculation, n value was chosen to be 10^4.

Step 1: Calculate Z^l and Z^v by the EOS with initially assumed value of P^{sat}

Manipulating **eq. (3.1)** to represent Z and expanding with respect to Z yields a cubic equation:

$$Z = \frac{PV}{RT}$$

$$Z^3 + \left((\epsilon+\sigma)\beta - 1 - beta\right)Z^2 + (\epsilon\sigma\beta^2 - (\epsilon+\sigma)\beta - (\epsilon+\sigma)\beta^2 + q\beta)Z - \epsilon\sigma\beta^2 - \epsilon\sigma\beta^3 - q\beta^2 = 0$$

Where β and q are calculated from parameters shown in **Table 1**.

$$\beta = \frac{bP}{RT}$$

and

$$q = \frac{a}{bRT}$$

In this project, the P range where the above equation has three real roots was considered, so that the largest real root is assigned as Z^v and the smallest Z^l .

$$Z^v = \max(Z_{real})$$

$$Z^l = \min(Z_{real})$$

Step 2: Calculate ϕ_i^l, ϕ_i^l and K with Z^l and Z^v

Once Z^l and Z^v are evaluated, fugacity coefficients for each phase can be acquired:

$$\phi^l = \exp(Z^l - 1 - \ln(Z^l - \beta) - qI^l)$$

$$\phi^v = \exp(Z^v - 1 - \ln(Z^v - \beta) - qI^v)$$

Where if $\epsilon = \sigma$, I is given by

$$I = \frac{\beta}{Z + \epsilon\beta}$$

And if $\epsilon \neq \sigma$,

$$I = \frac{1}{\sigma - \epsilon} \ln \frac{Z + \sigma\beta}{Z + \epsilon\beta}$$

At equilibrium, by definition,

$$\phi_i^l = \phi_i^v = \phi_i^{sat}$$

To apply this equality to find the vapor pressure, K value is introduced.

$$K = \frac{\phi_i^l}{\phi_i^v}$$

If the system is in equilibrium, K value must equal unity.

Step 3: Determine P^{sat} for T

If $K > 1$, P^{sat} is increased, and if $K < 1$, P^{sat} is lowered and **Step 1-2** are repeated until K converges to unity. And when the sign of $(K-1)$ changes, the step size of changing P^{sat} is decreased by 10^{-1} until K strictly converges to 1 by absolute error smaller than 10^{-10} . Once P^{sat} is determined, the properties of saturated vapor and saturated liquid can be acquired from Z^l and Z^v that are used to yield $K = 1$.

Step 4: Repeat Step 1-3 with different T

Steps 1-3 are repeated for T values increasing by small steps in each iteration to complete the phase diagram. For better performance, P^{sat} value acquired from $T(i)$ is used as the initial Psat value for $T(i+1)$, where i indicates the iteration number.

3.2 VLE Calculation of mixture from generic CEOS

VLE calculation of mixture from vdW, SRK, and PR EOS follows the same approach to calculate Z^v and Z^l. However, new equations are required here to calculate fugacity coefficients of the components in the mixture, which will be introduced in **Step 2**. Here, T and x_i are known values. The procedure is repeated for x_1 increasing from zero to the point where bubbleline and dewline meets, i.e., when the both vapor and liquid phases of component i does not coexist anymore.

Step 1: Calculate Z_i^l and Z_i^v by the EOS with initially assumed values of P and y_1

This step is identical to **Step 1** in **3.1**. The initial value for the very first calculation, $x_1 = 0$, is set as the approximate value taken from the experiment data available at each temperature.

$$Z^3 + \big((\epsilon+\sigma)\beta - 1 - beta\big)Z^2 + (\epsilon\sigma\beta^2 - (\epsilon+\sigma)\beta - (\epsilon+\sigma)\beta^2 + q\beta)Z - \epsilon\sigma\beta^2 - \epsilon\sigma\beta^3 - q\beta^2 = 0$$

$$\beta = \frac{bP}{RT}$$

$$q = \frac{a}{bRT}$$

$$Z^v = \max(Z_{real})$$

$$Z^l = \min(Z_{real})$$

Step 2: Calculate $\widehat{\boldsymbol{\phi}}_{\boldsymbol{i}}^{\boldsymbol{l}}$, $\widehat{\boldsymbol{\phi}}_{\boldsymbol{i}}^{\boldsymbol{v}}$ and $\boldsymbol{K}$ with $\boldsymbol{Z}_{\boldsymbol{i}}^{\boldsymbol{l}}$ and $\boldsymbol{Z}_{\boldsymbol{i}}^{\boldsymbol{v}}$

Once Z^l and Z^v are evaluated, fugacity coefficients for each species in the mixture can be acquired. Fugacity coefficients of species in mixture can be calculated from following equations:

$$\hat{\phi}_i^l = \exp(Z_i^l - 1 - \ln(Z_i^l - \beta_i) - q_i I_i^l)$$

$$\hat{\phi}_i^v = \exp(Z_i^v - 1 - \ln(Z_i^v - \beta_i) - q_i I_i^v)$$

Where if $\epsilon = \sigma$, I^l and I^v are given by

$$I_i = \frac{\beta_i}{Z_i + \epsilon\beta_i}$$

And if $\epsilon \neq \sigma$,

$$I_i = \frac{1}{\sigma - \epsilon} \ln \frac{Z_i + \sigma\beta_i}{Z_i + \epsilon\beta_i}$$

At equilibrium, by definition,

$$x_i \hat{\phi}_i^l = y_i \hat{\phi}_i^v$$

To apply this equality to find the correct pressure, K_i value is introduced similarly as in **3.1**. However, this time, the sum of $K_i = \sum_i K_i x_i$ is used instead for the procedure.

$$K = \sum_i y_i = \sum_i K_i x_i = \sum_i \frac{\hat{\phi}_i^l}{\hat{\phi}_i^v} x_i$$

If each component in the system is in VLE, K must converge to unity.

Step 3: Determine $\boldsymbol{P}$ and $\boldsymbol{y_1}$

If $K > 1$, P is increased, and if $K < 1$, P is lowered and **Step 1-2** are repeated until K converges to unity. And when the sign of $(K - 1)$ changes, the step size of changing P is decreased by 10^{-1} until K strictly converges to 1 by absolute error smaller than 10^{-10}. Once P^{sat} is determined, the

properties of saturated vapor and saturated liquid can be acquired from Z^l and Z^v that are used to yield $K = 1$.

Step 4: Repeat Step 1-3 with different T or x_1

Steps 1-3 are repeated for x_i values increasing by small steps in each iteration to complete the VLE diagram. For better performance, P value acquired from $x_i(k)$ is used as the initial Psat value for $T(k+1)$, where k indicates the iteration number.

3.3 Van der Waals EOS

3.3.1 Pure VLE Calculation by vdW EOS

The phase diagrams acquired by the procedure introduced above with vdW EOS are shown in **Figure 2-3** along with experimental data:

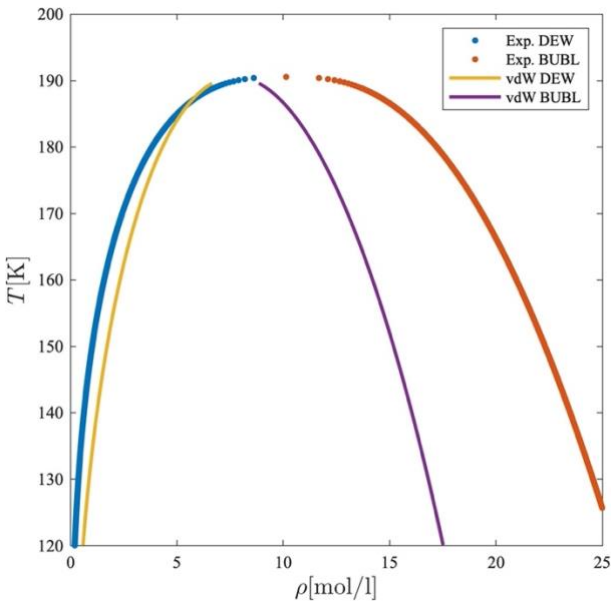

Figure 2. VLE of CH_4 by vdW EOS

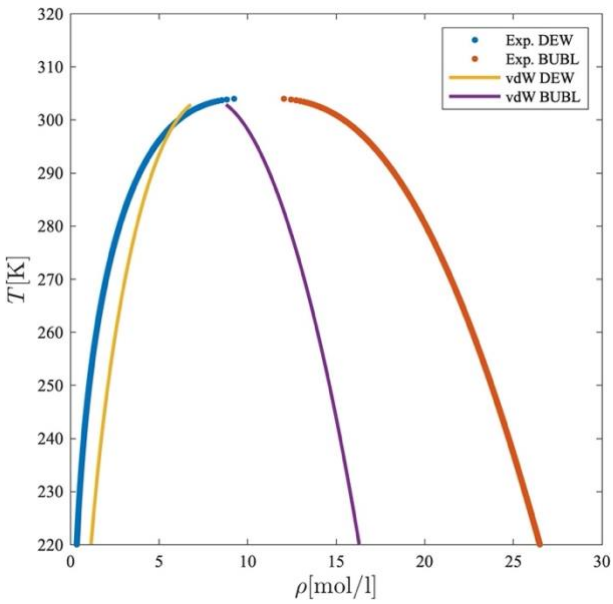

Figure 3. VLE of CO_2 by vdW EOS

As one can see from the figures, VLE calculation by vdW EOS deviates significantly in overall T range and only the dewline calculation gives reasonable approximation. As much as the equation itself is simple and intuitive, it fails to make precise prediction of dewline and yield practically unusable prediction for bubbleline. This is because of vdW EOS has only two constant parameters a and b for all T range. Although the parameters a and b do have physical meanings, namely that of molecular volumes and attraction [6], the limitation of vdW EOS arises from the lack of proper consideration of 1) deviation from the perfect gas (Ar, Xe, etc.) of each substance (i.e. acentric factor) and 2) temperature dependent behavior of substance. This limitation, however, has been overcome by succeeding modifications of the model, two of which are SRK and PR EOS discussed in this project.

3.3.2 Mixture VLE Calculation by vdW EOS

The binary VLE diagrams acquired by the procedure introduced above with vdW EOS are shown in **Figure 4-6** along with experimental data:

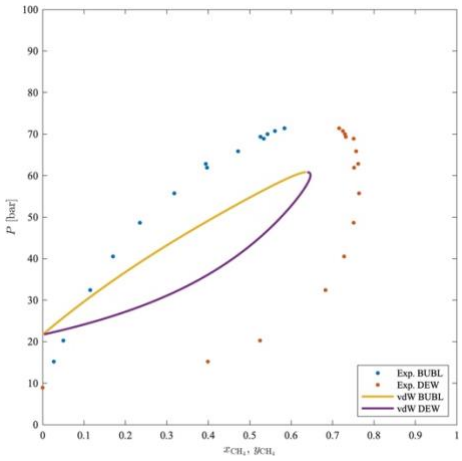

Figure 4. CH_4/CO_2 VLE by vdW EOS at 230K

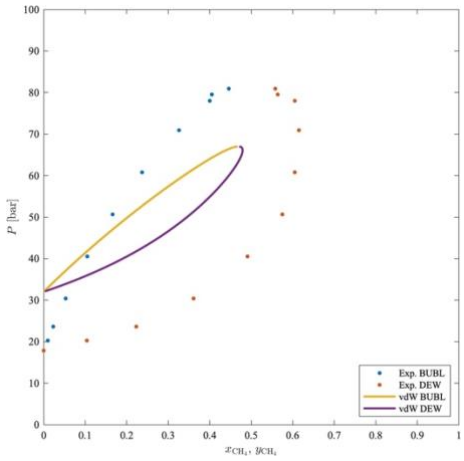

Figure 5. CH_4/CO_2 VLE by vdW EOS at 250K

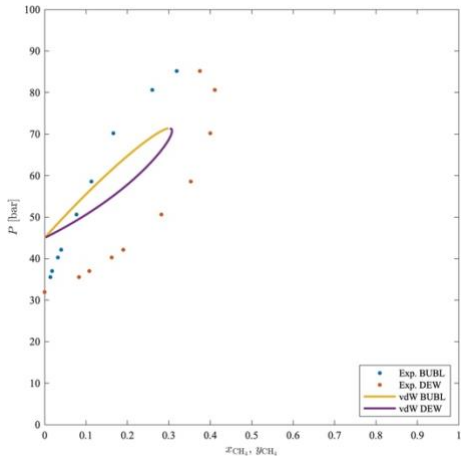

Figure 6. CH_4/CO_2 VLE by vdW EOS at 270K

The calculations by vdW EOS do yield a complete VLE diagram, however, it is practically useless for reasonable prediction of VLE since the accuracy is highly inaccurate. Thus, in-depth error analysis for calculations by vdW EOS are not included in this report. Only some general conclusions are acquired from this calculation by qualitative analysis:

- Although both bubbleline and dewline calculations are significantly inaccurate, the VLE diagram itself roughly lies within the actual VLE diagram, meaning vdW EOS predicts the liquid CH_4 to be more contained in the liquid mixture and the vapor CH_4 less contained in the vapor mixture. Thus, using vdW EOS to calculate VLE for distillation would predict less efficiency of separation process for CH_4/CO_2 mixture.
- The pressure value at $x_1 = 0$ calculated by vdW EOS is already highly inaccurate, being higher than the actual data. This is because the vapor pressure of CO_2 calculated by vdW EOS is higher than its actual vapor pressure. This error arises from the nonideality of CO_2 gas.

3.4 Modification to vdW EOS for Higher Accuracy

In order to make a modification to vdW EOS to improve its accuracy, it is important to review the history of cubic EOSs, starting from RK EOS. In 1949, Redlich and Kwong proposed a modified cubic EOS. This equation did not have a strong theoretical background but proved to give good results for many gaseous systems [7]:

$$P = \frac{RT}{V-b} - \frac{a/\sqrt{T}}{V(V+b)} \qquad \text{(RK EOS)}$$

In 1972, Soave proposed a modified version of RK EOS, which is now called SRK EOS, which satisfied great need of that time for a simple, generalized, and reasonably accurate EOS for the many repetitive calculations required in process simulations. It characterizes the alpha term multiplied to parameter a, which is a function of reduced temperature and acentric factor:

$$P = \frac{RT}{V-b} - \frac{a\alpha[T_r;\omega]}{V(V+b)} \qquad \text{(SRK EOS)}$$

$$\alpha_{SRK}[T_r;\omega] = \left(1 + (0.480 + 1.574\omega - 0.176\omega^2)\left(1-\sqrt{T}\right)\right)^2$$

In 1976, Peng and Robinson improved upon Soave's equation by recalculating the $\alpha(T_r,\omega)$ function and by modifying the volume dependency of the attractive term. These changes allowed them to obtain better results for liquid volumes and better representations of VLE for many mixtures [7]:

$$P = \frac{RT}{V-b} - \frac{a\alpha[T_r;\omega]}{V(V+b)+b(V-b)} \qquad \text{(PR EOS)}$$

$$\alpha_{PR}[T_r;\omega] = \left(1 + (0.37464 + 1.5422\omega - 0.26992\omega^2)\left(1-\sqrt{T}\right)\right)^2$$

Following the proposal of PR EOS, many modifications to the temperature-dependent attractive term has been made for higher accuracy in predicting vapor pressure of polar gases. There are also three-parameter EOSs too such as Patel-Teja EOS and its modifications which incorporates compressibility at the critical point to determine the third parameter [8], [9].

It has been generally agreed upon that two-parameter cubic EOSs can be adjusted to give good representations of PVT properties of pure polar fluids by modifying the temperature functionality of

the attractive term [7]. In this project, since both species in the given binary system are nonpolar substances, the original alpha functions proposed with SRK and PR EOS are kept.

In following sections **3.5** and **3.5**, VLE calculations previously done by vdW EOS in **3.3** will be repeated in the same manner with SRK and PR EOS to assess the accuracy of each EOS. After the calculations, the better model will be chosen to be further modified to predict the present system even more accurately with the introduction of binary interaction parameter (BIP) in **3.7**.

3.5 Soave-Redlich-Kwong (SRK) EOS

3.5.1 Pure VLE Calculation by SRK EOS

The phase diagrams acquired by the procedure introduced previously with SRK EOS are shown in **Figure 7** and **8** along with experimental data:

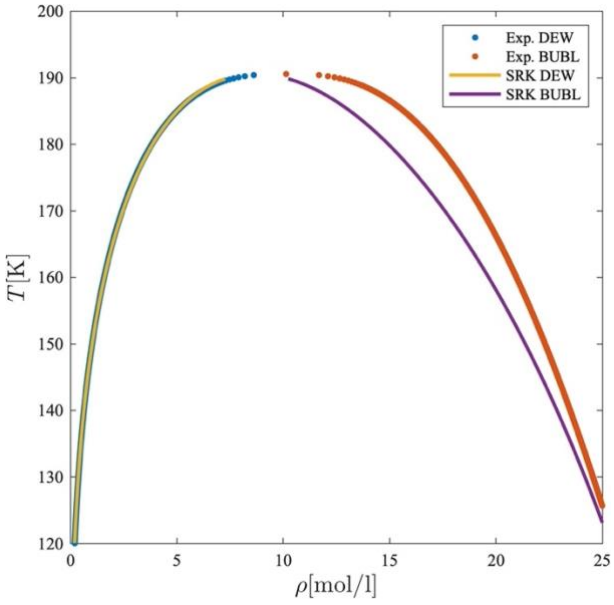

Figure 7. VLE of CH_4 by SRK EOS

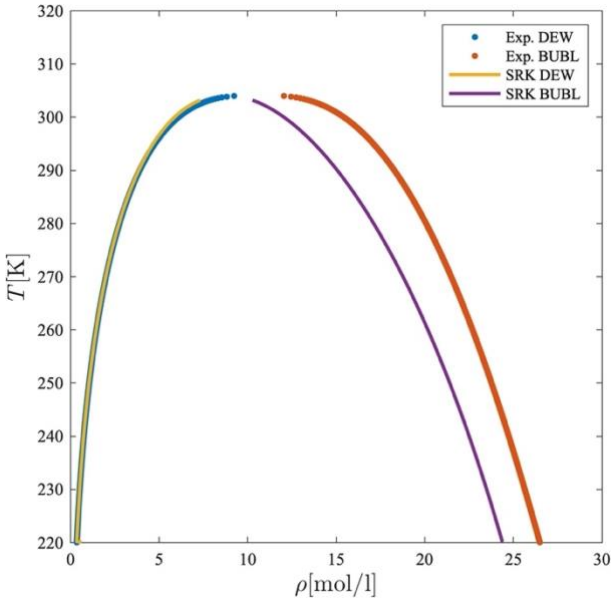

Figure 8. VLE of CO_2 by SRK EOS

In **Figure 7-8**, it can be found that SRK EOS gives a very accurate prediction for dewlines, i.e. saturated vapor phase of both species. However, it still fails to accurately predict bubblelines. This again shows the difficulty of liquid phase prediction by CEOS. However, it is noticeable that the calculation gives reasonably accurate result for bubbleline of CH_4 at lower temperature.

3.5.2 Mixture VLE Calculation by SRK EOS

The binary VLE diagrams acquired by the procedure introduced previously with SRK EOS are shown in **Figure 9-11** along with experimental data:

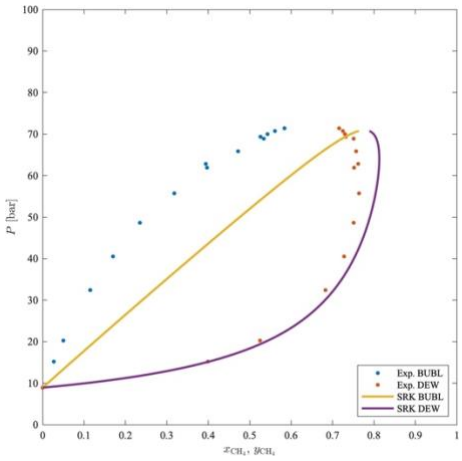

Figure 9. CH_4/CO_2 VLE by SRK EOS at 230K

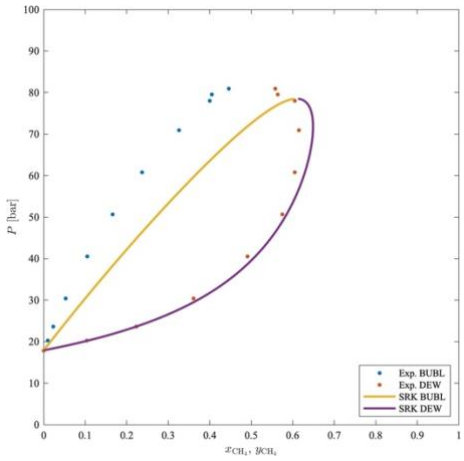

Figure 10. CH_4/CO_2 VLE by SRK EOS at 250K

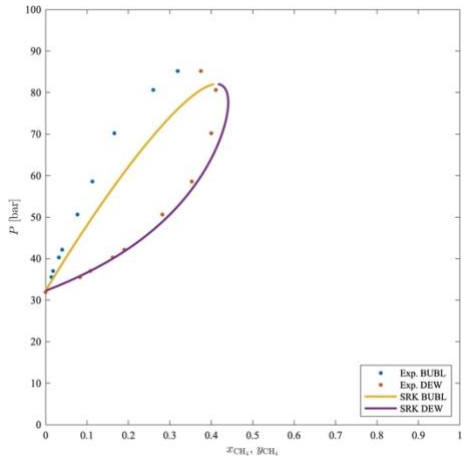

Figure 11. CH_4/CO_2 VLE by SRK EOS at 270K

Compared to the calculation by vdW EOS, it is noticeable that the VLE diagrams acquired by SRK EOS are in general more in agreement with the experimental data, especially at the dewline. This also corresponds with highly accurate prediction of dewline for pure species as shown in **Figure 7-8**. Because SRK EOS successfully predicts vapor pressure of pure CO_2, the calculation shows exact agreement with the experimental data at $x_1 = 0$.

By qualitative analysis,

- it is noted that the calculated bubblelines in general lie below the actual bubblelines, which indicates the calculation by SRK EOS predicts that CH_4 in the liquid mixture is more dissolved in the liquid mixture than reality, which would predict the efficiency of distillation to be lower than the actually possible maximum efficiency.

3.6 Peng-Robinson (PR) EOS

3.6.1 Pure VLE Calculation by PR EOS

The phase diagrams acquired by the procedure introduced previously with PR EOS are shown in **Figure 12-13** along with experimental data:

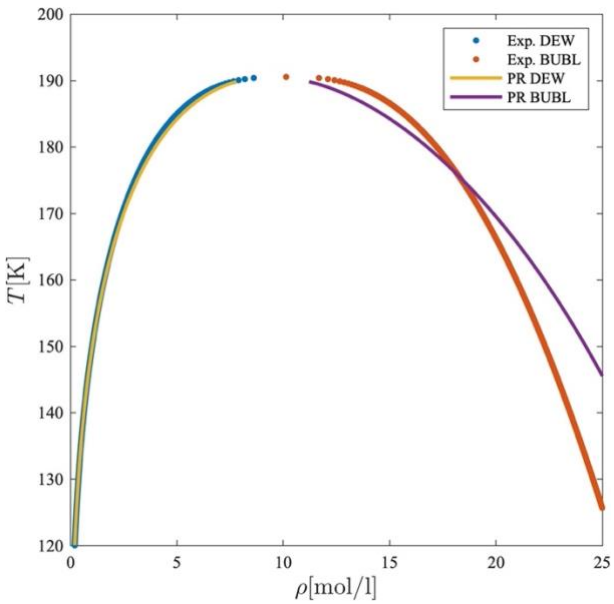

Figure 12. VLE of CH_4 by PR EOS

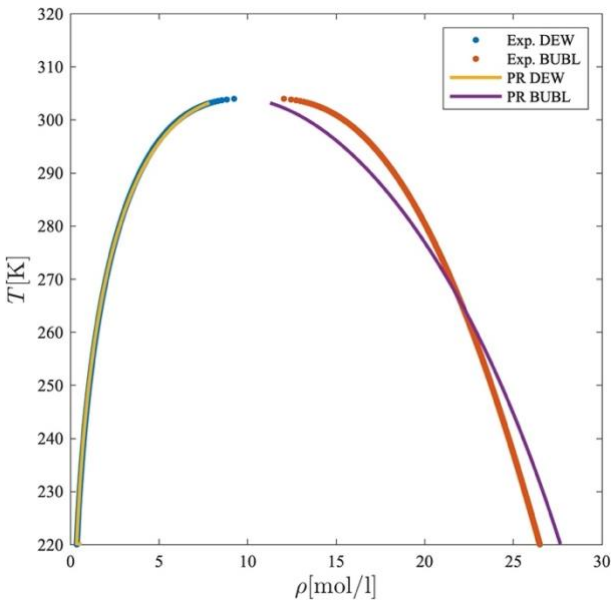

Figure 13. VLE of CO_2 by PR EOS

In **Figure 12-13**, similarly to SRK EOS, it can be found that PR EOS gives a very accurate prediction for dewlines. But it still fails to accurately predict bubblelines except for the point in each diagram where the calculated bubbleline and experimental bubbleline intersect each other. Although the modification from SRK EOS to PR EOS is purely arbitrary without no actual physical logic supporting it, the calculation result shows PR EOS does give a better approximation for the given system especially at higher temperature range near the critical temperature.

3.6.2 Mixture VLE Calculation by PR EOS

The binary VLE diagrams acquired by the procedure introduced previously with PR EOS are shown in **Figure 14-16** along with experimental data:

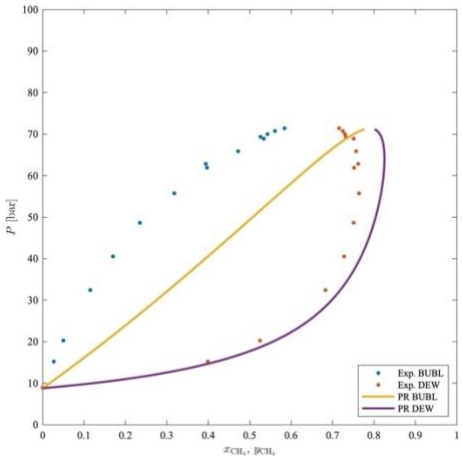

Figure 14. CH_4/CO_2 VLE by PR EOS at 230K

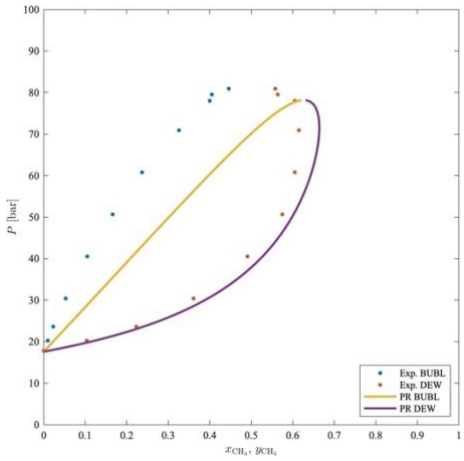

Figure 15. CH_4/CO_2 VLE by PR EOS at 250K

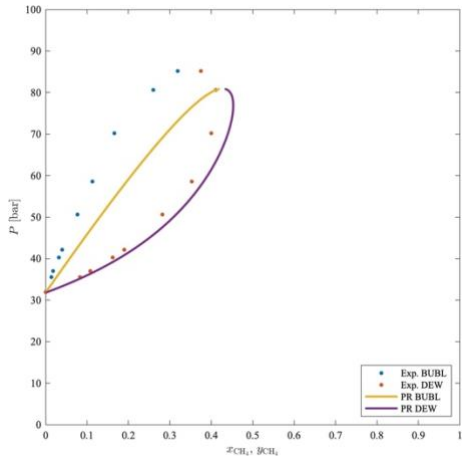

Figure 16. CH_4/CO_2 VLE by PR EOS at 270K

It is difficult to find any distinctive difference between the VLE calculations by SRK EOS and PR EOS. Similarly to SRK EOS, by qualitative analysis, it is noted that the calculated bubblelines in general lie below the actual bubblelines.

After comparing the accuracy of both SRK and PR models, PR EOS was chosen to be used with a modified mixing rule in this project for its successful prediction of pure substances, as shown in **Figures 12-13** compared to that of SRK EOS shown in **Figures 7-8**. This, however, is essentially an arbitrary and practical choice that is made to allow further modifications focused onto one single EOS.

3.7 Binary Interaction Parameter (BIP)

PR EOS can be used to carry out VLE calculations in even higher accuracy with introduction of binary interaction parameter (BIP) to the classical van der Waals mixing rules. Classical mixing rules are given as:

$$a_{12} = x_1 x_2 \sqrt{a_1 a_2}$$

With the introduction of BIP, the attractive term is modified as:

$$a_{12} = (1 - k_{12}) x_1 x_2 \sqrt{a_1 a_2}$$

Where k_{12} is BIP, models for which have been rigorously developed and studied. In this section, two temperature-independent models and a temperature-dependent model for BIP were reviewed. After applying the models for the system of interest in this project, the model with the least root-mean square error (RMSE) calculated by each average absolute relative deviation (AARD) of P and y_1 will be chosen, from which specific BIP for the given mixture at 230K, 250K, and 270K will be determined by regression.

3.7.1 Temperature-independent Models

Model 1: Zhang et al. [10] proposed a simple temperature-independent model of BIP for PR-vdW model (PR EOS with vdW one-parameter mixing rules) for 119 binary systems. In the model, every pure component was considered to have a constant contribution k_i to k_{ij}, with a contribution coefficient w_i. BIP for CH_4/CO_2 proposed in their work is 0.0917.

$$k_{12} = 0.0917$$

Model 2: Fateen et al. [11] proposed a semi-empirical correlation for BIPs of PR-vdW model for the prediction of high-pressure VLE. BIP for the present system was proposed as 0.0919 in their work.

$$k_{12} = 0.0919$$

3.7.2 Temperature-dependent Model

Model 3: Xu et al. [12] reviewed several typically accepted correlations for BIP of mixtures containing light hydrocarbon species, considering the effects of temperature, pressure, carbon

number of hydrocarbon components, and acentric factor, and revealed that the effect of temperature is the most dominant. A simple correlation proposed by Xu et al. represents k_{12} as:

$$k_{12} = 0.5219 - 0.8254T_r + 0.4494T_r^3$$

Where

$$T_r = \frac{1}{2}(T_{r1} + T_{r2})$$

For the present system, k_{12} is given as 0.1367 for 230K, 0.1870 For 250K, and 0.2583 for 270K.

3.7.3 Error Analysis of Each Model

The accuracy of BIP models were evaluated by average absolute relative deviation (AARD) of *P* and y_1 calculated for given T and x1 compared to the available experimental data, and the root mean square error (RMSE) was evaluated for each BIP model. The AARD and RMSE values for each model with PR EOS are given in **Table 2** with the lowest RMSE results for each temperature marked with underlines.

Table 2. AARD and RMSE of three BIP models reviewed from literature

	Model 1			**Model 2**			**Model 3**		
***T* [K]**	230	250	270	230	250	270	230	250	270
k_{12}		0.0917			0.0919		0.1367	0.1870	0.2583
AARD of *P* (%)	17.97	9.43	4.94	17.93	9.39	4.92	7.99	11.19	19.72
AARD of y_1 (%)	8.83	10.87	10.98	8.81	10.84	10.95	4.65	6.17	21.41
RMSE (%)	21.01	15.53	12.71	20.97	15.48	<u>12.67</u>	<u>9.71</u>	<u>14.30</u>	32.71

Following conclusions are given by the error analysis:

- Temperature-independent models have higher accuracy at higher temperature. This is because given temperatures, 230K, 250K, and 270K are somewhat too low compared to the critical temperature of CO_2 ($T_{cr2} = 304.21K$). PR EOS itself determines its parameters from critical properties of given substance. Thus, for temperature far from critical temperature, its validity inevitably declines.

- Temperature-independent model (**Model 3**) have higher accuracy at lower temperature. This result is expected as Model 3 was specifically developed for very low temperature range. However, applying the model to not low enough temperatures resulted in significant deviation.

3.7.4 Determination of BIP for the Systems of Interest

Although all three models that were reviewed above gave poor prediction of VLE behavior of given binary mixture, they do give reasonable initial guess for finding optimal BIP value for each temperature. Starting with the BIP values with lowest RMSE in each temperature as an initial guess, the optimal BIP that minimizes RMSE for each temperature can be determined by computation. In this report, the optimal BIP values were determined to the accuracy of fourth decimal point by simple programming by MATLAB®.

The values and resulting VLE diagrams are presented in **Figure 17-19**. In comparison to Figures **14-16** (VLE calculation by PR EOS with vdW mixing rules), the accuracy of bubble line calculation is significantly improved by making the overall shape of the diagrams a bit "fuller".

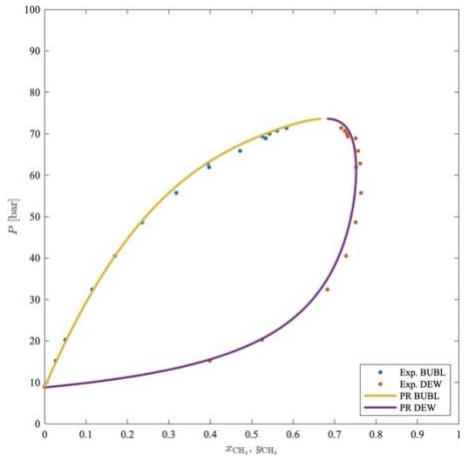

Figure 17. CH_4/CO_2 VLE by PR EOS at 230K, with $k_{12} = 0.1670$

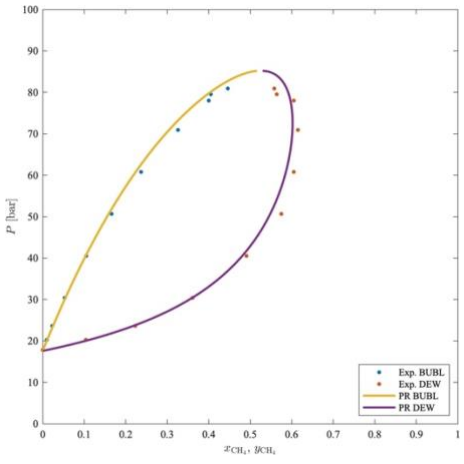

Figure 18. CH_4/CO_2 VLE by PR EOS at 250K, with $k_{12} = 0.1459$

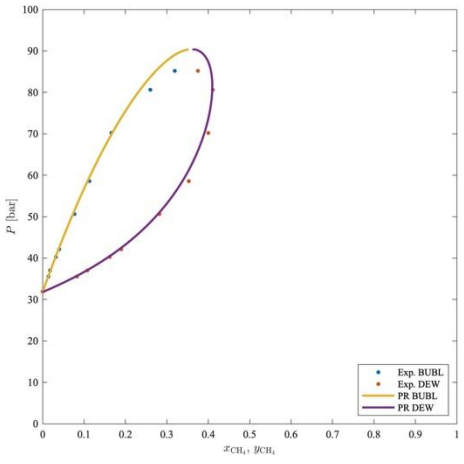

Figure 19. CH_4/CO_2 VLE by PR EOS at 270K, with $k_{12} = 0.1462$

However, in **Figure 19**, the diagram seems to deviate to the area above the experimental at high pressure range. Because the experimental data have limited number of datapoints (9 points for 270K), even small experimental measurement flaw can significantly affect the overall diagram. In addition, the regressed BIP values for 250K and 270K do not correspond with the generally accepted fact that BIP is a strong function of temperature [10]–[12], showing very little difference although temperature difference is relatively huge. This also supports that the experimental data themselves might have some flaw.

In this case, the bubblepoint and dewpoint that lie approximately at 70bar seemed to slightly depart from the overall shape, causing the distortion of regressed VLE diagram.

Therefore, for the calculation at 270K, an arbitrary value was found to bring the overall shape of the diagram closer to the experimental data other than the suspected flaw datapoints: $k_{12} = 0.13$ at 270K. The comparison of two calculation results is shown in **Figures 19-20** below:

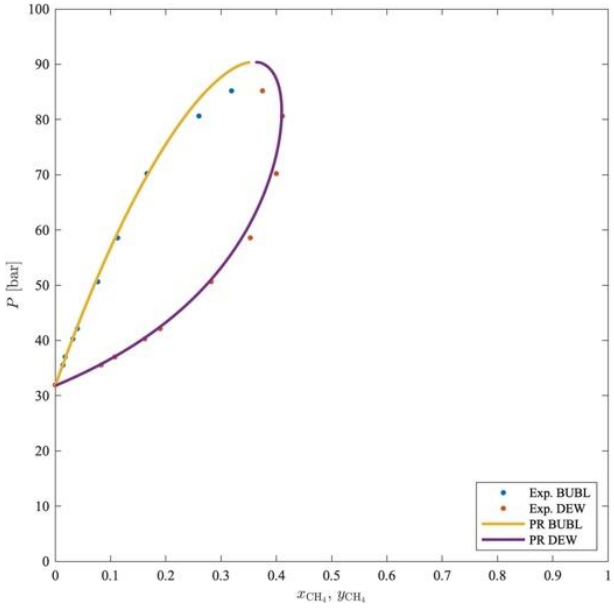

Figure 19. CH_4/CO_2 VLE by PR EOS at 270K, with $k_{12} = 0.1462$

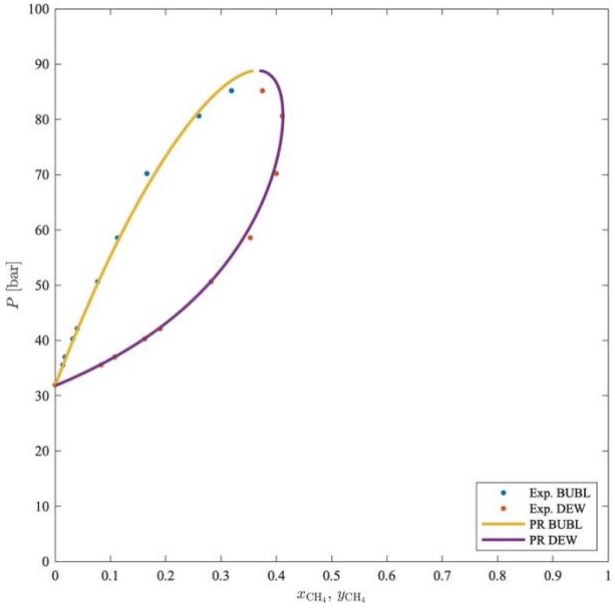

Figure 20. CH_4/CO_2 VLE by PR EOS at 270K, with $k_{12} = 0.13$

Comparing the result obtained from the calculation with newly determined BIP and the ones from the literature are summarized in **Table 3**. The RMSE have been significantly decreased by applying BIP values determined in this work.

Table 3. AARD and RMSE of calculations by BIP values from literature and from present work

	From Literature			Present Work			
T [K]	230	250	270	230	250	270	
k_{12}	0.1367	0.1870	0.0919	0.1670	0.1459	0.1462	0.13
AARD of P (%)	7.99	11.19	4.92	1.94	3.07	2.07	4.45
AARD of y_1 (%)	4.65	6.17	10.95	1.35	1.88	1.63	1.35
RMSE (%)	9.71	14.30	12.67	2.52	3.76	2.92	4.81

4 Conclusion

In this work, methods to calculate vapor-liquid equilibrium (VLE) of the CH_4/CO_2 binary system are reviewed, from simple models such as Henry's law and Lewis/Randall Rule to more complex and accurate models given by cubic equations of state (CEOS). Along with the first-ever CEOS, van der Waals equation of state (vdW EOS), two other two-parameter CEOSs were used to improve the accuracy of the calculation.

Three Two-parameter CEOSs, vdW, RK, and PR EOS were employed to carry out the VLE calculations for the given system at 230K, 250K, and 270K. While both SRK and PR EOS gave reasonable prediction for the systems with classical vdW mixing rules, this project focused on PR EOS to be further used with modified mixing rule in this project.

BIP reviewed from the literature gave predictions with RMSE ranges from 9% to 15%, which are not quite accurate, because of the discrepancy between the given system and the environments the models are specifically developed for. Therefore, new BIP for each temperature were determined in this project to successfully improve the accuracy, yielding RMSE ranging from 2% to 3%.

This project has limitation in that the BIP values are specifically viable for very specific systems. However, for the temperatures between/around 230K and 270K, the values can be used as reference values to regress a new model for BIP. For example, if a polynomial or exponential correlation of BIP and temperature for the temperature range is proposed with up to three parameters, the values determined in this project could be used to regress a reasonably accurate model.

5 References

[1] C. OLDENBURG, D. LAW, Y. LEGALLO, and S. WHITE, “Mixing of CO2 and CH4 in Gas ReservoirsCode Comparison Studies,” *Greenh. Gas Control Technol. - 6th Int. Conf.*, pp. 443–448, 2003, doi: 10.1016/b978-008044276-1/50071-4.

[2] J.-H. Wee, J.-I. Kim, I.-S. Song, B.-Y. Song, and K.-S. Choi, “Reduction of Carbon-Dioxide Emission Applying Carbon Capture and Storage Technology To Power Generation and Industry Sectors in Korea,” 대한환경공학회지, vol. 30, no. 9, pp. 961–972, 2008.

[3] J. M. Smith, H. C. Van Ness, M. M. Abbott, and M. T. Swihart, *Introduction to Chemical Engineering Thermodynamics*, 8th ed. McGraw-Hill Education, 2018.

[4] G. L. A. Forero and J. J. A. Velásquez, “A generalized cubic equation of state for non-polar and polar substances,” *Fluid Phase Equilib.*, vol. 418, pp. 74–87, 2016, doi: 10.1016/j.fluid.2015.09.045.

[5] A. Anderko, “4 Cubic and generalized van der waals equations,” *Exp. Thermodyn.*, vol. 5, no. C, pp. 75–126, 2000, doi: 10.1016/S1874-5644(00)80015-6.

[6] G. M. Kontogeorgis, R. Privat, and J. N. Jaubert, “Taking Another Look at the van der Waals Equation of State-Almost 150 Years Later,” *J. Chem. Eng. Data*, vol. 64, no. 11, pp. 4619–4637, 2019, doi: 10.1021/acs.jced.9b00264.

[7] J. O. Valderrama, “The State of the Cubic Equations of State,” *Ind. Eng. Chem. Res.*, vol. 42, no. 8, pp. 1603–1618, 2003, doi: 10.1021/ie020447b.

[8] N. C. Patel and A. S. Teja, “A new cubic equation of state for fluids and fluid mixtures,” *Chem. Eng. Sci.*, vol. 37, no. 3, pp. 463–473, 1982, doi: 10.1016/0009-2509(82)80099-7.

[9] L. A. Forero G. and J. A. Velásquez J., “A modified Patel-Teja cubic equation of state: Part I - Generalized model for gases and hydrocarbons,” *Fluid Phase Equilib.*, vol. 342, pp. 8–22, 2013, doi: 10.1016/j.fluid.2012.12.032.

[10] H. Zhang *et al.*, “A simple model for temperature-independent kij of the PR-vdW model for mixtures containing HCs, HFCs, PFCs, HFOs, CO2, RE170 and R13I1,” *Fluid Phase Equilib.*, vol. 425, pp. 374–384, 2016, doi: 10.1016/j.fluid.2016.06.029.

[11] S. E. K. Fateen, M. M. Khalil, and A. O. Elnabawy, “Semi-empirical correlation for binary interaction parameters of the Peng-Robinson equation of state with the van der Waals mixing rules for the prediction of high-pressure vapor-liquid equilibrium,” *J. Adv. Res.*, vol. 4, no. 2, pp. 137–145, 2013, doi: 10.1016/j.jare.2012.03.004.

[12] X. Xu, H. Chen, C. Liu, and C. Dang, “Prediction of the Binary Interaction Parameter of Carbon Dioxide/Alkanes Mixtures in the Pseudocritical Region,” *ACS Omega*, vol. 4, no. 8, pp. 13279–13294, 2019, doi: 10.1021/acsomega.9b01450.

6 Appendix

6.1 MATLAB® Codes

VLE_pure.m

```
clear all, clc
close all

load('datapure.mat')
set(groot,'defaultLineMarkerSize', 10, ...
    'defaultLineLineWidth', 2, ...
    'defaultAxesFontName', 'Times');

global R
R = 0.0831446261815324; % L bar / K mol

% CH4: species 1
Tc1 = 190.564;  % K
Pc1 = 45.992;   % bar
Dc1 = 10.139;   % mol/L
Vc1 = 1/Dc1;    % L/mol
 w1 = 0.01142;
% AC1 = [6.61184 389.9278 265.99];

% CO2: species 2
Tc2 = 304.1282;
Pc2 = 73.773;
Dc2 = 10.6249;
Vc2 = 1/Dc2;
 w2 = 0.22394;
AC2 = [7.5788, 863.35, 273.15];

% FIG1 = figure('Position',[0 10000 500 500]);

% EOS = input('vdW : 1\n SRK : 2\nPR : 3\n\n');
for EOS = 1:3

    EOSname = {'vdW' 'SRK' 'PR'};
    EOSname = EOSname{EOS};
    fprintf(['\n< ' EOSname ' EOS >\n'])
    plotsettings = {'interpreter','latex','fontsize',14};

    T_lim1 = [120 200];
    T_lim2 = [220 320];
    n = 10^3;

    [rho1_v, rho1_l, T1] = VLE(T_lim1(1), T_lim1(2), n, 10, Pc1, Tc1, w1, EOS);

    VLE1 = figure('Position',[0 10000 400 400]);
    plot(CH4_vap(:,3),CH4_vap(:,1),'.'); hold on
    plot(CH4_liq(:,3),CH4_liq(:,1),'.');
    plot(rho1_v,T1,'-'); hold on
    plot(rho1_l,T1,'-');
    axis([0 25 T_lim1(1) T_lim1(2)])
    pbaspect([1 1 1])
    xlabel('$\rho \mathrm{ [mol/l]}$',plotsettings{:})
    ylabel('$T \mathrm{ [K]}$',plotsettings{:})
    legend('Exp. DEW' ,'Exp. BUBL',[EOSname ' DEW'], [EOSname ' BUBL'])
    exportgraphics(gca,['VLE_1_' EOSname '.jpg'],'Resolution',300)

    [rho2_v, rho2_l, T2] = VLE(T_lim2(1), T_lim2(2), n, 20, Pc2, Tc2, w2, EOS);
```

```
    VLE2 = figure('Position',[0 0 400 400]);
    plot(CO2_vap(:,3),CO2_vap(:,1),'.'); hold on
    plot(CO2_liq(:,3),CO2_liq(:,1),'.');
    plot(rho2_v,T2,'-'); hold on
    plot(rho2_l,T2,'-');
    axis([0 30 T_lim2(1) T_lim2(2)])
    pbaspect([1 1 1])
    xlabel('$\rho \mathrm{ [mol/l]}$',plotsettings{:})
    ylabel('$T \mathrm{ [K]}$',plotsettings{:})
    legend('Exp. DEW' ,'Exp. BUBL',[EOSname ' DEW'], [EOSname ' BUBL'])
    exportgraphics(gca,['VLE_2_' EOSname '.jpg'],'Resolution',300)

end



function [rho_v, rho_l, T] = VLE(Tmin, Tmax, n, P_in, Pc, Tc, w, EOS)

    global R

    for i = 1:n
        T(i) = Tmin + (Tmax - Tmin)*(i/n);
        Tr = T(i)/Tc;

%         fprintf('\n Calculating for T = %.4fK >',T(i))

        if EOS == 1
            [Omega, Ksi, eps, sig, alpha] = vdW;
        elseif EOS == 2
            [Omega, Ksi, eps, sig, alpha] = SRK(Tr,w);
        elseif EOS == 3
            [Omega, Ksi, eps, sig, alpha] = PR(Tr,w);
        end

        a = Ksi*alpha*R^2*Tc^2/Pc;
        b = Omega*R*Tc/Pc;

        [rho_v(i), rho_l(i), P_sat(i), stop] = Psat(eps, sig, a, b, T(i), P_in);
        P_in = P_sat(i);
%         fprintf(' %7f, %7f, %7f', rho_v(i), rho_l(i), P_sat(i))

        if stop == 1
            break
        end
    end

    rho_v = rho_v(1:i-1);
    rho_l = rho_l(1:i-1);
    T = T(1:i-1);

end

function [rho_v, rho_l, Psat, stop] = Psat(eps, sig, a, b, T, P_in)
    global R
    P(1) = P_in;
    power = -3;
    stop = 0;
    for i = 1:10000

        beta = b*P(i)/(R*T);
        q = a/(b*R*T);

        [Z_l, Z_v] = Z_CEOS(eps, sig, beta, q);

        I_l = I_calc(eps, sig, beta, Z_l);
        I_v = I_calc(eps, sig, beta, Z_v);

        phi_l = exp( Z_l - 1 - log(Z_l - beta) - q*I_l);
```

```
        phi_v = exp( Z_v - 1 - log(Z_v - beta) - q*I_v);

        K(i) = phi_l / phi_v;

        if abs( K(i)-1 ) < 10^(-15)
            if Z_l == Z_v
                stop = 1;
                break
            end
            break
        elseif K(i) > 1
            P(i+1) = P(i) + 10^power;
        elseif K(i) < 1
            P(i+1) = P(i) - 10^power;
        end

        if i > 2
            if ( K(i) - 1 )*( K(i-1) - 1 ) < 0
                power = power - 1;
            end
        end

    end

    Psat = P(i);
    rho_l = Psat/(Z_l*R*T);
    rho_v = Psat/(Z_v*R*T);

end

function [Omega, Ksi, eps, sig, alpha] = vdW
    eps = 0;
    sig = 0;
    alpha = 1;
    Omega = 1/8;
    Ksi = 27/64;
end

function [Omega, Ksi, eps, sig, alpha] = SRK(Tr,w)
    eps = 0;
    sig = 1;
    alpha = ( 1 + (0.480 + 1.574*w - 0.176*w^2) * (1 - Tr^(0.5)) )^2;
%     alpha0 = Tr^(-0.201158) * exp(0.141599*(1-Tr^2.29528));
%     alpha1 = Tr^(-0.660145) * exp(0.500315*(1-Tr^2.63165));
%     alpha = alpha0 + w*(alpha1 - alpha0);
    Omega = 0.08664;
    Ksi = 0.42748;
end

function [Omega, Ksi, eps, sig, alpha] = PR(Tr,w)
    eps = 1-sqrt(2);
    sig = 1+sqrt(2);
    alpha = (1+(0.37464+1.57226*w-0.26992*w^2)*(1-Tr^(1/2)))^2;
    Omega = 0.07780;
    Ksi = 0.45724;
end

function I = I_calc(eps, sig, beta, Z)
    if sig == eps
        I = beta/(Z+eps*beta);
    else
        I = (1/(sig - eps)) * (log((Z + sig*beta)/(Z + eps*beta)));
    end
end

function [Zl, Zv] = Z_CEOS(eps,sig,beta,q)

    Z = roots([1 ...
```

```
        ( (eps+sig)*beta - 1 - beta ) ...
        ( eps*sig*beta^2 - (eps+sig)*beta - (eps+sig)*beta^2 + q*beta) ...
        ( - eps*sig*beta^2 - eps*sig*beta^3 - q*beta^2) ]);

    Zi = Z==real(Z);
    Z_real = Z(Zi);
    Zv = max(Z_real);
    Zl = min(Z_real);
end
```

< vdW EOS >

< SRK EOS >

< PR EOS >

Published with MATLAB® R2022a

VLE_binary.m

Contents


```
clear all, clc, close all

load('datamix.mat')
set(groot,'defaultLineMarkerSize', 10, ...
    'defaultLineLineWidth', 2, ...
    'defaultAxesFontName', 'Times');

global R

R = 0.0831446261815324; % L bar / K mol
T_data = [230, 250, 270]; % K

% CH4: species 1
Tc1 = 190.564;  % K
Pc1 = 45.992;   % bar
Dc1 = 10.139;   % mol/L
Vc1 = 1/Dc1;    % L/mol
 w1 = 0.01142;

% CO2: species 2
Tc2 = 304.21;
Pc2 = 73.829955;
Dc2 = 10.6249;
Vc2 = 1/Dc2;
 w2 = 0.22394;
```

Choice of an EOS and BIP (Binary Interaction Parameter)

```
EOS = input('\nChoose an EOS\n(1:vdW, 2:SRK, 3:PR)\n>> ');    % Choose which EOS you want
to use
EOSname = {'vdW' 'SRK' 'PR'};
EOSname = EOSname{EOS};
P_initial = [15 20 30]; % Initial values for faster calculation

fprintf('\nChoose a BIP model (0 to 4)\n');
fprintf(' 0: k12 = 0 (Classical Mixing Rule)\n')
fprintf(' 1: k12 = 0.919\n');
fprintf(' 2: k12 = 0.979\n');
fprintf(' 3: k12 = 0.5219 - 0.8254*Tr + 0.4494*Tr^3\n');

% this work
fprintf(' 4: k12 = 0.7834 - 0.9783493*Tr + 0.35693431*Tr^2\n');

BIP_model_num = input('>> ');

fprintf(['\n< ' EOSname ' EOS with BIP model no.' num2str(BIP_model_num) ' >\n'])
```

Calculate for three temperatures: 230K, 250K, 270K

```
for Ti = 1:3
    T = T_data(Ti);
    Tr1 = T/Tc1;
    Tr2 = T/Tc2;
    Tr = (Tr1+Tr2)/2;
    fprintf('Calculating VLE for %dK...\n',T)
```

```
if EOS == 1
    [Omega, Ksi, eps, sig, alpha1, alpha2] = vdW;
elseif EOS == 2
    [Omega, Ksi, eps, sig, alpha1, alpha2] = SRK(Tr1,Tr2,w1,w2);
elseif EOS == 3
    [Omega, Ksi, eps, sig, alpha1, alpha2] = PR(Tr1,Tr2,w1,w2);
end

k12 = BIP(BIP_model_num,Tr);

a1 = Ksi*alpha1*R^2*Tc1^2/Pc1;
a2 = Ksi*alpha2*R^2*Tc2^2/Pc2;
a12 = (1-k12)*a1^(1/2)*a2*(1/2);

b1 = Omega*R*Tc1/Pc1;
b2 = Omega*R*Tc2/Pc2;

% x1 = 0.0001, 0.0002, 0.0003, ...
n = 10^4;

for k = 1:n

    x1 = k/n;
    x2 = 1 - x1;

    if k == 1
        P_new(1) = P_initial(Ti);
        y1_new(1) = 0.1;
    else
        P_new(1) = P_res(k-1);
        y1_new(1) = y1_res(k-1);
    end

    power = - 2;

    for j = 1:10000

        P = P_new(j);
        y1 = y1_new(j);

        parameters = [a1, a2, a12, b1, b2, eps, sig];
        [K(j), y1_new(j+1)] = K_converge(T, x1, P, y1, parameters);

        if j > 1
            if ( K(j) -1 )*( K(j-1) -1 ) < 0
                power = power - 1;
            end
        end

        if abs( K(j) - 1 ) < 10^(-10)
            P_res(k) = P_new(j);
            y1_res(k) = y1_new(j+1);
            break
        elseif K(j) > 1
            P_new(j + 1) = P + 10^power;
        elseif K(j) < 1
            P_new(j + 1) = P - 10^power;
        end

        if j == 10000
            fprintf('K did not converge to 1\n')
        end
    end

    if k > 1        % Stop the procedure if P decreases with increasing x1
        if P_res(k) < P_res(k-1)
            break
        end
```

```
        end

    end

    x1 = linspace(1/n, 1, n);
    lim = k-2;
    x1 = x1(1:lim);

    VLE = figure('Position',[0 10000 500 500]);
    plot(dataset{Ti}(:,1),dataset{Ti}(:,3),'.'); hold on
    plot(dataset{Ti}(:,2),dataset{Ti}(:,3),'.');
    plot(x1,P_res(1:lim)); hold on
    plot(y1_res(1:lim),P_res(1:lim));
    axis([0 1 0 100])
    pbaspect([1 1 1])
    legend('Exp. BUBL','Exp. DEW',[EOSname ' BUBL'],[EOSname '
DEW'],'Location','southeast')
    xlabel('$x_{\mathrm{CH_4}}$, $y_{\mathrm{CH_4}}$','Interpreter','latex')
    ylabel('$P$ ${\mathrm {[bar]}}$','Interpreter','latex')
    exportgraphics(gca,['mixVLE_', EOSname, '_BIP_', num2str(BIP_model_num), '_',
num2str(T), 'K.jpg'],'Resolution',300)

end

fprintf('\nComplete!\n\n')
```

- **Functions**

```
function k12 = BIP(num, Tr)

    if num == 0
        k12 = 0;
    elseif num == 1
        k12 = 0.919;
    elseif num == 2
        k12 = 0.979;
    elseif num == 3
        k12 = 0.5219 - 0.8254*Tr + 0.4494*Tr^3;
    elseif num == 4
        k12 = 0.7834 - 0.9783493*Tr + 0.35693431*Tr^2;
    end

end

function [Omega, Ksi, eps, sig, alpha1, alpha2] = vdW
    eps = 0;
    sig = 0;
    alpha1 = 1;
    alpha2 = 1;
    Omega = 1/8;
    Ksi = 27/64;
end

function [Omega, Ksi, eps, sig, alpha1, alpha2] = SRK(Tr1,Tr2,w1,w2)
    eps = 0;
    sig = 1;
    alpha1 = ( 1 + (0.480 + 1.574*w1 - 0.176*w1^2) * (1 - Tr1^(0.5)) )^2;
    alpha2 = ( 1 + (0.480 + 1.574*w2 - 0.176*w2^2) * (1 - Tr2^(0.5)) )^2;
    Omega = 0.08664;
    Ksi = 0.42748;
end

function [Omega, Ksi, eps, sig, alpha1, alpha2] = PR(Tr1,Tr2,w1,w2)
    eps = 1-sqrt(2);
    sig = 1+sqrt(2);
    alpha1 = (1+(0.37464+1.57226*w1-0.26992*w1^2)*(1-Tr1^(1/2)))^2;
    alpha2 = (1+(0.37464+1.57226*w2-0.26992*w2^2)*(1-Tr2^(1/2)))^2;
```

```
    Omega = 0.07780;
    Ksi = 0.45724;
end

function I = I_calc(eps, sig, beta, Z)
    if sig == eps
        I = beta/(Z+eps*beta);
    else
        I = (1/(sig - eps)) * (log((Z + sig*beta)/(Z + eps*beta)));
    end
end

function Z_out = Zv_CEOS(eps,sig,beta,q)

    Z = roots([1 ...
        ( (eps+sig)*beta - 1 - beta ) ...
        ( eps*sig*beta^2 - (eps+sig)*beta - (eps+sig)*beta^2 + q*beta) ...
        ( - eps*sig*beta^2 - eps*sig*beta^3 - q*beta^2) ]);

    Zi = Z==real(Z);
    Z_real = Z(Zi);
    Z_out = max(Z_real);

end

function Z_out = Zl_CEOS(eps,sig,beta,q)

    Z = roots([1 ...
        ( (eps+sig)*beta - 1 - beta ) ...
        ( eps*sig*beta^2 - (eps+sig)*beta - (eps+sig)*beta^2 + q*beta) ...
        ( - eps*sig*beta^2 - eps*sig*beta^3 - q*beta^2) ]);

    Zi = Z==real(Z);
    Z_real = Z(Zi);
    Z_out = min(Z_real);

end

function [K_res, y1_res] = K_converge(T, x1, P, y1_in, parameters)
    global R

    y1_new(1) = y1_in;
    x2 = 1 - x1;

    a1  = parameters(1);
    a2  = parameters(2);
    a12 = parameters(3);
    b1  = parameters(4);
    b2  = parameters(5);
    eps = parameters(6);
    sig = parameters(7);

    for i = 1:100

        y1 = y1_new(i);
        y2 = 1 - y1;

        a_l = x1^2*a1 + 2*x1*x2*a12 + x2^2*a2;
        a_v = y1^2*a1 + 2*y1*y2*a12 + y2^2*a2;
        b_l = x1*b1 + x2*b2;
        b_v = y1*b1 + y2*b2;
        beta_l = b_l*P/(R*T);
        beta_v = b_v*P/(R*T);
        q_l = a_l/(b_l*R*T);
        q_v = a_v/(b_v*R*T);
        qbar1_l = q_l*( (2*x1*a1 + 2*x2*a12)/a_l - b1/b_l );
        qbar2_l = q_l*( (2*x2*a2 + 2*x1*a12)/a_l - b2/b_l );
        qbar1_v = q_v*( (2*y1*a1 + 2*y2*a12)/a_v - b1/b_v );
```

```
        qbar2_v = q_v*( (2*y2*a2 + 2*y1*a12)/a_v - b2/b_v );

        Z_l = Zl_CEOS(eps, sig, beta_l, q_l);
        Z_v = Zv_CEOS(eps, sig, beta_v, q_v);

        I_l = I_calc(eps, sig, beta_l, Z_l);
        I_v = I_calc(eps, sig, beta_v, Z_v);

        phi1_l = exp( (b1/b_l)*(Z_l - 1) - log(Z_l - beta_l) - qbar1_l*I_l);
        phi2_l = exp( (b2/b_l)*(Z_l - 1) - log(Z_l - beta_l) - qbar2_l*I_l);

        phi1_v = exp( (b1/b_v)*(Z_v - 1) - log(Z_v - beta_v) - qbar1_v*I_v);
        phi2_v = exp( (b2/b_v)*(Z_v - 1) - log(Z_v - beta_v) - qbar2_v*I_v);

        K1 = phi1_l/phi1_v;
        K2 = phi2_l/phi2_v;

        K(i) = K1*x1 + K2*x2;
        y1_new(i + 1) = (K1*x1) / (K1*x1 + K2*x2);

        if i > 1
            if abs ( (K(i)-K(i-1)) / K(i-1)) < 10^(-10)
                break
            end
        end
    end

    K_res = K(i);
    y1_res = y1_new(i);

end
```

Published with MATLAB® R2022a

Error_analysis.m

```
clear all, clc
close all

load('datamix.mat')
set(groot,'defaultLineMarkerSize', 10, ...
    'defaultLineLineWidth', 2, ...
    'defaultAxesFontName', 'Times');

global R

R = 0.0831446261815324; % L bar / K mol
T_data = [230, 250, 270]; % K

% CH4: species 1
Tc1 = 190.564;  % K
Pc1 = 45.992;   % bar
Dc1 = 10.139;   % mol/L
Vc1 = 1/Dc1;    % L/mol
 w1 = 0.01142;

% CO2: species 2
Tc2 = 304.21;
Pc2 = 73.829955;
Dc2 = 10.6249;
Vc2 = 1/Dc2;
 w2 = 0.22394;

VLE = figure('Position',[0 10000 500 1000]);
plotsettings = {'interpreter','latex','fontsize',14};
% dataset{3,1} = dataset{3,1}([1:7,9:10],:);

n_point = [14,10,9];
% k12_me_SRK = [0.1523, 0.1294, 0.1321];
% k12_me_SRK = [0.1523, 0.1294, 0.11];
% k12_me_PR = [0.1670, 0.1459, 0.1462];
k12_me_PR = [0.1670, 0.1459, 0.13];

for Ti = 1:3
    clear P_res y1_res
    T = T_data(Ti);
    Tr1 = T/Tc1;
    Tr2 = T/Tc2;
    Tr = (Tr1+Tr2)/2;
    fprintf('\n< T = %d K >',T)

%     [Omega, Ksi, eps, sig, alpha1, alpha2] = vdW;
    [Omega, Ksi, eps, sig, alpha1, alpha2] = PR(Tr1,Tr2,w1,w2);
%     [Omega, Ksi, eps, sig, alpha1, alpha2] = SRK(Tr1,Tr2,w1,w2);

%     k12 = k12_me_SRK(Ti);

%     k12 = 0.0919;
%     k12 = 0.05818 + 0.04117*Tr
%     k12 = 0.5219 - 0.8254*Tr + 0.4494*Tr^3;
    k12 = k12_me_PR(Ti);
%     k12 = 0;

    a1 = Ksi*alpha1*R^2*Tc1^2/Pc1;
    a2 = Ksi*alpha2*R^2*Tc2^2/Pc2;
    a12 = (1-k12)*a1^(1/2)*a2*(1/2);

    b1 = Omega*R*Tc1/Pc1;
    b2 = Omega*R*Tc2/Pc2;

    for k = 1:n_point(Ti)
```

```
        x1 = dataset{Ti}(k+1,1);
        x2 = 1 - x1;

        if k == 1
            P_new(1) = 5;
            y1_new(1) = 0.1;
        else
            P_new(1) = P_res(k-1);
            y1_new(1) = y1_res(k-1);
        end

        power = -0.5;

        for j = 1:10000

            P = P_new(j);
            y1 = y1_new(j);

            parameters = [a1, a2, a12, b1, b2, eps, sig];

            [K(j), y1_new(j+1)] = K_converge(T, x1, P, y1, parameters);
%             fprintf('%.10f\n',K(j))
            if j > 1
                if ( K(j) -1 )*( K(j-1) -1 ) < 0
                    power = power - 0.5;
                end
            end

            if abs( K(j) - 1 ) < 10^(-10)
                P_res(k) = P_new(j);
                y1_res(k) = y1_new(j+1);
                break
            elseif K(j) > 1
                P_new(j + 1) = P + 10^power;
            elseif K(j) < 1
                P_new(j + 1) = P - 10^power;
            end

            if j == 10000
                fprintf('shit\n')
            end
        end
    end
    subplot(3,1,Ti)
    plot(dataset{Ti}(:,1),dataset{Ti}(:,3),'.'); hold on
    plot(dataset{Ti}(:,2),dataset{Ti}(:,3),'.');
    plot(dataset{Ti}(2:1+n_point(Ti),1),P_res,'-');
    plot(y1_res,P_res);
    axis([0 1 0 100])
    pbaspect([1 1 1])
    xlabel('$x_{\mathrm{CH_4}}$, $y_{\mathrm{CH_4}}$','Interpreter','latex')
    ylabel('$P$ ${\mathrm {[bar]}}$','Interpreter','latex')

    P_error{Ti} = 100 * abs( (P_res' -
dataset{Ti}(2:1+n_point(Ti),3)) ./dataset{Ti}(2:1+n_point(Ti),3) );
    y1_error{Ti} = 100 * abs( (y1_res' -
dataset{Ti}(2:1+n_point(Ti),2)) ./dataset{Ti}(2:1+n_point(Ti),2) );
    error{Ti} = sqrt(P_error{Ti}.^2 + y1_error{Ti}.^2);

end

error{3}
fprintf('\n 230K: y1 AARD = %5.2f %%', mean(y1_error{1}))
fprintf('\n 250K: y1 AARD = %5.2f %%', mean(y1_error{2}))
fprintf('\n 270K: y1 AARD = %5.2f %%\n', mean(y1_error{3}))

fprintf('\n 230K: P AARD = %5.2f %%', mean(P_error{1}))
```

```
fprintf('\n 250K: p AARD = %5.2f %%', mean(P_error{2}))
fprintf('\n 270K: P AARD = %5.2f %%\n', mean(P_error{3}))

fprintf('\n 230K: RMSE = %5.2f %%', mean(error{1}))
fprintf('\n 250K: RMSE = %5.2f %%', mean(error{2}))
fprintf('\n 270K: RMSE = %5.2f %%\n', mean(error{3}))

function [Omega, Ksi, eps, sig, alpha1, alpha2] = vdW
    eps = 0;
    sig = 0;
    alpha1 = 1;
    alpha2 = 1;
    Omega = 1/8;
    Ksi = 27/64;
end

function [Omega, Ksi, eps, sig, alpha1, alpha2] = SRK(Tr1,Tr2,w1,w2)
    eps = 0;
    sig = 1;
    alpha1 = ( 1 + (0.480 + 1.574*w1 - 0.176*w1^2) * (1 - Tr1^(0.5)) )^2;
    alpha2 = ( 1 + (0.480 + 1.574*w2 - 0.176*w2^2) * (1 - Tr2^(0.5)) )^2;
    Omega = 0.08664;
    Ksi = 0.42748;
end

function [Omega, Ksi, eps, sig, alpha1, alpha2] = PR(Tr1,Tr2,w1,w2)
    eps = 1-sqrt(2);
    sig = 1+sqrt(2);
    alpha1 = (1+(0.37464+1.57226*w1-0.26992*w1^2)*(1-Tr1^(1/2)))^2;
    alpha2 = (1+(0.37464+1.57226*w2-0.26992*w2^2)*(1-Tr2^(1/2)))^2;
    Omega = 0.07780;
    Ksi = 0.45724;
end

function Z_out = Zv_CEOS(eps,sig,beta,q)

    Z = roots([1 ...
        ( (eps+sig)*beta - 1 - beta ) ...
        ( eps*sig*beta^2 - (eps+sig)*beta - (eps+sig)*beta^2 + q*beta) ...
        ( - eps*sig*beta^2 - eps*sig*beta^3 - q*beta^2) ]);

    Zi = Z==real(Z);
    Z_real = Z(Zi);
    Z_out = max(Z_real);

end

function Z_out = Zl_CEOS(eps,sig,beta,q)

    Z = roots([1 ...
        ( (eps+sig)*beta - 1 - beta ) ...
        ( eps*sig*beta^2 - (eps+sig)*beta - (eps+sig)*beta^2 + q*beta) ...
        ( - eps*sig*beta^2 - eps*sig*beta^3 - q*beta^2) ]);

    Zi = Z==real(Z);
    Z_real = Z(Zi);
    Z_out = min(Z_real);

end

function [K_res, y1_res] = K_converge(T, x1, P, y1_in, parameters)
    global R

    y1_new(1) = y1_in;
    x2 = 1 - x1;

    a1  = parameters(1);
    a2  = parameters(2);
```

```
    a12 = parameters(3);
    b1  = parameters(4);
    b2  = parameters(5);
    eps = parameters(6);
    sig = parameters(7);

    for i = 1:100

        y1 = y1_new(i);
        y2 = 1 - y1;

        % fprintf('with y1 = %.4f:\n', y1)

        a_l = x1^2*a1 + 2*x1*x2*a12 + x2^2*a2;
        a_v = y1^2*a1 + 2*y1*y2*a12 + y2^2*a2;
        b_l = x1*b1 + x2*b2;
        b_v = y1*b1 + y2*b2;
        beta_l = b_l*P/(R*T);
        beta_v = b_v*P/(R*T);
        q_l = a_l/(b_l*R*T);
        q_v = a_v/(b_v*R*T);
        qbar1_l = q_l*( (2*x1*a1 + 2*x2*a12)/a_l - b1/b_l );
        qbar2_l = q_l*( (2*x2*a2 + 2*x1*a12)/a_l - b2/b_l );
        qbar1_v = q_v*( (2*y1*a1 + 2*y2*a12)/a_v - b1/b_v );
        qbar2_v = q_v*( (2*y2*a2 + 2*y1*a12)/a_v - b2/b_v );

        Z_l = Zl_CEOS(eps, sig, beta_l, q_l);
        Z_v = Zv_CEOS(eps, sig, beta_v, q_v);

        I_l = I_calc(eps, sig, beta_l, Z_l);
        I_v = I_calc(eps, sig, beta_v, Z_v);

        phi1_l = exp( (b1/b_l)*(Z_l - 1) - log(Z_l - beta_l) - qbar1_l*I_l);
        phi2_l = exp( (b2/b_l)*(Z_l - 1) - log(Z_l - beta_l) - qbar2_l*I_l);

        phi1_v = exp( (b1/b_v)*(Z_v - 1) - log(Z_v - beta_v) - qbar1_v*I_v);
        phi2_v = exp( (b2/b_v)*(Z_v - 1) - log(Z_v - beta_v) - qbar2_v*I_v);

        K1 = phi1_l/phi1_v;
        K2 = phi2_l/phi2_v;

        K(i) = K1*x1 + K2*x2;
        y1_new(i + 1) = (K1*x1) / (K1*x1 + K2*x2);
%         fprintf('%.5f     %.5f, %d\n', y1_new(i), K(i), i);

        if i > 1
            if abs ( (K(i)-K(i-1)) / K(i-1)) < 10^(-10)
%                 fprintf('%d\n',i)
                break
            end
        end
    end

    K_res = K(i);
    y1_res = y1_new(i);

    function I = I_calc(eps, sig, beta, Z)
        if sig == eps
            I = beta/(Z+eps*beta);
        else
            I = (1/(sig - eps)) * (log((Z + sig*beta)/(Z + eps*beta)));
        end
    end
end
```

< T = 230 K >

< T = 250 K >

```
< T = 270 K >

ans =


    6.0844

   10.2841

    4.4495

    3.7925

    1.3958

    4.8348

    4.9700

    1.0244

    6.4643



 230K: y1 AARD =  1.94 %

 250K: y1 AARD =  3.05 %

 270K: y1 AARD =  4.45 %


 230K: P AARD =  1.35 %

 250K: p AARD =  1.88 %

 270K: P AARD =  1.35 %


 230K: RMSE =  2.52 %

 250K: RMSE =  3.76 %

 270K: RMSE =  4.81 %
```

Published with MATLAB® R2022a