ACS OMEGA

Open Access

http://pubs.acs.org/journal/acsodf

Article

Python-Based Algorithm for Estimating the Parameters of Physical Property Models for Substances Not Available in Database

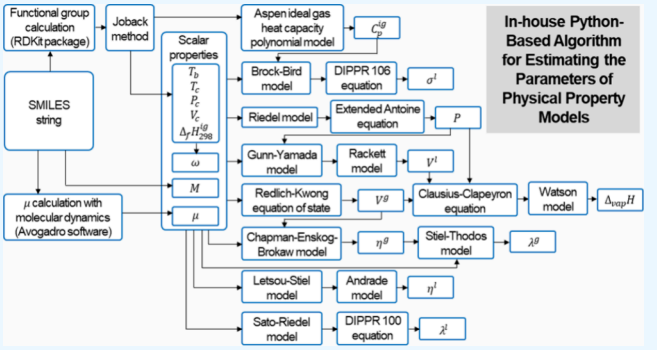
Jina Lee, Wangyun Won,* and Jun-Woo Kim*

Cite This: *ACS Omega* 2024, 9, 11895−11909

Read Online

ACCESS | Metrics & More | Article Recommendations | Supporting Information

ABSTRACT: An in-house Python-based algorithm was developed using simplified molecular-input line-entry specification (SMILES) strings and a dipole moment for estimating the normal boiling point, critical properties, standard enthalpy, vapor pressure, liquid molar volume, enthalpy of vaporization, heat capacity, viscosity, thermal conductivity, and surface tension of molecules. Normal boiling point, critical properties, and standard enthalpy were estimated by using the Joback group contribution method. Vapor pressure, liquid molar volume, enthalpy of vaporization, heat capacity, and surface tension were estimated by using the Riedel model, Gunn−Yamada model, Clausius−Clapeyron equation, Joback group contribution method, and Brock−Bird model, respectively. Viscosities of liquid and gas were estimated by using the Letsou−Stiel model and the Chapman−Enskog−Brokaw model, respectively. Thermal conductivities of liquid and gas were estimated by using the Sato−Riedel model and Stiel−Thodos model, respectively. Dipole moment was calculated through molecular dynamics simulation using the MMFF94 force field, performed with Avogadro software. A case study was conducted with dihydro-2-methyl-3-furanone (DHMF), 2-furaldehyde diethyl acetal (FDA), 1,1-diethoxy-3-methyl butane (DEMB), glutathione (GSH), vitamin B5 (VITB5), homocysteine (HCYS), and *O*-acetyl-L-homoserine (AH), which are not present in the existing property database. Cross-validation indicated that the developed Python-based algorithm provided pure component model parameters nearly identical with those obtained with the Aspen Property Constant Estimation System (PCES) method, except for the enthalpy of vaporization. The parameters for estimating the enthalpy of vaporization using the current Python-based algorithm accurately represented the behavior of the actual substances, as determined using the Clausius−Claperyon equation. This Python-based algorithm provides a detailed and clear reference for estimating pure property parameters.



1. INTRODUCTION

In recent years, studies on biobased fuels and chemicals have continued to increase in an effort to reduce the dependence of chemical industries on fossil fuels.[1,2] Because industrial bioprocesses are relatively less technologically mature than conventional chemical processes, modeling and optimizing bioprocesses through technical economic analysis (TEA) and life cycle assessment (LCA) are significant.[3] Determining the pure component properties is the initial step in the modeling and simulation of bioprocesses for TEA and LCA.[4,5] The pure component properties can be obtained from existing databases or can be determined through direct experiments, or predictions.[6] If an existing database is available, then cost and time savings can be accrued by conducting experiments. Although direct experimentation is the most precise method, it frequently involves time-consuming procedures. In some cases, obtaining a pure sample for a property analysis is expensive or practically impossible. In such cases, the prediction of the pure component properties is a viable alternative.

Bioethanol is a well-known commercial product prepared through biological conversion rather than from fossil fuels or platform chemicals via chemical reactions.[7−9] Bioethanol is an attractive alternative energy source for addressing concerns about the depletion of fossil fuels, climate change, and environmental pollution. The raw materials for bioethanol include not only high-purity feedstocks, such as sugar cane and starch, but also low-purity biomass, such as lignocellulose, which are being used in diverse fields. As a result, many unfamiliar components have been identified as impurities in bioethanol products owing to the diversity of the raw materials.[10,11] Recently reported biobased impurities usually lack a properties database, and obtaining measurement data through experiments often requires a substantial amount of resources.

Received: December 4, 2023
Revised: February 14, 2024
Accepted: February 21, 2024
Published: February 29, 2024

Table 1. Summary of Data for Chemicals Covered in This Study

name	abbreviation	chemical formula	CAS number	SMILES string
dihydro-2-methyl-3-furanone	DHMF	$C_5H_8O_2$	3188-00-9	CC1C(=O)CCO1
2-furaldehyde diethyl acetal	FDA	$C_9H_{14}O_3$	13529-27-6	CCOC(C1=CC=CO1)OCC
1,1-diethoxy-3-methyl butane	DEMB	$C_9H_{20}O_2$	3842-03-3	CCC(C)C(OCC)OCC
glutathione	GSH	$C_{10}H_{17}N_3O_6S$	70-18-8	C(CC(=O)N[C@@H](CS)C(=O)NCC(=O)O)[C@@H](C(=O)O)N
vitamin B5	VITB5	$C_9H_{17}NO_5$	599-54-2	CC(C)(CO)[C@H](C(=O)NCCC(=O)O)O
homocysteine	HCYS	$C_4H_9NO_2S$	6027-13-0	C(CS)[C@@H](C(=O)O)N
O-acetyl-L-homoserine	AH	$C_6H_{11}NO_4$	7540-67-2	CC(=O)OCC[C@@H](C(=O)O)N

Aspen, a commonly used commercial tool for developing bioprocess models,[12] has a comprehensive database of pure component properties.[13] However, for pure components for which there is no existing database, the properties need to be predicted using a property prediction algorithm called the property constant estimation system (PCES) method.[14] The PCES method can be easily used for both industrial and academic purposes with a commercial license fee. However, as a disadvantage, the complete algorithm has not been published.

In this study, a Python-based open-source algorithm with a reproducibility similar to that of the PCES method is developed. The current Python-based algorithm has the advantages of a clear theoretical reference, easy customization, and no license fee. The pure component properties considered in this study include the normal boiling point, critical properties, standard enthalpy, vapor pressure, heat capacity, heat of vaporization, viscosity, thermal conductivity, and surface tension. These properties are essential for process simulations. Parameter estimation is automatically conducted for these properties by entering a simplified molecular-input line-entry specification (SMILES) string.[15]

To verify the accuracy of the model, the proposed Python-based algorithm was cross-validated by using the PCES method as a reference for various substances. Some impurities in biobased ethanol for which there is no existing property database, such as dihydro-2-methyl-3-furanone (DHMF), 2-furaldehyde diethyl acetal (FDA), and 1,1-diethoxy-3-methyl butane (DEMB), were examined and compared. In addition, biobased active substances for which there is no existing property database, such as glutathione (GSH), vitamin B5 (VITB5), homocysteine (HCYS), and *O*-acetyl-L-homoserine (AH), were examined and compared.

2. METHODS

2.1. Chemicals. The properties of substances not present in the Aspen Property Database, including DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, were examined. The chemical formulas, SMILES strings, and Chemical Abstracts Service (CAS) numbers of these chemicals are summarized in Table 1. Because almost all of these chemicals are synthesized through biological processes, racemic compounds were not considered.

2.2. Unit System. Aspen covers a wide range of property models, and its built-in templates have different unit systems, which can lead to confusion. Table 2 provides a summary of the unit systems based on the property models and Aspen models,[14] to prevent confusion. Because industrial bioprocesses are performed in aqueous solutions, a unit system of the Aspen electrolyte template was used in this study. Although the unit system used the Aspen electrolyte template, the models employed in this study do not include pure property predictions for ionic species. The proposed algorithm in this study operates only for molecular species, similar to the Aspen PCES method.

2.3. Scalar Properties. The term "scalar property" refers to properties that are independent of temperature or pressure, and this term is used in Aspen software to collectively describe such properties. In the Aspen PCES method, various scalar properties are estimated using the Joback group contribution method.[16]

$$T_b = 198.2 + \sum T_{b,i} \tag{1}$$

$$T_c = T_b\left[0.584 + 0.965\sum T_{c,i} - \left(\sum T_{c,i}\right)^2\right]^{-1} \tag{2}$$

$$P_c = \left[0.113 + 0.0032N_a - \sum P_{c,i}\right]^{-2} \tag{3}$$

$$V_c = 17.5 + \sum V_{c,i} \tag{4}$$

$$\Delta_f H^{ig}_{298} = 68.29 + \sum \Delta_f H^{ig}_{298,i} \tag{5}$$

where T_b, T_c, P_c, V_c, and $\Delta_f H^{ig}_{298}$ are the normal boiling temperature, critical temperature, critical pressure, critical volume, and enthalpy of formation of the ideal gas at 298 K, respectively. $T_{b,i}$, $T_{c,i}$, $P_{c,i}$, $V_{c,i}$, and $\Delta_f H^{ig}_{298}$ are the values for each property associated with a specific functional group. N_a is the total number of atoms in the molecule, excluding hydrogen. The number of functional groups was estimated using an automated algorithm in the JRgui software proposed by Shi and Borchardt.[17]

The Pitzer acentric factor, ω, was calculated according to the following definition:[18]

$$\omega = -\log_{10} P_r - 1 \text{ at } T_r = 0.7 \tag{6}$$

$$P_r = \frac{P}{P_c} \tag{7}$$

$$T_r = \frac{T}{T_c} \tag{8}$$

where P, T, P_r, and T_r are the vapor pressure, temperature, reduced vapor pressure, and reduced temperature, respectively. The method for predicting P_r is described in Section 2.4.

The dipole moment, μ, was estimated by molecular dynamics simulation using Avogadro software.[19] The molecular structure was built from SMILES strings by using the built-in function in Avogadro software. The molecular structure was optimized using the MMFF94 force field, known for its high accuracy for analyzing organic compounds.[20] For the geometry optimization, a number of steps of 500, the steepest descent algorithm, and a convergence of 10^{-7} options were used. μ was automatically calculated from the electric charge and the optimized molecular structure.

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

Table 2. Summary of Unit Systems for Reference Models and Aspen Models Based on Properties

property	reference model	model in Aspen electrolyte template with their identifiers
normal boiling temperature, T_b	Joback group contribution method in eq 1 with temperature in K.	scalar property, TB, with temperature in °C.
critical temperature, T_c	Joback group contribution method in eq 2 with temperature in K.	scalar property, TC, with temperature in °C.
critical pressure, P_c	Joback group contribution method in eq 3 with pressure in bar.	scalar property, PC, with pressure in bar.
critical volume, V_c	Joback group contribution method in eq 4 with volume in $cm^3\ mol^{-1}$.	scalar property, VC, with volume in $cm^3\ mol^{-1}$.
enthalpy of formation of ideal gas at 298 K, $\Delta_f H^{ig}_{298}$	Joback group contribution method in eq 5 with enthalpy in kJ mol^{-1}.	scalar property, DHFORM, with enthalpy in kcal mol^{-1}.
dipole moment, μ	dipole moment in Debye is calculated based on an electric charge and an energy-optimized molecular structure, and a molecular dynamics simulation using the MMFF94 force field is performed with Avogadro software.	scalar property, MUP, with dipole moment in Debye.
vapor pressure, P	Riedel model in eq 9 with dimensionless reduced temperature, T_r, and dimensionless reduced pressure, P_r.	extended Antoine equation, PLXANT, in eq 18 with pressure in bar and temperature in °C. If any of $C_{P,5}$−$C_{P,7}$ has a nonzero value, the temperature unit of $C_{P,1}$−$C_{P,7}$ is changed to K.
liquid molar volume, V^l	Gunn−Yamada model in eq 28 with volume in $m^3\ kmol^{-1}$, temperature in K, and pressure in bar.	Rackett model with scalar property, RKTZRA, in eq 35 with volume in $cm^3\ mol^{-1}$, temperature in K, and pressure in bar.
enthalpy of vaporization, $\Delta_{vap}H$	Clausius−Clapeyron equation in eq 37 with enthalpy in kJ mol^{-1}, temperature in K, pressure in bar, and volume in $m^3\ kmol^{-1}$.	Watson model, DHVLWT, in eq 41 with enthalpy in kcal mol^{-1} and temperature in °C.
ideal gas heat capacity, C^{ig}_P	Joback group contribution method in eq 42 with heat capacity in J $mol^{-1}\ K^{-1}$ and temperature in K.	Aspen ideal gas heat capacity polynomial model, CPIG, in eqs 47 and 48 with heat capacity in cal $mol^{-1}\ K^{-1}$ and temperature in °C. If any of $C_{Cp,10}$ and $C_{Cp,11}$ has a nonzero value, the temperature unit of $C_{Cp,9}$−$C_{Cp,11}$ is changed to K.
liquid viscosity, η^l	Letsou−Stiel model in eq 64 with viscosity in cP, temperature in K, and pressure in bar.	Andrade model, MULAND, in eq 68 with viscosity in cP and temperature in °C. If any $C_{\eta,2}$ and $C_{\eta,3}$ has a nonzero value, the temperature unit of $C_{\eta,1}$−$C_{\eta,3}$ is changed to K.
gas viscosity, η^g	Chapman−Enskog−Brokaw model in eq 69 with viscosity in cP, dipole moment in Debye, temperature in K, and volume in $m^3\ kmol^{-1}$.	the same method as the reference model with viscosity in cP.
liquid thermal conductivity, λ^l	Sato−Riedel model in eq 76 with thermal conductivity in kcal m $h^{-1}\ m^{-2}\ K^{-1}$ and temperature in K.	DIPPR eq 100 model, KLDIP, in eq 77 with thermal conductivity in kcal m $h^{-1}\ m^{-2}\ K^{-1}$ and temperature in K.
gas thermal conductivity, λ^g	Stiel−Thodos model in eq 78 with thermal conductivity in W $m^{-1}\ K^{-1}$, heat capacity in J $mol^{-1}\ K^{-1}$, and viscosity in cP.	the same method as the reference model with thermal conductivity in kcal m $h^{-1}\ m^{-2}\ K^{-1}$.
surface tension, σ^l	Brock−Bird model in eq 79 with surface tension in dyne cm^{-1}, temperature in K, and pressure in bar.	DIPPR eq 106 model, SIGDIP, in eq 81 with surface tension in dyne cm^{-1} and temperature in °C. If any $C_{\sigma,1}$−$C_{\sigma,5}$ has a nonzero value, the temperature unit of $C_{\sigma,1}$−$C_{\sigma,5}$ is changed to K.

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

2.4. Vapor Pressure. The reference value for P was predicted by using the Riedel model.[21]

$$\ln P_r = A - \frac{B}{T_r} + C \ln T_r + D T_r^6 \tag{9}$$

$$\Psi_b = -35 + \frac{36}{T_{br}} + 42 \ln T_{br} - T_{br}^6 \tag{10}$$

$$\alpha_c = \frac{3.758 K \Psi_b + \ln(P_c/1.01325)}{K\Psi_b - \ln T_{br}} \tag{11}$$

$$Q = K(3.758 - \alpha_c) \tag{12}$$

$$A = -35Q \tag{13}$$

$$B = -36Q \tag{14}$$

$$C = 42Q + \alpha_c \tag{15}$$

$$D = -Q \tag{16}$$

$$T_{br} = \frac{T_b}{T_c} \tag{17}$$

where T_{br} denotes the reduced boiling temperature. A, B, C, D, Q, α_c, Ψ_b, and K are parameters of the Riedel model. K depends on the type of material, and 0.0838 is recommended for general purposes.[21]

The extended Antoine equation model was used to simulate P in the PCES method and the current Python-based algorithm.

$$\ln P = C_{P,1} + \frac{C_{P,2}}{T + C_{P,3}} + C_{P,4} T + C_{P,5} \ln T + C_{P,6} T^{C_{P,7}} \quad \text{for } C_{P,8} \le T \le C_{P,9} \tag{18}$$

where $C_{P,1}$, $C_{P,2}$, $C_{P,3}$, $C_{P,4}$, $C_{P,5}$, $C_{P,6}$, $C_{P,7}$, $C_{P,8}$, and $C_{P,9}$ are parameters of the extended Antoine equation. If the temperature is out of bound, linear extrapolation is conducted up to 7 + ln P at $T = C_{P,9}$ where the slope is determined by ln P versus $1/T$; beyond this limit, the vapor pressure remains constant. The Riedel model parameters can be analytically converted into extended Antoine equation parameters as follows:

$$C_{P,1} = A + \ln P_c - C \ln T_c \tag{19}$$

$$C_{P,2} = -BT_c \tag{20}$$

$$C_{P,3} = 0 \tag{21}$$

$$C_{P,4} = 0 \tag{22}$$

$$C_{P,5} = C \tag{23}$$

$$C_{P,6} = \frac{D}{T_c^6} \tag{24}$$

$$C_{P,7} = 6 \tag{25}$$

$$C_{P,8} = T_b \tag{26}$$

$$C_{P,9} = T_c \tag{27}$$

where the temperature boundary between $C_{P,8}$ and $C_{P,9}$ was defined according to the PCES method.

2.5. Liquid Molar Volume. The reference liquid molar volume for the pure component was predicted by applying the Gunn−Yamada model.[22]

$$V^l = V_r^{(0)}(1 - \omega\delta) V_{SC} \tag{28}$$

$$Z_{SC} = 0.2920 - 0.0967\omega \tag{29}$$

$$V_{SC} = Z_{SC} \frac{RT_c}{P_c} \tag{30}$$

$$\delta = 0.29607 - 0.09045 T_r - 0.04842 T_r^2 \quad \text{for } T_r > 0.2 \tag{31}$$

$$V_r^{(0)} = 0.33593 - 0.33593 T_r + 1.51941 T_r^2 - 2.02512 T_r^3 + 1.11422 T_r^4 \quad \text{for } 0.8 > T_r \ge 0.2 \tag{32}$$

$$V_r^{(0)} = 1.0 + 1.3(1 - T_r)^{0.5} \log_{10}(1 - T_r) - 0.50879(1 - T_r) - 0.91534(1 - T_r)^2 \quad \text{for } 1 > T_r \ge 0.8 \tag{33}$$

$$V_r^{(0)} = 1 \quad \text{for } T_r = 1 \tag{34}$$

where V^l is the volume of the pure liquid. δ, $V_r^{(0)}$, Z_{SC}, and V_{SC} are parameters of the Gunn−Yamada model. R is the ideal gas constant, and a value of 0.0831446 m^3 bar K^{-1} kmol^{-1} was used in eqs 28−34.

The Rackett model was used to simulate V^l using the PCES method and the current Python-based algorithm.[23]

$$\log_{10} V^l = (1 + (1 - T_r)^{2/7}) \log_{10} Z^{RA} - \log_{10} P_c/(RT_c) \tag{35}$$

where Z^{RA} is the parameter of the Rackett model. For the PCES method, a value of 83.1446 m^3 bar K^{-1} kmol^{-1} was used for R in eq 35. If T_r is greater than 0.99, a special form of extrapolation is used to obtain a smooth curve according to the PCES method.[24] The Gunn−Yamada and Rackett models are analytically inconsistent. A detailed document explaining the clear algorithm used to evaluate Z^{RA}in PCES could not be found. Instead, the empirical correlation for the critical compressibility factor proposed by Gunn and Yamada was applied to estimate Z^{RA}.[22]

$$Z^{RA} = 0.2918 - 0.0928\omega \tag{36}$$

2.6. Enthalpy of Vaporization. The reference value for the enthalpy of vaporization was predicted by applying the Clausius−Clapeyron equation:[25]

$$\Delta_{vap} H = \frac{dP}{dT} T(V^g - V^l) \tag{37}$$

where $\Delta_{vap}H$ is the enthalpy of vaporization. V^g is the volume of the pure gas. P and V^l can be obtained from eqs 18 and 35, respectively. dP/dT was obtained by numerical differentiation with 1×10^{-5} dT. V^g was obtained by PCES using the Redlich−Kwong equation of state (RKEOS):[26]

$$P = \frac{RT}{(V^g - b)} - \frac{a}{V^g(V^g + b)\sqrt{T}} \tag{38}$$

$$a = 0.42748 \frac{R^2 T_c^{2.5}}{P_c} \tag{39}$$

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

$$b = 0.08664\frac{RT_c}{P_c} \tag{40}$$

where a and b are the RKEOS parameters. V^g under the given T and P conditions was determined by using the well-known Newton−Raphson method. An objective function that involves multiplying both terms of eq 37 by $(V^g - b)V^g(V^g + b)\sqrt{T}$ was used, along with the initial value of RT/P.

The Watson model was used to simulate $\Delta_{vap}H$ in both the PCES method and the current Python-based algorithm:[27]

$$\Delta_{vap}H = C_{WT,1}\left(\frac{1 - T_r}{1 - C_{WT,2}/T_c}\right)^{C_{WT,3}+C_{WT,4}(1-T_r)} \quad \text{for } T > C_{WT,5} \tag{41}$$

where $C_{WT,1}$, $C_{WT,2}$, $C_{WT,3}$, $C_{WT,4}$, and $C_{WT,5}$ are the parameters of the Watson model. If the temperature condition is out of bound, linear extrapolation is performed. Previously, in PCES, the Clausius−Clapeyron equation was used as a reference model to estimate the parameters in eq 41.[14] However, a detailed explanation of the algorithm could not be found. In this study, $C_{WT,1}$ was determined as $\Delta_{vap}H$ at T_b by using eq 37. $C_{WT,2}$ was determined as T_b. $C_{WT,3}$ and $C_{WT,4}$ were determined through regression analysis of the results obtained from eq 41. These results were simulated at 10 uniformly spaced intervals between T_b and T_c using eq 37. The Nelder−Mead method was employed for the regression analysis with a reflection parameter of 1, an expansion parameter of 2, a contraction parameter of 0.5, a shrink parameter of 0.5, an initial simplex parameter for nonzero values of 5×10^{-2}, and an initial simplex parameter for zero values of 2.5×10^{-4}.[28] The initial values of $C_{WT,3}$ and $C_{WT,4}$ for the Nelder−Mead method were analytically determined and used to interpolate the two points dividing the temperature range between T_b and T_c into thirds according to eq 37. $C_{WT,5}$ was determined by multiplying T_b by 0.4, according to the PCES method.

2.7. Ideal Gas Heat Capacity. The reference ideal gas heat capacity was simulated using the Joback group contribution method:[16]

$$C_p^{ig} = \sum C_{p,a,i}^{ig} - 37.93 + \left(\sum C_{p,b,i}^{ig} + 0.210\right)T + \left(\sum C_{p,c,i}^{ig} - 3.91 \times 10^{-4}\right)T^2 + \left(\sum C_{p,d,i}^{ig} + 2.06 \times 10^{-7}\right)T^3 \tag{42}$$

$$C_{p,a}^{ig} = \sum C_{p,a,i}^{ig} \tag{43}$$

$$C_{p,b}^{ig} = \sum C_{p,b,i}^{ig} \tag{44}$$

$$C_{p,c}^{ig} = \sum C_{p,c,i}^{ig} \tag{45}$$

$$C_{p,d}^{ig} = \sum C_{p,d,i}^{ig} \tag{46}$$

where C_p^{ig} is the ideal gas heat capacity. $C_{p,a,i}^{ig}$, $C_{p,b,i}^{ig}$, $C_{p,c,i}^{ig}$, and $C_{p,d,i}^{ig}$ are the ideal gas heat capacities associated with a specific functional group. $C_{p,a}^{ig}$, $C_{p,b}^{ig}$, $C_{p,c}^{ig}$, and $C_{p,d}^{ig}$ are parameters of the Joback ideal gas heat capacity model, which is the sum of the values corresponding to the functional groups.

In both the PCES method and the current Python-based algorithm, C_p^{ig} was simulated using the following empirical equation, known as the Aspen ideal gas heat capacity polynomial model:

$$C_p^{ig} = C_{Cp,1} + C_{Cp,2}T + C_{Cp,3}T^2 + C_{Cp,4}T^3 + C_{Cp,5}T^4 + C_{Cp,6}T^5 \text{ for } C_{Cp,7} \le T \le C_{Cp,8} \tag{47}$$

$$C_p^{ig} = C_{Cp,9} + C_{Cp,10}T^{C_{Cp,11}} \text{ for } T \le C_{Cp,7} \tag{48}$$

where $C_{Cp,1}$, $C_{Cp,2}$, $C_{Cp,3}$, $C_{Cp,4}$, $C_{Cp,5}$, $C_{Cp,6}$, $C_{Cp,7}$, $C_{Cp,8}$, $C_{Cp,9}$, $C_{Cp,10}$, and $C_{Cp,11}$ are parameters of the Aspen ideal gas heat capacity polynomial model. If the temperature condition is out of bound, then a linear extrapolation is performed. The Joback model parameters can be analytically converted into the Aspen ideal gas heat capacity polynomial model parameters. A conversion factor of 4.1868 for converting the units from calorie to Joule was obtained from the International Standard.[29]

$$C_{p,a}^{ig}/\text{cal mol}^{-1}\text{ K}^{-1} = \frac{C_{p,a}^{ig}/\text{J mol}^{-1}\text{ K}^{-1}}{-37.93 \times 4.1868} \tag{49}$$

$$C_{p,b}^{ig}/\text{cal mol}^{-1}\text{ K}^{-1} = \frac{C_{p,b}^{ig}/\text{J mol}^{-1}\text{ K}^{-1} + 0.210}{4.1868} \tag{50}$$

$$C_{p,c}^{ig}/\text{cal mol}^{-1}\text{ K}^{-1} = \frac{C_{p,c}^{ig}/\text{J mol}^{-1}\text{ K}^{-1} - 3.91 \times 10^{-4}}{4.1868}c_3 \tag{51}$$

$$C_{p,d}^{ig}/\text{cal mol}^{-1}\text{ K}^{-1} = \frac{C_{p,d}^{ig}/\text{J mol}^{-1}\text{ K}^{-1} + 2.06 \times 10^{-7}}{4.1868}c_4 \tag{52}$$

$$C_{Cp,1} = C_{p,a}^{ig} + 273.15C_{p,b}^{ig} + 273.15^2C_{p,c}^{ig} + 273.15^3C_{p,d}^{ig} \tag{53}$$

$$C_{Cp,2} = C_{p,b}^{ig} + 2 \times 273.15C_{p,c}^{ig} + 3 \times 273.15^2C_{p,d}^{ig} \tag{54}$$

$$C_{Cp,3} = c_3 + 3 \times 273.15c_4 \tag{55}$$

$$C_{Cp,4} = c_4 \tag{56}$$

$$C_{Cp,5} = 0 \tag{57}$$

$$C_{Cp,6} = 0 \tag{58}$$

$$C_{Cp,7} = 6.85 \tag{59}$$

$$C_{Cp,8} = 826.85 \tag{60}$$

$$C_{Cp,9} = 8.60543 \tag{61}$$

$$C_{Cp,10} = \frac{(C_{Cp,1} + C_{Cp,2}C_{Cp,7} + C_{Cp,3}C_{Cp,7}^2 + C_{Cp,4}C_{Cp,7}^3) - C_{Cp,9}}{(C_{Cp,7} + 273.15)^{1.5}} \tag{62}$$

$$C_{Cp,11} = 1.5 \tag{63}$$

Eqs 49 and 52 are the unit conversion for the heat capacity from J mol^{-1} K^{-1} to cal mol^{-1} K^{-1}. Note that the temperature unit of $C_{Cp,1} - C_{Cp,8}$ in eqs 47, 48, and 53−63is °C whereas the temperature unit for $C_{Cp,9} - C_{Cp,11}$ is K. The temperature

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

boundary between $C_{Cp,7}$ and $C_{Cp,8}$ was defined according to the PCES method.

2.8. Viscosity. The reference liquid viscosity was simulated using the Letsou−Stiel model:[30]

$$\eta^{l}\xi = (\eta\xi)^{(0)} + \omega(\eta\xi)^{(1)} \tag{64}$$

$$(\eta\xi)^{(0)} = 0.015174 - 0.02135T_r + 0.0075T_r^2 \tag{65}$$

$$(\eta\xi)^{(1)} = 0.042552 - 0.07674T_r + 0.0340T_r^2 \tag{66}$$

$$\xi = \frac{T_c^{1/6}}{M^{1/2}P_c^{2/3}} \tag{67}$$

where η and M are the liquid viscosity and molecular weight, respectively. $(\eta\xi)^{(0)}$, $(\eta\xi)^{(1)}$, and ξ are the parameters of the Letsou−Stiel model. ω can be calculated by using eq 6.

The Andrade model was used to simulate η^{l} in both the PCES method and the current Python-based algorithm:[31]

$$\ln \eta^{l} = C_{\eta,1} + \frac{C_{\eta,2}}{T} + C_{\eta,3}\ln T \text{ for } C_{\eta,4} \le T \le C_{\eta,5} \tag{68}$$

where $C_{\eta,1}$, $C_{\eta,2}$, $C_{\eta,3}$, $C_{\eta,4}$, and $C_{\eta,5}$ are parameters of the Andrade model. If the temperature condition is out of bound, linear extrapolation is conducted with a slope determined by ln η^{l} versus $1/T$. It was not possible to find a detailed explanation of the algorithm used to evaluate $C_{\eta,1}$, $C_{\eta,2}$, and $C_{\eta,3}$ in the PCES method. $C_{\eta,4}$ was determined as T_b according to the PCES method. $C_{\eta,5}$ was determined as $0.99T_c$ according to the PCES method. $C_{\eta,1}$, $C_{\eta,2}$, and $C_{\eta,3}$ values were determined through regression analysis of the results obtained from eq 68. These results were simulated at 10 uniformly spaced intervals between T_b and $0.99T_c$ using eq 64. The well-known ordinary least-squares regression method was employed for regression analysis.

The Chapman−Enskog−Brokaw model was used to simulate the gas viscosity in the PCES method and the current Python-based algorithm.[32,33]

$$\eta^{g} = 26.693 \times 10^{-4}\frac{(MT)^{0.5}}{\sigma^2\Omega_p^{2,2}} \tag{69}$$

$$\delta = 1.94 \times 10^{3}\frac{\mu^2}{T_bV_b} \tag{70}$$

$$\sigma^3 = \frac{1.585V_b}{1 + 1.3\delta^2} \tag{71}$$

$$\frac{\epsilon}{k_B} = 1.18(1 + 1.3\delta^2)T_b \tag{72}$$

$$T^* = \frac{T}{\epsilon/k_B} \tag{73}$$

$$\Omega_n^{2,2} = 1.16145(T^*)^{-0.14874} + 0.52487\exp(-0.7732T^*) + 2.16178\exp(-2.43787T^*) \tag{74}$$

$$\Omega_p^{2,2} = \Omega_n^{2,2} + 0.2\frac{\delta^2}{T^*} \tag{75}$$

where η^{g} is the gas viscosity. δ and V_b are the polarity parameter and gas volume at T_b, respectively. σ is a dimensional parameter related to the intermolecular potential. ϵ and k_B are energy parameters related to the intermolecular potential and Boltzmann constant (1.38×10^{-18} erg/K), respectively. These parameters were used directly in the form of the Lennard-Jones energy parameter (ϵ/k_B) without the requirement of separate calculations. T^* is the reduced temperature defined in the model. $\Omega_p^{2,2}$ and $\Omega_n^{2,2}$ are the polar Lennard-Jones (12−6) potential and nonpolar Lennard-Jones (12−6) potential, respectively. μ was calculated using molecular dynamics, as described in Section 2.3. When μ is small, it is anticipated that Aspen will employ a different model or algorithm to calculate the gas viscosity. However, a clear reference specifying the exact model to be used could not be found.

2.9. Thermal Conductivity. The reference liquid thermal conductivity was simulated using the Sato−Riedel model:[34]

$$\lambda^{l} = \frac{0.9510}{M^{0.5}}\frac{3 + 20(1 - T_r)^{2/3}}{3 + 20(1 - T_{br})^{2/3}} \tag{76}$$

where λ^{l} is the thermal conductivity of the pure liquid.

The Design Institute for Physical Properties (DIPPR) eq 100 model was employed to simulate λ^{l} in the PCES method and the current Python-based algorithm.

$$\lambda^{l} = C_{\lambda,1} + C_{\lambda,2}T + C_{\lambda,3}T^2 + C_{\lambda,4}T^3 + C_{\lambda,5}T^4 \text{ for } C_{\lambda,6} \le T \le C_{\lambda,7} \tag{77}$$

where $C_{\lambda,1}$, $C_{\lambda,2}$, $C_{\lambda,3}$, $C_{\lambda,4}$, $C_{\lambda,5}$, $C_{\lambda,6}$, and $C_{\lambda,7}$ are parameters of the DIPPR eq 100 model. Linear extrapolation is conducted for T outside the range $C_{\lambda,6}$ to $C_{\lambda,7}$. $C_{\lambda,6}$ was determined as T_b according to the PCES method. $C_{\lambda,7}$ was determined as $0.99T_c$ according to the PCES method. In case of the current Python algorithm, $C_{\lambda,1}$, $C_{\lambda,2}$, $C_{\lambda,3}$, $C_{\lambda,4}$, and $C_{\lambda,5}$ were determined using the well-known ordinary least-squares regression with 10 temperature points uniformly distributed between $C_{\lambda,6}$ and $C_{\lambda,7}$.

The Stiel−Thodos model was employed to simulate the gas thermal conductivity using both the PCES method and the current Python-based algorithm:[35]

$$\lambda^{g} = \left(1.15 + \frac{2.03}{C_p^{ig}/R - 1}\right)\frac{\eta^{g}(C_p^{ig} - R)}{M} \tag{78}$$

where λ^{g} is the gas thermal conductivity. C_p^{ig} values can be obtained using eqs 47 and 48. A value of 8.134 J mol^{-1} K^{-1} was used for R in eq 78 to obtain a reference value of λ^{g} in W m^{-1} K^{-1}. In the Aspen electrolyte template, because the units of the gas thermal conductivity and heat capacity are kcal m h^{-1} m^{-2} K^{-1} and cal mol^{-1} K^{-1}, respectively, the reference value of λ^{g} must be converted by applying 3.6/4.1868.

2.10. Surface Tension. The reference liquid surface tension for the pure component was simulated using the Brock−Bird model:[36,37]

$$\sigma^{l} = P_c^{2/3}T_c^{1/3}(-0.281 + 0.133Y_c)(1 - T_r)^{11/9} \tag{79}$$

$$Y_c = 0.9076\left(1 + \frac{T_{br}\ln\left(\frac{P_c}{1.01325}\right)}{1 - T_{br}}\right) \tag{80}$$

where σ^{l} is the liquid surface tension. Y_c is a parameter of the Brock−Bird model.

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

Table 3. Estimated Scalar Properties from the PCES Method and the Current Python-Based Algorithm

substance	method	T_b (°C)	T_c (°C)	P_c (bar)	V_c(cm^3 mol^{-1})	$\Delta_f H^{ig}_{298}$ (kcal mol^{-1})	ω
DHMF	PCES	150.7	370.6	44.9	0.285	−84.97	0.349
	this work	150.9	370.9	44.9	0.285	−84.97	0.349
FDA	PCES	226.8	425.2	29.3	0.505	−106.5	0.583
	this work	227.0	425.5	29.3	0.505	−106.5	0.583
DEMB	PCES	176.1	346.5	22.7	0.564	−120.4	0.533
	this work	176.3	346.8	22.7	0.564	−120.4	0.533
GSH	PCES	788.7	1031.4	34.2	0.797	−240.5	1.92
	this work	788.9	1031.6	34.2	0.797	−240.5	1.92
VITB5	PCES	562.4	757.8	34.4	0.626	−228.6	1.86
	this work	562.6	758.1	34.4	0.626	−228.6	1.86
HCYS	PCES	298.2	527.9	52.2	0.361	−97.73	0.833
	this work	298.4	528.2	52.2	0.361	−97.73	0.833
AH	PCES	339.4	545.9	37.4	0.445	−192.6	1.02
	this work	339.6	546.2	37.4	0.445	−192.6	1.02

Table 4. Estimation of Extended Antoine Equation Parameters Using the PCES Method and the Current Python-Based Algorithm

substance	method	$C_{P,1}$	$C_{P,2}$	$C_{P,3}$	$C_{P,4}$	$C_{P,5}$	$C_{P,6}$	$C_{P,7}$	$C_{P,8}$	$C_{P,9}$
DHMF	PCES	52.00	-7.323×10^3	0	0	−5.742	4.439×10^{-18}	6.0	150.7	370.6
	this work	52.00	-7.325×10^3	0	0	−5.741	4.426×10^{-18}	6.0	150.9	370.9
FDA	PCES	74.54	-1.038×10^4	0	0	−8.659	3.559×10^{-18}	6.0	226.8	425.2
	this work	74.54	-1.039×10^4	0	0	−8.658	3.550×10^{-18}	6.0	227.0	425.5
DEMB	PCES	68.55	-8.754×10^3	0	0	−8.041	6.929×10^{-18}	6.0	176.1	346.5
	this work	68.55	-8.758×10^3	0	0	−8.040	6.910×10^{-18}	6.0	176.3	346.8
GSH	PCES	226.9	-4.689×10^4	0	0	−26.26	2.026×10^{-19}	6.0	788.7	1031
	this work	226.9	-4.690×10^4	0	0	−26.26	2.023×10^{-19}	6.0	788.9	1032
VITB5	PCES	213.9	-3.602×10^4	0	0	−25.42	8.081×10^{-19}	6.0	562.4	757.8
	this work	213.9	-3.603×10^4	0	0	−25.42	8.069×10^{-19}	6.0	562.6	758.1
HCYS	PCES	100.6	-1.487×10^4	0	0	−11.75	1.951×10^{-18}	6.0	298.2	527.9
	this work	100.6	-1.488×10^4	0	0	−11.75	1.947×10^{-18}	6.0	298.4	528.2
AH	PCES	119.3	-1.755×10^4	0	0	−14.14	1.971×10^{-18}	6.0	339.4	545.9
	this work	119.3	-1.755×10^4	0	0	−14.14	1.967×10^{-18}	6.0	339.6	546.2

The DIPPR eq 106 model was employed to simulate the liquid surface tension in both the PCES method and the current Python-based algorithm:

$$\sigma^l = C_{\sigma,1}(1 - T_r)^{(C_{\sigma,2}+C_{\sigma,3}T_r+C_{\sigma,4}T_r^2+C_{\sigma,5}T_r^3)} \quad \text{for } C_{\sigma,6} \le T \le C_{\sigma,7} \tag{81}$$

where $C_{\sigma,1}$, $C_{\sigma,2}$, $C_{\sigma,3}$, $C_{\sigma,4}$, $C_{\sigma,5}$, $C_{\sigma,6}$, and $C_{\sigma,7}$ are parameters of the DIPPR eq 106 model. Linear extrapolation is conducted for T outside the range $C_{\sigma,6}$ to C_{σ}. Because the DIPPR 106 model and Brock−Bird models have mathematically identical structures, the parameters can be obtained analytically; the relationships are as follows:

$$C_{\sigma,1} = P_c^{2/3}T_c^{1/3}(-0.281 + 0.133Y_c) \tag{82}$$

$$C_{\sigma,2} = 11/9 \tag{83}$$

$$C_{\sigma,3} = 0 \tag{84}$$

$$C_{\sigma,4} = 0 \tag{85}$$

$$C_{\sigma,5} = 0 \tag{86}$$

$$C_{\sigma,6} = T_b \tag{87}$$

$$C_{\sigma,7} = 0.98T_c \tag{88}$$

where the temperature boundary between $C_{\sigma,6}$ and $C_{\sigma,7}$ was defined according to the PCES method.

2.11. Python-Based Algorithm Code. In this study, a Python-based algorithm code for estimating the property parameters was developed. SMILES arbitrary target specification (SMARTS) codes corresponding to each functional group proposed in JRgui software were applied.[17] The open-source Chemoinformatics package RDKit automatically detects and counts functional groups.[38] The Numpy package was used for array calculations.[39] All of the algorithms introduced in this study were developed in Python; the source codes are provided in the Supporting Information.

3. RESULTS AND DISCUSSION

3.1. Scalar Properties. Table 3 lists the calculated data for the scalar properties. The percent absolute residuals between the PCES method and the current Python-based algorithm for T_b are 0.13, 0.09, 0.11, 0.03, 0.04, 0.07, and 0.06% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively. The percent absolute residuals between the PCES method and the current Python-based algorithm for T_c are 0.08, 0.07, 0.08, 0.02, 0.03, 0.05, and 0.05% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively. There were slight differences in the cases of T_b and T_c because the first parameter in eq 1 was set to 198.0 instead of 198.2 in the PCES method. In the original study, a value of 198.2 was used

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

as the first parameter.[16] The current Python-based algorithm used the same values as in the original study. The simulated values of P_c, V_c, $\Delta_f H^{ig}_{298}$, and ω were all the same regardless of the method used. Therefore, the percent absolute residuals are zero for all of these properties.

3.2. Vapor Pressure. The extended Antoine equation parameters for the vapor pressure model were analytically derived from the Riedel model using the values of T_b, T_c, and P_c listed in Table 3. The values of all parameters were similar for both the PCES and the current Python-based algorithms, as listed in Table 4. Slight differences were observed in $C_{P,2}$, $C_{P,6}$, $C_{P,8}$, and $C_{P,9}$ owing to variations in T_b and T_c, as mentioned in Section 3.1. The percent mean absolute residuals between the PCES method and the current Python-based algorithm for P are 0.47, 0.44, 0.46, 0.34, 0.60, 0.67, and 0.53% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively.

3.3. Liquid Molar Volume. For the current Python-based algorithm, Z^{RA} was obtained from the empirical model of eq 36 proposed by Gunn and Yamada. However, there is no clear published algorithm for estimating Z^{RA} for the PCES method. Nevertheless, the PCES method and the current Python-based algorithm yielded similar results, as shown in Table 5. This example demonstrates the versatility of the empirical model proposed by Gunn and Yamada (eq 36) for obtaining the Rackett parameter. Figure 1 shows the calculated liquid molar volumes for FDA and HCYS, demonstrating that the simulation results were almost identical. The calculation results for the liquid molar volume of all substances can be seen in Figure S1 in the Supporting Information. The percent mean absolute residuals between the PCES method and the current Python-based algorithm for V^l are 0.48, 0.89, 0.66, 0.62, 0.61, 0.78, and 1.14% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively.

Table 5. Estimation of Rackett Model Parameters from the PCES Method and the Current Python-Based Algorithm

substance	method	Z^{RA}
DHMF	PCES	0.259
	this work	0.259
FDA	PCES	0.236
	this work	0.238
DEMB	PCES	0.241
	this work	0.242
GSH	PCES	0.113
	this work	0.114
VITB5	PCES	0.199
	this work	0.120
HCYS	PCES	0.213
	this work	0.215
AH	PCES	0.196
	this work	0.197

3.4. Enthalpy of Vaporization. As shown in Table 6, the values of $C_{WT,1}$, $C_{WT,3}$, and $C_{WT,4}$ estimated by using the PCES method and the current Python-based algorithm were significantly different. $C_{WT,1}$ can be derived from the enthalpy of vaporization at T_b by using the Clausius−Clapeyron equation (eq 37). Figure 2 shows the results for DHMF and GSH obtained with the Clausius−Clapeyron equation, calculated by using both Aspen software and the current Python-based algorithm. The calculation results for the Clausius−Clapeyron equation of all substances can be seen in Figure S2 in the Supporting Information. Both algorithms yielded the same results. Nevertheless, the estimated value of $C_{WT,1}$ was different, although the exact cause could not be analyzed, owing to a lack of a clear reference. $C_{WT,3}$ and $C_{WT,4}$ are parameters that represent the influence of $\Delta_{vap}H$ on the temperature; the difference between the PCES method and the current Python-based algorithm is more pronounced for these parameters. Figure 3 shows the $\Delta_{vap}H$ for DHMF and GSH, simulated using the Watson model with the parameters estimated using the PCES method and the current Python-based algorithm. The calculation results for the $\Delta_{vap}H$ of all substances can be seen in Figure S3 in the Supporting Information. In the case of the PCES method, $C_{WT,4}$ had a significantly negative value. As a result, in some cases (including for GSH), $\Delta_{vap}H$ reached a maximum value at a certain temperature and tended to decrease as the temperature decreased. For common substances (such as water, ethyl alcohol, carbon disulfide, ethyl ether, *n*-pentane, and sulfur dioxide), the heat of vaporization gradually decreases with increasing temperature until it approaches zero near the critical temperature. This phenomenon is consistent with predictions based on the Clausius−Clapeyron equation.[40] As mentioned earlier, in the case of the PCES method, there are regions that do not align with the trends predicted by the Clausius−Clapeyron equation. However, the results obtained with the

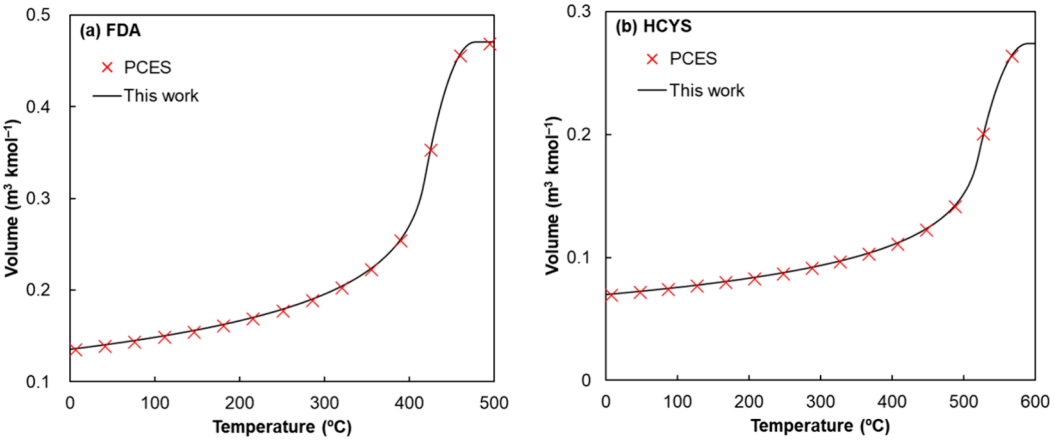


Figure 1. Simulation of liquid molar volume for FDA (a) and HCYS (b) using the Rackett model with parameters estimated by the PCES method and the current Python-based algorithm.

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

Table 6. Data Estimated by Applying the Watson Model Using Parameters from the PCES Method and the Current Python-Based Algorithm

substance	method	$C_{WT,1}$	$C_{WT,2}$	$C_{WT,3}$	$C_{WT,4}$	$C_{WT,5}$
DHMF	PCES	9.391	150.7	0.4354	−0.2049	−103.6
	this work	9.564	150.9	0.3446	0.09174	−103.5
FDA	PCES	11.66	226.8	0.4246	−0.3539	−73.18
	this work	11.93	227.0	0.3230	0.1449	−73.10
DEMB	PCES	9.802	176.1	0.4277	−0.3463	−93.44
	this work	10.06	176.3	0.3268	0.1463	−93.36
GSH	PCES	39.66	788.7	0.3657	−1.357	151.6
	this work	40.62	788.9	0.2283	−0.1184	151.7
VITB5	PCES	30.72	562.4	0.3678	−1.307	61.07
	this work	31.46	562.6	0.2316	−0.1038	61.15
HCYS	PCES	16.05	298.2	0.4106	−0.4604.	−44.60
	this work	16.35	298.4	0.3028	0.1578	−44.52
AH	PCES	17.60	339.4	0.4025	−0.6282	−28.15
	this work	17.99	339.6	0.2875	0.1207	−28.07

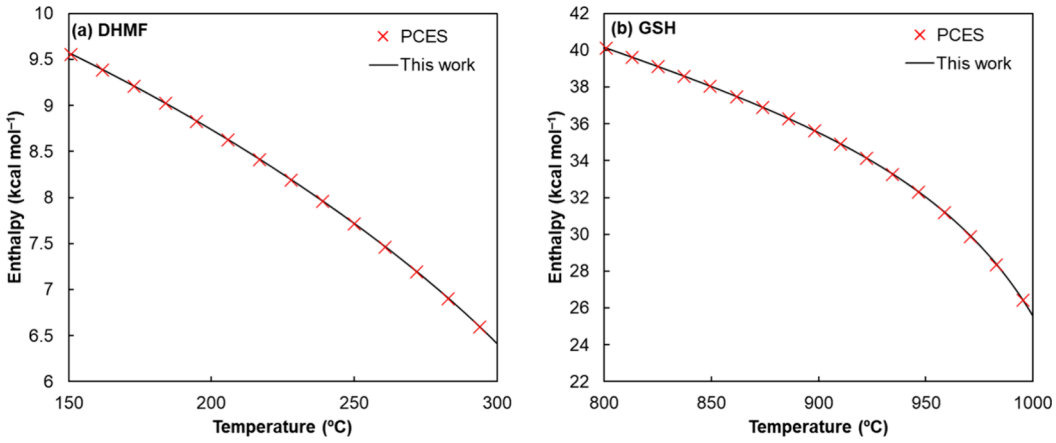


Figure 2. Enthalpy of vaporization for DHMF (a) and GSH (b) simulated using the Clausius−Clapeyron equation.

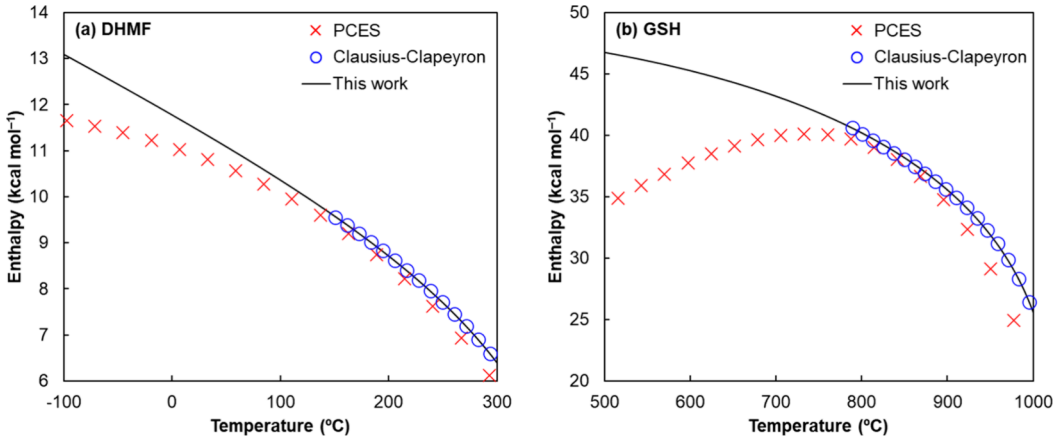


Figure 3. Enthalpy of vaporization for DHMF (a) and GSH (b) simulated using the Watson model with parameters estimated by the PCES method and the current Python-based algorithm.

Watson model using the parameters estimated from the current Python-based algorithm exhibited the typical behavior wherein $\Delta_{vap}H$ does not decrease, but the slope of $\Delta_{vap}H$ versus temperature decreased as the temperature decreased. The percent mean absolute residuals between the Clausius−Clapeyron equation and the PCES method for $\Delta_{vap}H$ are 6.82, 7.15, 7.53, 7.42, 7.38, 6.43, and 6.86% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively. The percent mean absolute residuals between the Clausius−Clapeyron equation and the current Python-based algorithm for $\Delta_{vap}H$ are 0.20, 0.24, 0.25, 0.31, 0.32, 0.22, and 0.19% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively. Based on the results, it can be asserted that the regression using this Python-based algorithm shows better alignment with the Clausius−Clapeyron equation. The value of P in the Clausius−Clapeyron equation can be obtained from

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

Table 7. Data Estimated by Applying an Aspen Ideal Gas Heat Capacity Polynomial Model with Parameters from the PCES Method and Current Python-Based Algorithm

substance	method	$C_{Cp,1}$	$C_{Cp,2}$	$C_{Cp,3}$	$C_{Cp,4}$	$C_{Cp,5}$	$C_{Cp,6}$	$C_{Cp,7}$	$C_{Cp,8}$	$C_{Cp,9}$	$C_{Cp,10}$	$C_{Cp,11}$
DHMF	PCES	24.39	8.774×10^{-2}	-3.475×10^{-5}	-8.097×10^{-9}	0	0	6.85	826.9	8.605	3.497×10^{-3}	1.5
	this work	24.39	8.774×10^{-2}	-3.475×10^{-5}	-8.097×10^{-9}	0	0	6.85	826.9	8.605	3.497×10^{-3}	1.5
FDA	PCES	47.40	1.318×10^{-1}	-7.171×10^{-5}	6.759×10^{-9}	0	0	6.85	826.9	8.605	8.472×10^{-3}	1.5
	this work	47.40	1.318×10^{-1}	-7.171×10^{-5}	6.759×10^{-9}	0	0	6.85	826.9	8.605	8.472×10^{-3}	1.5
DEMB	PCES	54.14	1.443×10^{-1}	-6.264×10^{-5}	-3.511×10^{-9}	0	0	6.85	826.9	8.605	9.929×10^{-3}	1.5
	this work	54.14	1.443×10^{-1}	-6.264×10^{-5}	-3.511×10^{-9}	0	0	6.85	826.9	8.605	9.929×10^{-3}	1.5
GSH	PCES	76.37	2.049×10^{-1}	-1.548×10^{-4}	4.355×10^{-8}	0	0	6.85	826.9	8.605	1.476×10^{-2}	1.5
	this work	76.37	2.049×10^{-1}	-1.548×10^{-4}	4.355×10^{-8}	0	0	6.85	826.9	8.605	1.476×10^{-2}	1.5
VITB5	PCES	59.83	1.831×10^{-1}	-1.497×10^{-4}	5.168×10^{-8}	0	0	6.85	826.9	8.605	1.120×10^{-2}	1.5
	this work	59.83	1.831×10^{-1}	-1.497×10^{-4}	5.168×10^{-8}	0	0	6.85	826.9	8.605	1.120×10^{-2}	1.5
HCYS	PCES	34.60	8.706×10^{-2}	-6.435×10^{-5}	1.923×10^{-8}	0	0	6.85	826.9	8.605	5.674×10^{-3}	1.5
	this work	34.60	8.706×10^{-2}	-6.435×10^{-5}	1.923×10^{-8}	0	0	6.85	826.9	8.605	5.674×10^{-3}	1.5
AH	PCES	43.67	1.118×10^{-1}	-7.001×10^{-5}	9.936×10^{-9}	0	0	6.85	826.9	8.605	7.648×10^{-3}	1.5
	this work	43.67	1.118×10^{-1}	-7.001×10^{-5}	9.936×10^{-9}	0	0	6.85	826.9	8.605	7.648×10^{-3}	1.5

Table 8. Data Estimated by Applying Andrade Model with Parameters from the PCES Method and Current Python-Based Algorithm

substance	method	$C_{\eta,1}$	$C_{\eta,2}$	$C_{\eta,3}$	$C_{\eta,4}$	$C_{\eta,5}$
DHMF	PCES	101.1	-5.636×10^{3}	−14.73	150.7	364.2
	this work	101.1	-5.638×10^{3}	−14.73	150.9	364.5
FDA	PCES	151.5	-9.730×10^{3}	−21.45	226.8	418.3
	this work	151.5	-9.733×10^{3}	−21.45	227.0	418.5
DEMB	PCES	146.1	-8.514×10^{3}	−21.08	176.1	340.4
	this work	146.1	-8.518×10^{3}	−21.08	176.3	340.6
GSH	PCES	551.3	-7.079×10^{4}	−69.69	788.7	1018
	this work	551.4	-7.080×10^{4}	−69.68	788.9	1019
VITB5	PCES	510.7	-5.317×10^{4}	−66.60	562.4	747.5
	this work	510.7	-5.318×10^{4}	−66.60	562.6	747.8
HCYS	PCES	186.0	-1.364×10^{4}	−25.69	298.2	519.9
	this work	186.1	-1.364×10^{4}	−25.69	298.4	520.2
AH	PCES	239.0	-1.864×10^{4}	−32.66	339.4	537.7
	this work	239.0	-1.864×10^{4}	−32.66	339.6	538.0

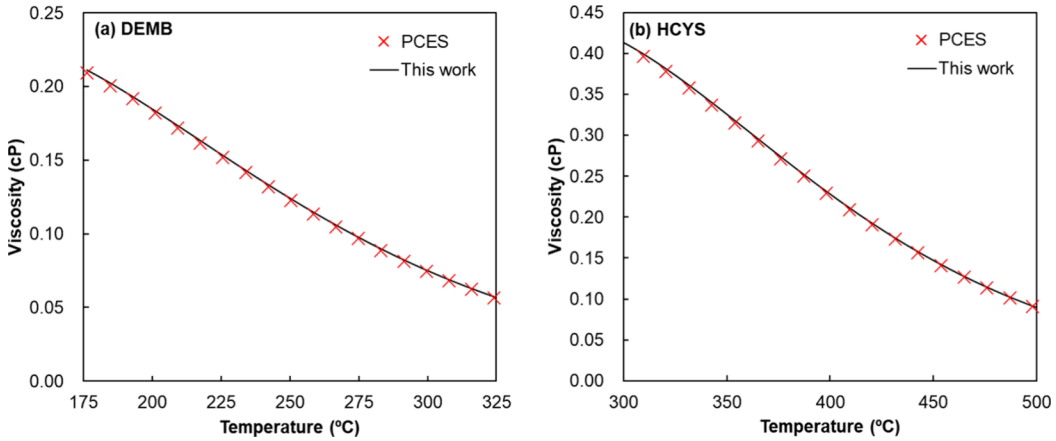


Figure 4. Liquid viscosity for DEMB (a) and HCYS (b) simulated using the Andrade model with parameters estimated by the PCES method and the current Python-based algorithm.

the extended Antoine equation (eqs 26 and 27); the extended Antoine equation performs extrapolation beyond the T_b and T_c ranges. Anticipating the potential for heightened physical inconsistency, simulations of the Clausius−Clapeyron equation and regression with the Watson model were performed within the temperature range of T_b to T_c.

3.5. Ideal Gas Heat Capacity. Table 7 presents the results of the analytical conversion of the ideal gas heat capacity model parameters obtained through the Joback method into the Aspen ideal gas heat capacity polynomial model parameters. The results obtained with PCES and the current Python-based algorithm were identical. Therefore, the percent mean absolute residuals are zero.

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

3.6. Viscosity. Table 8 lists the parameters of the Andrade liquid viscosity model estimated using the PCES method and the current Python-based algorithm. Although the exact algorithm for the PCES method is unknown, the values were nearly identical to the results obtained with the Python-based algorithm. As shown in Figure 4, the simulations employing the Andrade model with the parameters obtained from the PCES method and the current Python-based algorithm yielded nearly identical results for DEMB and HCYS. The calculation results for the liquid viscosity of all substances can be seen in Figure S4 in the Supporting Information. The percent mean absolute residuals between the PCES method and the Python-based algorithm for η^{l} are 1.05, 1.06, 1.08, 1.04, 1.08, 1.06, and 1.07% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively.

The dipole moment is essential for calculating the gas viscosity by using the Chapman−Enskog−Brokaw model. Figure 5 shows the vector of the dipole moment and the energy-optimized molecular structures of DEMB and HCYS obtained by using Avogadro software. The calculation results for the dipole moment of all substances can be seen in Figure S5 in the Supporting Information. Table 9 summarizes the dipole moments predicted by using Avogadro software. The PCES method does not include an algorithm for estimating the dipole moments; therefore, unless the user provides this value, the dipole moment is treated as zero. Group contribution methods are available for calculating dipole moments,[41,42] but relatively low accuracy is expected owing to the three-dimensional characteristics of the dipole moment. Therefore, free software, such as Avogadro, which can simulate three-dimensional structural information using molecular dynamics, may be a useful alternative for calculating dipole moments. Figure 6 shows the gas viscosities of DEMB and HCYS simulated by using the Chapman−Enskog−Brokaw model. The calculation results for the gas viscosity of all substances can be seen in Figure S6 in the Supporting Information. The results obtained with PCES and the current Python-based algorithm are identical. The percent mean absolute residuals between the PCES method and the current Python-based algorithm for η^{g} are 0.40, 0.67, 0.66, 0.48, 0.61, 0.79, and 0.98% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively.

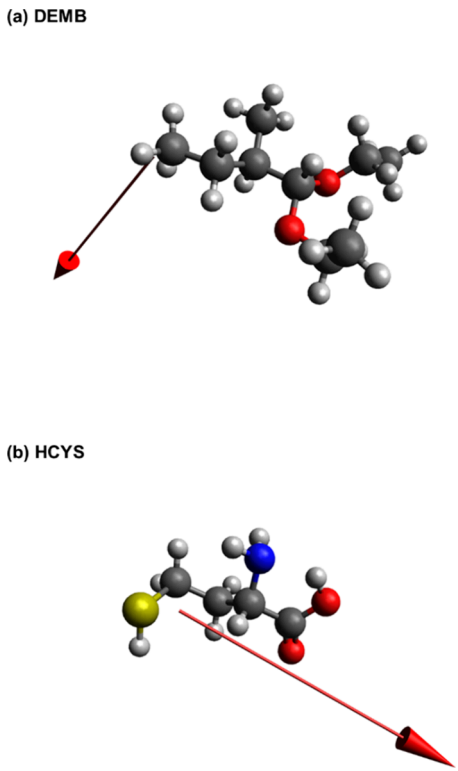


Figure 5. Dipole moment vector with energy-optimized molecular structure for DEMB (a) and HCYS (b) simulated using Avogadro software. Light gray, dark gray, red, blue, and yellow spheres indicate hydrogen, carbon, oxygen, nitrogen, and sulfur atoms, respectively. Red arrow indicates the vector of the dipole moment.

Table 9. Dipole Moments Obtained by Molecular Dynamics Simulation Using Avogadro Software

substance	μ (Debye)
DHMF	1.500
FDA	2.313
DEMB	2.859
GSH	4.767
VITB5	3.425
HCYS	3.330
AH	3.790

3.7. Thermal Conductivity. Table 10 lists the DIPPR equation with 100 model parameters obtained from the Sato−Riedel model. Although the exact data interval, quantity, and data-fitting method used by the PCES algorithm remain undisclosed, the values were almost identical to the results obtained using the current Python-based algorithm. As shown in Figure 7, the liquid thermal conductivity data from the DIPPR eq 100 model using the parameters estimated by the PCES method and the current Python-based algorithms were almost identical for DHMF and AH. The calculation results for the liquid thermal conductivity of all substances can be seen in Figure S7 in the Supporting Information. Figure 8 shows the gas thermal conductivities of DHMF and HCYS simulated by using the Stiel−Thodos model. The results from the PCES method and the current Python-based algorithm are almost identical. The calculation results for the gas thermal conductivity of all substances can be seen in Figure S8 in the Supporting Information. The percent mean absolute residuals between the PCES method and the current Python-based algorithm for λ^{l} are 0.21, 0.21, 0.23, 0.15, 0.17, 0.19%, and 0.19% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively. The percent mean absolute residuals between the PCES method and the current Python-based algorithm for λ^{g} are 0.72, 0.93, 0.93, 0.80, 0.89, 0.97, and 1.32% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively.

3.8. Surface Tension. Table 11 shows the DIPPR eq 106 model parameters estimated from the PCES method and the current Python-based algorithm. Notably, the PCES method and the current Python-based algorithm yielded different results, with a notable difference in $C_{\sigma,1}$. In the current Python-based algorithm, $C_{\sigma,1}$ was analytically determined using eq 82; therefore, the aforementioned difference may be attributed to Y_c in eq 80. Equation 80 is an empirical expression proposed by Miller and Thodos[37] based on experimental data from various substances and was not directly proposed by Brock and Bird.[36] Unfortunately, a model for Y_c that predicts the same $C_{\sigma,1}$ as the PCES method could not be found. Nevertheless, as shown in Figure 9, the surface tension data simulated by using

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

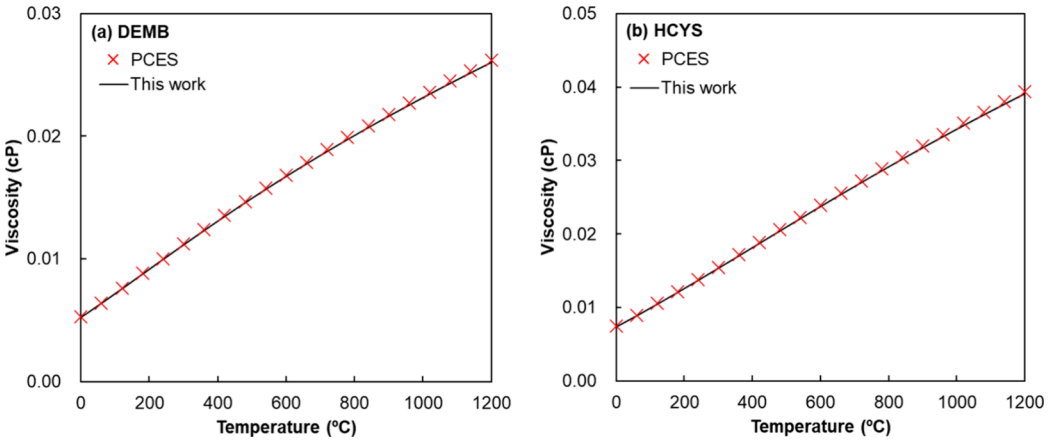


Figure 6. Gas viscosity for DEMB (a) and HCYS (b) simulated by using the Chapman−Enskog−Brokaw model.

Table 10. Estimation of DIPPR Equation 100 Liquid Thermal Conductivity Model Parameters Using the PCES Method and the Current Python-Based Algorithm

substance	method	$C_{\lambda,1}$	$C_{\lambda,2}$	$C_{\lambda,3}$	$C_{\lambda,4}$	$C_{\lambda,5}$	$C_{\lambda,6}$	$C_{\lambda,7}$
DHMF	PCES	6.615×10^{-2}	9.119×10^{-4}	-7.561×10^{-6}	2.216×10^{-8}	-2.482×10^{-11}	150.7	364.2
	this work	6.607×10^{-2}	9.140×10^{-4}	-7.568×10^{-6}	2.216×10^{-8}	-2.479×10^{-11}	150.9	364.5
FDA	PCES	-7.555×10^{-2}	2.416×10^{-3}	-1.326×10^{-5}	3.006×10^{-8}	-2.583×10^{-11}	226.8	418.3
	this work	-7.586×10^{-2}	2.419×10^{-3}	-1.327×10^{-5}	3.006×10^{-8}	-2.581×10^{-11}	227.0	418.5
DEMB	PCES	-2.718×10^{-2}	2.209×10^{-3}	-1.557×10^{-5}	4.437×10^{-8}	-4.802×10^{-11}	176.1	340.4
	this work	-2.746×10^{-2}	2.213×10^{-3}	-1.558×10^{-5}	4.436×10^{-8}	-4.797×10^{-11}	176.3	340.6
GSH	PCES	−3.498	1.666×10^{-2}	-2.902×10^{-5}	2.238×10^{-8}	-6.487×10^{-12}	788.7	1018
	this work	−3.500	1.666×10^{-2}	-2.903×10^{-5}	2.238×10^{-8}	-6.485×10^{-12}	788.9	1019
VITB5	PCES	−2.664	1.783×10^{-2}	-4.314×10^{-5}	4.609×10^{-8}	-1.853×10^{-11}	562.4	747.5
	this work	−2.667	1.784×10^{-2}	-4.315×10^{-5}	4.609×10^{-8}	-1.852×10^{-11}	562.6	747.8
HCYS	PCES	−0.1856	3.243×10^{-3}	-1.346×10^{-5}	2.420×10^{-8}	-1.624×10^{-11}	298.2	519.9
	this work	−0.1860	3.246×10^{-3}	-1.365×10^{-5}	2.420×10^{-8}	-1.623×10^{-11}	298.4	520.2
AH	PCES	−0.4443	5.500×10^{-3}	-2.084×10^{-5}	3.407×10^{-8}	-2.102×10^{-11}	339.4	537.7
	this work	−0.4451	5.505×10^{-3}	-2.085×10^{-5}	3.406×10^{-8}	-2.101×10^{-11}	339.6	538.0

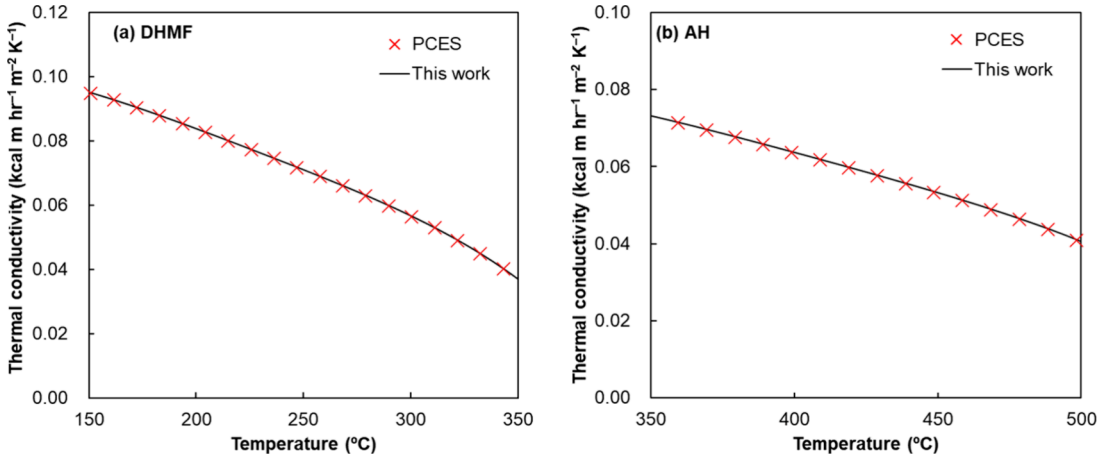


Figure 7. Liquid thermal conductivity for DHMF (a) and AH (b) simulated using the DIPPR eq 100 model with parameters estimated by the PCES method and the current Python-based algorithm.

the parameters obtained from the PCES method and the current Python-based algorithm were almost identical. The calculation results for the surface tension of all substances can be seen in Figure S9 in the Supporting Information. The percent mean absolute residuals of the PCES method and the current Python-based algorithm for σ are 2.76, 2.76, 2.88, 2.51, 2.59, 2.63, and 2.66% for DHMF, FDA, DEMB, GSH, VITB5, HCYS, and AH, respectively.

3.9. Comparative Summary and Future Work. The primary objective of this study is to compare the PCES method with the current Python-based algorithm and to make it available to the public. It was found that the various scalar properties, vapor pressure, liquid molar volume, ideal gas heat capacity, viscosity, thermal conductivity, and surface tension predicted by both methods show almost identical results. However, as seen in Figure 3, the results predicted for the

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

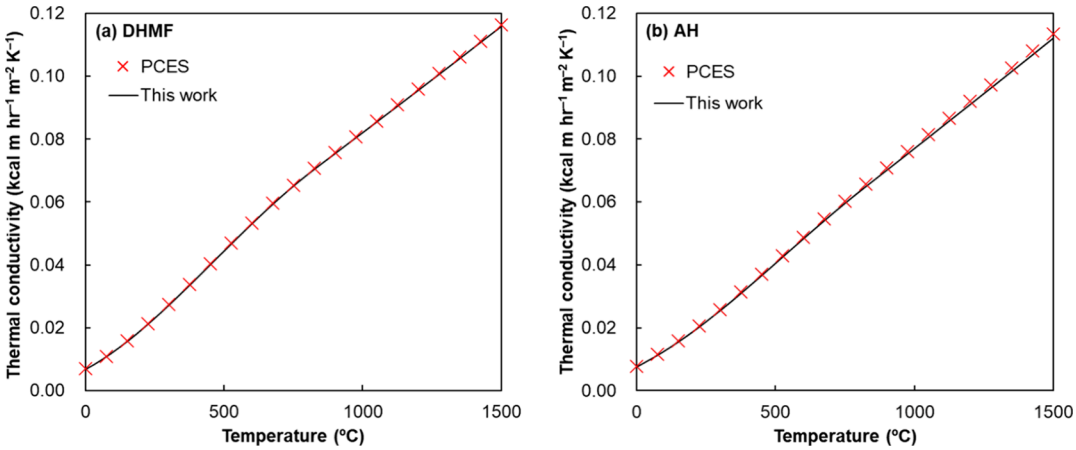


Figure 8. Gas thermal conductivity for DHMF (a) and AH (b) simulated by using the Stiel−Thodos model.

Table 11. Estimation Results of DIPPR Equation 106 Surface Tension Model Parameters Using the PCES Method and the Current Python-Based Algorithm

substance	method	$C_{\sigma,1}$	$C_{\sigma,2}$	$C_{\sigma,3}$	$C_{\sigma,4}$	$C_{\sigma,5}$	$C_{\sigma,6}$	$C_{\sigma,7}$
DHMF	PCES	77.30	1.222	0	0	0	150.7	357.7
	this work	79.12	1.222	0	0	0	150.9	358.0
FDA	PCES	71.43	1.222	-5.819×10^{-10}	6.540×10^{-10}	-2.592×10^{-10}	226.8	411.3
	this work	73.12	1.222	0	0	0	227.0	411.6
DEMB	PCES	55.56	1.222	-3.030×10^{-10}	3.406×10^{-10}	-1.348×10^{-10}	176.1	334.2
	this work	56.91	1.222	0	0	0	176.3	334.4
GSH	PCES	191.7	1.222	2.542×10^{-9}	-2.837×10^{-9}	1.083×10^{-9}	788.7	1005
	this work	196.1	1.222	0	0	0	788.9	1006
VITB5	PCES	173.8	1.222	-1.201×10^{-8}	1.342×10^{-8}	-5.131×10^{-9}	562.4	737.2
	this work	177.1	1.222	0	0	0	562.6	737.4
HCYS	PCES	130.3	1.222	-1.319×10^{-9}	1.482×10^{-9}	-5.878×10^{-10}	298.2	511.9
	this work	133.3	1.222	0	0	0	298.4	512.2
AH	PCES	116.2	1.222	2.273×10^{-9}	-2.557×10^{-9}	1.003×10^{-9}	339.4	529.5
	this work	118.9	1.222	0	0	0	339.6	529.8

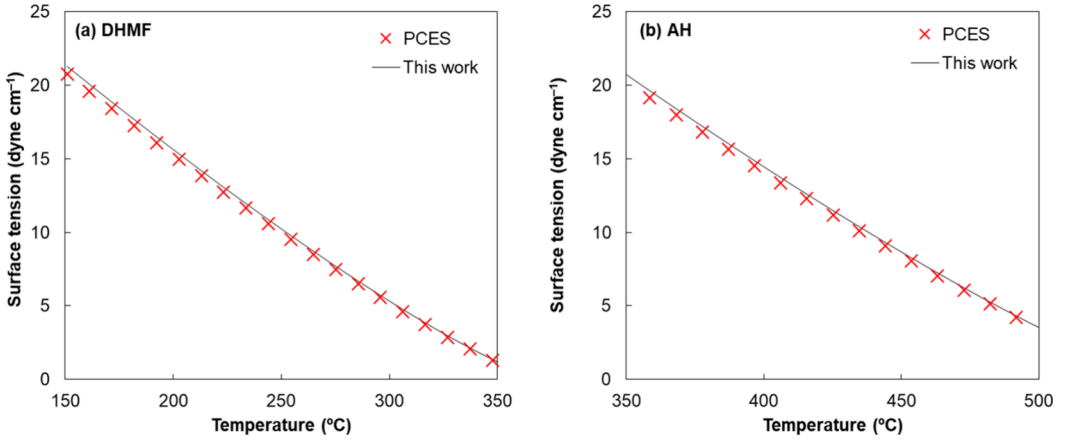


Figure 9. Surface tension of DHMF (a) and AH (b) simulated using the DIPPR eq 106 model with parameters estimated by the PCES method and the current Python-based algorithm.

enthalpy of vaporization by these two methods differed from each other. Although the current Python-based algorithm more accurately matched the Clausius−Clapeyron equation compared to the PCES method, this does not necessarily prove that it is more consistent with the actual experimental results.

If experimental values for the properties of pure substances can be obtained, then comparing them with simulation results is the most useful method for demonstrating accuracy. While the original works for each pure property were developed based on the experimental values of various substances,[22] the value of such comparative studies remains valid for newly predicted substances. Unfortunately, as mentioned in the introduction, the substances applied in our study make it challenging to obtain pure substances through experiments. Our future plans involve developing an in-house Python-based algorithm for the mixture model of the Aspen electrolyte

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909

template. Through this study, several binary mixture simulation results will be compared with various binary mixture experimental values, such as density, heat capacity, viscosity, and thermal conductivity.[43]

Additionally, we compared the experimental values of the well-known enthalpy of vaporization for ethanol[44−46] with the predictions made by the PCES method and the current Python-based method, as shown in Figure 10. The percent

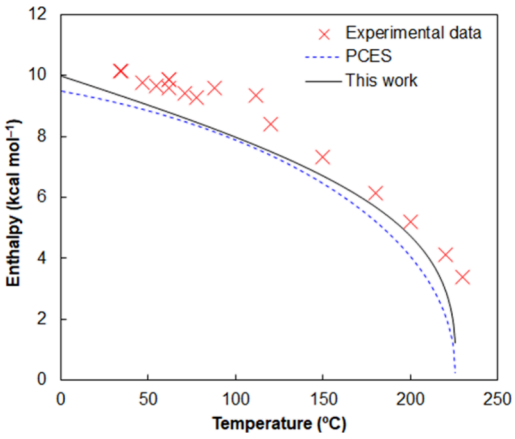


Figure 10. Enthalpy of vaporization for ethanol simulated using the PCES model and the current Python-based algorithm with the experimental data.

mean absolute residuals for the vaporization enthalpy predicted by the PCES method and the current Python-based algorithm were 14.7 and 10.4%, respectively. This suggests that the current Python-based algorithm can be expected to predict acceptable results. Although comparison with experimental data for more substance would be useful in generalizing, it is beyond the scope of this study.

4. CONCLUSIONS

The Aspen PCES method and an in-house Python-based algorithm were compared to estimate the parameters of the pure component property models for substances not registered in the Aspen software. The impurities found in biobased ethanol (DHMF, FDA, and DEMB) and biobased active substances (GSH, VITB5, HCYS, and AH) were analyzed and compared. The estimated parameters for the normal boiling point, critical properties, standard enthalpy, vapor pressure, liquid molar volume, heat capacity, viscosity, thermal conductivity, and surface tension models were nearly identical with those of the PCES method and the current Python-based algorithm. In the case of the enthalpy of vaporization, the current Python-based algorithm estimated parameters that exactly matched the Clausius−Clapeyron equation but yielded different results from the PCES method. The current Python-based algorithm accurately represented the temperature dependence of the enthalpy of vaporization for common substances. Furthermore, the dipole moment was determined using the Avogadro software, and it was verified that the gas viscosity could be calculated using this calculated value. The methods presented in this study provide detailed and clear references for estimating the parameters of pure component property models.

■ ASSOCIATED CONTENT

Supporting Information

The Supporting Information is available free of charge at https://pubs.acs.org/doi/10.1021/acsomega.3c09657.

An in-house Python-based algorithm capable of estimating the properties of pure components (ZIP)

A comparative graph of property simulations using parameters estimated by the PCES method and the current Python-based algorithm for seven substances—dihydro-2-methyl-3-furanone, 2-furaldehyde diethyl acetal, 1,1-diethoxy-3-methyl butane, glutathione, vitamin B5, homocysteine, and *O*-acetyl-L-homoserine (PDF)

■ AUTHOR INFORMATION

Corresponding Authors

Wangyun Won − *Department of Chemical and Biological Engineering, Korea University, Seoul 02841, Republic of Korea*; Email: wwon@korea.ac.kr

Jun-Woo Kim − *CJ BIO Research Institute, CJ CheilJedang Corp., Suwon-Si, Gyeonggi-do 16495, Republic of Korea*; Email: junwoo.kim1@cj.net

Author

Jina Lee − *CJ BIO Research Institute, CJ CheilJedang Corp., Suwon-Si, Gyeonggi-do 16495, Republic of Korea*; orcid.org/0009-0002-8217-7039

Complete contact information is available at: https://pubs.acs.org/10.1021/acsomega.3c09657

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work is supported by the CJ BIO Research Institute, CJ CheilJedang, South Korea. Thanks are given to CJ CheilJedang for granting permission to publish this article.

■ REFERENCES

(1) Khan, M. A. H.; Bonifacio, S.; Clowes, J.; Foulds, A.; Holland, R.; Matthews, J. C.; Percival, C. J.; Shallcross, D. E. Investigation of biofuel as a potential renewable energy source. *Atmosphere.* **2021**, *12*, 1289.

(2) Lee, G. N.; Na, J. The impact of synthetic biology. *ACS Synth. Biol.* **2013**, *2*, 210−212.

(3) Oh, M. Y.; Gujjala, L. K. S.; Won, W. Process development for production of platform chemicals from white birch: Insights from techno-economic and life-cycle assessment. *Chem. Eng. J.* **2023**, *472*, No. 144955.

(4) Wooley, R. J.; Putsche, V. Development of an ASPEN PLUS physical property database for biofuels components. *NREL/TP-425−20685* 1996. National Renewable Energy Lab. (NREL), Golden, CO, United States.

(5) Humbird, D.; Davis, R.; Tao, L.; Kinchin, C.; Hsu, D.; Aden, A.; Schoen, P.; Lukas, J.; Olthof, B.; Worley, M.; Sexton, D.; Dudgeon, D. Process design and economics for biochemical conversion of lignocellulosic biomass to ethanol: dilute-acid pretreatment and enzymatic hydrolysis of corn stover. *NREL/TP-5100−47764* 2011. National Renewable Energy Lab. (NREL), Golden, CO, United States,.

(6) O'Connell, J. P.; Gani, R.; Mathias, P. M.; Maurer, G.; Olson, J. D.; Crafts, P. A. Thermodynamic property modeling for chemical process and product engineering: Some perspectives. *Ind. Eng. Chem. Res.* **2009**, *48*, 4619−4637.

(7) Vohra, M.; Manwar, J.; Manmode, R.; Padgilwar, S.; Patil, S. Bioethanol production: Feedstock and current technologies. *J. Environ. Chem. Eng.* **2014**, *2*, 573−584.
(8) Jang, Y.-S.; Kim, B.; Shin, J. H.; Choi, Y. J.; Choi, S.; Song, C. W.; Lee, J.; Park, H. G.; Lee, S. Y. Bio-based production of C2−C6 platform chemicals. *Biotechnol. Bioeng.* **2012**, *109*, 2437−2459.
(9) Amornraksa, S.; Subsaipin, I.; Simasatitkul, L.; Assabumrungrat, S. Systematic design of separation process for bioethanol production from corn stover. *BMC Chem. Eng.* **2020**, *2*, 1−16.
(10) Habe, H.; Shinbo, T.; Yamamoto, T.; Sato, S.; Shimada, H.; Sakaki, K. Chemical analysis of impurities in diverse bioethanol samples. *J. Jpn. Pet. Inst.* **2013**, *56*, 414−422.
(11) Sánchez, C.; Santos, S.; Sánchez, R.; Lienemann, C. P.; Todolí, J. L. Profiling of organic compounds in bioethanol samples of different nature and the related fractions. *ACS Omega* **2020**, *5*, 20912−20921.
(12) Ferdous, J.; Bensebaa, F.; Pelletier, N. Integration of LCA, TEA, process simulation and optimization: A systematic review of current practices and scope to propose a framework for pulse processing pathways. *J. Cleaner Prod.* **2023**, *402*, No. 136804.
(13) Ureta, M. M.; Salvadori, V. O. A review of commercial process simulators applied to food processing. *J. Food Process Eng.* **2023**, *46*, No. e14225.
(14) "*Property parameter estimation*", Aspen physical property system reference manuals, Version 11.1, Aspen Technology.
(15) Weininger, D. SMILES, a chemical language and information system. 1. Introduction to methodology and encoding Rules. *J. Chem. Inf. Comput. Sci.* **1988**, *28*, 31−36.
(16) Joback, K. G.; Reid, R. C. Estimation of pure-component properties from group-contributions. *Chem. Eng. Commun.* **1987**, *57*, 233−243.
(17) Shi, C.; Borchardt, T. B. JRgui: A python program of Joback and Reid method. *ACS Omega* **2017**, *2*, 8682−8688.
(18) Pitzer, K. S.; Lippmann, D. Z.; Curl, R. F., Jr.; Huggins, C. M.; Petersen, D. E. The volumetric and thermodynamic properties of fluids. II. Compressibility factor, vapor pressure, and entropy of vaporization. *J. Am. Chem. Soc.* **1955**, *77*, 3433−3440.
(19) Hanwell, M. D.; Curtis, D. E.; Lonie, D. C.; Vandermeersch, T.; Zurek, E.; Hutchison, G. R. Avogadro: An advanced semantic chemical editor, visualization, and analysis platform. *J. Chem. Inf.* **2012**, *4*, 1−17.
(20) Halgren, T. A. Merck molecular force field. I. Basis, form, scope, parameterization, and performance of MMFF94. *J. Comput. Chem.* **1996**, *17*, 490−519.
(21) Vetere, A. Again the Riedel equation. *Fluid Phase Equilib.* **2006**, *240*, 155−160.
(22) Gunn, R. D.; Yamada, T. A corresponding states correlation of saturated liquid volumes. *AIChE J.* **1971**, *17*, 1341−1345.
(23) Rackett, H. G. Equation of state for saturated liquids. *J. Chem. Eng. Data* **1970**, *15*, 514−517.
(24) "*Rackett extrapolation method*", Aspen plus reference manuals, Veraion 11.1, Aspen Technology.
(25) Anderson, G. K. Enthalpy of dissociation and hydration number of carbon dioxide hydrate from the Clapeyron equation. *J. Chem. Thermodyn.* **2003**, *35*, 1171−1183.
(26) Redlich, O.; Kwong, J. N. On the thermodynamics of solutions. V. An equation of state. fugacities of gaseous solutions. *Chem. Rev.* **1949**, *44*, 233−244.
(27) Watson, K. Prediction of critical temperatures and heats of vaporization. *Ind. Eng. Chem.* **1931**, *23*, 360−364.
(28) Gao, F.; Han, L. Implementing the Nelder-Mead simplex algorithm with adaptive parameters. *Comput. Optim. Appl.* **2012**, *51*, 259−277.
(29) International Organization for Standardization. ISO 31−4:1992, *Quantities and units, Part 4: Heat*. Annex B (informative): Other units given for information, especially regarding the conversion factor., 1992.
(30) Letsou, A.; Stiel, L. I. Viscosity of saturated nonpolar liquids at elevated pressures. *AIChE J.* **1973**, *19*, 409−411.
(31) Andrade, E. D. C. The viscosity of liquids. *Nature* **1930**, *125*, 309−310.
(32) Brokaw, R. S. Predicting transport properties of dilute gases. *Ind. Eng. Chem. Process Des. Dev.* **1969**, *8*, 240−253.
(33) Neufeld, P. D.; Janzen, A. R.; Aziz, R. Empirical equations to calculate 16 of the transport collision integrals $\Omega(l, s)^*$ for the Lennard-Jones (12−6) potential. *J. Chem. Phys.* **1972**, *57*, 1100−1102.
(34) Latini, G.; Di Nicola, G.; Pierantozzi, M. A critical survey of thermal conductivity literature data for organic compounds at atmospheric pressure and an equation for aromatic compounds. *Energy Procedia* **2014**, *45*, 616−625.
(35) Stiel, L. I.; Thodos, G. The thermal conductivity of nonpolar substances in the dense gaseous and liquid regions. *AIChE J.* **1964**, *10*, 26−30.
(36) Brock, J. R.; Bird, R. B. Surface tension and the principle of corresponding states. *AIChE J.* **1955**, *1*, 174−177.
(37) Miller, D. G.; Thodos, G. Correspondence. Reduced Frost-Kalkwarf vapor pressure equation. *Ind. Eng. Chem. Fundam.* **1963**, *2*, 78−80.
(38) *RDKit: Open-source cheminformatics*. https://www.RDKit.org.
(39) Harris, C. R.; Millman, K. J.; Van Der Walt, S. J.; Gommers, R.; Virtanen, P.; Cournapeau, D.; Wieser, E.; Taylor, J.; Berg, S.; Smith, N. J.; Kern, R.; Picus, M.; Hoyer, S.; van Kerkwijk, M. H.; Brett, M.; Haldane, A.; del Río, J. F.; Wiebe, M.; Peterson, P.; Gérard-Marchant, P.; Sheppard, K.; Reddy, T.; Weckesser, W.; Abbasi, H.; Gohlke, C.; Oliphant, T. E. Array programming with NumPy. *Nature* **2020**, *585*, 357−362.
(40) Poling, B. E.; Robert, C. R.; Prausnitz, J. M. *The Properties of Gases and Liquids*, 4th ed.; McGraw-Hill: New York, 1987.
(41) Müller, K.; Mokrushina, L.; Arlt, W. Second-order group contribution method for the determination of the dipole moment. *J. Chem. Eng. Data* **2012**, *57*, 1231−1236.
(42) Sheldon, T. J.; Adjiman, C. S.; Cordiner, J. L. Pure component properties from group contribution: Hydrogen-bond basicity, hydrogen-bond acidity, hildebrand solubility parameter, macroscopic surface tension, dipole moment, refractive index, and dielectric constant. *Fluid Phase Equilib.* **2005**, *231*, 27−37.
(43) Kim, J.-W.; Lee, K. H.; Park, W. H.; Hong, S. B.; Park, C.; Kim, M.; Kim, J.-K. Development of thermophysical property models for aqueous amino acid solutions. *Chem. Eng. Technol.* **2023**, *46*, 702−710.
(44) Stephenson, R. M.; Malanowski, S. *Handbook of the Thermodynamics of Organic Compounds*; Elsevier: New York, 1987.
(45) Dong, J. Q.; Lin, R. S.; Yen, W. H. Heats of vaporization and gaseous molar heat capacities of ethanol and the binary mixture of ethanol and benzene. *Can. J. Chem.* **1988**, *66*, 783−790.
(46) Vine, M. D.; Wormald, C. J. The enthalpy of ethanol. *J. Chem. Thermodyn.* **1989**, *21*, 1151−1157.

https://doi.org/10.1021/acsomega.3c09657
ACS Omega 2024, 9, 11895−11909