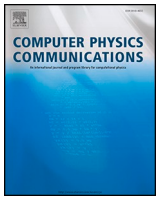
Computer Physics Communications 297 (2024) 109068

Contents lists available at ScienceDirect

Computer Physics Communications

journal homepage: www.elsevier.com/locate/cpc



Computational Physics

GenEOS: An accurate equation of state for the fast calculation of two-phase geofluids properties based on gene expression programming

Morteza Esmaeilpour *, Fabian Nitschke, Thomas Kohl

Geothermal Energy and Reservoir Technology, Institute of Applied Geosciences, Karlsruhe Institute of Technology, Karlsruhe, Germany

ARTICLE INFO

Keywords:
Equation of state
Gene expression programming
Machine learning
Two-phase flow
Multicomponent geofluids

ABSTRACT

Numerical simulation of two-phase multicomponent flows requires solving continuity, momentum, energy, and transport equations. Typically, these conservation equations are solved for computing the main variables of pressure, enthalpy, velocity, and composition. Variation of thermophysical properties (e.g., density, viscosity, etc.) as functions of the main variables necessitates introducing equations of state (EOS) to the modeling scheme, equating the number of unknowns and equations. The problem arises here as almost all the available EOSs in the literature receive temperature as an input, which is not a main variable. Guessing temperature, as an unknown input, imposes more iterations on the already iterative algorithm of the EOS and increases the computational cost. The primary focus of this study is to provide highly-precise, but fast EOS scheme for calculating two-phase fluid properties using artificial intelligence algorithms. In the first step, a Fugacity-Activity model is implemented to supply a supervised learning algorithm with a large dataset. The provided data are fed into a machine-learning (ML) model called gene expression programming (GEP). The outputs of this GEP model are high-preciseness explicit formulas for non-iterative computing of temperature and equilibrium constants. Testing the proposed GEP equations for 1,000,000 arbitrary sets of inputs revealed high accuracy in predicting desired outputs (e.g., $<$ 0.6% error in calculating temperature). Implementing GEP equations in modeling platforms can result in ~90% reduction in EOS-related computational cost. This ML-based EOS is a transparent box for computing thermophysical properties of two-phase mixtures containing H_2O, CO_2, CH_4, N_2, H_2S, NaCl, KCl, $CaCl_2$, and $MgCl_2$.

1. Introduction

Control of thermal fluids is essential for producing renewable geothermal energy [1,2] or storing/extracting solutes in deep reservoirs [3,4]. Depending on the specific application (e.g., leaching, underground CO_2 storage, heat extraction), the circulated fluid should be engineered to be a proper carrier of energy to maximize the power production rate [5], have high mobility factor (inverse kinematic viscosity) to be easily circulated in the system [6], be a good solvent of target minerals to enhance absorption rate [7], and not cause any environmental hazard. Two-phase conditions may occur under a specific P-T situation, like gas injection into geothermal brines [8] or under production conditions when non-condensable gases (NCG) are released [9]. Injecting CO_2 for carbon capture, utilization, and storage (CCUS) [10,11] and enhancing oil recovery (EOR) [12,13] are the most-renown examples of two-phase flows in reservoirs. It is also noteworthy that the high pressure of deep geothermal reservoirs typically prevents water from boiling. However, the co-occurring high temperature and low pressure in production wellbores may cause the fluid to enter the two-phase regime [14].

Calculating two-phase geofluids properties is difficult, as they are complex functions of pressure, temperature, and the composition of both liquid and gas phases. The computational cost of modeling fluid flow and heat transfer in large 3D reservoirs escalates due to the need for determining these properties. This challenge arises from using highly iterative algorithms to calculate the composition of each phase. The equilibrium condition, when the chemical potential of each component in the aqueous phase (AqP) equals that in the non-AqP (NaqP), can be expressed by either Fugacity–Fugacity (F–F) or Fugacity–Activity (F–A) models. These two methods use different thermodynamic properties for defining the equilibrium state of each component in a two-phase mixture. The F–A model introduces equilibrium constants and ion activities to represent the chemical potentials in the AqP, while the fugacity coefficients of gas components are calculated using an equation of state (EOS). However, The F–F model uses the classical cubic EOSs (e.g., Soave-Redlich-Kwong and Peng-Robinson) to compute the chemical potential in both phases.

* Corresponding author.
E-mail address: morteza.esmaeilpour@kit.edu (M. Esmaeilpour).

https://doi.org/10.1016/j.cpc.2023.109068
Received 4 August 2023; Received in revised form 13 November 2023; Accepted 15 December 2023
Available online 16 December 2023

Nomenclature

Latin Symbols

R	gas universal constant
T	temperature
P	pressure
y	mole fraction in the gas phase
x	mole fraction in the liquid phase
K_H	Henry's constant
Nw	number of moles per kilogram of water
V_{H_2O}	average partial molar volume of the water
K	equilibrium constant
Z	compressibility factor
m_C	cation molality
m_A	anion molality
n^v	mole fraction of non-AqP
H	enthalpy
V	specific volume
w	mass fraction
M	molar mass
b	molality
cm	gas-dependent volume shift factor
Mw	molecular weight
k_B	Boltzmann's constant
N_A	Avogadro's number

Greek symbols

μ	chemical potential
$\varnothing$	fugacity coefficient
γ	activity coefficient
θ	temperature in °C
ω	acentric factor
ρ	density
λ	second-order interaction parameter
ζ	third-order interaction parameter
ε	thermal expansion coefficient
ν	molar volume
η	viscosity
λ	thermal conductivity
σ	Lennard-Jones size parameter
Ω	collision integral

Subscripts and superscripts

0	reference condition
AqP	aqueous phase
NaqP	non-aqueous phase
i	component
c	critical condition
solution	brine mixture without dissolved gases
$'$	brine mixture with dissolved gases
sv	saturated water vapor
sl	saturated liquid water
ps	pseudocritical

Various numerical software packages apply EOS algorithms to calculate two-phase fluid properties. The TOUGH software family is widely used for modeling non-isothermal multiphase multicomponent flows [15]. These codes contain several equations of state modules that calculate the thermodynamic properties of various fluid systems using F-A and F-F algorithms. The ECO2N [16] and ECO2N V2.0 [17] modules were specifically developed to simulate the flow of CO_2-brine systems at temperatures of up to 110°C and 250°C, respectively. Neither of these two modules has the capability to calculate gas-mixture properties. In order to incorporate additional gas components, such as N_2 and CH_4, in the two-phase mixture, Oldenburg et al. [18] developed another module called EOS7C. Using a highly-iterative thermodynamic model to solve the mutual solubility of gases in brine makes EOS7C computationally expensive. Pruess and Battistelli [19] proposed TMVOC as a TOUGH2 module to model three-phase systems comprising gas, aqueous, and non-aqueous phase liquids. However, TMVOC was developed to simulate near-surface contamination, characterized by low pressure and temperature conditions. TMGAS is another TOUGH2 module that has been developed based on the TMVOC module to simulate the injection of gas mixtures into deep geological sites [20]. Unlike other modules, TMGAS uses the F-F model to calculate phase equilibrium in gas mixtures and brine systems. It poses a considerable computational cost when included in the numerical modeling of multiphase multicomponent flows, and its accuracy decreases for salinities greater than 2mol/kg water. EWASG is a TOUGH module developed for modeling non-condensable gas-brine systems with temperatures ranging between 100 and 350 °C [21]. It assumes perfect gas behavior, which simplifies the calculation of gas properties. However, it is important to note that this assumption may not accurately represent the behavior of real gases at extreme pressures or low temperatures.

Further studies have been conducted applying EOS algorithms to determine the phase composition. Numerous non-iterative F-A models predict the phase equilibrium of binary systems, such as CO_2-brine [22–30], H_2S-brine [31], and CH_4-brine [32]. However, calculating the solubility in two-phase mixtures with several gas components is iterative and requires solving the complex Rachford-Rice equation. Ziabakhsh-Ganji and Kooi [33] used this iterative F-A approach to model the thermodynamic equilibrium in brine-gas mixtures containing CO_2, CH_4, N_2, H_2S, and SO_2. In their proposed model, fugacity coefficients of gas components are calculated by Peng-Robinson (PR) EOS, while Pitzer formalism and Henry's law are implemented to compute the activity coefficients in the AqP. Appelo *et al.* [34] introduced a more general model for calculating the apparent molar volumes of single ions. While their modifications are now embedded in the F-A algorithm of PHREEQC, the validity of H_2S solubility is still unclear. Francke *et al.* [35] tried to unify Duan's single-gas solubility functions [22,32,36] to model the gas dissolution in CO_2–N_2–CH_4-brine mixtures. However, assuming ideal gas behavior (fugacity coefficients equal to unity) makes this model inappropriate for geothermal applications with a high range of pressures and temperatures. Zirrahi *et al.* [37] tried to develop a non-iterative F-A model to describe the phase equilibrium behavior of brine-gas mixtures, including CO_2, H_2S, and CH_4. However, their proposed method cannot accurately predict CO_2 and H_2S solubility in the AqP [38]. In 2015, Li *et al.* [39] formulated an iterative F-F model to calculate the mutual solubility of gas mixtures (CO_2 - SO_2 - H_2S - CH_4 - N_2) in brine. Their suggested model is suitable for a wide range of pressures, temperatures, and salinity. However, the significant computational costs involved in iteratively solving the Rachford-Rice equation render it unsuitable for use in reactive transport simulations. Li *et al.* [40] compared the calculation speed of the F-F and F-A models in predicting the mutual solubility of CO_2 - H_2S - CH_4 mixtures in brine and confirmed that the F-A models are much faster than the F-F models.

All the EOS algorithms discussed above use pressure, temperature, and two-phase composition as inputs. However, most of the conducted studies on numerical modeling of multiphase multicomponent flows solve continuity, momentum, energy, and transport equations to calculate the main variables of pressure, enthalpy, velocity, and two-phase composition. Using these EOSs in the simulation process of

multiphase reactive transports considerably increases the computational cost since temperature, as an unknown, needs to be guessed at the beginning of the algorithm. In other words, considering enthalpy instead of temperature as an input multiplies the iterations required for calculations of gas solubility in the AqP.

The main focus of the current study is to propose a novel EOS, applicable to the fast simulation of multiphase/multicomponent flows. This work is distinguished from the existing body of knowledge with the following novelties:

(1). For the first time, an artificial intelligence (AI) technique, called gene expression programming (GEP), is implemented to propose a new formulation for the non-iterative calculation of thermodynamic properties.
Unlike other thermodynamic algorithms, GenEOS takes enthalpy as an input, making it suitable for fast numerical modeling of multiphase and multicomponent transport.
(2). Focusing mainly on geothermal applications, GenEOS is developed for two-phase mixtures of water, CO_2, CH_4, N_2, H_2S, NaCl, KCl, $CaCl_2$, and $MgCl_2$.
(3). In contrast to some other EOSs (e.g., TMGAS), GenEOS presents a transparent box for the fast computation of fluid properties.

Accuracy, computation speed, applicability, and *transparency* are the key characteristics that make an EOS suitable for implementation in reactive transport modeling. Since it is challenging to achieve all of these targets simultaneously, most studies tend to prioritize some of these targets at the expense of others. However, the four aforementioned points strongly indicate that we have achieved our primary goal of developing a suitable EOS without compromising any of its desired characteristics. The development of this novel calculation scheme is organized in two steps: 1) using the F-A model to provide large datasets for training the GEP model, 2) assessing the accuracy of GEP equations in predicting target outputs, and 3) discussing the methods/equations for computing other thermophysical properties.

2. Methodology

While the F-A algorithm is less time-expensive than the F-F model in calculating gas solubility, it still imposes a considerable computational cost when implemented in the numerical modeling of two-phase multicomponent flows. The primary suggestion of this study for accelerating the computation process is to use an artificial intelligence technique called gene expression programming, GEP. In contrast to widely-used machine learning methods like Neural Networks (NN), GEP has a unique ability to produce precise, *explicit formulas* for calculating desired outputs. These generated equations can be effortlessly incorporated into computer programs, bypassing the need for complex iterative algorithms, such as those used in thermodynamic modeling. GEP models are known for their transparency and interpretability, making it easier to understand the underlying logic behind the predictions. Furthermore, GEP typically involves fewer hyperparameters to adjust compared to neural networks, which often demand intricate fine-tuning of various aspects like layers, units, and learning rates. This streamlined approach simplifies the model development process.

The following sections provide detailed explanations of both, the conventional thermodynamic and the novel GEP algorithms. Primarily, we provide a comprehensive outline of the calculation procedure for the major components of the F-A model, explaining the necessity of employing an iterative approach. Despite the absence of sufficient experimental data, this time-consuming conventional method can provide substantial and reliable inputs for machine learning. In the subsequent section, the GEP model is fed by the outputs of the F-A model.

2.1. Thermodynamic algorithm based on the iterative F-A model

- Chemical potential

The F-A model used in this study [33] is capable of describing thermodynamic equilibrium between a NaqP in gas/supercritical/condensed conditions and an AqP including water and dissolved gases/solids. In this model, the chemical potentials of each component in the AqP and NaqP are assumed to be equal to each other and calculated by the following equations:

$$\mu^{NaqP}(T,P) = \mu_0^{NaqP}(T,P) + RT\ \ln(f) \tag{1}$$

$$\mu^{AqP}(T,P) = \mu_0^{AqP}(T,P) + RT\ \ln(a) \tag{2}$$

Where μ_0, R, T, and P represent the chemical potential at the reference temperature, the gas constant, temperature, and pressure, respectively. f, the fugacity of gas components, is calculated by:

$$f = P\varnothing y \tag{3}$$

In which y is the mole fraction of each component in the NaqP and $\varnothing$ denotes the fugacity coefficient. Equating the two chemical potentials Eqs. (1) and ((2)) results in:

$$\frac{\mu_0^{AqP}(T,P) - \mu_0^{NaqP}(T,P)}{RT} = \ln\left(\frac{a}{P\varnothing y}\right) = \ln\left(K^0\right) \tag{4}$$

Given that the equilibrium constant ($K^0 = Nw/K_H$) can be defined by Henry's constant (K_H) and the number of moles per kilogram of water ($Nw = 55.508$), it is possible to re-express Eq. (4) as:

$$\frac{Nw}{K_H} = \frac{a}{P\varnothing y} \tag{5}$$

Assuming that the solubility of gas species in the AqP is small, a in Eq. (5) can be defined by $a = Nw\gamma x$, where γ and x indicate the activity coefficient and mole fraction of components in the AqP [27]. Hence, the new form of Eq. (5) for each gas is [41,42]:

$$(P\varnothing_i y_i)_{NaqP} = (K_{Hi}\gamma_i x_i)_{AqP} \tag{6}$$

Similar to Battistelli and Marcolini [20], the binary interaction between different dissolved gases in the AqP is disregarded, which allows for the non-iterative calculation of activity coefficients. The procedure of calculating $K_i = \frac{y_i}{x_i}$, $\varnothing_i$, K_{Hi}, and γ_i in Eq. (6) are explained in the following.

- Equilibrium constant ($K_i = \frac{y_i}{x_i}$)

While the equilibrium constants of CH_4, CO_2, N_2, and H_2S can be calculated by Eq. (6), the more straightforward and accurate approach of Spycher *et al.* [28] is chosen for computing the water equilibrium constant:

$$K_{H_2O} = \frac{f_{H_2O(g)}}{a_{H_2O(l)}} = K^0_{H_2O}(T,P_0)\exp\left[\frac{(P-P_0)V_{H_2O}}{RT}\right] \tag{7}$$

Where V_{H_2O} represents the average partial molar volume of the water in the AqP (18.1). P_0 is the reference pressure (1bar). $K^0_{H_2O}(T,P_0)$, the equilibrium constant of water at reference pressure is obtained by:

$$\log\left(K^0_{H_2O}\right) = -2.209 + 3.097\ \times 10^{-2}\theta - 1.098\ \times 10^{-4}\theta^2 + 2.048 \times 10^{-7}\theta^3 \tag{8}$$

Where θ is the temperature in °C. Combining Eqs. (3) and (7) results in:

$$y_{H_2O} = \frac{K^0_{H_2O}\ a_{H_2O}}{\varnothing_{H_2O}}\exp\left[\frac{(P-P_0)V_{H_2O}}{RT}\right] \tag{9}$$

Because of the low solubility of gases, water activity can be approximated by its mole fraction in the AqP Consequently, the equation for the equilibrium state of H_2O in the two-phase system can be written as:

$$K^0_{H_2O}\exp\left[\frac{(P-P_0)V_{H_2O}}{RT}\right]x_{H_2O} = \varnothing_{H_2O}Py_{H_2O} \tag{10}$$

- Fugacity coefficient ($\varnothing_i$)

Calculating the fugacity coefficient of gas components as a function of compressibility necessitates solving the classical cubic EOS of Peng-Robinson:

$$Z^3-(1-B)Z^2+\left(A-2B-3B^2\right)Z-\left(AB-B^2-B^3\right)=0 \tag{11}$$

Parameters A and B are functions of temperature and pressure:

$$A = \frac{a(T)P}{(RT)^2} \tag{12}$$

$$B = \frac{bP}{RT} \tag{13}$$

In which

$$a(T) = 0.45724\frac{R^2T_c^2}{P_c}\alpha(T) \tag{14}$$

$$b = 0.07780\frac{RT_c}{P_c} \tag{15}$$

$$\alpha(T) = \left[1+\left(0.37646\ +1.4522\omega\ -0.26992\omega^2\right)\left(1\ -\ \sqrt{\frac{T}{T_c}}\right)\right]^2 \tag{16}$$

In the above equations, ω, P_c, and T_c stand for acentric factor, critical pressure, and critical temperature, respectively. For a gas mixture, the parameters of a and b can be calculated by the following mixing rules:

$$a = \sum_i\sum_j y_iy_ja_{ij}, a_{ij} = \sqrt{a_ia_j}\left(1-k_{ij}\right), b = \sum_i b_iy_i \tag{17}$$

It is noteworthy that in these calculations (Eq. (17)), the mole fraction of water in the gas phase is neglected, which allows for the non-iterative computation of solubility in binary systems (e.g., brine-CO_2, brine-CH_4). However, to accurately determine the compressibility factor, the interaction coefficient (k_{ij}) between water and other gases are modified [33]. The other interaction coefficients are taken from the study conducted by Li and Yan [43]. Finally, the fugacity coefficient can be determined by:

$$\ln(\varnothing_i) = \frac{B_i}{B}(Z-1)-\ln(Z-B)+\frac{A}{2.828B}\left[\frac{B_i}{B}-\frac{2\sum_j y_ja_{ij}}{a}\right]\ln\left[\frac{Z+2.414B}{Z-0.414B}\right] \tag{18}$$

- Henry's constant (K_h)

Henry's constant and activity coefficient are the remaining unknowns on the right-hand side of Eq. (6). The virial-type equation established by Akinfiev and Diamond [41] can be used for calculating thermodynamic properties of the AqP species at infinite dilution. This model gives Henry's constant by:

$$\ln(K_h) = (1-\eta)\ln\left(f^0_{H_2O}\right)+\eta\ln\left(\frac{RT}{M_W}\rho^0_{H_2O}\right)+2\rho^0_{H_2O}\Delta B \tag{19}$$

Where

$$\Delta B = \tau+\Gamma P+\beta\sqrt{\frac{10^3}{T}} \tag{20}$$

For more information about the calculation procedure of fugacity and density of pure water, $f^0_{H_2O}$ and $\rho^0_{H_2O}$, as well as adjustable parameters of η, τ, Γ, and β refer to Fine and Millero [44] and Ziabakhsh-Ganji and Kooi [33].

- Activity coefficient (γ_i)

The reduction of the activity coefficient, caused by the interaction between solutes in the brine, is determined by a virial expansion of Gibbs excess energy. This expansion is derived using the Pitzer model [45]:

$$\ln(\gamma_i) = \sum_C 2m_C\lambda_{i-C}+\sum_A 2m_A\lambda_{i-A}+\sum_C\sum_A m_Am_C\zeta_{i-A-C} \tag{21}$$

Where i refers to the dissolved gases of CH_4, CO_2, N_2, and H_2S. The calculated activity factors will be used for computing the equilibrium constants Eq. (6). m_C and m_A in Eq. (21) denote cations and anions molality in the AqP. The second (λ) and third-order (ζ) gas-dependent interaction parameters are calculated by:

$$Par(T,P) = c_1+c_2T+\frac{c_3}{T}+c_4P+\frac{c_5}{P}+c_6\frac{P}{T}+c_7\frac{T}{P^2}+\frac{c_8P}{630-T}+c_9Tln(P)+c_{10}\frac{P}{T^2} \tag{22}$$

In Eq. (22), $Par(T,P)$ can be either λ or ζ. Following Duan and Sun [22] and Ziabakhsh-Ganji and Kooi [33], we assumed that ($\lambda_{i-A}=0$, $\lambda_{i-C}=\lambda_{i-Na}$, *and* $\zeta_{i-A-C}=\zeta_{i-Na-Cl}$). For the other constant factors and the procedure of calculating the molality of the ions, refer to Ziabakhsh-Ganji and Kooi [33]. As an example, in a system of sodium (Na), calcium (Ca), potassium (K), and magnesium (Mg) salts, the activity coefficient is given by:

$$\ln(\gamma_i) = 2\lambda_{i-Na}\left(m_{Na}+2m_{Ca}+m_K+2m_{Mg}\right)+\zeta_{i-Na-Cl}m_{Cl}\left(m_{Na}+m_K+m_{Mg}+m_{Ca}\right) \tag{23}$$

- H_2O—CO_2 binary mixture

Since the fugacity of a gas component in a binary gas-brine system does not depend on the composition, its solubility in the AqP can be non-iteratively determined. For a binary mixture of H_2O—CO_2, the mole fraction of water in the NaqP and the CO_2 solubility in AqP can be calculated by the following equations:

$$y_{H_2O} = \frac{\left(1-\frac{P\varnothing_{CO_2}}{K_{HH_2O}\gamma_{CO_2}}\right)}{\left(\left(1\Big/\frac{K^0_{H_2O}}{\varnothing_{H_2O}P}\exp\left[\frac{(P-P_0)V_{H_2O}}{RT}\right]\right)-\frac{P\varnothing_{CO_2}}{K_{HH_2O}\gamma_{CO_2}}\right)} \tag{24}$$

$$x_{CO_2} = \frac{P\varnothing_{CO_2}}{K_{HH_2O}\gamma_{CO_2}}\left(1-\frac{\left(1-\frac{P\varnothing_{CO_2}}{K_{HH_2O}\gamma_{CO_2}}\right)}{\left(\left(1\Big/\frac{K^0_{H_2O}}{\varnothing_{H_2O}P}\exp\left[\frac{(P-P_0)V_{H_2O}}{RT}\right]\right)-\frac{P\varnothing_{CO_2}}{K_{HH_2O}\gamma_{CO_2}}\right)}\right) \tag{25}$$

Eqs. (24) and (25) are provided by combining Eqs. (6) and (10) following the general rule:

$$\sum_{i=CO_2,\ H_2O} x_i = 1, \sum_{i=CO_2,\ H_2O} y_i = 1 \tag{26}$$

- Flash calculation and general iterative algorithm

Nevertheless, vapor-liquid flash calculations are required to compute the gas mixture mutual solubility. In this case, the total pressure, temperature, and mole fraction of each component in the two-phase mixture

(z_i) are received as inputs. Then, the Rachford–Rice equation is implemented to calculate the mole fraction of NaqP (n^v):

$$\sum_{i=1}^{N} \frac{Z_i(K_i - 1)}{1 + (K_i - 1)n^v} = 0 \tag{27}$$

Where K_i is defined as:

$$K_i = \frac{y_i}{x_i},\ i = H_2O, CO_2, CH_4, N_2, H_2S \tag{28}$$

Using the formerly derived equations Eqs. (6) and ((10)), the k-value of water and other components can be alternatively calculated by:

$$K_{H_2O} = \frac{K^0_{H_2O}}{P\varnothing_{H_2O}} \exp\left[\frac{(P - P_0)V_{H_2O}}{RT}\right] \tag{29}$$

$$K_i = \frac{K_{Hi}\gamma_i}{P\varnothing_i},\ i = CO_2, CH_4, N_2, H_2S \tag{30}$$

With the known values of Z_i, K_i, and n^v, it is now possible to compute the composition of each phase:

$$x_i = \frac{Z_i}{1 + (K_i - 1)n^v}, y_i = \frac{Z_i K_i}{1 + (K_i - 1)n^v} \tag{31}$$

Using pressure, temperature, and two-phase composition as inputs makes this F-A algorithm an unattractive option to be included in the numerical simulation of two-phase multicomponent flows. Indeed, most of the conducted studies on modeling two-phase flows in reservoirs solve the partial differential equations of continuity, momentum, and energy to calculate the main variables of pressure, enthalpy, and velocity. Therefore, the fluid temperature cannot be provided as input to the F-A model. As shown in Fig. 1, using enthalpy instead of temperature as an input multiplies the required iteration for calculating solubility. This iterative F-A algorithm is highly precise in predicting the mutual solubilities of gas mixtures in brine. Therefore, it is suitable for producing a large dataset to train a new ML-based equation of state that can perform this calculation without the need for iteration. Gene expression programming is the artificial intelligence technique employed by this study to develop the new EOS. This method is elaborately introduced in the next section.

2.2. Gene expression programming

The GEP method, an extension of genetic programming (GP), was introduced by Ferriera [46] as an evolutionary AI technique. Its purpose is to enhance the performance of the traditional GP approach and address its limitations, which include inadequate exploration of the research space, limited regression strategies, and a slow convergence rate. This method is chosen for the development of the equation of state as it is capable of generating high-preciseness explicit formulas for the non-iterative prediction of target outputs. The new formulation can be easily implemented in other codes and software systems, eliminating the need for iteration in conventional F-A and F-F algorithms.

The operators employed by the GEP method are all inspired by the biological evolution in nature. They range from fundamental genetic operators (e.g., mutation, crossover, selection) to some advanced operators like transposition, insertion, and recombination. Each GEP model uses the three primary components of chromosomes, genes, and expression trees (ET) in the optimization process. Chromosomes are composed of one or several genes. They possess a fixed length and mimic candidate solutions within the code. Genes themselves consist of terminals (tails) which can be either some variables (e.g., pressure, temperature) or functions (heads) such as (+, -, /, ×, tan, log). The expression trees also represent the real candidate expressions. In establishing a general GEP framework, control parameters such as population size, gene length, and mutation rate are defined. Following this, an initial population is created, comprising randomly encoded potential solutions represented as chromosomes. Each individual chromosome is assessed by a fitness function, and the fittest (best) solutions are selected for reproduction in the new population. The genetic operators are subsequently applied to the chosen individuals to generate new offspring. As shown in Fig. 2, the processes of selection, replication, mutation, inversion, transposition, and recombination are re-iterated until a stopping criterion is fulfilled. This study utilized GeneXpro-Tools v5.0, a gene expression algorithm software, to generate precise formulas that correlate input and output parameters. For more details about the implementation process alongside with code examples, refer to Ferriera [46] and Gao *et al.* [47].

3. Results and discussion

Although not favoring fast computation of fluid properties, the iterative algorithm of the F-A model shows high accuracy in the reproduction of experimental data [33]. Herein, we use this thermodynamic scheme to generate a large amount of reliable solubility data, which is required for training the GEP model. In the next step, the GEP functions can entirely replace the time-consuming iterative F-A algorithm or at least decrease the number of iterations for determining each phase composition. This study introduces two sets of GEP functions to calculate fluid temperature and equilibrium constants. The presented GEP equations can predict the fluid temperature as a function of pressure, enthalpy, and two-phase composition. Nevertheless, even using these functions, the algorithm is still iterative as computing equilibrium constant as a function of temperature requires some iterations. Therefore, a new set of straightforward GEP functions are developed to directly calculate the equilibrium constants as functions of pressure,

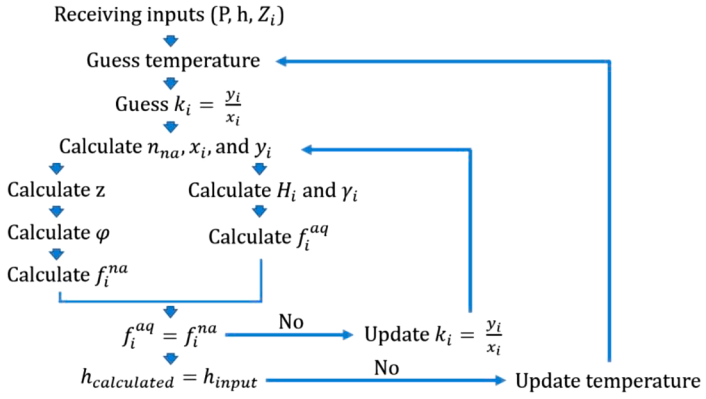


Fig. 1. Flow chart for the conventional iterative calculation of temperature and equilibrium constants as functions of pressure, enthalpy, and two-phase fluid composition.

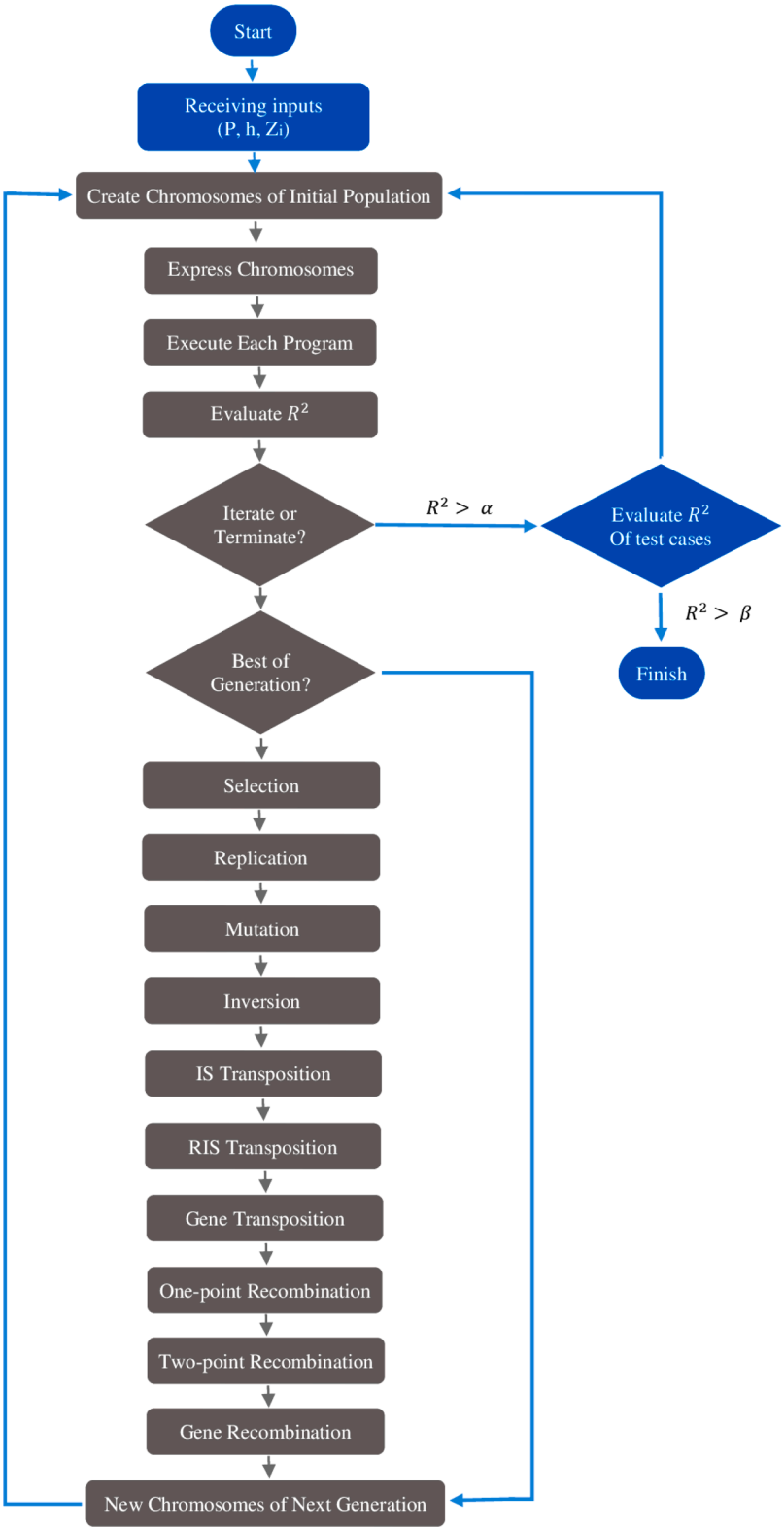


Fig. 2. The flowchart of gene expression algorithm. Pressure, enthalpy, and two-phase fluid composition are assumed inputs, while the R-squared in calculating temperature and equilibrium constants is considered as a fitness function.

temperature, and two-phase composition and bypass the iterations. The following section discusses the precision and computational speed of the new GEP-based EOS.

The predicted temperatures derived from the GEP are compared to those calculated by the iterative F-A algorithm for 1000,000 arbitrary sets of [pressure, enthalpy, two-phase composition]. The same comparison is conducted for the equilibrium constant values, with the difference that the inputs were [pressure, temperature, two-phase composition]. The range of input values is mentioned in Table 1. The accuracy of suggested GEP equations in predicting target values is

Table 1
Range of input parameters for training GEP functions (mf stands for mole fraction in two-phase mixture).

variable	min	max
Pressure (MPa)	1	50
Enthalpy ($MJ{\cdot}Kg^{-1}{\cdot}K^{-1}$)	−13	114
mf (H_2O)	0.5	0.93
mf (CH_4)	0	0.27
mf (CO_2)	0	0.26
mf (N_2)	0	0.32
mf (H_2S)	0	0.28
mf (NaCl)	0	0.027
mf (KCl)	0	0.026
mf ($CaCl_2$)	0	0.025
mf ($MgCl_2$)	0	0.026

elaborately discussed in the next sections.

3.1. Comparing fast GEP-based EOS to iterative method

3.1.1. Temperature

In our approach, 10,000 data sets are used for training the GEP equations with the settings addressed in Table 2. Then, the precision of developed equations in predicting 1000,000 other data points (i.e., validation cases) is meticulously evaluated. The GEP method shows a better performance in predicting fluid temperature when it is normalized by enthalpy. Therefore, the presented GEP function is trained to calculate the value of temperature/enthalpy (displayed as T/H in Fig. 3). Since fluid enthalpy is assumed to be a known input, it is easy to subsequently compute the fluid temperature ($T{=}T/H{\times}H$). The strong nonlinear behavior of T/H as a function of pressure does not allow for developing a single high-preciseness equation for predicting temperature over the whole range of 1MPa $< P <$ 50MPa. For improved accuracy in temperature estimation, we partitioned the pressure range (1MPa to 50MPa) into two distinct sub-domains. To determine the optimal boundary within this range, we conducted a comprehensive evaluation by testing eleven different pressure values: 1, 5, 10, 15, and so forth, up to 50MPa. The analysis revealed that the 10MPa pressure threshold yields the most precise results. Consequently, we recommend employing two separate GEP equations for temperature calculation within the specified pressure intervals: 1MPa $< P <$ 10MPa and 10MPa $< P <$ 50MPa. Fig. 3 shows the R-squared and relative error of these two equations in computing the temperature of validation cases (the 1000,000 data sets). Both equations can predict the fluid temperature with the R-squared of $\approx$ 0.998 and the median relative error of less than 0.6%, which indicates their reliability for accurate computation of fluid temperature.

Even this small error may be unacceptable in some applications. In this case, the proposed GEP equations can be used as an initial guess for temperature in the F-A algorithm shown in Fig. 1. It is guaranteed that, after the first iteration, the error will be almost zero (Fig. 4).

For ease of implementation, the GEP functions are provided as simple C++ codes in (Appendix A: C++ codes)

3.1.2. Equilibrium constants

The new GEP equations for calculating equilibrium constants are developed with the settings mentioned in Table 3. Like other machine learning techniques, the performance of GEP models depends on hyperparameters. Fortunately, GeneXproTools addresses many hyperparameters (e.g., mutation, inversion, IS transformation) within the optimal evolution strategy. Nevertheless, some crucial factors, such as the number of chromosomes, genes, and composing functions, which influence the length and complexity of the formula, must still be selected by the user. In cases of highly nonlinear input-output relationships, equations can become lengthy and intricate. The choice of these parameters typically involves a trial-and-error approach. In the case of equilibrium constants, it has been observed that for chromosome sizes exceeding 30 and gene sizes greater than 8, predictive accuracy does not improve, but the learning process becomes more time-consuming.

The equilibrium constant exhibits extremely nonlinear behavior and covers a broader range of values (0.1 - 10,000) compared to the temperature domain (10 - 100°C), making it a formidable challenge to develop highly precise functions. Nevertheless, the herein-introduced GEP equations can compute the equilibrium constants of the 1000,000 (validation) data set with a median relative error of $\approx$ 6% (Fig. 5 and Table 4). Using these equations as an initial guess for solving the Rachford-Rice equation (Fig. 1) is highly recommended as it results in a median relative error of only $\approx$ 1% after the first iteration. Accurate determination of equilibrium constants ensures consistent adherence to the conservation laws previously stated (Eq. (26)). Indeed, the equilibrium constants and the fractional composition of each component in the gas/liquid phase are interrelated parameters (Eq. (31)). This correlation arises from the Rashford-Rice equation (Eq. (27)), which ensures that the sum of mole fractions in the liquid and gas phases equals 1.

The F-A algorithm typically needs nine iterations for the simultaneous calculation of fluid temperature and equilibrium constants as functions of enthalpy, pressure, and two-phase composition. However, using the developed GEP functions as initial guesses for temperature and equilibrium constant can yield accurate results in only one iteration. This leads to an 89% reduction (8/9×100) in the number of iterations, which now makes GenEOS an exciting option to be included in the fast simulation of two-phase multicomponent flows.

3.2. GenEOS calculation scheme for fluid properties

After determining temperature and phase composition using the non-iterative GEP algorithm, it is also possible to calculate the thermophysical properties of these fluids. This section presents the GEP scheme for calculating fluid properties, with three main objectives: 1) to enhance the transparency of GenEOS as an EOS, 2) to demonstrate its accuracy through validation against international standards, experimental data, and numerical methods, and 3) to show that its capabilities are beyond computing gas solubility in brine.

3.2.1. Enthalpy

- Liquid phase

For the pressure range of (1MPa $< P <$ 50MPa) and temperature domain of (10°C $<$ T $<$ 100°C), the enthalpy of pure liquid water can be calculated by:

$$H^{Aq}_{H_2O}(T,\ P+\Delta P) - H^{SL}_{H_2O}(T,\ P) = V[1-\varepsilon T]\Delta P \tag{32}$$

In which $H^{SL}_{H_2O}$, V, and ε represent the enthalpy of saturated liquid (reference condition), specific volume, and thermal expansion

Table 2
GEP settings for the calculation of Temperature.

Number of chromosomes	30	Head size	10	Number of genes	4
Linking function	/	Fitness function	R^2	Mutation	0.00138
IS Transposition	0.00546	RIS Transposition	0.00546	Inversion	0.00546
One-point Recombination	0.00277	Two-point Recombination	0.00277	Gene Transposition	0.00277
Constants per gene	10	Range of constants	−10 to +10	Data type	Floating point

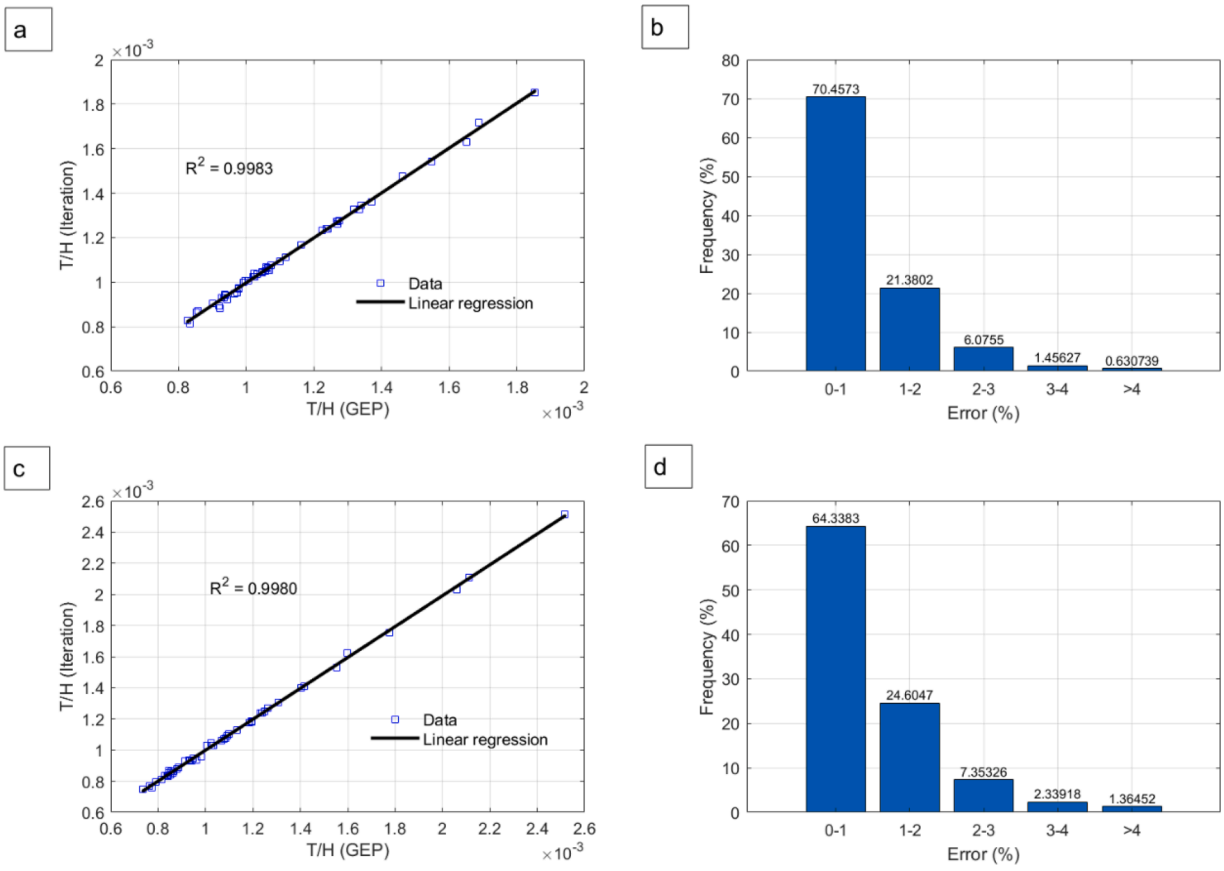


Fig. 3. The R-squared and relative error of the GEP equation in the calculation of fluid temperature when (a-b) pressure > 10MPa and (c-d) pressure < 10MPa.

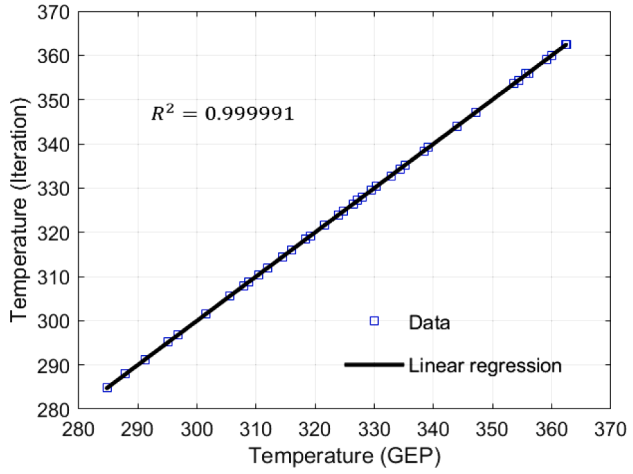


Fig. 4. Accuracy of calculated temperatures after the first iteration when the GEP equation is used as the initial guess in the Fugacity-activity algorithm (Fig. 1).

coefficient, respectively. The procedure for calculating these parameters is discussed in Appendix A.1. The average relative error of Eq. (32) in calculating water enthalpy is below 1% when compared to the IAPWS [48] database.

The enthalpy of Water+NaCl solution is computed through a model developed by Driesner [49].

$$H_{solution}(T, P, X_{NaCl}) = H^{Aq}_{H_2O}(T^*_h, P) \tag{33}$$

This simple model declares that the enthalpy of brine mixture ($H_{solution}$) at a specific pressure, temperature, and salinity equals pure water enthalpy at another temperature (T^*_h), calculated by:

$$T^*_h = q_1 + q_1 T \tag{34}$$

Where q_1 and q_2 are functions of pressure and temperature [49]. Specific enthalpies computed using Eq. (33) typically agree within 1–3% with those obtained from experimental studies. Calculation of enthalpy of a brine mixture containing other salts ($H_{solution}$) and dissolved gases in the AqP ($H^{'}_{solution}$) are explained in Appendix A.2 and Appendix A.3.

- Gas phase

The classic Peng-Robinson EOS is employed for computing gas

Table 3
GEP settings for the calculation of equilibrium constant.

Number of chromosomes	30	Head size	10	Number of genes	8
Linking function	/	Fitness function	R^2	Mutation	0.00138
IS Transposition	0.00546	RIS Transposition	0.00546	Inversion	0.00546
One-point Recombination	0.00277	Two-point Recombination	0.00277	Gene Transposition	0.00277
Constants per gene	10	Range of constants	−10 to +10	Data type	Floating point

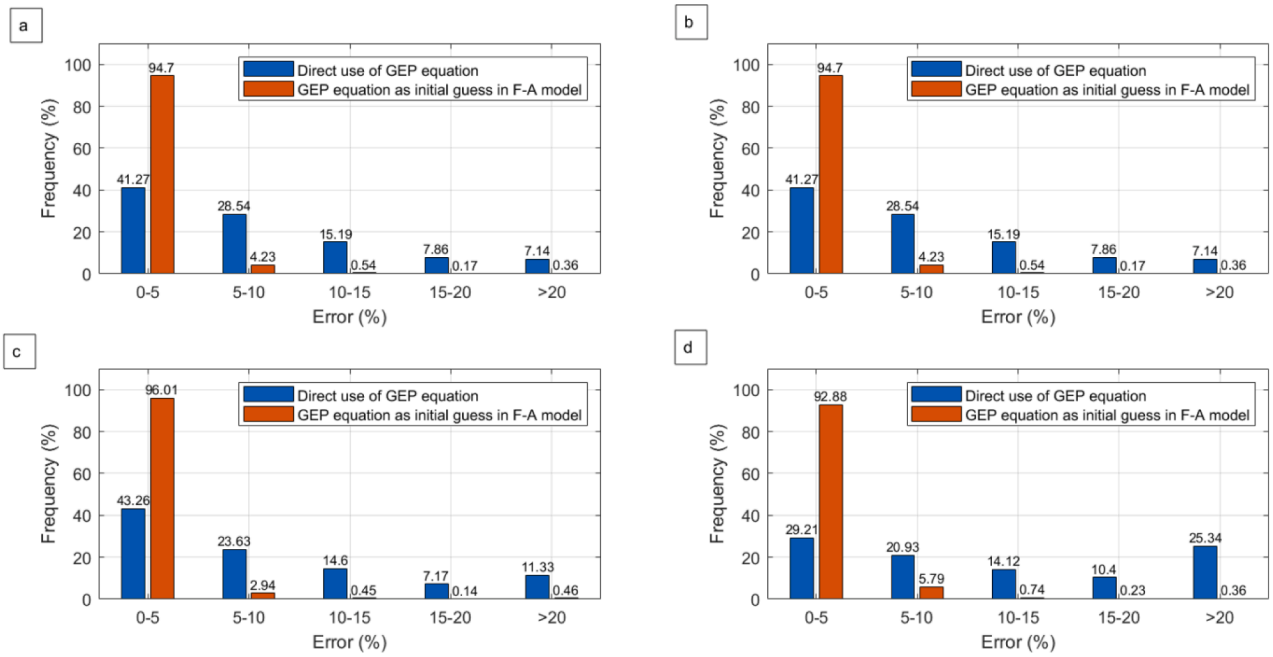


Fig. 5. The relative error in calculating the equilibrium constant of (a) Methane (b) Carbon dioxide (c) Nitrogen (d) Hydrogen sulfide. The orange columns display the outcomes obtained from the F-A model after the first iteration, using GEP equations as the initial guess.

Table 4
The median relative error of GEP equations in computing equilibrium constants.

component	Relative error (%)	
	GEP equation	After first iteration
CH_4	6.2	0.8
CO_2	5.6	1.4
N_2	6.1	0.5
H_2S	10.0	1.1

enthalpy [50]:

$$H^{Naq} - H_0^{Naq} = RT(Z-1) + \left(\frac{a - T\left(\frac{da}{dT}\right)}{2\sqrt{2}b}\right)\ln\left(\frac{Z-0.414B}{Z+2.414B}\right) \tag{35}$$

Where H_0^{Naq} represents ideal gas enthalpy. For a gas mixture, the term of $\left(\frac{da}{dT}\right)$ is calculated by:

$$\frac{da}{dT} = \frac{1}{2}\sum_i\sum_j w_i w_j (1-K_{ij})\sqrt{(a_i a_j)}\left[\frac{1}{a_i}\frac{da_i}{dT} + \frac{1}{a_j}\frac{da_j}{dT}\right] \tag{36}$$

Where

$$\frac{da(T)}{dT} = -ka(T_c)\sqrt{\frac{a(T)}{TT_C}} \tag{37}$$

The PR EOS cannot describe the behavior of *non-boiling* water in NaqP. As reported by Pan *et al.* [51], the small water content of the NaqP behaves like a mixture of "vapor-like" and "liquid-like" components. The gas phase pressure increment plays an essential role in deviating its properties from "vapor-like" to "liquid-like". The proposed method for computing water enthalpy is elaborately discussed in Appendix A.4.

- Two-phase mixture

The mass-average two-phase enthalpy (H) is calculated through [52]:

$$H = w^{Aq} H'_{solution} + w^{Naq} H^{Naq} \tag{38}$$

The calculated enthalpies by GenEOS are validated against experimental data, accurate numerical studies, and international standards like IAPWS [48] and NIST [53]. Some validation cases are addressed in Fig. 6.

3.2.2. Density

- Liquid phase

Al Ghafri *et al.* [54] performed a series of experiments with a vibrating-tube densimeter to measure the density of brine containing various salts (i.e., $MgCl_2$, $CaCl_2$, KI, NaCl, KCl, and $AlCl_3$). Their proposed correlations for calculating density are very accurate for pressures up to 68.5MPa, temperature range of (10°C to 200°C), and salt molality of ($b < 5$). This model is implemented in GenEOS and expresses the brine density as:

$$\rho(T,P,b) = \rho_{ref}(T,b)\left\{1 - C(b)\ln\left[\frac{B(T,b)+P}{B(T,b)+P_{ref}(T)}\right]\right\}^{-1} \tag{39}$$

and the density of a brine mixture containing various salts can be computed by:

$$\rho_{solution}(T,P,b) = \left[\sum_i x_i(1+bM_i)\right] \times \left[\sum_i \frac{x_i(1+bM_i)}{\rho_i(T,P,b)}\right]^{-1} \tag{40}$$

Here, x_i and M_i represent the mole fraction of electrolyte i in the mixed salt and the molar mass of salt i. ρ_i is the density of the single electrolyte solution at the pressure, temperature, and molality of the mixed electrolyte solution. For further details regarding the computation of parameters used in Eqs. (39) and (40), refer to Appendix B.1. This model re-produces experimental data with absolute average relative deviations of 0.03, 0.06, 0.04, 0.02, and 0.02 percent for the MgCl2(aq), $CaCl_2$(aq), KI(aq), NaCl(aq), and KCl(aq) systems, respectively. However, it's worth noting that this error can increase to 0.05 percent for mixed electrolyte solutions. The procedure of computing brine density after gas dissolution ($\rho'_{solution}$) is addressed in Appendix B.2.

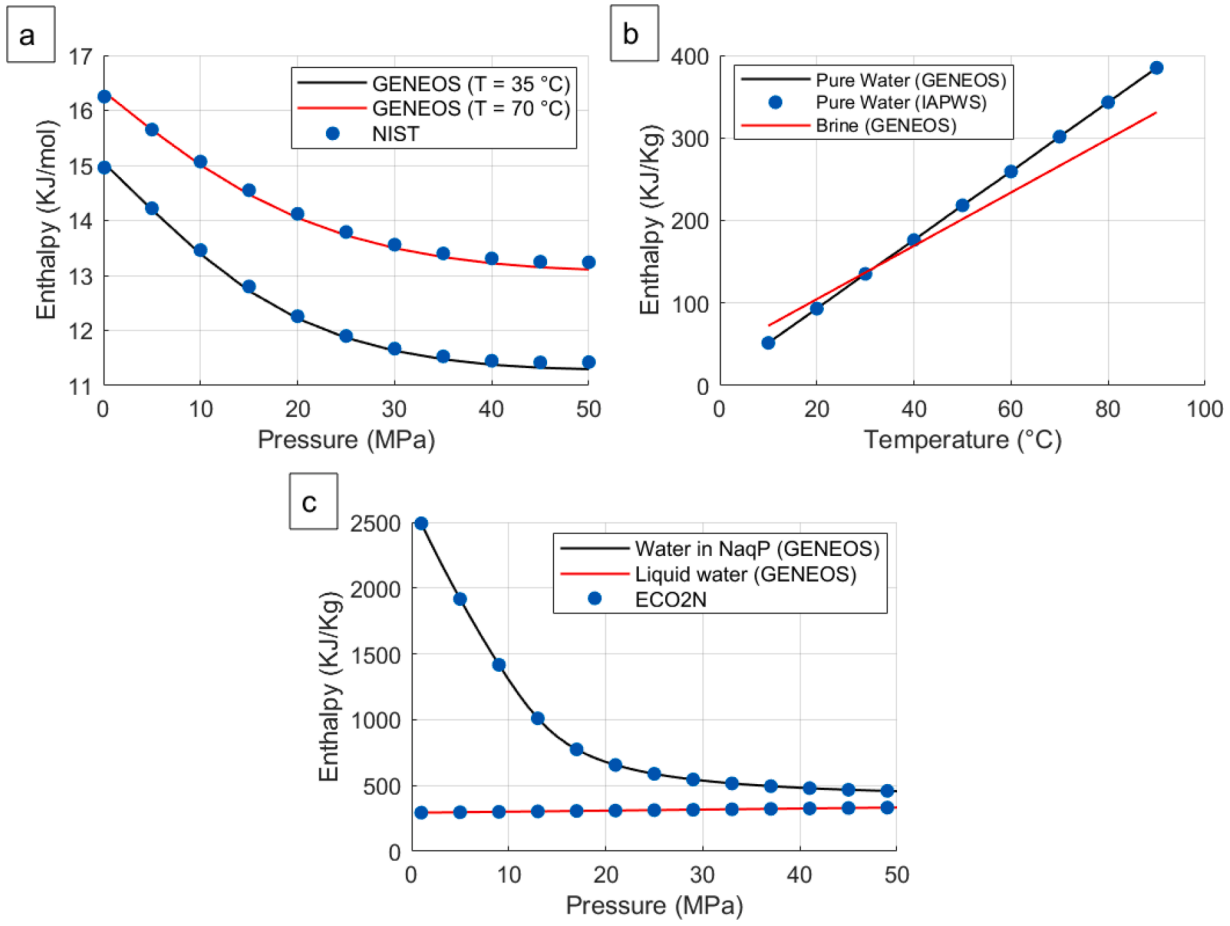


Fig. 6. Calculation and validation of enthalpy of (a) single gas [CH_4, P=10MPa] (b) pure liquid water and brine [P=10MPa, mole fraction of NaCl = 0.1]. The red line is solely included to illustrate the behavior of brine enthalpy. (c) non-boiling water in NaqP (T=70°C).

- Gas phase

The density of a gas mixture is given as:

$$\rho = \frac{MW_g}{\nu_g + cm} \tag{41}$$

Where

$$\nu_g = \frac{Z_g RT}{P} \tag{42}$$

In these equations, Z_g is the compressibility factor computed by the PR EOS, ν_g represents molar volume, and MW_g denotes the molecular weight of the gas phase. *cm*, the gas-dependent volume shift factor, is introduced by Shabani and Vilcáez [55] to increase the accuracy of these equations in calculating gas density. By employing this approach, we successfully re-produced densities of pure gases, as well as gas mixtures, with an error of less than 1%, assuming NIST data as the reference.

As mentioned before, the PR EOS cannot be used to determine non-boiling water density in the NaqP. The corresponding calculation is outlined in Appendix B.3.

- Two-phase mixture

The mass-average two-phase density (ρ) is calculated through:

$$\rho = w^{Aq}\rho^{'}_{solution} + w^{Naq}\rho^{Naq} \tag{43}$$

Some validated/calculated densities by GenEOS are shown in Fig. 7.

3.2.3. Viscosity

- Liquid phase

Laliberte' [56] introduced an experimental correlation for computing brine viscosity:

$$\eta^{Aq} = \eta_w^{w_w}\prod\eta_i^{w_i} \tag{44}$$

In Eq. (44), η_w and η_i are water and salt viscosities, respectively. These properties are defined by:

$$\eta_w \Big/ mPa{\cdot}s = \frac{T/^{\circ}C + 246}{(0.05594T/^{\circ}C + 5.2842)T/^{\circ}C + 137.37} \tag{45}$$

$$\eta_i \Big/ mPa{\cdot}s = \frac{e^{\left(\frac{\nu_1(1-w_w)^{\nu_2}+\nu_3}{\nu_4(T/^{\circ}C)+1}\right)}}{\nu_5(1-w_w)^{\nu_6}+1} \tag{46}$$

Where ν_1-ν_6 are salt-dependent constants. The average relative error of Eq. (44) in predicting an experimental viscosity database with 1700 points is reported to be 2.7% [56].

- Gas phase

To the best of our knowledge, there is no single high-preciseness equation for calculating the viscosities of all the gases included in GenEOS (i.e., CO_2, CH_4, N_2, and H_2S). Therefore, various models are employed for computing gas viscosities. Subsequently, a mixing rule is

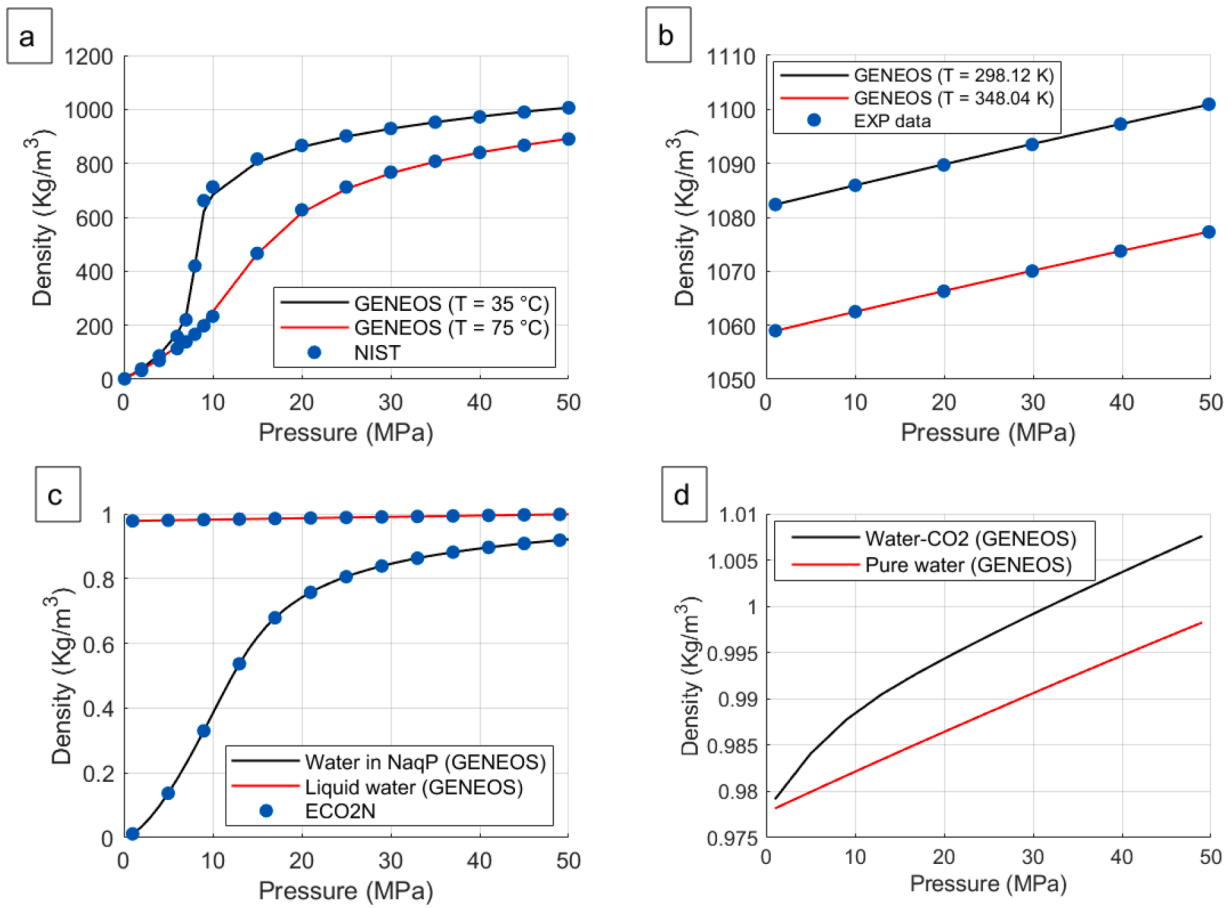


Fig. 7. Calculation and validation of density of (a) single gas [CO_2, P=10MPa] (b) H_2O—$CaCl_2$ solution [b=1 mol·Kg^{-1}], experimental data are taken from Al Ghafri et al. [54] (c) non-boiling water in NaqP [T=70°C] d) CO_2-saturated water [T=70°C].

used to compute gas mixture viscosity.

- Carbon dioxide

Fenghour and Wakeham [57] proposed an empirical equation for computing CO_2 viscosity, which is valid for pressures up to 300MPa and temperatures below 1000K. according to this model, CO_2 viscosity is decomposed into three separate terms:

$$\eta(\rho,T)=\eta_0(T)+\Delta\eta(\rho,T)+\Delta\eta_c(\rho,T) \tag{47}$$

Where η_0, $\Delta\eta$, and $\Delta\eta_c$ stand for viscosity in the zero-density limit, the viscosity increase at elevated density over the dilute gas value, and the viscosity alteration in the immediate vicinity of the critical point. The calculation of these terms is explained in Appendix C.1.

- Methane

The viscosity of methane is calculated using a simple empirical equation suggested by Tadashi *et al.* [58]:

$$\eta=\sum_{i=0}^{4}B_{0i}T^i+P\sum_{i=0}^{2}B_{1i}T^i+P^2\sum_{i=0}^{3}B_{2i}T^i+P^3\sum_{i=0}^{3}B_{3i}T^i+P^4\sum_{i=0}^{2}B_{4i}T^i \tag{48}$$

Eq. (48) re-produces experimental data with a standard deviation of 0.52.

- Nitrogen

Stephan and Krauss [59] developed a model that splits the viscosity of nitrogen into two contributions of zero-density limit (dilute-gas function, η_0) and residual part (excess function, $\Delta\eta_R$):

$$\eta(\rho,T)=\eta_0(T)+\Delta\eta_R(\rho) \tag{49}$$

The process of computing η_0 and $\Delta\eta_R$ is explained in Appendix C.2. Eq. (49) yielded a mean error of 0.165%, a standard deviation of 1.15%, and an absolute mean error of 0.8% in computing nitrogen viscosity.

- Hydrogen sulfide

An empirical correlation proposed by Giri *et al.* [60] is employed by GenEOS for computing H_2S viscosity:

$$\frac{\eta}{\mu Pas}=a_0+a_1\left(\frac{T}{K}\right)\exp\left[\left(a_2+\frac{a_3}{(T/K)}+\frac{a_4}{(T/K)^2}\right)\cdot\left(\frac{\rho(T,P)}{kgm^{-3}}\right)\right] \tag{50}$$

This model is valid for pressures up to 100MPa and temperatures below 483K. Eq. (50) can replicate experimental data with an Average Absolute Deviation (AAD) of around 5 percent.

- Gas mixture

A mixing rule developed by Wilke [61] is used for computing the gas mixture viscosity:

$$\eta^{Naq}=\sum_{i=1}^{n}\frac{\eta_i}{1+\frac{1}{x_i}\sum_{\substack{j=1\\ j\neq i}}^{j=n}x_j\phi_{ij}} \tag{51}$$

where ϕ_{ij} is defined as:

$$\phi_{ij} = \frac{\left[1 + \left(\eta_i/\eta_j\right)^{1/2}\left(M_j/M_i\right)^{1/4}\right]^2}{\left(4/\sqrt{2}\right)\left[1 + \left(M_i/M_j\right)\right]^{1/2}} \tag{52}$$

M in Eq. (52) denotes the molecular weight of each component.

- Two-phase mixture

The mass-average two-phase viscosity (ρ) is written as:

$$\eta = w^{Aq}\eta^{Aq} + w^{Naq}\eta^{Naq} \tag{53}$$

Fig. 8 depicts some validated viscosities by GenEOS.

3.2.4. Thermal conductivity

- Liquid phase

The following equation can be used for calculating the thermal conductivity of multicomponent electrolyte solutions [62]:

$$\lambda^{Aq} = \lambda_0\left(1 + \sum_{i=1}^{k}\beta_i c_i\right) \tag{54}$$

Where β, c, and k denote the gas-dependent constant, the mass content of electrolyte in the solution, and the number of components, respectively. λ_0, the thermal conductivity of pure water, is written as:

$$\lambda_0 = 10^{-3}\left(L_0 + L_1\Psi + L_2\Psi^{1.5} + L_3\Psi^{2.5} + L_4\Psi^3\right) \tag{55}$$

Eq. (55) is valid for the temperature range of ($0°C < T < 135°C$), and Ψ is given by ($\Psi = 0.01T(°C)$).

- Gas phase

The thermal conductivity of each gas is described by a specific model, and subsequently, a mixing rule is implemented for computing the gas mixture's thermal conductivity.

- Carbon dioxide

Amooey [63] developed a model that computes the CO_2 thermal conductivity as a function of density and temperature:

$$\lambda = \frac{A_1 + A_2\rho + A_3\rho^2 + A_4\rho^3T^3 + A_5\rho^4 + A_6T + A_7T^2}{\sqrt{T}} \tag{56}$$

The constant factors of A_1-A_7 are calibrated to cover the thermal conductivity in the temperature range of ($290K < T < 800K$) and densities below 1200 $Kg{\cdot}m^{-3}$. The average relative error in computing methane carbon dioxide thermal conductivity using Eq. (56) is approximately 2.74 percent.

- Methane

Prasad *et al.* [64] proposed a correlation for calculating CH_4 thermal conductivity, which is valid for the temperature range of ($120K < T < 400K$) and pressure domain of ($2MPa < P < 70MPa$). According to this model, the total thermal conductivity is composed of two primary terms:

$$\lambda = \lambda_1 + \Delta\lambda_e \tag{57}$$

In which λ_1, the thermal conductivity at low pressures, is calculated by:

$$\lambda_1 = \sqrt{T_r} \Big/ \sum_{k=0}^{n}\left(a_k / T_r^k\right) \tag{58}$$

And the excess value of thermal conductivity in high-density regions ($\Delta\lambda_e$) is given by:

$$\Delta\lambda_e = \sum_{i=0}^{m}\sum_{j=0}^{n} b_{ij}T_r^i\rho_r^j \tag{59}$$

In Eqs. (58) and (59), T_r and ρ_r are temperature and density, normalized by the critical value of the corresponding property. For the constant factors of a_k and b_{ij} refer to the reference paper. The maximum estimated uncertainties in computing methane thermal conductivity stand at 3 percent.

- Nitrogen

Lemmon and Jacobsen [65] proposed a model for computing the Nitrogen thermal conductivity as a function of temperature and density:

$$\lambda = \lambda_0(T) + \lambda_r(\tau,\delta) + \lambda_c(\tau,\delta) \tag{60}$$

Where $\tau = T_c/T$ and $\delta = \rho/\rho_c$. The procedure for computing (λ_0, λ_r, and λ_c) is explained in Appendix D.1. The uncertainties of calculated values derived from Eq. (60) typically fall within a 2% margin, with the exception of the critical region, where uncertainties are greater.

- Gas mixture

The thermal conductivity of a gas mixture can be computed by [66]:

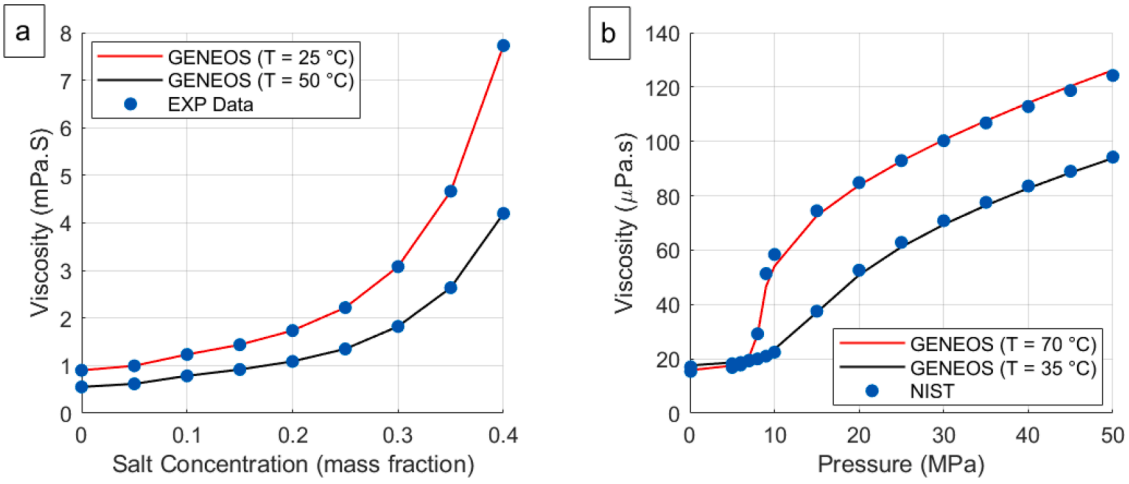


Fig. 8. Calculation and validation of viscosity of (a) binary H_2O—$CaCl_2$ brine [b=1 mol•Kg^{-1}], the experimental data are taken from Laliberte' [56] (b) single gas (CO_2).

$$\lambda^{Naq} = \sum \frac{y_i \lambda_i}{\sum y_i A_{ij}} \tag{61}$$

Where the interaction parameter (A_{ij}) is defined by:

$$A_{ij} = \left(\frac{MW_j}{MW_i}\right)^{0.5} \tag{62}$$

- Two-phase mixture

The mass-average two-phase thermal conductivity (λ) is calculated through:

$$\lambda = w^{Aq}\lambda^{Aq} + w^{Naq}\lambda^{Naq} \tag{63}$$

Some validated/calculated thermal conductivities by GenEOS are exhibited in Fig. 9.

4. Conclusion

Determining the phase composition plays an essential role in the accurate computation of two-phase mixtures properties. Fugacity-Fugacity and Fugacity-Activity are popular thermodynamic models to quantify the impact of gas dissolution on phase composition. However, their iterative algorithms make them computationally expensive. GenEOS is a novel approach to overcome the limitations of conventional Fugacity-Fugacity and Fugacity-Activity models. Employing a distinct AI approach enables the utilization of advanced and precise EOS methods, even for large-scale problems. This approach extends to a wide range of applications, including reservoir-like environments, characterized by high pressure, high temperature, and high salinity conditions. Gene expression programming is used in this study to provide explicit equations for the prediction of target outputs and support this fast calculation of fluid properties. Developing high-preciseness GEP functions requires a lot of accurate data for the training process. Hence, in the first step of this study, a large database is provided by employing a robust Fugacity-Activity method. The generated data points are fed into established GEP models to non-iteratively anticipate the fluid temperature and equilibrium constants as functions of enthalpy, pressure, and two-phase composition. The introduced EOS in this study (GenEOS) uses these novel GEP equations to calculate fluid properties quickly. GenEOS shows benefits in terms of:

- Accuracy: The average relative error of 0.6% in predicting fluid temperature for 1000,000 arbitrary sets of [pressure, enthalpy, two-phase composition] indicates the high accuracy of introduced GEP functions in calculating target outputs. This error reduces to zero after the first iteration when the GEP function is used as an initial guess in the F-A algorithm. Accounting for the highly-nonlinear pressure-dependent thermodynamic behavior of non-boiling water in the NaqP phase and quantifying the impact of gas dissolution on the AqP composition makes GenEOS very accurate in predicting fluid properties.
- Computation speed: The F-A algorithm typically needs three iterations for calculating equilibrium constants as functions of temperature. However, in the numerical modeling of two-phase flows, the conservation equations are solved for computing velocity, pressure, enthalpy, and two-phase composition. Therefore, temperature, as an input for the EOS, should be guessed at the beginning of the algorithm. It can increase the total number of iterations to nine. Using the proposed GEP equations as initial guesses leads to convergence in only one iteration and considerably reduces the EOS-related computational costs.
- Applicability: Focusing on geothermal applications, the nine primary components of water, carbon dioxide, methane, nitrogen, hydrogen sulfide, sodium chloride, potassium chloride, magnesium chloride, and calcium chloride are included in GenEOS. This C++ code can work stand-alone, without coupling to any other chemical solver. Consequently, it can be easily implemented in other modeling platforms while avoiding the complexity of calling multiple linked codes/software. GenEOS is also provided as an object in a multi-physics object-oriented simulation environment called MOOSE [68]. Therefore, all MOOSE-based applications can now benefit from this open-access UserObject. So far, GenEOS is included in a MOOSE-based wellbore simulator called MOSKITO [69–71].
- Transparency: GenEOS is a transparent box for computing fluid properties. All the new GEP equations for computing fluid temperature and equilibrium constants are presented as simple C++ codes in the Appendix. Thus, they can directly and freely be used or converted to any other programming language. Moreover, the implemented equations for calculating other properties and corresponding references are clearly addressed in the context of the paper.

The development of GenEOS opens up new pathways for the accurate and efficient computation of fluid properties in two-phase mixtures, with broad applications in various fields. In the numerical modeling of two-phase fluid flow, GenEOS has a key advantage over traditional EOS

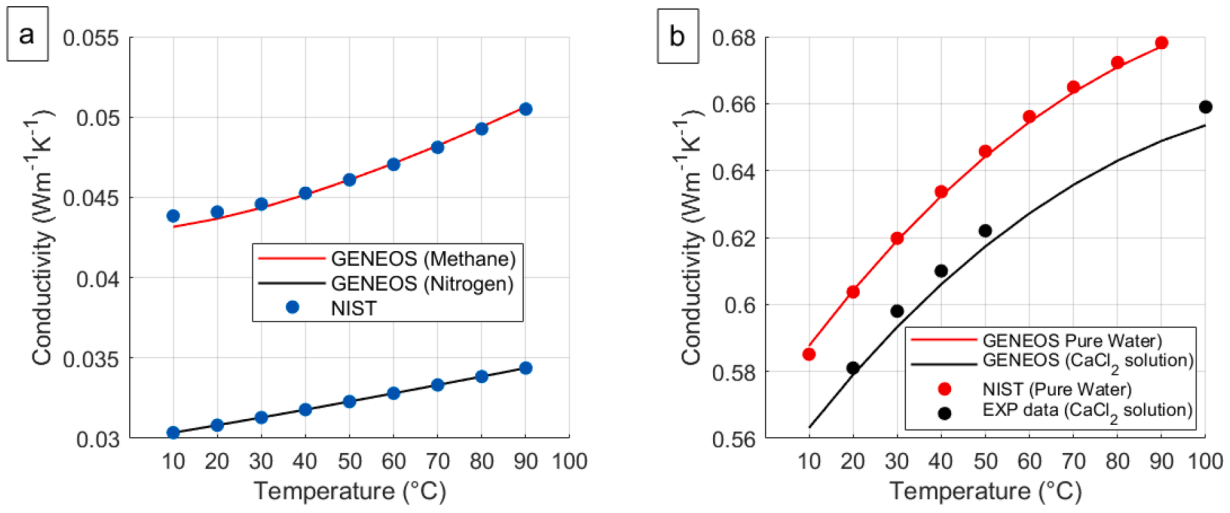


Fig. 9. Calculation and validation of thermal conductivity of (a) a single gas [P=10MPa] (b) pure water and binary H_2O—$CaCl_2$ brine (mass fraction = 20%, P=10MPa), experimental data are taken from Akhmedova-Azizova and Abdulagatov [67].

methods. It achieves this by rapidly calculating fluid properties using novel GEP equations. This capability enables more precise simulations of complex flow phenomena. Furthermore, the application of GenEOS can extend to the forecasting of complex processes in pipes, where the accurate prediction of fluid properties is critical for optimizing pipeline design and operation.

Another promising area of application for GenEOS is in carbon capture and storage, where it can be used to accurately predict the behavior of fluids under varying pressure and temperature conditions, facilitating the design of efficient and cost-effective carbon capture and storage systems. In the field of geothermal reservoirs, GenEOS's ability to accurately calculate fluid properties can aid in the prediction of reservoir behavior, improving reservoir management and maximizing energy production.

Repository information

Repository: https://github.com/morteza374/GenEOS.git.

License

GenEOS is licensed under the terms of the MIT License, which allows for flexibility in how the software is used and distributed.

Usage permissions

Modification: Users are free to modify the GenEOS code according to their requirements and preferences.

Distribution: Users can distribute the modified or unmodified GenEOS code, either privately or publicly.

Commercial Use: GenEOS can be used for both non-commercial and commercial purposes without any restrictions.

We encourage the community to contribute to GenEOS and help improve its functionality and usability.

Declaration of Competing Interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

Acknowledgment

The study is part of the subtopic "Geoenergy" in the program "MTET - Materials and Technologies for the Energy Transition" of the Helmholtz Association.

Appendix A. Elaborate calculation of enthalpy

Appendix A.1 Liquid water

Enthalpy of saturated liquid water can be expressed as:

$$H^{SL}_{H_2O}(T,\ P) = \ d_1 + d_2T + d_3T^2 + d_4T^3 + d_5T^4 + d_6T^5 \tag{A.1}$$

Nevertheless, to calculate the enthalpy of liquid water at any given pressure and temperature (Eq. (32)), it is essential to have the specific volume and thermal expansion coefficient, defined by:

$$V = \ \frac{V^0 - V^0P}{B + A_1P + A_2P^2} \tag{A.2}$$

$$\varepsilon = \ r_1 + r_2T + r_3T^{1.5} + r_4T^2 \tag{A.3}$$

More details about the constant factors of d_1-d_6, r_1-r_4, and the equations for computing V^0, A_1, A_2, and B can be found in Fine and Millero [44] and Popiel and Wojtkowiak [72].

Appendix A.2 Brine mixture

Nusiaputra [52] proposed a formula for the calculation of the enthalpy of a brine mixture containing KCl, $CaCl_2$, and $MgCl_2$:

$$H_{solution}(P,T) = \ w^{Aq}_{H_2O}H^{Aq}_{H_2O} + \sum_{N_i}\frac{w^{Aq}_i}{M_i}h^{\varphi}_i \tag{A.4}$$

Where w, M, and h^{φ}_i stand for mass fraction, molar mass, and apparent molar enthalpy, respectively.

$$h^{\varphi}_i = \ g_{01} + g_{02}b_i + g_{03}(\theta + 273.15) + g_{04}(\theta + 273.15)^2 + \left(b_i^{c_1} + c_2\right)\left(c_3(\theta + 273.15) - c_4\ln\left(1 - \frac{\theta + 273.15}{c_5}\right)\right) \tag{A.5}$$

b in Eq. (A.5) denotes salt molality in the liquid phase. For other constants, refer to Nusiaputra [52].

Appendix A.3 Dissolved gases in aqueous phase

The enthalpy of the dissolved gases in the AqP can be expressed by the summation of gas enthalpy at the corresponding total pressure (h_j) and dissolution enthalpy ($h^{\varphi}_{sol,j}$). Therefore, the AqP enthalpy after gas dissolution ($H^{'}_{solution}$) is governed by [52]:

$$H^{'}_{solution}(P,T) = \ H_{solution}(P,T) + \sum_{N_j} w^{Aq}_j\left(h_j + \frac{h^{\varphi}_{sol,j}}{M_j}\right) \tag{A.6}$$

Where M_j represents gas molar mass. The dissolution enthalpy (i.e., the enthalpy change associated with the dissolution of gas in water at constant

pressure resulting in infinite dilution) can be computed by using the first derivative of the standard chemical potential ($\frac{\partial}{\partial T}(\mu_j^{1(0)}/RT)$). Following Pitzer *et al.* [45], the standard chemical potential can be calculated by:

$$\frac{\mu_j^{1(0)}}{RT} = s_1 + s_2T + s_3 \Big/ T + s_4T^2 + s_5 \Big/ (630 - T) + s_6P + s_7Pln(T) + s_8P \Big/ T + s_9P \Big/ (630 - T) + s_{10}P^2 \Big/ (630 - T)^2 + s_{11}Tln(P) \tag{A.7}$$

The constant parameters of s_1-s_{11} can be found in studies conducted by Duan *et al.* [22,31,32,36].

Appendix A.4 Non-boiling water in non-aqueous phase

Pan *et al.* [51] proposed an equation for computing water enthalpy:

$$h_{H_2O} = (1 - X_L)u_{sv} + X_L u_{sl} + \frac{P_{H_2O}}{\rho_{H_2O}} \tag{A.8}$$

Where

$$X_v = 1 - X_L = \begin{cases} 1 \ if \ P_{H_2O} \le P_{sat} \\ \dfrac{P_{sat}}{P_{H_2O}} \ if \ P_{H_2O} > P_{sat} \end{cases} \tag{A.9}$$

$$P_{H_2O} = y_{H_2O}P \tag{A.10}$$

$$P_{sat} = P_C \exp\left\{ \left[\frac{T_c}{273.15 + \theta} \right] \left(a_1 + a_2 Y^{1.5} + a_3 Y^3 + a_4 Y^{3.5} + a_5 Y^4 + a_6 Y^{7.5} \right) \right\} \tag{A.11}$$

In Eq. (A.8), u_{sv} and u_{sl} denote the specific enthalpies of water vapor and liquid water, respectively. The constant factors of a_1-a_6 for a binary mixture of H_2O—CO_2 are addressed in the reference paper. In this case, the enthalpy of NaqP can be computed by:

$$H^{Naq} = y_{H_2O}h_{H_2O} + \left(1 - y_{H_2O}\right)h_{CO_2} + w_{H_2O}\left(\frac{P}{\rho^{Naq}} - \frac{P_{H_2O}}{\rho_{H_2O}} \right) \tag{A.12}$$

For gas enthalpies falling in the range of 5.1E+5 to 6.1E+5, Equation (A.12) can determine the enthalpy with an R-squared value of 0.9995.

Appendix B. Elaborate calculation of density

Appendix B.1 Brine mixture

The method introduced by Al Ghafri *et al.* [54] calculates brine density as a function of reference density (ρ_{ref}), reference pressure (P_{ref}) and some pressure- molality- dependent factors (C and B). These parameters are expressed as:

$$\rho_{ref}(T, b) - \rho_0(T) = \sum_{i=1}^{i=3} \alpha_{i0} b^{\frac{i+1}{2}} + \sum_{i=1}^{i=3} \sum_{j=1}^{j=3} \alpha_{ij} b^{\frac{i+1}{2}} (T/T_c)^{\frac{j+1}{2}} \tag{B.1}$$

$$C(b) = \gamma_0 + \gamma_1 b + \gamma_2 b^{3/2} \tag{B.2}$$

$$B(T, b) = \sum_{i=0}^{i=1} \sum_{j=0}^{j=3} \beta_{ij} b^i (T/T_c)^j \tag{B.3}$$

$$\ln\left(\frac{P_{ref}(T)}{P_c} \right) = (T_c / T)\left(\sigma_1\varphi + \sigma_2\varphi^{1.5} + \sigma_3\varphi^3 + \sigma_4\varphi^{3.5} + \sigma_5\varphi^4 + \sigma_6\varphi^{7.5} \right) \tag{B.4}$$

In Eq. (B.4), ($T_c = 647.10$K) and ($P_c = 22.064$MPa) stand for critical temperature and pressure of pure water. φ is defined by ($\varphi = 1 - T/T_c$) and ρ_0, the density of saturated liquid water, is given by the auxiliary equation of Wagner and Kretzschmar [48]:

$$\rho_0(T)/\rho_c = 1 + n_1\varphi^{1/3} + n_2\varphi^{2/3} + n_3\varphi^{5/3} + n_4\varphi^{16/3} + n_5\varphi^{43/3} + n_6\varphi^{110/3} \tag{B.5}$$

In which ($\rho_c = 322\ kg{\cdot}m^{-3}$) denotes the critical density of water.

Appendix B.2 Dissolved gases in aqueous phase

Laliberte' [56] and Francke *et al.* [35] introduced a mixing rule that converts the apparent molar volume into AqP density. Using this model, brine density after gas dissolution ($\rho'_{solution}$) can be written as:

$$\rho'_{solution}(T, P, b) = \left(\left(1 - \sum_j w_j^{Aq} \right) \rho_{solution}(T, P, b) + \sum_j \frac{w_j^{Aq}}{M_j} V_j^{\varnothing} \right)^{-1} \tag{B.6}$$

Where the apparent molar volume of dissolved gases is computed by [52]:

$$V_j^{\varnothing} = g_1 f_1 + f_2 \tag{B.7}$$

$$g_1 = \sum_{n=1}^{4} c_{0n} T^{(n-1)} \tag{B.8}$$

$$f_1(T) = \left(1 + \exp\left(\left(T - T_{ps}\right)/\left(c_1 T_{ps}\right)\right)\right)^{-1} \tag{B.9}$$

$$f_2(T) = c_{21}\exp\left(c_{22}.\left|\left(T - T_{ps}\right)/T_{ps}\right|^{c_{23}}\right) \tag{B.10}$$

In Eqs. (B.9) and (B.10), T_{ps} denotes the pseudocritical temperature of 395°C. The gas-dependent constants used in Eqs. (B.6) to (B.10) are addressed in the study conducted by Nusiaputra [52].

Appendix B.3 Non-boiling water in non-aqueous phase

The suggested empirical equation by Pan *et al.* [51] for computing water density is:

$$\rho_{H_2O}(P,T) = \rho_v(P,T) + (1 - X_v)^{1.8}\rho_l(P,T) \tag{B.11}$$

In which vapor density (ρ_v) is given by:

$$\rho_v(P,T) = \begin{cases} \dfrac{P}{P_{H_2O}}\rho_{sv}(P_{H_2O},T) \ if \ P_{H_2O} \le P_{sat} \\ \dfrac{P}{P_{H_2O}}\rho_{sv}(P_{sat},T) \ if \ P_{H_2O} > P_{sat} \end{cases} \tag{B.12}$$

In Eq. (B.12), ρ_{sv} represents the vapor density. For a binary mixture of H_2O—CO_2, the density of NaqP is written as [51]:

$$\rho^{Naq} = y_{H_2O}\rho_{H_2O} + \left(1 - y_{H_2O}\right)\rho_{CO_2} \tag{B.13}$$

For gas densities within the range of 1 Kg·m^{-3} to 800 Kg·m^{-3}, Eq. (B.13) can calculate the density with an R-squared value of 0.9992.

Appendix C. Elaborate calculation of viscosity

Appendix C.1 Carbon dioxide viscosity

The proposed model by Fenghour and Wakeham [57] (Eq. (47)) computes nitrogen viscosity as a function of viscosity in the zero-density limit (η_0), the viscosity increase at elevated density over the dilute gas value ($\Delta\eta$), and the viscosity alteration in the immediate vicinity of the critical point ($\Delta\eta_c$). The term η_0 is given by:

$$\eta_0(T) = \frac{1.00697T^{1/2}}{\varrho_\eta^*(T^*)} \tag{C.1}$$

In which

$$\ln\left(\varrho_\eta^*(T^*)\right) = \sum_{i=0}^{4} a_i(\ln(T^*))^i \tag{C.2}$$

$$T^* = kT/\varepsilon \tag{C.3}$$

The energy scaling parameter (ε/k) in Eq. (C.3) is 251.196K. The term ($\Delta\eta$) is defined as a function of density (ρ):

$$\Delta\eta(\rho,T) = d_{11}\rho + d_{21}\rho^2 + \frac{d_{64}\rho^6}{T^{*3}} + d_{81}\rho^8 + \frac{d_{82}\rho^8}{T^*} \tag{C.4}$$

The ratio of $\Delta\eta_c(\rho,T)/\eta(\rho,T)$ may be greater than 0.01 only within 1% (~5K) of the critical temperature. For more information about the calculation of $\Delta\eta_c$ as well as the constant parameters in Eqs. (47) to (C.4) refer to Vesovic *et al.* [73] and Fenghour and Wakeham [57]. The uncertainties in computing carbon dioxide viscosity range from 0.3% for the viscosity of the dilute gas near room temperature to 5.0% at the highest pressures (≈ 300MPa).

Appendix C.2 Nitrogen viscosity

The model introduced by Stephan and Krauss [59] computes nitrogen viscosity as a function of viscosity at zero-density limit (dilute-gas function, η_0) and a residual part (excess function, $\Delta\eta_R$):

η_0 is expressed as:

$$\eta_0(T) = 5\Big/16[MkT/(\pi N_A)]^{0.5}\Big/\left[\sigma^2\Omega(T^*)\right] \tag{C.5}$$

Here, M, k, and N_A denote molecular weight, Boltzmann's constant, and Avogadro's number, respectively. The constant parameters of π and σ are assumed to be 3.14159 and 0.36502496nm. Furthermore, the function $\Omega(T^*)$ is written as:

$$\ln(\Omega(T^*)) = \sum_{i=0}^{4} A_i(\ln(T^*))^i \tag{C.6}$$

In which T^* is the normalized temperature (Eq. (C.3)) by the energy scaling parameter of ($\varepsilon/k = 100.01654\ K$). The residual part of viscosity ($\Delta\eta_R$) is given by:

$$\frac{\Delta\eta_R(\rho)}{\eta_c} = \frac{C_1}{(\chi - C_2)} + \frac{C_1}{C_2} + \sum_{i=3}^{5} C_i\chi^{i-2} \tag{C.7}$$

Where η_c represents the critical viscosity of nitrogen, and χ stands for normalized density ($\chi = \frac{\rho}{\rho_c = 314.0\ kg\cdot m^{-3}}$). For all the constant parameters used in Eqs. (49) to (C.7), refer to Stephan and Krauss [59].

Appendix D. Elaborate calculation of thermal conductivity

Appendix D.1 Nitrogen thermal conductivity

The proposed model by Lemmon and Jacobsen [65] calculates nitrogen thermal conductivity as a function of dilute gas thermal conductivity (λ_0), the residual part (λ_r), and alteration in the vicinity of the critical point (λ_c). The dilute gas thermal conductivity can be computed by:

$$\lambda_0(T) = N_1\left[\frac{\eta_0(T)}{1\ \mu Pa{\cdot}s}\right] + N_2\tau^{t_2} + N_3\tau^{t_3} \tag{D.1}$$

In which η_0 represents the dilute gas viscosity, described by:

$$\eta_0(T) = \frac{0.0266958\sqrt{MT}}{\sigma^2\Omega(T^*)} \tag{D.2}$$

In Eq. (D.2), σ stand for the Lennard-Jones size parameter (0.3656nm), and Ω is the collision integral given by Eq. (C.6). The energy scaling parameter of nitrogen is ($\varepsilon/k = 98.94\ K$). The residual contribution to the thermal conductivity (λ_r) is expressed as:

$$\lambda_r(\tau,\delta) = \sum_{i=4}^{n} N_i\tau^{t_i}\delta^{d_i}\exp\left(-\gamma_i\delta^{l_i}\right) \tag{D.3}$$

Where γ_i is zero when l_i is zero and one when l_i is not zero. For all the coefficients of N_i, t_i, d_i, l_i, and the calculation procedure of λ_c refer to the reference paper.

References

[1] J.W. Lund, A.N. Toth, Direct utilization of geothermal energy 2020 worldwide review, Geothermics 90 (2021), 101915, https://doi.org/10.1016/j.geothermics.2020.101915.

[2] K. Salhein, C.J. Kobus, M. Zohdy, Forecasting installation capacity for the top 10 countries utilizing geothermal energy by 2030, Thermo 2 (2022) 334–351, https://doi.org/10.3390/thermo2040023.

[3] W.T. Stringfellow, P.F. Dobson, Technology for the recovery of lithium from geothermal brines, Energies 14 (2021) 6805, https://doi.org/10.3390/en14206805.

[4] J. Miao, K. Zhao, F. Guo, L. Xu, Y. Xie, T. Deng, Novel LIS-doped mixed matrix membrane absorbent with high structural stability for sustainable lithium recovery from geothermal water, Desalination 527 (2022), 115570, https://doi.org/10.1016/j.desal.2022.115570.

[5] Y. Zhang, C. Yu, G. Li, X. Guo, G. Wang, Y. Shi, C. Peng, Y. Tan, Performance analysis of a downhole coaxial heat exchanger geothermal system with various working fluids, Appl. Therm. Eng. 163 (2019), 114317, https://doi.org/10.1016/j.applthermaleng.2019.114317.

[6] R.M. Abraham-A, F. Taioli, A.I. Nzekwu, Physical properties of sandstone reservoirs: implication for fluid mobility, Energy Geosci. 3 (2022) 349–359, https://doi.org/10.1016/j.engeos.2022.06.001.

[7] I. Warren. Techno-Economic Analysis of Lithium Extraction from Geothermal Brines, United States, 2021, https://doi.org/10.2172/1782801.

[8] T.M. Ratouis, S.Ó. Snæbjörnsdóttir, M.J. Voigt, B. Sigfússon, G. Gunnarsson, E. S. Aradóttir, V. Hjörleifsdóttir, Carbfix 2: a transport model of long-term CO_2 and H_2S injection into basaltic rocks at Hellisheidi, SW-Iceland, Int. J. Greenh. Gas Control 114 (2022), 103586, https://doi.org/10.1016/j.ijggc.2022.103586.

[9] A. Khaghani, A. Date, A. Akbarzadeh, Sustainable removal of non-condensable gases from geothermal waters, Renew. Sustain. Energy Rev. 21 (2013) 204–214, https://doi.org/10.1016/j.rser.2012.12.001.

[10] S. Chen, J. Liu, Q. Zhang, F. Teng, B.C. McLellan, A critical review on deployment planning and risk analysis of carbon capture, utilization, and storage (CCUS) toward carbon neutrality, Renew. Sustain. Energy Rev. 167 (2022), 112537, https://doi.org/10.1016/j.rser.2022.112537.

[11] N. Wang, K. Akimoto, G.F. Nemet, What went wrong? Learning from three decades of carbon capture, utilization and sequestration (CCUS) pilot and demonstration projects, Energy Policy 158 (2021), 112546, https://doi.org/10.1016/j.enpol.2021.112546.

[12] D. Li, S. Saraji, Z. Jiao, Y. Zhang, CO_2 injection strategies for enhanced oil recovery and geological sequestration in a tight reservoir: an experimental study, Fuel 284 (2021), 119013, https://doi.org/10.1016/j.fuel.2020.119013.

[13] O. Massarweh, A.S. Abushaikha, A review of recent developments in CO_2 mobility control in enhanced oil recovery, Petroleum 8 (2022) 291–317, https://doi.org/10.1016/j.petlm.2021.05.002.

[14] I. Stober, K. Bucher, Geothermal systems in high-enthalpy regions, in: I. Stober, K. Bucher (Eds.). Geothermal Energy, Springer International Publishing, Cham, 2021, pp. 227–256.

[15] S. Finsterle, E.L. Sonnenthal, N. Spycher, Advances in subsurface modeling using the TOUGH suite of simulators, Comput. Geosci. 65 (2014) 2–12, https://doi.org/10.1016/j.cageo.2013.06.009.

[16] K. Pruess, ECO2N: A TOUGH2 Fluid Property Module For Mixtures of Water, NaCl, and CO2, Lawrence Berkeley National Laboratory Berkeley, CA, 2005.

[17] Pan, L.; Spycher, N.; Doughty, C.; Pruess, K., ECO2N V2. 0: a TOUGH2 fluid property module for mixtures of water, NaCl, and CO2, Scientific report LBNL-6930E (2015).

[18] Oldenburg, C.M.; Moridis, G.J.; Spycher, N.; Pruess, K., EOS7C version 1.0: TOUGH2 module for carbon dioxide or nitrogen in natural gas (methane) reservoirs, 2004.

[19] K. Pruess, A. Battistelli, TMVOC, a numerical simulator for three-phase non-isothermal flows of multicomponent hydrocarbon mixtures in saturated-unsaturated heterogeneous media (2002).

[20] A. Battistelli, M. Marcolini, TMGAS: a new TOUGH2 EOS module for the numerical simulation of gas mixtures injection in geological structures, Int. J. Greenh. Gas Control 3 (2009) 481–493, https://doi.org/10.1016/j.ijggc.2008.12.002.

[21] A. Battistelli, C. Calore, K. Pruess, The simulator TOUGH2/EWASG for modelling geothermal reservoirs with brines and non-condensible gas, Geothermics 26 (1997) 437–464, https://doi.org/10.1016/S0375-6505(97)00007-2.

[22] Z. Duan, R. Sun, An improved model calculating CO2 solubility in pure water and aqueous NaCl solutions from 273 to 533K and from 0 to 2000bar, Chem. Geol. 193 (2003) 257–271, https://doi.org/10.1016/S0009-2541(02)00263-2.

[23] Z. Duan, R. Sun, C. Zhu, I.M. Chou, An improved model for the calculation of CO2 solubility in aqueous solutions containing Na+, K+, Ca2+, Mg2+, Cl−, and SO42−, Mar. Chem. 98 (2006) 131–139, https://doi.org/10.1016/j.marchem.2005.09.001.

[24] S.X. Hou, G.C. Maitland, J.M. Trusler, Measurement and modeling of the phase behavior of the (carbon dioxide+water) mixture at temperatures from 298.15K to 448.15K, J. Supercrit. Fluids 73 (2013) 87–96, https://doi.org/10.1016/j.supflu.2012.11.011.

[25] S. Mao, D. Zhang, Y. Li, N. Liu, An improved model for calculating CO2 solubility in aqueous NaCl solutions and the application to CO2–H2O–NaCl fluid inclusions, Chem. Geol. 347 (2013) 43–58, https://doi.org/10.1016/j.chemgeo.2013.03.010.

[26] N. Spycher, K. Pruess, A phase-partitioning model for CO2–brine mixtures at elevated temperatures and pressures: application to CO2-enhanced geothermal systems, Transp Porous Med 82 (2010) 173–196, https://doi.org/10.1007/s11242-009-9425-y.

[27] N. Spycher, K. Pruess, CO2-H2O mixtures in the geological sequestration of CO2. II. Partitioning in chloride brines at 12–100°C and up to 600bar, Geochim. Cosmochim. Acta 69 (2005) 3309–3320, https://doi.org/10.1016/j.gca.2005.01.015.

[28] N. Spycher, K. Pruess, J. Ennis-King, CO2-H2O mixtures in the geological sequestration of CO2. I. Assessment and calculation of mutual solubilities from 12 to 100°C and up to 600bar, Geochim. Cosmochim. Acta 67 (2003) 3015–3031, https://doi.org/10.1016/S0016-7037(03)00273-4.

[29] W. Yan, S. Huang, E.H. Stenby, Measurement and modeling of CO2 solubility in NaCl brine and CO2–saturated NaCl brine density, Int. J. Greenh. Gas Control 5 (2011) 1460–1477, https://doi.org/10.1016/j.ijggc.2011.08.004.

[30] H. Zhao, M.V. Fedkin, R.M. Dilmore, S.N. Lvov, Carbon dioxide solubility in aqueous solutions of sodium chloride at geological conditions: experimental results at 323.15, 373.15, and 423.15K and 150bar and modeling up to 573.15K and 2000bar, Geochim. Cosmochim. Acta 149 (2015) 165–189, https://doi.org/10.1016/j.gca.2014.11.004.

[31] Z. Duan, R. Sun, R. Liu, C. Zhu, Accurate thermodynamic model for the calculation of H2S solubility in pure water and brines, Energy Fuels 21 (2007) 2056–2065, https://doi.org/10.1021/ef070040p.

[32] Z. Duan, S. Mao, A thermodynamic model for calculating methane solubility, density and gas phase composition of methane-bearing aqueous fluids from 273 to 523K and from 1 to 2000bar, Geochim. Cosmochim. Acta 70 (2006) 3369–3386, https://doi.org/10.1016/j.gca.2006.03.018.

[33] Z. Ziabakhsh-Ganji, H. Kooi, An equation of state for thermodynamic equilibrium of gas mixtures and brines to allow simulation of the effects of impurities in subsurface CO2 storage, Int. J. Greenh. Gas Control 11 (2012) S21–S34, https://doi.org/10.1016/j.ijggc.2012.07.025.

[34] C. Appelo, D.L. Parkhurst, V. Post, Equations for calculating hydrogeochemical reactions of minerals and gases such as CO2 at high pressures and temperatures, Geochim. Cosmochim. Acta 125 (2014) 49–67, https://doi.org/10.1016/j.gca.2013.10.003.
[35] H. Francke, M. Kraume, A. Saadat, Thermal–hydraulic measurements and modelling of the brine circuit in a geothermal well, Environ. Earth Sci. 70 (2013) 3481–3495, https://doi.org/10.1007/s12665-013-2612-8.
[36] S. Mao, Z. Duan, A thermodynamic model for calculating nitrogen solubility, gas phase composition and density of the N2–H2O–NaCl system, Fluid Phase Equilib. 248 (2006) 103–114, https://doi.org/10.1016/j.fluid.2006.07.020.
[37] M. Zirrahi, R. Azin, H. Hassanzadeh, M. Moshfeghian, Prediction of water content of sour and acid gases, Fluid Phase Equilib. 299 (2010) 171–179, https://doi.org/10.1016/j.fluid.2010.10.012.
[38] B. Shabani, J. Vilcáez, Prediction of CO_2-CH_4-H_2S-N_2 gas mixtures solubility in brine using a non-iterative fugacity-activity model relevant to CO_2-MEOR, J. Pet. Sci. Eng. 150 (2017) 162–179, https://doi.org/10.1016/j.petrol.2016.12.012.
[39] J. Li, L. Wei, X. Li, An improved cubic model for the mutual solubilities of CO2–CH4–H2S–brine systems to high temperature, pressure and salinity, Appl. Geochem. 54 (2015) 1–12, https://doi.org/10.1016/j.apgeochem.2014.12.015.
[40] J. Li, L. Wei, X. Li, Modeling of CO2-CH4-H2S-brine based on cubic EOS and fugacity-activity approach and their comparisons, Energy Procedia 63 (2014) 3598–3607, https://doi.org/10.1016/j.egypro.2014.11.390.
[41] N.N. Akinfiev, L.W. Diamond, Thermodynamic description of aqueous nonelectrolytes at infinite dilution over a wide range of state parameters, Geochim. Cosmochim. Acta 67 (2003) 613–629, https://doi.org/10.1016/S0016-7037(02)01141-9.
[42] M. Zirrahi, R. Azin, H. Hassanzadeh, M. Moshfeghian, Mutual solubility of CH4, CO2, H2S, and their mixtures in brine under subsurface disposal conditions, Fluid Phase Equilib. 324 (2012) 80–93, https://doi.org/10.1016/j.fluid.2012.03.017.
[43] H. Li, J. Yan, Evaluating cubic equations of state for calculation of vapor–liquid equilibrium of CO2 and CO2-mixtures for CO2 capture and storage processes, Appl. Energy 86 (2009) 826–836, https://doi.org/10.1016/j.apenergy.2008.05.018.
[44] R.A. Fine, F.J. Millero, Compressibility of water as a function of temperature and pressure, J. Chem. Phys. 59 (1973) 5529–5536, https://doi.org/10.1063/1.1679903.
[45] K.S. Pitzer, Thermodynamics of electrolytes. I. Theoretical basis and general equations, J. Phys. Chem. 77 (1973) 268–277, https://doi.org/10.1021/j100621a026.
[46] C. Ferreira, R. Roy, M. Köppen, S. Ovaska, T. Furuhashi, F. Hoffmann, Gene expression programming in problem solving, Eds.. Soft Computing and Industry, Springer London, London, 2002, pp. 635–653.
[47] S. Gao, geppy: a Python framework for gene expression programming, Zenodo (2020). https://github.com/ShuhuaGao/geppy.
[48] W. Wagner, H.J. Kretzschmar, International Steam Tables, Springer Berlin Heidelberg, Berlin, Heidelberg, 2008.
[49] T. Driesner, The system H2O–NaCl. Part II: correlations for molar volume, enthalpy, and isobaric heat capacity from 0 to 1000°C, 1 to 5000bar, and 0 to 1 XNaCl, Geochim. Cosmochim. Acta 71 (2007) 4902–4919, https://doi.org/10.1016/j.gca.2007.05.026.
[50] D.Y. Peng, D.B. Robinson, A new two-constant equation of state, Ind. Eng. Chem. Fund. 15 (1976) 59–64, https://doi.org/10.1021/i160057a011.
[51] L. Pan, N. Spycher, C. Doughty, K. Pruess, ECO_2N V2.0: a TOUGH2 fluid property module for modeling CO_2 -H_2O-NaCl systems to elevated temperatures of up to 300°C, Greenhouse Gas Sci Technol 7 (2017) 313–327, https://doi.org/10.1002/ghg.1617.
[52] Y.Y. Nusiaputra, Coupled hydraulic, thermal and chemical simulations for geothermal installations, Ph.D. thesis,Karlsruhe (2017), https://doi.org/10.5445/IR/1000068708.
[53] P.J. Linstrom, W.G. Mallard, The NIST Chemistry WebBook: a chemical data resource on the internet, J. Chem. Eng. Data 46 (2001) 1059–1063, https://doi.org/10.1021/je000236i.
[54] S. Al Ghafri, G.C. Maitland, J.P.M. Trusler, Densities of aqueous MgCl 2 (aq), CaCl 2 (aq), KI(aq), NaCl(aq), KCl(aq), AlCl 3 (aq), and (0.964 NaCl + 0.136 KCl)(aq) at temperatures between (283 and 472) K, pressures up to 68.5MPa, and molalities up to 6 mol·kg –1, J. Chem. Eng. Data 57 (2012) 1288–1304, https://doi.org/10.1021/je2013704.
[55] B. Shabani, J. Vilcáez, TOUGHREACT-CO2Bio – A new module to simulate geological carbon storage under biotic conditions (Part 1): the multiphase flow of CO_2-CH_4-H_2-H_2S gas mixtures, J. Nat. Gas Sci. Eng. 63 (2019) 85–94, https://doi.org/10.1016/j.jngse.2019.01.013.
[56] M. Laliberté, Model for calculating the viscosity of aqueous solutions, J. Chem. Eng. Data 52 (2007) 321–335, https://doi.org/10.1021/je0604075.
[57] A. Fenghour, W.A. Wakeham, V. Vesovic, The viscosity of carbon dioxide, J. Phys. Chem. Ref. Data 27 (1998) 31–44, https://doi.org/10.1063/1.556013.
[58] M. Tadashi, T. Yoshiyuki, N. Akira, Evaluation and correlation of viscosity data the most probable values of the viscosity of gaseous methane, Rev. Phys. Chem. Jpn. 43 (1973) 54–61.
[59] K. Stephan, R. Krauss, A. Laesecke, Viscosity and thermal conductivity of nitrogen for a wide range of fluid states, J. Phys. Chem. Ref. Data 16 (1987) 993–1023, https://doi.org/10.1063/1.555798.
[60] B.R. Giri, R.A. Marriott, P. Blais, Y. Wu, J.J. Carroll, W. Zhu, H_2S viscosities and densities at high-temperatures and pressures, Eds.. Sour Gas and Related Technologies, Wiley, 2012, pp. 37–47.
[61] C.R. Wilke, A viscosity equation for gas mixtures, J. Chem. Phys. 18 (1950) 517–519, https://doi.org/10.1063/1.1747673.
[62] L.A. Watts, Electrolytes. properties of solutions. methods for calculation of multicomponent systems and experimental data on thermal conductivity and surface tension. by G. G. Aseyev, Begell House, Inc., New York. 1998. 611 pp. $275.50. ISBN 1-56700-106-8, J. Chem. Eng. Data 44 (1999) 1435, https://doi.org/10.1021/je990473p.
[63] A.A. Amooey, A simple correlation to predict thermal conductivity of supercritical carbon dioxide, J. Supercrit. Fluids 86 (2014) 1–3, https://doi.org/10.1016/j.supflu.2013.11.016.
[64] R.C. Prasad, J.E.S. Venart, N. Mani, T. Ashworth, D.R. Smith, Thermal conductivity of methane and ethane, Eds.. Thermal Conductivity 18, Springer US, Boston, MA, 1985, pp. 81–91.
[65] E.W. Lemmon, R.T. Jacobsen, Viscosity and thermal conductivity equations for nitrogen, oxygen, argon, and air, Int. J. Thermophys. 25 (2004) 21–69, https://doi.org/10.1023/B:IJOT.0000022327.04529.f3.
[66] B.E. Poling, J.M. Prausnitz, J.P. O'Connell, The Properties of Gases and Liquids, 5thth ed., McGraw-Hill, New York, 2007.
[67] L.A. Akhmedova-Azizova, I.M. Abdulagatov, Thermal conductivity of aqueous $CaCl_2$ solutions at high temperatures and high pressures, J. Solut. Chem. 43 (2014) 421–444, https://doi.org/10.1007/s10953-014-0141-z.
[68] C.J. Permann, D.R. Gaston, D. Andrš, R.W. Carlsen, F. Kong, A.D. Lindsay, J. M. Miller, J.W. Peterson, A.E. Slaughter, R.H. Stogner, R.C. Martineau, MOOSE: enabling massively parallel multiphysics simulation, SoftwareX 11 (2020), 100430, https://doi.org/10.1016/j.softx.2020.100430.
[69] Esmaeilpour, M.; Gholami Korzani, M.; Kohl, T., Evaluation of the advantages of multilateral closed deep geothermal frameworks over conventional geothermal systems: 47th workshop on Geothermal Reservoir Engineering Stanford University, Stanford, California, 2022.
[70] Esmaeilpour, M.; Gholami Korzani, M.; Kohl, T., Performance analyses of deep closed-loop u-shaped heat exchanger system with a long horizontal extension: 46th Workshop on Geothermal Reservoir Engineering Stanford University, Stanford, California 2021.
[71] M. Esmaeilpour, M. Gholami Korzani, T. Kohl, Impact of thermosiphoning on long-term behavior of closed-loop deep geothermal systems for sustainable energy exploitation, Renew. Energy 194 (2022) 1247–1260, https://doi.org/10.1016/j.renene.2022.06.014.
[72] C.O. POPIEL, J. WOJTKOWIAK, Simple formulas for thermophysical properties of liquid water for heat transfer calculations (from 0°C to 150°C), Heat Transf. Eng. 19 (1998) 87–101, https://doi.org/10.1080/01457639808939929.
[73] V. Vesovic, W.A. Wakeham, G.A. Olchowy, J.V. Sengers, J.T.R. Watson, J. Millat, The transport properties of carbon dioxide, J. Phys. Chem. Ref. Data 19 (1990) 763–808, https://doi.org/10.1063/1.555875.