Fluid Phase Equilibria 599 (2026) 114528

Contents lists available at ScienceDirect

Fluid Phase Equilibria

journal homepage: www.sciencedirect.com/journal/fluid-phase-equilibria

Solving the UVN-flash problem in TVN-space

Pardeep Kumar [a,b,*], Patricio I. Rosen Esquivel [c]

[a] Delft University of Technology, Delft, The Netherlands
[b] Centrum Wiskunde & Informatica, Amsterdam, The Netherlands
[c] Shell Projects and Technology, Amsterdam, The Netherlands

ARTICLE INFO

Keywords:
UV-flash
UVN reformulation
Flash
Entropy maximization
Stability analysis
Phase equilibrium calculations

ABSTRACT

In this paper, we investigate the phase equilibrium problem for multicomponent mixtures under specified internal energy (U), volume (V), and mole numbers ($N_1, N_2, \ldots, N_n$), commonly known as the UVN-flash problem. While conventional phase equilibrium calculations typically use pressure–temperature-mole number (PTN) specifications, the UVN formulation is essential for dynamic simulations of closed systems and energy balance computations. Existing approaches, including those based on iterative pressure–temperature updates and direct entropy maximization, can suffer from computational inefficiencies due to inner Newton iterations needed to solve for temperature T at specified internal energy U and volume V.

In this work, we present a reformulation of the UVN-flash problem that eliminates the need for the inner Newton iterations, addressing a computational bottleneck. We begin with stability analysis and discuss a strategy to generate the initial guess for the UVN-flash from the stability analysis results. We then reformulate the UVN-flash problem in TVN-space as constrained entropy maximization. We provide a detailed derivation of Michelsen's Q-function using the method of Lagrange multipliers, illustrating its direct application in solving the UVN-flash problem. Furthermore, we discuss the numerical methods used, including gradient and Hessian computations. The reformulation is validated against benchmark cases, demonstrating improved efficiency.

1. Introduction

We investigate the phase equilibrium calculations for multicomponent mixtures under specified internal energy (U), volume (V), and mole numbers ($N_1, N_2, \ldots, N_n$), commonly referred to in the literature as the UVN-flash problem. Compared to the more conventional PTN-flash (e.g., [1–3]) (where pressure, temperature, and mole numbers are specified), the UVN specification is less commonly addressed. However, it plays a crucial role in various thermodynamic applications where energy and volume are specified, such as in the dynamic simulation of closed systems and energy balance calculations in process design. Notably, the UVN-flash formulation proves to be particularly valuable in non-isothermal problems, such as those encountered in the dynamic simulation of tanks and CO_2 injection in geological storage. Key contributions in this area include [4–8].

Additionally, the UVN-flash problem has been explored in the work of several researchers, often alongside other non-isothermal flash formulations such as PHN, PSN, or TVN flashes. Early foundational work by Brantferger et al. [9] introduced the unconstrained minimization approach for PHN-flash calculations, later extended by Castier [10], Smejkal et al. [11] and Paterson et al. [12]. Further developments in RAND-based formulations for non-isothermal flashes, including UVN-flash, were presented by Paterson et al. [13–15]. Furthermore, Lipovac et al. [16] recently proposed a unified framework for PHN and UVN flashes, solving phase stability and flash problems simultaneously. Fathi et al. [17] investigated volume-based flash methods applicable to UVN scenarios. These studies have collectively advanced the understanding of multiphase equilibrium under non-isothermal constraints, though UVN-flash remains less explored compared to its PHN counterpart.

Michelsen [18] proposed a general framework to address flash problems under various specifications, including UVN. His approach utilizes the PTN-flash in an inner loop while iteratively updating pressure and temperature in an outer loop. This strategy combines both a nested-loop method using the Q-function and a direct Newton iteration approach, with the latter being employed whenever possible for efficiency. The advantage of this method is that it leverages existing PTN-flash solvers. However, the nested iterations inherent in the Q-function method can become computationally expensive. Notably, within the same seminal work, Michelsen also tabulated Q-functions for UVN flash based on temperature and volume (TV-based) variables (see Table 3 in [18]), though their explicit derivations were not provided.

* Corresponding author.
E-mail address: pardeep@cwi.nl (P. Kumar).

https://doi.org/10.1016/j.fluid.2025.114528
Received 26 February 2025; Received in revised form 24 June 2025; Accepted 7 July 2025
Available online 19 July 2025

One of the earliest works addressing the UVN-flash problem is by Saha et al. [19]. In this paper, the authors developed heuristics to estimate pressure and temperature corresponding to specified UVN conditions. Their approach combined successive substitution (fixed-point iteration) for updating equilibrium K-values with Newton's method for pressure and temperature updates. However, they often encountered convergence to trivial solutions, limiting the robustness of their method.

Bi et al. [20] reformulated the UVN-flash problem using the Rachford-Rice equation while ensuring pressure equilibrium and enforcing internal energy and volume constraints. Their approach employed fixed-point iteration with soft tolerance, followed by Newton's method for refinement.

A significant contribution on UVN flash is found in the work of Castier [10], who proposed direct entropy maximization as an alternative approach. In his method, the algorithm adaptively adds or removes phases as needed during the computation. However, obtaining a good initial phase split requires a reasonable estimate of pressure and temperature, which has to be determined using heuristics. In cases of numerical difficulties, Castier's method switches to a PTN-flash solver for the inner loop while adjusting pressure and temperature in the outer loop, ensuring that U and V approach their specified values. Once sufficient estimates for P and T are found, the algorithm returns to direct entropy maximization.

Another important contribution is by Smejkal et al. [11], who also applied direct entropy maximization for both stability and flash calculations. They used the stability analysis results as initial guesses for the flash calculations, demonstrating the utility of entropy-based methods in UVN-flash scenarios. This approach requires an inner Newton iteration to determine temperature by solving $U(T, V, \mathbf{N}) = U$ for given $U, V, \mathbf{N}$ which poses an additional computational burden within the overall optimization, particularly in challenging scenarios or with poor initial estimates, as noted by Castier [10]. While Newton's method is typically efficient and converges in a few iterations with a good initial guess, even a couple of additional iterations per flash computation can become costly in large-scale applications such as reservoir simulations, where millions of such evaluations are required; avoiding inner Newton iterations can offer significant computational advantages.

Building upon the theoretical groundwork laid by Michelsen, Medeiros et al. [21] significantly extended the Q-function methodology to address open thermodynamic systems, encompassing both reactive and non-reactive chemical species. Their approach leveraged Legendre transformations to derive generalized formulations applicable under a variety of imposed thermodynamic specifications. In addition, they provided a derivation for the Q-functions previously introduced by Michelsen [18]. Their derivation mandates the use of two Lagrange multipliers, and they provided the expressions for these Lagrange multipliers.

In this work, we revisit the UVN-flash problem by reformulating it as a constrained entropy maximization problem in the TVN-space. We then cast it as an unconstrained saddle-point problem using a single Lagrange multiplier. Furthermore, we provide an explicit derivation of the Lagrange multiplier without using the Legendre transforms. This reformulation of the UVN-flash problem within the TVN-space eliminates the need for nested Newton iterations to compute the temperature T consistent with a given internal energy U, volume V and mole numbers $\mathbf{N}$. It thus removes a computational bottleneck inherent in both Michelsen's Q-function approach [18] and the method of Castier's unconstrained minimization [10]. This advantage becomes especially significant when using complex equations of state.

The structure of the paper is as follows. We begin with a precursor to stability analysis in Section 3, where we present the relevant formulations and the generation of initial guesses for both stability and flash calculations. This is followed by a recap of the UVN flash formulation in natural variables in Section 4. Next, in Section 5, we reformulate the UVN-flash problem in TVN-space by using the method of Lagrange multipliers. In Section 6, we provide a detailed derivation of Michelsen's Q-function, illustrating its direct application in solving the UVN-flash problem. We then discuss the numerical approach, including the necessary gradient and Hessian computations (derived in Appendix A). Finally, we present the results in Section 7 and conclude with key findings and implications in Section 8.

2. Preliminaries

For the sake of clarity, in this section, we define the following concepts in the context of UVN-flash.

2.1. Trial phase

The trial phase is an incipient phase introduced to assess the thermodynamic stability of a system. It involves perturbing the composition of the system and evaluating whether the introduction of this new phase leads to an increase in entropy (for UVN flash calculations). If the entropy increases, the system is unstable as a single phase, and phase separation is favorable.

2.2. Reference phase

The reference phase ($\star$) represents a hypothetical single-phase system characterized by the total internal energy $U^\star$, volume $V^\star$ and total mole numbers $\mathbf{N}^\star = (N_1^\star, \ldots, N_n^\star)$.

2.3. Stability analysis

Stability analysis is typically performed prior to flash calculations, as it determines the stability of a multicomponent mixture across p phases, where p represents the number of phases in the system. The primary objective of this analysis is to establish whether the mixture will remain in p phases or separate into $p + 1$ phases. In most cases, we focus on systems with at most two phases, typically a vapor–liquid mixture. In such cases, stability analysis determines whether the mixture can remain as a single phase or will separate into a vapor–liquid equilibrium.

A crucial aspect of stability analysis is its role in providing an initial guess for subsequent flash calculations. If instability is detected, the analysis often yields valuable information about the incipient phase, such as its temperature, concentration, and internal energy density, which can significantly aid in the convergence of the flash calculation algorithm.

2.4. Flash calculation

When a stability test indicates that a mixture is thermodynamically unstable, a flash calculation is performed to determine the phase equilibrium of the multicomponent mixture under specified conditions, such as pressure and temperature, internal energy and volume, or entropy and volume. The flash calculation calculates the amounts and compositions of each phase, assuming the system reaches equilibrium.

In this study, the UVN flash problem is addressed using an equation of state derived from the Helmholtz energy function. Specifically, we will use the Peng–Robinson [22] Equation of State (EOS) for all our results. This EOS enables the computation of thermodynamic properties for given T, V, and $\mathbf{N}$. For specified values of internal energy U, volume V, and mole numbers $\mathbf{N}$, the phase split calculations proceed as follows. The details of the algorithm can be found in Castier [10] and Smejkal et al. [11].

1. **Initial Stability Test:** Assess the stability of the single-phase mixture.

2. **Iterative Phase Adjustment and Equilibrium:**

   (a) **If unstable:** Introduce a new phase, using stability analysis results for an initial guess.
   (b) **Equilibrium Calculation:** Determine phase equilibrium (temperature, volumes, compositions) for the current number of phases.
   (c) **Stability Re-evaluation and Phase Count Update:** Check the stability of the resulting phase split by testing the stability of one arbitrarily selected phase from the equilibrium.

      - **Phase Addition:** If the selected phase is unstable, add another phase.
      - **Delayed Phase Merging:** After at least three iterations of equilibrium calculations for the current number of phases, attempt to merge a phase whose mole fraction is below a specified threshold (e.g., 10^{-6}) and distribute its total energy, volume and moles uniformally among remaining phases. If the total entropy increases[1] upon merging, remove the candidate phase.

   (d) **Repeat:** Continue equilibrium calculation and stability re-evaluation until a stable phase split is achieved.

3. **Termination:** Output the stable equilibrium state (temperature, phase volumes, and compositions).

Accordingly, we begin with stability analysis in the next section, followed by the flash procedure in the subsequent section.

3. Stability analysis precursor

Stability analysis is a fundamental step in assessing the thermodynamic stability of a mixture and determining whether phase separation occurs. A detailed discussion and performance comparison of various phase stability analysis methods are beyond the scope of this paper; interested readers are referred to dedicated literature on this topic (e.g., Nichita [23], Smejkal et al. [11], Michelsen and Mollerup [3]). In this section, we briefly present the formulation of the UVN stability problem, discuss initialization strategies, and outline an algorithm to generate initial guesses for flash calculations based on stability analysis results.

The UVN stability problem can be reduced to the TVN stability problem as follows. For given $U^\star, V^\star$ and $\mathbf{N}^\star$, we can solve

$$U(T, V^\star, \mathbf{N}^\star) = U^\star, \tag{1}$$

for T, as discussed by Mikyska [24] and Nichita [25]. Therefore, we provide the formulation of TVN stability analysis in the following subsection.

3.1. TVN stability formulation

In this section, we briefly discuss the formulation of TVN stability as UVN stability can be reduced to TVN stability [23]. The TPD (Tangent Plane Distance) function for volume-based stability analysis was originally introduced by Nagarajan et al. [26] for PT conditions and later extended to TVN conditions by Nichita et al. [27,28]. The TPD function, denoted by D at temperature T, is given by:

$$D(T, \boldsymbol{c}') = -\left(P' - P^\star\right) + \sum_{i=1}^{n} \left(\mu_i' - \mu_i^\star\right) c_i', \tag{2}$$

where $\mu_i' = \mu_i(T, 1, \boldsymbol{c}')$ and $\mu_i^\star = \mu_i(T, 1, \boldsymbol{c})$ are the chemical potentials of component i in the trial and reference phases, respectively. Similarly, $P' = P(T, 1, \boldsymbol{c}')$ and $P^\star = P(T, 1, \boldsymbol{c})$ denote the pressures in the two phases. The molar concentration of component i in the *trial phase* is given by $c_i' = \frac{N_i'}{V'}$, where N_i' is the mole number of component i and V' is the molar volume of the trial phase. Similarly, the molar concentration of component i in the *reference phase* is given by $c_i = \frac{N_i}{V}$, where N_i is the mole number of component i and V is the molar volume of the reference phase. The concentration vectors are denoted by

$\boldsymbol{c} = \{c_1, \ldots, c_n\}$ for the reference phase, and
$\boldsymbol{c}' = \{c_1', \ldots, c_n'\}$ for the trial phase.

A non-negative value of D indicates that the reference phase is stable. The existence of a state with $D < 0$ can be detected by examining the stationary points of D. The stationary points of D are given by

$$\frac{\partial D}{\partial c_i'} = 0, \quad \forall i \in \{1, \ldots, n\}. \tag{3}$$

Mikyska et al. [24] reformulated the stability condition (Eq. (3)) using the volume function Φ_i, related to the fugacity coefficient ϕ_i via

$$\Phi_i = \frac{1}{Z\phi_i}, \qquad Z = \frac{PV}{nRT}. \tag{4}$$

Their stability equation in terms of Φ_i is given by:

$$\ln \frac{c_i'}{c_i} + \ln \Phi_i(\boldsymbol{c}) - \ln \Phi_i(\boldsymbol{c}') = 0, \quad \forall i = 1, \ldots, n. \tag{5}$$

This system (5) can be solved using the Newton–Raphson method. At each iteration step k, the update direction $\Delta \boldsymbol{c}'^{,k}$ is obtained by solving:

$$\mathbf{J}(\boldsymbol{c}'^{,k}) \Delta \boldsymbol{c}'^{,k} = -\mathbf{F}(\boldsymbol{c}'^{,k}), \tag{6}$$

where the residual vector $\mathbf{F}$ has components:

$$F_i(\boldsymbol{c}') = \ln \frac{c_i'}{c_i} + \ln \Phi_i(\boldsymbol{c}) - \ln \Phi_i(\boldsymbol{c}'), \quad \forall i \in \{1, \ldots, n\}, \tag{7}$$

and the Jacobian matrix $\mathbf{J}$ is given by:

$$J_{ij}(\boldsymbol{c}') = \frac{\delta_{ij}}{c_j'} - \frac{\partial \ln \Phi_i(\boldsymbol{c}')}{\partial c_j'}, \tag{8}$$

where δ_{ij} is the Kronecker delta. The update step is then performed as:

$$\boldsymbol{c}'^{k+1} = \boldsymbol{c}'^{,k} + \lambda^k \Delta \boldsymbol{c}'^{,k}, \tag{9}$$

where $\lambda^k \in (0, 1]$ is a step size, possibly determined by line search. For further details on this, we refer the reader to [24]. The pseudocode for stability analysis is provided in the appendix as Algorithm 2. The convergence of the stability test strongly depends on the choice of an appropriate initial guess for $\boldsymbol{c}'$. Effective initialization is crucial for numerical stability and robustness in stability analysis.

3.2. Initialization for stability analysis

In this section, we discuss the initialization strategy for TVN stability analysis. We adopt the simplex-based initialization method proposed by Smejkal et al. [11], which leverages the geometric properties of the feasibility domain of admissible molar concentrations. In this approach, the feasible domain is represented as an n-simplex, where n denotes the number of components in the mixture. Initial guesses are generated by computing the barycenter of the simplex and the midpoints between the barycenter and each of the $n + 1$ vertices. This procedure yields

[1] For UVN flash problems, the total entropy should increase. For other specifications, the corresponding thermodynamic potential should evolve appropriately from one iteration to the next: for example, Helmholtz free energy in TVN, Gibbs free energy in TPN, and enthalpy in PSN should decrease, whereas entropy in PHN should increase.

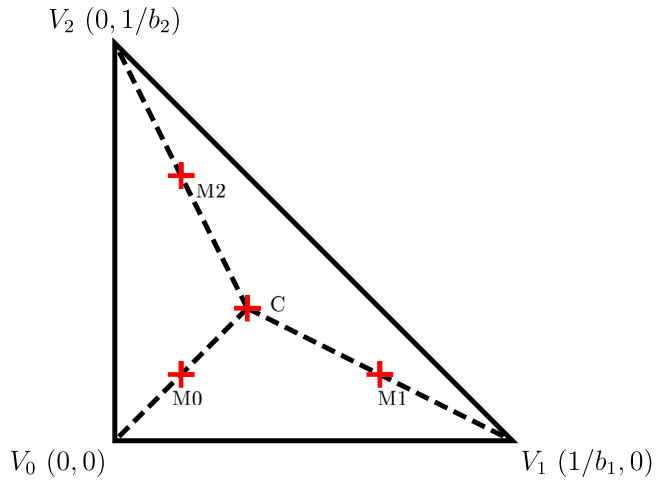

Fig. 1. Depiction of the initial guesses for $\mathbf{c}'$ in a binary mixture where the two components have molar volumes b_1 and b_2, respectively.
Source: Adapted from [29].

$n+2$ initial estimates, ensuring a well-distributed set of starting points for the stability analysis.

The admissible molar concentrations c_i' must satisfy the following conditions:

$$\sum_{i=1}^{n} c_i' b_i < 1, \quad c_i' \geq 0, \quad b_i > 0, \quad \forall i \in \{1, \ldots, n\}, \tag{10}$$

where b_i denotes the co-volume of the component i from the Peng–Robinson EOS. Fig. 1 (adapted from [29]) illustrates the initial concentration guesses for a binary mixture. The four points marked with circles: the barycenter C and the midpoints M_0, M_1, and M_2, are used sequentially as initial guess to perform the stability analysis. The results of the stability analysis are then used to generate the initial guess for phase split calculations. These results, however, are in the form of concentration and temperature of the trial phase. A procedure is needed to convert the stability analysis results into the initial guess for phase split calculations, which is addressed in the following subsection.

3.3. Initial guess for flash from stability analysis

The results of the stability analysis provide the initial guess required for phase split calculations. A good initial guess is crucial for ensuring convergence in the numerical optimization procedures used in phase split calculations, as discussed in Section 6. However, the results from stability analysis are not immediately suitable as initial guesses for flash calculations. Stability analysis provides the concentrations of the incipient phase along with the specific internal energy; but an additional parameter, the volume of the trial phase, is needed to initiate phase split calculations.

We begin by assuming that the trial phase occupies half of the total system volume. The mole numbers of each phase are determined by multiplying the phase volume with the species concentrations obtained from the stability test. The internal energy of each phase is then computed using the internal energy density and phase volume. Next, the phase temperature is determined by solving Eq. (1) and finding a temperature consistent with the given internal energy, volume, and mole numbers. This provides a complete initial estimate.

This initial estimate is then iteratively refined to find a two-phase split with a higher total entropy than the reference single phase, while satisfying the feasibility conditions (10) at each step. At each step, the total entropy of the two-phase system is evaluated. If the entropy increases and all feasibility conditions are met, the solution is accepted. If these criteria are not satisfied, the trial phase volume is further halved, and the internal energy and mole numbers are adjusted accordingly. This iterative process continues until a feasible phase split is achieved or until the predefined iteration limit is reached. This pseudocode is outlined in Algorithm 1. Alternatively, the procedure can be initiated with a small volume of the incipient phase, which is then iteratively doubled until convergence. However, a comparative analysis of these initialization strategies is beyond the scope of the present work. The stability analysis in our implementation serves exclusively to generate a good initial guess for the subsequent flash calculations.

Algorithm 1 Initial Guess Generation and Feasibility Check for Phase Equilibrium

Require: Total internal energy $U^\star$ [J], volume $V^\star$ [m^3], and mole numbers $\mathbf{N}^\star = [N_1^\star, \ldots, N_n^\star]$, the trial phase concentration vector $\mathbf{c}$ [mol/m^3] and the trial phase internal energy density u [J/m^3].
Ensure: Feasible initial guess for phase split or termination if no solution exists

Compute temperature $T^\star$ = for given $U^\star, V^\star, \mathbf{N}^\star$ of the reference phase by solving Eq. (1).
Compute entropy $S^\star = S(T^\star, V^\star, \mathbf{N}^\star)$ of the reference phase using the equation of state (EOS)
Initialize the trial phase I as follows:

$$V^I = 0.5 \cdot V^\star$$
$$\mathbf{N}^I = V^I \cdot \mathbf{c}$$
$$U^I = u \cdot V^I$$

Initialize iteration count: $n_{\text{iters}} \leftarrow 0$
while $n_{\text{iters}} <$ max_iters **do**
    Compute total entropy for the two-phase system:

$$S_{\text{two-phase}} = S(U^I, V^I, \mathbf{N}^I) + S(U^\star - U^I, V^\star - V^I, \mathbf{N}^\star - \mathbf{N}^I)$$

    Compute entropy difference: $\Delta S = S_{\text{two-phase}} - S^\star$
    Update phase properties vector $\mathbf{x}$ for the trial phase:

$$\mathbf{x} = \begin{bmatrix} \mathbf{N}^I, & V^I, & U^I \end{bmatrix}$$

    Check feasibility of $\mathbf{x}$ using equations Eq. (10).
    if $\Delta S > 0$ **and** $\mathbf{x}$ is feasible **then**
        Return feasible initial guess $\mathbf{x}$
    if $V^I/V^\star < 10^{-8}$ **then**
        Terminate: No feasible solution found
    Update phase properties:

$$V^I \leftarrow V^I/2$$
$$U^I \leftarrow u \cdot V^I$$
$$\mathbf{N}^I \leftarrow V^I \cdot \mathbf{c}$$

    Increment iteration count: $n_{\text{iters}} \leftarrow n_{\text{iters}} + 1$
Return failure: No feasible solution found

The final feasible solution obtained from the stability analysis serves as the initial guess for phase split calculations. However, since the initialization is adapted from a TVN stability formulation, it may not always provide a suitable ascent direction for entropy in UVN problems. In such rare cases where Algorithm 1 fails to yield a good initial estimate, a fallback to the nested loop approach proposed by Castier [10] can be employed. This method involves solving an inner isothermal flash problem to compute phase volumes and internal energies, thereby generating a coarse UVN initial estimate. The practicality of this fallback is supported by the availability of established PTN flash routines. For the test problems considered in this work, this fallback was not necessary. In the following section, we review the phase split calculation method presented by Castier [10] and Smejkal et al. [11], which serves as the foundational framework for our work.

4. Direct entropy maximization formulation for UVN flash calculations

The UVN flash problem can be formulated as a direct entropy maximization problem, constrained by specified system properties as discussed by Castier [10], Smejkal et al. [11] and more recently by Paterson et al. [12]. Consider a multicomponent mixture composed of n species, distributed across p phases and a total energy $U^\star$, total volume $V^\star$ and the mole numbers vector $\mathbf{N}^\star$. The total entropy of the system, denoted as $S^{(\mathrm{UVN})}$, can be expressed as:

$$S^{(\mathrm{UVN})} = \sum_{k=1}^{p} S(U^k, V^k, \mathbf{N}^{(k)}), \tag{11}$$

where U^k, V^k, and $\mathbf{N}^{(k)} = \{N_1^{(k)}, \ldots, N_n^{(k)}\}$ represent the internal energy, volume, and mole numbers of each component in phase k, respectively. The superscript (UVN) highlights the fact that entropy here is expressed as a function of $U, V, \mathbf{N}$. Additionally, the problem is subject to the following constraints:

$$U^\star = \sum_{k=1}^{p} U^k, \quad V^\star = \sum_{k=1}^{p} V^k, \quad N_i^\star = \sum_{k=1}^{p} N_i^{(k)}, \quad i = 1, \ldots, n. \tag{12}$$

To simplify the problem, we can apply these constraints and reformulate the problem as an unconstrained optimization problem. This is done by writing the properties of phase p as a function of the properties in the other phases. For the entropy function, this reads:

$$S_{\mathrm{unc}}^{(\mathrm{UVN})} = \left[\sum_{k=1}^{p-1} S(U^k, V^k, \mathbf{N}^{(k)})\right] + S(\mathbf{x}^{(\xi)}), \tag{13}$$

where

$$\mathbf{x}^{(\xi)} := \left(U^{(\xi)}, V^{(\xi)}, \mathbf{N}^{(\xi)}\right), \tag{14}$$

where $U^{(\xi)} := U^\star - \sum_{k=1}^{p-1} U^k, V^{(\xi)} := V^\star - \sum_{k=1}^{p-1} V^k$ and $\mathbf{N}^{(\xi)} := \{N_1^\star - \sum_{k=1}^{p-1} N_1^{(k)}, \ldots, N_n^\star - \sum_{k=1}^{p-1} N_n^{(k)}\}$. Throughout this text, we denote the total system entropy, when expressed as a function of the reduced set of independent variables $\{U^k, V^k, \mathbf{N}^{(k)}\}_{k=1}^{p-1}$ for the unconstrained optimization problem, as S_{unc}. Here, the subscript unc signifies the use of this reduced variable set, and the superscript (ξ) denotes properties of the remaining phase p as defined by (14).

The unconstrained optimization problem now involves solving for the $(p-1)(n+2)$ unknowns: U^k, V^k, and $\mathbf{N}^{(k)}$ for each phase $k \in \{1, 2, \ldots, p-1\}$. Formally, we want to solve the following unconstrained optimization problem:

$$\mathbf{x} = \arg\max_{\mathbf{y}} S_{\mathrm{unc}}^{(\mathrm{UVN})}(\mathbf{y}), \tag{15}$$

where $\mathbf{y}$ now entails all the $(p-1)(n+2)$ unknowns. The solution to this optimization problem will be discussed in detail in Section 6. We will refer to this approach as the **UVN approach** throughout the rest of the paper.

Before proceeding further, we highlight a difficulty inherent to this approach. Given that the equation of state is in the form $f(T, V, \mathbf{N})$, Eq. (13) requires writing the entropy S as a function of U, V and $\mathbf{N}$. Therefore, the approach requires determining the temperature T by solving the equation:

$$U(T, V, \mathbf{N}) = U$$

for given U, V and $\mathbf{N}$. Once T is determined, the equation of state can then be used to compute the corresponding entropy. This process results in a nested Newton method where each iteration of the outer optimization problem requires multiple iterations of the inner solver to achieve convergence, thereby increasing the computational complexity of the solution procedure. We refer to this process as **implicit temperature calculation** throughout the paper.

To circumvent this difficulty, we reformulate the optimization problem directly in terms of T, V, and $\mathbf{N}$. This reformulation avoids the need to perform multiple implicit temperature calculations and will be discussed in the next section.

5. Reformulation of entropy maximization: Transition from unconstrained UVN to constrained TVN space

This section discusses the reformulation of UVN-flash problem in terms of the variables inherent to the Helmholtz energy-based equation of state (EOS), specifically in the TVN-space. This reformulation circumvents the need for repeated implicit temperature calculation to determine the temperature at each iteration. The objective function in this formulation is given by:

$$S_{\mathrm{unc}}^{(\mathrm{TVN})} = \left[\sum_{k=1}^{p-1} S\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)\right] + S\left(T, V^{(\xi)}, \mathbf{N}^{(\xi)}\right), \tag{16}$$

subject to the constraint $U^\star = \sum_{k=1}^{p} U\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)$. Note that the constraints for the total volume $V^\star$ and total moles $\mathbf{N}^\star$ are directly incorporated in the arguments of the entropy of phase p. However, the constraint of internal energy has not yet been incorporated. Rewriting this constraint in functional form yields

$$C(\mathbf{x}) := \left[\sum_{k=1}^{p-1} U\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)\right] + U\left(T, V^{(\xi)}, \mathbf{N}^{(\xi)}\right) - U^\star. \tag{17}$$

The solution $\hat{\mathbf{x}}$ satisfies the following constrained optimization problem:

$$\max_{\mathbf{x}} \quad S_{\mathrm{unc}}^{(\mathrm{TVN})}(\mathbf{x}) \quad \text{subject to } C(\mathbf{x}) = 0, \tag{18}$$

where the optimization variable $\mathbf{x} := \left(T, V^{(1)}, \mathbf{N}^1, \ldots, V^{(p-2)}, \mathbf{N}^{p-2}, \ldots, V^{(p-1)}, \mathbf{N}^{p-1}\right)$ is a vector of $(p-1)(n+1)+1$ unknowns.

This constrained optimization problem can be reformulated as an unconstrained saddle point problem using the method of Lagrange multipliers. The Lagrangian function is defined as:

$$\mathcal{L}(\mathbf{x}, \lambda) = S_{\mathrm{unc}}(\mathbf{x}) + \lambda\, C(\mathbf{x}), \tag{19}$$

where λ is the Lagrange multiplier. For convenience, where possible we omit the superscript (TVN) from the objective function. However, it is included when needed to ensure clarity. To find the optimum of the original constrained problem, we need to find the stationary points of the Lagrangian by solving the following system of equations:

$$\nabla_{\mathbf{x},\lambda}\mathcal{L} = 0,$$

where $\nabla_{\mathbf{x},\lambda}\mathcal{L} = \left(\nabla_{\mathbf{x}}\mathcal{L}, \frac{\partial \mathcal{L}}{\partial \lambda}\right)$. The gradient $\nabla_{\mathbf{x}}\mathcal{L}$ of the Lagrangian with respect to $\mathbf{x}$ is given by:

$$\nabla_{\mathbf{x}}\mathcal{L} = \left(\frac{\partial \mathcal{L}}{\partial \mathbf{x}_1}, \frac{\partial \mathcal{L}}{\partial \mathbf{x}_2}, \ldots, \frac{\partial \mathcal{L}}{\partial \mathbf{x}_{(p-1)(n+1)+1}}\right).$$

The condition $\nabla_{\mathbf{x},\lambda}\mathcal{L} = 0$ leads to two sets of equations.

1. Stationarity Condition:

$$\nabla S_{\mathrm{unc}}(\mathbf{x}) = -\lambda \nabla C(\mathbf{x}). \tag{20}$$

This ensures that the gradient of the objective function S_{unc} is parallel to the gradient of the constraint C.

2. Primal Feasibility Condition:

$$C(\mathbf{x}) = 0. \tag{21}$$

This ensures that the constraint is satisfied.

This formulation leads to a system of $(p-1)(n+1)+2$ equations. Specifically, for $p = 2$, the system contains one additional equation compared to the approach of Smejkal et al. [11]. For $p = 3$, the number of equations is the same in both approaches. However, for $p \geq 4$, our approach requires solving $p-3$ fewer equations as compared to Smejkal et al. [11].

To summarize, this section presented the reformulation of entropy maximization in TVN-space for the UVN-flash problem. The next section begins with a discussion of the numerical optimization procedure, followed by a derivation of the Lagrange multiplier, a simplification of the Lagrangian, and an analysis of the stationary conditions of the resulting objective function.

6. Computational framework

This section outlines the optimization of the objective function, as defined in Eq. (13) for Smejkal's approach and Eq. (19) for our method. Additionally, we derive an explicit expression for the Lagrange multiplier and use it to simplify the Lagrangian in our approach, making the implementation more straightforward. Finally, we conclude this section by performing a consistency check of the formulation. For simplicity, we assume the number of phases is known a priori, as determined by a stability analysis.

6.1. Numerical optimization

Formally, we seek to solve the following unconstrained optimization problem:

$$\mathbf{x} = \arg\max_{\mathbf{y}} \mathcal{F}(\mathbf{y}), \tag{22}$$

where $\mathcal{F}$ is the objective function. To solve this, we need to find the stationary points of the gradient of the objective function, denoted as $g(\mathbf{x})$. This gradient is expressed as:

$$g(\mathbf{x}) = \begin{cases} \nabla S_{\text{unc}}^{(\text{Smejkal})}(\mathbf{x}_{\text{UVN}}), & \text{UVN}, \\ \nabla S_{\text{unc}}^{(\text{Ours})}(\mathbf{x}_{\text{TVN}}) + \lambda \nabla C(\mathbf{x}_{\text{TVN}}), & \text{Ours}, \end{cases} \tag{23}$$

where $\mathbf{x}_{\text{UVN}}$ and $\mathbf{x}_{\text{TVN}}$ are the respective independent variables for each case:

$$\mathbf{x}_{\text{UVN}} := (\mathbf{N}^1, V^{(1)}, U^{(1)}, \ldots, \mathbf{N}^{p-1}, V^{(p-1)}, U^{(p-1)}), \tag{24a}$$

$$\mathbf{x}_{\text{TVN}} := (\mathbf{N}^1, V^{(1)}, \ldots, \mathbf{N}_{p-2}, V^{(p-2)}, \mathbf{N}^{p-1}, V^{(p-1)}, T). \tag{24b}$$

Here, $\mathbf{N}^k$ represents the mole vector in phase k, while $U^{(k)}$ and $V^{(k)}$ correspond to the internal energy and volume of phase k, respectively. We will revisit the alternate forms of $g(\mathbf{x})$ for our approach in the next section where we derive the Lagrange multiplier λ.

The optimization problem can now be written as solving $g(\mathbf{x}) = 0$. This is a nonlinear system which can be solved using a nonlinear solver, such as Newton–Raphson or a variant. To apply the Newton–Raphson method, we need the gradient of $g(\mathbf{x})$. The gradient of $g(\mathbf{x})$ is the Hessian of the objective function $\mathcal{F}$, given by:

$$\mathbb{H}(\mathbf{x}) = \left[\frac{\partial^2 \mathcal{F}}{\partial \mathbf{x}_i \partial \mathbf{x}_j} \right].$$

In this context, the Hessian for both optimization approaches is expressed as:

$$\mathbb{H}(\mathbf{x}) = \begin{cases} \dfrac{\partial^2 S_{\text{unc}}^{(\text{UVN})}(\mathbf{x})}{\partial \mathbf{x}_i \partial \mathbf{x}_j}, & \text{UVN} \\ \dfrac{\partial^2 S_{\text{unc}}^{(\text{TVN})}(\mathbf{x})}{\partial \mathbf{x}_i \partial \mathbf{x}_j} + \lambda \dfrac{\partial^2 C(\mathbf{x})}{\partial \mathbf{x}_i \partial \mathbf{x}_j}, & \text{Ours} \end{cases} \tag{25}$$

where $\mathbf{x}$ is defined as per Eq. (24), and the entropy function $S_{\text{unc}}(\mathbf{x})$ is given by Eq. (13) in Smejkal's formulation and by Eq. (16) in our approach. The additional term in our formulation accounts for the contribution of the constraint function $C(\mathbf{x})$ through the Lagrange multiplier λ, ensuring that the optimization respects the imposed constraints.

Eq. (23) can be solved using a non-linear solver. We employ Newton's method, which updates the solution iteratively as follows:

$$\mathbf{x}_{k+1} = \mathbf{x}_k + \alpha_k \Delta \mathbf{x}_k, \tag{26}$$

where α_k is the step size and the update direction $\Delta \mathbf{x}_k$ satisfies:

$$\mathbb{H}(\mathbf{x}_k) \Delta \mathbf{x}_k = -g(\mathbf{x}_k), \tag{27}$$

where $\mathbb{H}(\mathbf{x}_k)$ is the Hessian and $\mathbf{g}(\mathbf{x}_k)$ is the gradient. If $\mathbb{H}(\mathbf{x}_k)$ is singular or ill-conditioned, alternative approaches such as Levenberg–Marquardt regularization, modified Cholesky decomposition, or quasi-Newton methods (e.g., BFGS) can be employed [11,23]. However, no such issues were encountered in our test cases. For implementation, we use Newton's method from `NLsolve.jl` in Julia, with third-order backtracking Line Search. It is important to note that when Newton's method is combined with Line Search, the underlying optimization often minimizes a merit function (typically the sum of squares of the residuals). Stationary points of this merit function can include not only the true solutions to the original system but also local maxima or spurious solutions, which is a critical consideration in problems like phase stability where identifying correct equilibria is vital (see, e.g., [30]). While this potential issue exists, in our test cases, the algorithm consistently converged to valid solutions without encountering spurious ones. The good initial guess generated from stability analysis results likely contributed to this robust behavior.

The gradients and Hessian can be computed using automatic differentiation (AD).[2] However, we provide the derivations of the gradients for our approach in Appendix A along with the outline of the Hessian matrix, as we intend to use these gradients (of entropy and the constraint function) to compute the Lagrange multiplier, which is further discussed in the following section.

6.2. Derivation of the Lagrange multiplier

In this section, we discuss the computation of the Lagrange multiplier λ. Expanding the stationarity condition (20), we get

$$\frac{\partial S_{\text{unc}}}{\partial N_1^{(k)}} = -\lambda \frac{\partial C}{\partial N_1^{(k)}}, \quad \ldots, \quad \frac{\partial S_{\text{unc}}}{\partial N_n^{(k)}} = -\lambda \frac{\partial C}{\partial N_n^{(k)}}, \tag{28a}$$

$$\frac{\partial S_{\text{unc}}}{\partial V^{(k)}} = -\lambda \frac{\partial C}{\partial V^{(k)}}, \tag{28b}$$

$$\frac{\partial S_{\text{unc}}}{\partial T} = -\lambda \frac{\partial C}{\partial T}. \tag{28c}$$

From Eq. (28c), we isolate λ as:

$$\lambda = -\frac{\partial S_{\text{unc}}/\partial T}{\partial C/\partial T} \tag{29}$$

Substituting the expressions from Eqs. (A.3) and (A.10) into Eq. (29), we get

$$\lambda = -\frac{1}{T}. \tag{30}$$

The explicit dependence of the Lagrange multiplier λ on temperature T removes the need to treat λ as an independent optimization variable. This simplification reduces the dimensionality of the problem, as λ is no longer an unknown but is instead directly determined by T. By substituting (30) into the stationarity condition (20), the optimization process becomes more efficient, as we discuss in detail in the following section.

6.3. Objective function reformulation

With this choice of the Lagrange multiplier λ, Eq. (19) simplifies to

$$\mathcal{L}(\mathbf{x}) = S_{\text{unc}}(\mathbf{x}) - \frac{1}{T} C(\mathbf{x}), \tag{31}$$

where $\mathbf{x}$ is given by (24). As λ is now a fixed parameter and not a free variable to be determined, the function $\mathcal{L}(\mathbf{x})$ no longer corresponds to

[2] Automatic differentiation computes exact derivatives of a function by applying the chain rule to the sequence of operations performed in the program, without the approximation errors associated with numerical methods like finite differences. In our case, we use forward-mode AD to obtain derivatives at machine precision.

the standard Lagrangian used in constrained optimization. To avoid confusion, we refer to it as the Lagrange function. Substituting the expressions for the reduced entropy $S_{\text{unc}}(\mathbf{x})$ from (16) and the constraint $C(\mathbf{x})$ from (17), we get

$$\mathcal{L}(\mathbf{x}) = \sum_{k=1}^{p-1} S\left(T, V^{(k)}, \mathbf{N}^{(k)}\right) + S\left(T, V^{(\xi)}, \mathbf{N}^{(\xi)}\right) - \frac{1}{T}\left(\sum_{k=1}^{p-1} U\left(T, V^{(k)}, \mathbf{N}^{(k)}\right) + U\left(T, V^{(\xi)}, \mathbf{N}^{(\xi)}\right) - U^{\star}\right). \tag{32}$$

Rearranging terms and combining the entropy and internal energy contributions, we obtain

$$\mathcal{L}(\mathbf{x}) = \sum_{k=1}^{p}\left[S\left(T, V^{(k)}, \mathbf{N}^{(k)}\right) - \frac{U\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)}{T}\right] + \frac{U^{\star}}{T}. \tag{33}$$

Next, recalling the thermodynamic relation $A = U - TS$, where A is the Helmholtz free energy, we get

$$\mathcal{L}(\mathbf{x}) = \sum_{k=1}^{p}\left[S\left(T, V^{(k)}, \mathbf{N}^{(k)}\right) - \frac{A\left(T, V^{(k)}, \mathbf{N}^{(k)}\right) + TS\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)}{T}\right] + \frac{U^{\star}}{T}. \tag{34}$$

Upon simplifying the terms involving entropy and Helmholtz energy, we arrive at

$$\mathcal{L}(\mathbf{x}) = -\sum_{k=1}^{p}\frac{A\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)}{T} + \frac{U^{\star}}{T}. \tag{35}$$

Rearranging the terms gives

$$\mathcal{L}(\mathbf{x}) = \frac{U^{\star}}{T} - \sum_{k=1}^{p}\frac{A\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)}{T}. \tag{36}$$

Finally, segregating the residual terms corresponding to the pth phase, we get

$$\mathcal{L}(\mathbf{x}) = \frac{U^{\star} - \left(\sum_{k=1}^{p-1} A\left(T, V^{(k)}, \mathbf{N}^{(k)}\right) + A\left(T, V^{(\xi)}, \mathbf{N}^{(\xi)}\right)\right)}{T}. \tag{37}$$

Notably, this function is identical in form to the Q-function introduced by Michelsen [18]. We refer to this as the **Helmholtz energy-based Q-function** (ACQ, for short). Correspondingly, we designate the function represented by (32) as **Entropy-based Q-function** (SCQ for short). Michelsen presented this function without a derivation. Medeiros et al. [21] later presented the derivation of the Q-function within TPN-framework using two Legendre transformations. They also mentioned that a single Legendre transform would be required in TVN-space. In this paper, we have shown that in TVN-space, the Q-function can be derived using only one Lagrange multiplier. While the final forms of the Q-functions are similar, the derivation paths differ due to different underlying thermodynamic basis, i.e., TPN vs TVN framework.

Both SCQ and ACQ are mathematically equivalent as ACQ is derived directly from SCQ by using the relation $A = U - TS$. The Lagrange multiplier λ is explicitly determined as $-1/T$, effectively reducing the number of unknowns by one compared to the standard constrained optimization formulation Eq. (23), where λ is treated as an additional unknown alongside the state variables. Additionally, the SCQ approach requires more function evaluations compared to the ACQ. Specifically, SCQ involves evaluating both the entropy and the internal energy of each phase, while ACQ requires only the evaluation of the Helmholtz energy. This suggests that the ACQ formulation is computationally more efficient than SCQ. Consequently, we restrict our numerical results in Section 7 to ACQ formulation.

To maximize the entropy, the saddle point of the Lagrange function must be found by solving the system of equations $\nabla\mathcal{L}(\mathbf{x}) = 0$. The Hessian matrix in this formulation simplifies significantly as below:

$$\mathbb{H}(\mathbf{x}) = \left[\frac{\partial^2\mathcal{L}(\mathbf{x})}{\partial\mathbf{x}_i\partial\mathbf{x}_j}\right]. \tag{38}$$

Both Michelsen and Mederios et al. discussed a nested loop approach where a PT-flash can be solved in the inner loop. Additionally, they mentioned the use of a more efficient Newton's method and employing the nested loop approach as a fallback strategy. In this work, we restrict our focus to Newton's method. However, the convergence of Newton's method relies on the availability of sufficiently accurate initial estimates, as also noted by [10,11,18,21,30]. For a broader discussion of numerical techniques for saddle point problems, we refer the interested reader to the review by Benzi et al. [31]. Fortunately, the results obtained from the stability analysis provide high-quality initial guesses. Furthermore, for transient simulations e.g., pipeline transportation of multicomponent mixture, the results from previous time steps serve as a good initial guess. The analysis of the stationary conditions of the Lagrange function will be presented in the following subsection.

6.4. Analysis of stationary conditions and thermodynamic consistency

In this subsection, we analyze the stationary conditions of the function $\mathcal{L}$ defined in Eq. (31) to verify the thermodynamic consistency of our formulation. We demonstrate that setting the gradient of $\mathcal{L}$ with respect to all variables to zero recovers the necessary and sufficient conditions for thermodynamic equilibrium, including satisfaction of the total internal energy constraint and adherence to conditions of thermal, mechanical, and chemical equilibrium. To demonstrate consistency, we first show that the formulation with the Lagrange function defined as per Eq. (31) inherently satisfies the constraint of the total internal energy. The gradient of the Lagrange function with respect to temperature T is given by:

$$\frac{\partial\mathcal{L}}{\partial T} = \frac{\partial S_{\text{unc}}}{\partial T} - \frac{1}{T}\frac{\partial\mathcal{C}}{\partial T} + \frac{\mathcal{C}}{T^2}. \tag{39}$$

Substituting expressions from (A.3) and (A.10), we get

$$\frac{\partial\mathcal{L}}{\partial T} = \frac{1}{T}\sum_{k=1}^{p} C_v\left(T, V^{(k)}, \mathbf{N}^{(k)}\right) - \frac{1}{T}\sum_{k=1}^{p} C_v\left(T, V^{(k)}, \mathbf{N}^{(k)}\right) + \frac{\mathcal{C}}{T^2} = \frac{\mathcal{C}}{T^2}, \tag{40}$$

where C_v represents the heat capacity at constant volume. Using the optimality condition, $\frac{\partial\mathcal{L}}{\partial T} = 0$ yields:

$$\frac{\mathcal{C}}{T^2} = 0 \implies \mathcal{C} = 0. \tag{41}$$

In words, setting the derivative $\frac{\partial\mathcal{L}}{\partial T}$ to zero at a stationary point recovers the constraint $\mathcal{C} = 0$, ensuring that stationary points satisfy the total internal energy balance. Next, we verify consistency with respect to thermodynamic equilibrium by computing the gradients with respect to the volume and the mole numbers. The gradient of the Lagrange function with respect to the volume $V^{(k)}$:

$$\frac{\partial\mathcal{L}}{\partial V^{(k)}} = \frac{\partial S_{\text{unc}}}{\partial V^{(k)}} - \frac{1}{T}\frac{\partial\mathcal{C}}{\partial V^{(k)}}. \tag{42}$$

Substituting the expressions from Eqs. (A.6) and (A.14), we obtain

$$\frac{\partial\mathcal{L}}{\partial V^{(k)}} = \frac{\partial P^{(k)}}{\partial T} - \frac{\partial P^{(\xi)}}{\partial T} - \frac{1}{T}\left(T\left(\frac{\partial P^{(k)}}{\partial T}\right)_{V^{(k)},\mathbf{N}} - P^{(k)} - \left(T\left(\frac{\partial P^{(\xi)}}{\partial T}\right)_{V^{(\xi)},\mathbf{N}} - P^{(\xi)}\right)\right).$$

Here, the superscript (ξ) denotes evaluation at $\mathbf{x}^{(\xi)}$. After simplification, this reduces to

$$\frac{\partial\mathcal{L}}{\partial V^{(k)}} = \frac{P^{(k)}}{T} - \frac{P^{(\xi)}}{T}. \tag{43}$$

Similarly, the gradient of the Lagrange function with respect to the mole number $N_i^{(k)}$ of component i in phase k is:

$$\frac{\partial\mathcal{L}}{\partial N_i^{(k)}} = -\frac{\partial\mu_i^{(k)}}{\partial T} + \frac{\partial\mu_i^{(\xi)}}{\partial T} - \frac{1}{T}\left(\mu_i^{(k)} - T\frac{\partial\mu_i^{(k)}}{\partial T} - \left(\mu_i^{(\xi)} - T\frac{\partial\mu_i^{(\xi)}}{\partial T}\right)\right)$$

$$= -\frac{\mu_i^{(k)}}{T} + \frac{\mu_i^{(\xi)}}{T}. \tag{44}$$

Combining these results, the full gradient of the Lagrange function $\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x})$ is:

$$\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}) = \begin{pmatrix} \nabla_{\mathbf{x}}\mathcal{L}^{(1)} \\ \vdots \\ \nabla_{\mathbf{x}}\mathcal{L}^{(p-2)} \\ \nabla_{\mathbf{x}}\mathcal{L}^{(p-1)} \\ \frac{\partial \mathcal{L}}{\partial T} \end{pmatrix}, \tag{45}$$

where the individual entry

$$\nabla_{\mathbf{x}}\mathcal{L}^{(k)} = \begin{pmatrix} -\frac{\mu_1^{(k)}}{T} + \frac{\mu_1^{(\xi)}}{T} \\ \vdots \\ -\frac{\mu_n^{(k)}}{T} + \frac{\mu_n^{(\xi)}}{T} \\ \frac{P^{(k)}}{T} - \frac{P^{(\xi)}}{T} \end{pmatrix}, \tag{46}$$

and $\frac{\partial \mathcal{L}}{\partial T}$ is given by Eq. (40). The final gradients of the Lagrange function are structurally identical to those reported by Smejkal et al. [11], with the key distinction that our formulation allows all functions to be evaluated directly as a function of T, V and $\mathbf{N}$, whereas, Smejkal's formulation requires an inner Newton iteration to first determine the temperature. The optimality condition $\nabla_{\mathbf{x}}\mathcal{L}(\mathbf{x}) = 0$ leads to the following system of equations.

$$\mu_1^{(1)} = \mu_1^{(2)} = \cdots = \mu_1^{(\xi)}, \tag{47a}$$
$$\mu_2^{(1)} = \mu_2^{(2)} = \cdots = \mu_2^{(\xi)}, \tag{47b}$$
$$\vdots \tag{47c}$$
$$\mu_n^{(1)} = \mu_n^{(2)} = \cdots = \mu_n^{(\xi)}, \tag{47d}$$
$$P^{(1)} = P^{(2)} = \cdots = P^{(\xi)}, \tag{47e}$$
$$C = 0 \quad \text{(recovered from } \frac{\partial \mathcal{L}}{\partial T} = 0\text{)}. \tag{47f}$$

These conditions represent the necessary and sufficient criteria for thermodynamic equilibrium: equality of temperature (implicit as T is a single variable), equality of pressure across all coexisting phases, equality of chemical potential for each component across all coexisting phases, and satisfaction of the total internal energy constraint. This correspondence demonstrates that the stationary points of $\mathcal{L}$ coincide with thermodynamic equilibrium states, thereby validating the consistency of our formulation with established thermodynamic principles. The same analysis applies to the Helmholtz-based Q-function ACQ in (37). Please refer to Appendix B for details.

With the theoretical framework established, we now proceed to the results section, where we present numerical results obtained using the methodology discussed in this section which leverages the Lagrange function defined in Eq. (37).

7. Results

In this section, we present the results obtained using the approach discussed in previous section and compare them with existing literature. Our treatment focuses exclusively on the two-phase test cases examined by Castier [10], Smejkal et al. [11], and Bi et al. [20]. These problems have also been discussed by Nichita [23] and Bi et al. [29] in the context of stability analysis. Specifically, we consider Problems 1–6 from these studies, along with a pure component test case introduced by Smejkal et al. [11]. These problems are defined in Table 1, 2 and 3. Notably, no variable scaling was employed during the optimization process in our approach. This contrasts with methods like that of Smejkal et al. [11], who used variable scaling via a Jacobi preconditioner.

Table 1
Specification: Problems 1–4.

Property	Problem 1	Problem 2	Problem 3	Problem 4
U [J]	−756500.8	−1511407.6	−331083.7	−636468.0
V [cm^3]	52869.0	4268.1	80258.1	9926.71
N_{c_1} [mol]	10.0	0.95	15.1	10.0
N_{H_2S} [mol]	90.0	99.05	84.9	90.0

Table 2
Specification: Problems 5-6.

Property	Problem 5	Problem 6
U [J]	−16272506.4	24858.2
V [cm^3]	479845	289380.3
N_{C_2} [mol/m^3]	10.8	10.8
$N_{C_3H_6}$ [mol/m^3]	360.8	360.8
N_{C_3} [mol/m^3]	146.5	146.5
N_{iC_4} [mol/m^3]	233	233
N_{nC_4} [mol/m^3]	233	233
N_{C_5} [mol/m^3]	15.9	15.9

Table 3
Specification: Pure component CO_2.

Property	U [J]	V [m^3]	N_{CO_2} [mol]
Value	−87211375.744478	1	10 000

We begin by discussing the outcomes of the stability analysis, which serve as the foundation for determining the initial phase split. These results are then used to perform flash calculations, the details of which are presented subsequently. Finally, we validate our results with literature, followed by a discussion of the speedup gains.

For all calculations, the Peng–Robinson equation of state (EOS) [22], based on Helmholtz energy, is employed. Additional details regarding this EOS can be found in Appendix D.

7.1. Stability analysis

While phase stability analysis is a necessary preliminary step to obtain suitable initial guesses for the flash calculations in our framework, a detailed analysis of phase stability methods is outside the scope of this paper. For completeness, we first present the results of the stability analysis before proceeding to the flash results. We have obtained these results (for all the formulations) using the methodology discussed in Section 3. Our study reports the local minimum for each problem, with the results summarized in Tables 4–6. For each case, we report the computed values of temperature, component concentrations, and the tangent plane distance function D, as defined in Eq. (2), with the results reported to two significant digits. However, for values smaller than 1, the results are reported to four significant digits. In all cases, our local minima are in close agreement with the values (either global or local) reported by Nichita [23] for the multicomponent case and Smejkal et al. [11] for the single component case. The stability analysis reveals minimal discrepancies in concentration values, with errors remaining below 0.085%. The largest errors occur in Problem 6, with the highest being 0.085% for c'_{C_5}. In the following section, we utilize these stability results to initialize the phase split calculations.

7.2. Flash calculations

In this section, we present the initial guesses derived from stability analysis, generated using the Algorithm 1 described in Section 3.3.

Table 4
Results of stability analysis: Nichita [23] vs our results.

	Problem 1		Problem 2		Problem 3		Problem 4	
Property	Nichita	Current	Nichita	Current	Nichita	Current	Nichita	Current
T [K]	151.83	151.83	291.91	291.91	297.84	297.84	361.80	361.80
c'_{c_1} [mol/m^3]	104.13	104.12	146.11	146.18	188.14	188.14	1011.37	1011.36
c'_{H_2S} [mol/m^3]	564.39	564.35	736.15	736.58	1057.84	1057.84	10056.7	10037.91
D [Pa/K]	875.34	875.45	26771.1	26722	0.0	2.08e−12	0.5063	0.467

Table 5
Results of stability analysis Nichita [23] vs our results.

	Problem 5		Problem 6	
Property	Nichita	Current	Nichita	Current
T[K]	122.97	122.97	394.54	394.54
c'_{C_2} [mol]	0.3294	0.3294	46.41	46.41
$c'_{C_3H_6}$ [mol]	3.10	3.10	1739.38	1738.53
c'_{C_3} [mol]	0.9066	0.9066	719.16	718.79
c'_{iC_4} [mol]	0.3860	0.3860	1262.45	1261.59
c'_{nC_4} [mol]	0.2934	0.2934	1305.65	1304.69
c'_{C_5} [mol]	0.0038	0.0038	101.09	101.00
D[Pa/K]	35298.75	35298.74	16.3045	16.10

Table 6
Results for pure CO_2 from stability analysis. Smejkal et al. [11] vs our results.

Property	Smejkal	Current
T [K]	280.0	280.0
c' [mol/m^3]	19469.17	19487.12
D [Pa/K]	4608.22	4608.27

While Smejkal et al. [11] highlight the use of stability analysis to obtain initial guesses for flash calculations, their work does not explicitly provide these values for all the test cases, limiting the reproducibility of their results. To bridge this gap, we report the detailed initial guesses obtained from our stability analysis, followed by the results of the corresponding flash calculations. The initial guesses are comprehensively summarized in Tables 7–9, with the results reported to four significant digits.

Flash calculations are performed based on these initial guesses. For all results presented here, we have used the Helmholtz energy-based Q-function defined as per Eq. (37). Tables 10–12 present the results using Newton method with a third-order backtracking line search. The stopping criterion is set to a relative tolerance of 1×10^{-8}. The results are reported to six significant digits. In addition to the internal energy, volume and mole numbers, we also report the entropy of the reference phase and the two-phase system, denoted as S^I and S^{II}, respectively. A reasonable agreement is observed with the results reported by Smejkal et al. [11] for problems 1–6.

To further evaluate the generality of our method, we also consider a single-component test case, as discussed by Smejkal et al. [11], with specifications defined in Table 3. The stability analysis (see Table 6) reveals that the fluid is unstable as a single-phase fluid. Based on this analysis, an initial phase split was obtained, as shown in Table 9. Flash calculations are subsequently performed using this initial phase split, and the results are presented in Table 13. Our findings show excellent agreement with the results reported in the literature [11].

7.2.1. Speedup

We now turn our attention to the computational speedup achieved by our TVN approach compared to the UVN formulation. Both formulations are compared by directly using the same nonlinear solver in `Julia` employing Newton's method with line search. The results, summarized in Tables 14 and 15, were obtained using Newton–Raphson with a relative tolerance of 1×10^{-6} and per-variable, scale-invariant convergence criteria. Specifically, convergence is declared when, for each variable i,

$$\frac{|F_i(x^k)|}{|F_i(x^0)| + \epsilon} < 10^{-6}, \quad \frac{|x_i^{k+1} - x_i^k|}{|x_i^k| + \epsilon} < 10^{-6},$$

where $F : \mathbb{R}^n \to \mathbb{R}^n$ is the nonlinear system of equations being solved, F_i denotes its i^{th} component, x_i^k is the i^{th} component of the iterate at step k and ϵ is a small constant (e.g., machine precision) added to prevent division by zero. While moderate, a tolerance of $1{\times}10^{-6}$ is standard in comparative studies and sufficient to obtain physically meaningful results, ensuring a fair comparison of computational performance.

Both the TVN and UVN formulations require a comparable number of outer iterations across all test cases. A notable advantage of the TVN approach, however, lies in its circumvention of inner iterations, which represent a significant computational bottleneck in the UVN method. These inner iterations correspond to the total number of nonlinear function evaluations invoked by automatic differentiation (AD) during the inner Newton steps—specifically in the evaluation of the gradient and Hessian. Corresponding to each outer iteration in the UVN method, there are four evaluations of the inner nonlinear function defined by Eq. (D.4): two gradients (one per phase) and two Hessians (one per phase). Each of these inner evaluations entails determination of the phase temperatures T_1 and T_2 by solving Eq. (1). It is important to note that the number of inner iterations reported in Table 14 is implementation-dependent. A potential strategy is to initialize the inner Newton solver with temperature values carried over from the previous outer iteration. This approach, however, introduces a risk of numerical instability, especially in early iterations where the temperatures of the two phases may differ significantly. Consequently, the present study adopted a fixed initial temperature guess, e.g., $T = 300.0$, for all inner solves. However, for dynamic simulations, employing the temperature from the previous time step as the initial guess is a generally recommended practice. Under such conditions, a reduction in the number of inner iterations is anticipated. Nonetheless, the theoretical minimum number of inner iterations remains $4O_I$, where O_I denotes the number of outer iterations.

In contrast, the TVN approach circumvents this nested computational structure entirely, thereby diminishing both memory allocation and computational expenditure. Whereas the UVN method necessitates between 26 and 149 inner iterations across all test cases, the TVN formulation incurs no such computational overhead. For example, in Problem P5, the TVN approach achieves convergence about 30 times faster than the UVN method. This demonstrates the efficiency of the TVN formulation for the considered set of test problems.

It is worthwhile to note that variable scaling, where all variables are normalized by their respective total specified quantities, can improve the numerical stability and convergence behavior of the UVN method. However, as shown in Table 15, the UVN implementation exhibits scale-invariant results, with minimal impact of scaling on execution times across all test problems. While scaling does not significantly affect performance or convergence on the set of problems examined in this study, it may still offer benefits for numerical robustness in challenging cases.

Table 7
Initial guesses obtained from stability analysis for Problems 1, 2, 3, and 4.

Property	Problem 1		Problem 2		Problem 3		Problem 4	
	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2	Phase 1	Phase 2
N_{c_1} [mol]	0.0003	9.9997	0.0195	0.9305	0.0005	15.0995	1.2549	8.7451
N_{H_2S} [mol]	28.5304	61.4696	0.0982	98.9518	0.0583	84.8417	12.4787	77.5213
V [m^3]	0.0008	0.0520	0.0001	0.0041	2.45e−6	0.08026	0.0012	0.0087
U [J]	−717694	−38806	−379.56	−1.511e6	−891.17	−330193	−94307.8	−542160

Table 8
Initial guesses from stability analysis for Problems 5 and 6.

Property	Problem 5		Problem 6	
	Phase 1	Phase 2	Phase 1	Phase 2
N_{C_2} [mol]	0.6328	10.2672	0.8394	9.9624
$N_{C_3H_6}$ [mol]	84.6926	276.1074	31.4435	329.3565
N_{C_3} [mol]	22.4028	124.0972	13.0002	133.4998
N_{iC_4} [mol]	26.3615	206.6385	22.8175	210.1825
N_{nC_4} [mol]	82.5333	150.4667	23.5972	209.4028
N_{C_5} [mol]	9.1875	6.7125	1.8268	14.0732
V [m^3]	0.0150	0.4648	0.0181	0.2713
U [J]	−8.2275e6	−8.0450e6	−211881.92	236740.12

Table 9
Initial guesses obtained from stability analysis for pure component (CO_2).

Property	Phase 1	Phase 2
N_{CO_2} [mol]	2435.89	7564.11
V [cm^3]	0.125	0.875
U [J]	-3.129×10^7	-5.592×10^7

Furthermore, the TVN approach benefits from having inherently "well-behaved" optimization variables. For instance, temperature typically varies within a relatively narrow range-on the order of a few hundred Kelvins, whereas internal energy spans a much broader domain, often involving large-magnitude negative and positive values. Consequently, no explicit variable scaling was applied during the optimization process for the test cases considered.

8. Conclusion

In this work, we presented a reformulation of the UVN-flash problem in TVN-space. We simplified the numerical approach by transitioning from the unconstrained UVN space to the constrained TVN space. This reformulation eliminates the need for implicit temperature determination for given U, V and $\mathbf{N}$ at inner iterations of UVN-flash calculations, thereby significantly improving the efficiency of flash calculations.

We applied the method of Lagrange multipliers to transform the constrained optimization problem into a saddle point problem. By deriving the necessary gradients and Hessian, we obtained an explicit expression for the Lagrange multiplier in terms of temperature, eliminating the need to treat it as an independent variable. This led to the entropy-based Q-function (SCQ), which upon further simplification yielded the Helmholtz energy-based Q-function (ACQ) originally proposed by Michelsen. The saddle points of the ACQ function correspond to the maximization of entropy and are found by solving a system of nonlinear equations resulting from the stationarity conditions of the Lagrangian.

We also provided an explicit algorithm for generating high-quality initial guesses directly from stability analysis results. This crucial step greatly facilitates the convergence of the flash algorithm. We subsequently applied the reformulated approach to a set of test cases from the literature and validated the results against published data. We employed the Newton method with line search for solving the resulting nonlinear system and observed consistent convergence on all test cases.

Finally, we compared TVN reformulation against the UVN approach, which involves the entropy maximization in its natural variables (U, V, and $\mathbf{N}$). Our results show that the TVN formulation delivers substantial improvements in computational performance on all the test cases, making it a promising alternative for efficient and scalable UVN-flash calculations.

CRediT authorship contribution statement

Pardeep Kumar: Writing – original draft, Validation, Software, Methodology, Formal analysis, Conceptualization. **Patricio I. Rosen Esquivel:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Declaration of Generative AI and AI-assisted technologies in the writing process

During the preparation of this work the authors used GitHub Copilot in order to propose wordings and mathematical typesetting. After using this tool/service, the authors reviewed and edited the content as needed. The authors take full responsibility for the content of the publication.

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.

Acknowledgments

This research was generously supported by Shell Projects and Technology, and we deeply appreciate their invaluable contribution.

We would like to express our sincere gratitude to **Prof. Ruud Henkes** (TU Delft) and **Prof. Benjamin Sanderse** (CWI Amsterdam, TU Eindhoven) for their expert guidance and insightful contributions throughout this research.

Our heartfelt thanks also go to **Dr. Jannis Teunissen** (CWI Amsterdam), and **Dr. Marius Kurz** (CWI Amsterdam) for their constructive feedback on the manuscript.

We are grateful to **Prof. Jiri Mikyska** (Czech Technical University in Prague) for his invaluable insights and stimulating discussions on UVN-flash.

We also thank the anonymous reviewers for their valuable feedback.

Table 10
Comparison of flash results for Problems 1 and 2: Smejkal et al. [11] vs. our results.

	Problem 1		Problem 2	
	Smejkal	Current	Smejkal	Current
U [J]	−211544.585681	−211544.596326	−1510985.753624	−1510985.755666
V [cm^3]	51366.638771	51366.638597	4165.673900	4165.674425
N_{c_1} [mol]	9.664320	9.664319	0.930730	0.930730
N_{H_2S} [mol]	54.315978	54.315976	98.941685	98.941685
S^I [J K^{-1}]	−4847.824318	−4847.824867	−7391.709463	−7391.709647
S^{II} [J K^{-1}]	−4335.499136	−4335.499558	−7390.326639	−7390.326837
$T_{\text{phase 1}}$ [K]	297.997716	297.997717	298.000861	298.000876
$T_{\text{phase 2}}$ [K]	297.997716	297.997717	298.000856	298.000875
$P_{\text{phase 1}}$ [Pa]	2500170.787203	2500170.880449	2500317.847486	2500318.645643
$P_{\text{phase 2}}$ [Pa]	2500170.787153	2500170.880622	2500317.776275	2500318.640970

Table 11
Comparison of flash results for Problems 3 and 4: Smejkal et al. [11] vs. our results.

	Problem 3		Problem 4	
	Smejkal	Current	Smejkal	Current
U [J]	−330516.922985	−330516.953672	−390660.034825	−390689.64236
V [cm^3]	80256.537494	80256.537579	6414.083981	6414.415486
N_{c_1} [mol]	15.099651	15.099651	6.448582	6.448928
N_{H_2S} [mol]	84.862887	84.862889	56.390527	56.394270
S^I [J K^{-1}]	−2613.988230	−2613.988418	−4579.402758	−4579.403289
S^{II} [J K^{-1}]	−2613.987835	−2613.988023	−4579.402147	−4579.402679
$T_{\text{phase 1}}$ [K]	297.996887	297.99689	361.997885	361.997922
$T_{\text{phase 2}}$ [K]	297.996887	297.99689	361.997885	361.997922
$P_{\text{phase 1}}$ [Pa]	2500125.243552	2500125.055235	10130505.626170	1013051.326715
$P_{\text{phase 2}}$ [Pa]	2500124.858262	2500125.511354	10130505.626049	1013051.327506

Table 12
Comparison of flash results for Problems 5 and 6: Smejkal et al. [11] vs. our results.

	Problem 5		Problem 6	
	Smejkal	Current	Smejkal	Current
U [J]	−379886.931385	−380012.963119	174870.975415	174842.436972
V [cm^3]	401197.390420	401192.630291	273147.423428	273150.189814
N_{C_2} [mol]	4.203436	4.242459	10.064693	10.066498
$N_{C_3H_6}$ [mol]	68.225832	68.231202	333.710698	333.715455
N_{C_3} [mol]	24.416960	24.419097	135.325654	135.327702
N_{iC_4} [mol]	18.529159	18.531724	213.665513	213.668936
N_{nC_4} [mol]	13.885437	13.887650	213.118914	213.122442
N_{C_5} [mol]	0.325600	0.325674	14.391190	14.391459
S^I [J K^{-1}]	−73647.697512	−73640.643944	−9052.552759	−9052.541673
S^{II} [J K^{-1}]	−54939.068244	−54937.804163	−9052.431373	−9052.420341
$T_{\text{phase 1}}$ [K]	299.999735	300.004829	394.998501	394.998498
$T_{\text{phase 2}}$ [K]	299.999735	300.004829	394.998501	394.998498
$P_{\text{phase 1}}$ [Pa]	700082.833469	700360.612384	4230233.608414	4230243.484716
$P_{\text{phase 2}}$ [Pa]	700082.833469	700360.612385	4230233.576530	4230243.508068

Table 13
Comparison of flash results for pure CO_2: Smejkal vs. our results.

Property	Smejkal	Current
U [J]	−16873789.390417	−16873791.656255
V [cm^3]	481283.619636	481283.486064
N_{CO_2} [mol]	2818.038884	2818.038719
S^I [J/K]	−584388.217059	−584388.23982
S^{II} [J/K]	−583476.321606	−583476.346351
$T_{\text{phase 1}}$ [K]	299.040785	299.04079
$T_{\text{phase 2}}$ [K]	299.040785	299.04079
$P_{\text{phase 1}}$ [Pa]	6570486.596964	6570487.390738
$P_{\text{phase 2}}$ [Pa]	6570486.595448	6570487.390738

Table 14
Iteration counts for TVN (ACQ) and UVN formulations using line search.

Problem	TVN	UVN Outer	UVN Inner
P1	10	9	110
P2	4	4	40
P3	4	4	26
P4	9	10	69
P5	10	10	149
P6	5	5	52
PCO2	9	8	108

Table 15
Execution time comparison (in milliseconds) for TVN (ACQ) and UVN formulations using line search. The execution time reported is the average over 100 repetitions.

Problem	TVN (ms)	UVN no scale (ms)	UVN scaled (ms)
P1	0.18	2.20	2.06
P2	0.07	0.92	2.00
P3	0.04	0.75	0.70
P4	0.14	1.83	2.98
P5	1.23	40.19	39.99
P6	0.68	15.93	17.04
PCO2	0.05	1.00	0.99

Appendix A. Gradient computation for the Lagrangian function

In this section, we discuss the evaluation of the gradients of the entropy function defined by Eq. (16) and the constraint function defined by Eq. (17). These gradients are needed to compute the value of the Lagrange multiplier λ. The gradient $\nabla S_{\text{unc}}(\mathbf{x})$ of the entropy function $S_{\text{unc}}(\mathbf{x})$ is defined as:

$$\nabla S_{\text{unc}}(\mathbf{x}) = \begin{pmatrix} \nabla S_{\text{red}}^{(1)} \\ \vdots \\ \nabla S_{\text{red}}^{(p-2)} \\ \nabla S_{\text{red}}^{(p-1)} \\ \dfrac{\partial S_{\text{unc}}}{\partial T} \end{pmatrix}, \forall k \in \{1, \dots, p-1\}, \tag{A.1}$$

where $\nabla S_{\text{unc}}^{(k)} \in \mathbb{R}^{n+1}$, and $\frac{\partial S_{\text{unc}}}{\partial T} = \sum_{k=1}^{p} \frac{\partial S^{(k)}}{\partial T} \in \mathbb{R}$, and where $S^{(k)} = S\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)$ is the entropy of the phase k. The individual entries of $\nabla S_{\text{unc}}^{(k)}$ are given as below.

$$\nabla S_{\text{red}}^{(k)} = \begin{pmatrix} \dfrac{\partial S_{\text{unc}}}{\partial N_1^{(k)}} \\ \vdots \\ \dfrac{\partial S_{\text{unc}}}{\partial N_n^{(k)}} \\ \dfrac{\partial S_{\text{unc}}}{\partial V^{(k)}} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial S_{\text{red}}^{(k)}}{\partial N_1^{(k)}} - \dfrac{\partial S_{\text{red}}^{(\xi)}}{\partial N_1^{(\xi)}} \\ \vdots \\ \dfrac{\partial S_{\text{red}}^{(k)}}{\partial N_n^{(k)}} - \dfrac{\partial S_{\text{red}}^{(\xi)}}{\partial N_n^{(\xi)}} \\ \dfrac{\partial S_{\text{red}}^{(k)}}{\partial V^{(k)}} - \dfrac{\partial S_{\text{red}}^{(\xi)}}{\partial V^{(\xi)}} \end{pmatrix}. \tag{A.2}$$

We can simplify the partial derivatives using thermodynamic identities as follows:

$$\begin{aligned} \frac{\partial S_{\text{unc}}}{\partial T} &= \sum_{k=1}^{p-1} \frac{\partial S\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)}{\partial T} + \frac{\partial S\left(T, V^{\star} - \sum_{k=1}^{p-1} V^{(k)}, \mathbf{N}^{(\xi)}\right)}{\partial T} \\ &= \frac{1}{T} \sum_{k=1}^{p} C_v\left(T, V^{(k)}, \mathbf{N}^{(k)}\right). \end{aligned} \tag{A.3}$$

Furthermore, the thermodynamic identity for the volume derivative of entropy is given by:

$$\left(\frac{\partial S}{\partial V}\right)_{T,\mathbf{N}} = \left(\frac{\partial P}{\partial T}\right)_{V,\mathbf{N}}. \tag{A.4}$$

Next, for the derivative with respect to N, we can substitute S in terms of Helmholtz energy A as follows:

$$\left(\frac{\partial S}{\partial N}\right)_{T,V} = \left(\frac{\partial\left(-\frac{\partial A}{\partial T}\right)_{V,N}}{\partial N}\right)_{T,V}. \tag{A.5}$$

Since V is constant, we consider only T and N as variables, yielding:

$$\left(\frac{\partial S}{\partial N}\right)_{T} = \left(\frac{\partial\left(-\frac{\partial A}{\partial T}\right)_{N}}{\partial N}\right)_{T} = -\frac{\partial \mu}{\partial T}.$$

Finally, the gradient of the reduced entropy for phase k is given by:

$$\nabla S_{\text{red}}^{(k)} = \begin{pmatrix} \dfrac{\partial S_{\text{unc}}}{\partial N_1^{(k)}} \\ \vdots \\ \dfrac{\partial S_{\text{unc}}}{\partial N_n^{(n)}} \\ \dfrac{\partial S_{\text{unc}}}{\partial V^{(k)}} \end{pmatrix} = \begin{pmatrix} -\dfrac{\partial \mu_1(T, V^{(k)}, \mathbf{N}^{(k)})}{\partial T} + \dfrac{\partial \mu_1}{\partial T}(\mathbf{x}^{(\xi)}) \\ \vdots \\ -\dfrac{\partial \mu_n(T, V^{(k)}, \mathbf{N}^{(k)})}{\partial T} + \dfrac{\partial \mu_n}{\partial T}(\mathbf{x}^{(\xi)}) \\ \dfrac{\partial P(T, V^{(k)}, \mathbf{N}^{(k)})}{\partial T} - \dfrac{\partial P}{\partial T}(\mathbf{x}^{(\xi)}) \end{pmatrix}. \tag{A.6}$$

It is interesting to note that, while Smejkal's approach expresses the gradient of S_{unc} using the terms of the form $\frac{a}{b}$, our formulation instead involves partial derivatives of the form $\frac{\partial a}{\partial b}$. For instance, in our approach, the derivative of S_{unc} with respect to $N_1^{(k)}$ is expressed as

$$\frac{\partial S_{\text{unc}}}{\partial N_1^{(k)}} = -\frac{\partial \mu_1(T, V^{(k)}, \mathbf{N}^{(k)})}{\partial T} + \frac{\partial \mu_1(\mathbf{x}^{(\xi)})}{\partial T},$$

whereas in Smejkal's approach, it is given by

$$\frac{\partial S_{\text{unc}}}{\partial N_1^{(k)}} = -\frac{\mu_1(U^{(k)}, V^{(k)}, \mathbf{N}^{(k)})}{T} + \frac{\mu_1(\mathbf{x}^{(\xi)})}{T}.$$

This pattern persists across other derivatives as well, underscoring a fundamental difference in the treatment of thermodynamic variable dependencies between the two methodologies. We remark here that $U^{(k)}$ is defined differently for both approaches. For Smejkal's approach, it is the unknown of the optimization problem, whereas for our approach it is defined as $U^{(k)} := U(T, V^{(k)}, \mathbf{N}^{(k)})$. Having computed the gradients of the entropy function, we now turn our attention to the computation of the gradient of the constraint function C:

$$\nabla C(\mathbf{x}) = \begin{pmatrix} \nabla C^{(1)} \\ \vdots \\ \nabla C^{(p-2)} \\ \nabla C^{(p-1)} \\ \dfrac{\partial C}{\partial T} \end{pmatrix}, \quad \forall k \in \{1, \dots, p-1\}, \tag{A.7}$$

where $\nabla C^{(k)} \in \mathbb{R}^{n+1}$ and $\frac{\partial C}{\partial T} = \sum_{k=1}^{p} \frac{\partial U^{(k)}}{\partial T} \in \mathbb{R}, U^{(k)} := U(T, V^{(k)}, \mathbf{N}^{(k)})$, and

$$\nabla C^{(k)} = \begin{pmatrix} \dfrac{\partial C}{\partial N_1^{(k)}} \\ \vdots \\ \dfrac{\partial C}{\partial N_n^{(k)}} \\ \dfrac{\partial C}{\partial V^{(k)}} \end{pmatrix} = \begin{pmatrix} \dfrac{\partial C^{(k)}}{\partial N_1^{(k)}} - \dfrac{\partial C^{(\xi)}}{\partial N_1^{(\xi)}} \\ \vdots \\ \dfrac{\partial C^{(k)}}{\partial N_n^{(k)}} - \dfrac{\partial C^{(\xi)}}{\partial N_n^{(\xi)}} \\ \dfrac{\partial C^{(k)}}{\partial V^{(k)}} - \dfrac{\partial C^{(\xi)}}{\partial V^{(k)}} \end{pmatrix}, \tag{A.8}$$

We can simplify these gradients using standard thermodynamic identities. First, recall that the heat capacity at constant volume, C_v, is given by the following thermodynamic relation:

$$C_v = \left(\frac{\partial U}{\partial T}\right)_{V,\mathbf{N}}. \tag{A.9}$$

Consequently, the partial derivative of the constraint with respect to temperature becomes:

$$\frac{\partial C}{\partial T} = \sum_{k=1}^{p} C_v^{(k)}. \tag{A.10}$$

Next, utilizing the thermodynamic identity

$$\left(\frac{\partial U}{\partial V}\right)_{T,\mathbf{N}} = T\left(\frac{\partial P}{\partial T}\right)_{V,\mathbf{N}} - P, \tag{A.11}$$

we obtain the following expression for the partial derivative of the constraint with respect to volume:

$$\frac{\partial C^{(k)}}{\partial V^{(k)}} = \left(\frac{\partial U^{(k)}}{\partial V^{(k)}}\right)_{T,\mathbf{N}} = T\left(\frac{\partial P}{\partial T}\right)_{V^{(k)},\mathbf{N}} - P. \tag{A.12}$$

For the partial derivative of the constraint with respect to the mole number of component 1 in phase k, we proceed as follows:

$$\begin{aligned}\frac{\partial C^{(k)}}{\partial N_1^{(k)}} &= \left(\frac{\partial U^{(k)}}{\partial N_1^{(k)}}\right)_{T,V^{(k)}} = \left(\frac{\partial A^{(k)} + TS^{(k)}}{\partial N_1^{(k)}}\right)_{T,V^{(k)}} \\ &= \left(\frac{\partial A^{(k)}}{\partial N_1^{(k)}}\right)_{T,V^{(k)}} + T\left(\frac{\partial S^{(k)}}{\partial N_1^{(k)}}\right)_{T,V^{(k)}} \\ &= \mu_1^{(k)} - T\left(\frac{\partial \mu_1^{(k)}}{\partial T}\right),\end{aligned} \tag{A.13}$$

where $\mu_1^{(k)}$ is the chemical potential of component 1 in phase k. Similar expressions hold for components $2, \dots, p-1$. Finally, we can summarize the gradient of the constraint with respect to the generalized state variables:

$$\nabla C^{(k)}(\mathbf{x}) = \begin{pmatrix} \mu_1^{(k)} - T\dfrac{\partial \mu_1^{(k)}}{\partial T} - \left(\mu_1^{(\xi)} - T\dfrac{\partial \mu_1^{(\xi)}}{\partial T}\right) \\ \vdots \\ \mu_n^{(k)} - T\dfrac{\partial \mu_n^{(k)}}{\partial T} - \left(\mu_n^{(\xi)} - T\dfrac{\partial \mu_n^{(\xi)}}{\partial T}\right) \\ T\left(\dfrac{\partial P^{(k)}}{\partial T}\right)_{V^{(k)},\mathbf{N}} - P^{(k)} - \left(T\left(\dfrac{\partial P^{(\xi)}}{\partial T}\right)_{V^{(\xi)},\mathbf{N}} - P^{(\xi)}\right) \end{pmatrix}. \tag{A.14}$$

Appendix B. Gradient of the Helmholtz-based Q-function

This appendix provides analytical expressions for the gradient of the Helmholtz-energy-based Q-function $\mathcal{L}(\mathbf{x})$, defined as:

$$\mathcal{L}(\mathbf{x}) = \frac{U^\star - \left(\sum_{k=1}^{p-1} A\left(T, V^{(k)}, \mathbf{N}^{(k)}\right) + A\left(T, V^{(\xi)}, \mathbf{N}^{(\xi)}\right)\right)}{T}, \tag{B.1}$$

where $V^{(\xi)} := V^\star - \sum_{k=1}^{p-1} V^{(k)}$, $\mathbf{N}^{(\xi)} := \mathbf{N} - \sum_{k=1}^{p-1} \mathbf{N}^{(k)}$.

Partial derivative with respect to temperature:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial T} &= -\frac{U^\star}{T^2} - \frac{1}{T}\frac{\partial A}{\partial T} + \frac{A}{T^2} \\ &= -\frac{U^\star}{T^2} + \frac{S}{T} + \frac{A}{T^2} \\ &= \frac{A + TS - U^\star}{T^2} = \frac{U - U^\star}{T^2},\end{aligned} \tag{B.2}$$

where $S = \sum_{k=1}^{p} S^{(k)}$, $A = \sum_{k=1}^{p} A^{(k)}$, and $U = \sum_{k=1}^{p} U^{(k)}$. Thus, the stationarity condition $\frac{\partial \mathcal{L}}{\partial T} = 0$ enforces conservation of total internal energy, i.e., $U = U^\star$.

Partial derivatives with respect to phase volumes: For $k = 1, \dots, p-1$,

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial V^{(k)}} &= \frac{1}{T}\left[\frac{\partial A}{\partial V^{(k)}}\left(T, V^{(\xi)}, \mathbf{N}^{(\xi)}\right) - \frac{\partial A}{\partial V^{(k)}}\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)\right] \\ &= \frac{P^{(k)} - P^{(\xi)}}{T},\end{aligned} \tag{B.3}$$

where $V^{(\xi)} = V^\star - \sum_{k=1}^{p-1} V^{(k)}$ is the residual phase volume, and $P^{(k)} = -\frac{\partial A}{\partial V}(T, V^{(k)}, \mathbf{N}^{(k)})$ denotes the pressure in phase k.

Partial derivatives with respect to mole numbers: For each component i in phase $k = 1, \dots, p-1$, the gradient with respect to $N_i^{(k)}$ is:

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial N_i^{(k)}} &= \frac{1}{T}\left[\frac{\partial A}{\partial N_i^{(k)}}\left(T, V^{(\xi)}, \mathbf{N}^{(\xi)}\right) - \frac{\partial A}{\partial N_i^{(k)}}\left(T, V^{(k)}, \mathbf{N}^{(k)}\right)\right] \\ &= \frac{\mu_i^{(\xi)} - \mu_i^{(k)}}{T},\end{aligned} \tag{B.4}$$

where $\mu_i^{(k)} = \frac{\partial A}{\partial N_i^{(k)}}(T, V^{(k)}, \mathbf{N}^{(k)})$ denotes the chemical potential of component i in phase k.

Interpretation. These expressions demonstrate that stationarity of $\mathcal{L}$ corresponds to thermodynamic equilibrium, enforcing:

- Equality of pressure across all phases: $P^{(1)} = \cdots = P^{(\xi)}$,
- Equality of chemical potentials for each component across phases: $\mu_i^{(1)} = \cdots = \mu_i^{(\xi)}$ for all i,
- Conservation of total internal energy: $U = U^\star$.

These conditions are both necessary and sufficient for thermodynamic equilibrium under the specified constraints, thereby confirming the consistency of the Helmholtz-based Q-function. The pseudocode for this Q-function is provided in Algorithm 3.

Appendix C. Hessian of the Helmholtz-based Q-function

The Hessian matrix $H(\mathbf{x}) \in \mathbb{R}^{[(p-1)(n+1)+1]\times[(p-1)(n+1)+1]}$ of the Helmholtz-Based Q-function $\mathcal{L}$ defined by Eq. (37), admits the following block structure:

$$\mathbb{H}(\mathbf{x}) = \begin{bmatrix} \mathbb{H}_{N,N} & \mathbb{H}_{N,V} & \mathbb{H}_{N,T} \\ \mathbb{H}_{V,N} & \mathbb{H}_{V,V} & \mathbb{H}_{V,T} \\ \mathbb{H}_{T,N} & \mathbb{H}_{T,V} & \mathbb{H}_{T,T} \end{bmatrix}.$$

Each block has the following structure:

$\mathbb{H}_{N,N}$ $((p-1)n \times (p-1)n)$:

$$\mathbb{H}_{N,N}^{(k,\ell)} = \begin{bmatrix} \dfrac{\partial^2 \mathcal{L}}{\partial N_1^{(k)} \partial N_1^{(\ell)}} & \cdots & \dfrac{\partial^2 \mathcal{L}}{\partial N_1^{(k)} \partial N_n^{(\ell)}} \\ \vdots & \ddots & \vdots \\ \dfrac{\partial^2 \mathcal{L}}{\partial N_n^{(k)} \partial N_1^{(\ell)}} & \cdots & \dfrac{\partial^2 \mathcal{L}}{\partial N_n^{(k)} \partial N_n^{(\ell)}} \end{bmatrix}.$$

$\mathbb{H}_{N,V}$ $((p-1)n \times (p-1))$ and $\mathbb{H}_{N,T}$ $((p-1)n \times 1)$:

$$\mathbb{H}_{N,V}^{(k,\ell)} = \begin{bmatrix} \dfrac{\partial^2 \mathcal{L}}{\partial N_1^{(k)} \partial V^{(\ell)}} \\ \vdots \\ \dfrac{\partial^2 \mathcal{L}}{\partial N_n^{(k)} \partial V^{(\ell)}} \end{bmatrix}, \quad \mathbb{H}_{N,T}^{(k)} = \begin{bmatrix} \dfrac{\partial^2 \mathcal{L}}{\partial N_1^{(k)} \partial T} \\ \vdots \\ \dfrac{\partial^2 \mathcal{L}}{\partial N_n^{(k)} \partial T} \end{bmatrix}.$$

$\mathbb{H}_{V,V}$ $((p-1) \times (p-1))$ and $\mathbb{H}_{V,T}$ $((p-1) \times 1)$:

$$\mathbb{H}_{V,V}^{(k,\ell)} = \frac{\partial^2 \mathcal{L}}{\partial V^{(k)} \partial V^{(\ell)}}, \quad \mathbb{H}_{V,T}^{(k)} = \frac{\partial^2 \mathcal{L}}{\partial V^{(k)} \partial T}.$$

$\mathbb{H}_{T,T}$ (1×1):

$$\mathbb{H}_{T,T} = \frac{\partial^2 \mathcal{L}}{\partial T^2}.$$

For practical implementation, the computation of required gradients and Hessians is performed using automatic differentiation (AD). AD tools are employed due to their established robustness, computational efficiency, and their ability to evaluate analytical derivatives to machine precision. In this work, we have used the `ForwardDiff.jl` package [32] for AD.

Appendix D. Peng–Robinson equation of state

We employ the Peng–Robinson equation of state (EOS) [11], which is formulated as follows:

$$P(T, V, N_1, \dots, N_n) = \frac{NRT}{V - B} - \frac{a(T)N^2}{V^2 + 2BV - B^2}, \tag{D.1}$$

where T is the temperature, V is the volume, N_i represents the number of moles of component i in the system, R is the universal gas constant and N is the total number of moles in the system. The parameters $a(T)$ and B characterize intermolecular forces and volume exclusion, respectively. The parameters $a(T)$ and B are defined as follows:

$$B = bN, \tag{D.2a}$$

$$a = \sum_{i=1}^{n}\sum_{j=1}^{n} x_i x_j a_{ij}, \tag{D.2b}$$

$$a_{ij} = (1-\delta_{ij})\sqrt{a_i a_j}, \tag{D.2c}$$

$$a_i(T) = 0.45724\frac{R^2 T_{\mathrm{crit},i}^2}{P_{\mathrm{crit},i}}\left[1 + m_i\left(1-\sqrt{T_{r,i}}\right)\right]^2, \tag{D.2d}$$

$$b = \sum_{i=1}^{n} x_i b_i, \tag{D.2e}$$

$$b_i = 0.0778\frac{RT_{\mathrm{crit},i}}{P_{\mathrm{crit},i}}, \tag{D.2f}$$

where $x_i = N_i/N$ is the mole fraction of component i. $T_{\mathrm{crit},i}$, $P_{\mathrm{crit},i}$ and $T_{r,i} = T/T_{\mathrm{crit},i}$ are the critical temperature, critical pressure and the reduced temperature of component i, and δ_{ij} is the Kronecker delta. The parameter m_i accounts for the acentric factor ω_i as:

$$m_i = \begin{cases} 0.37464 + 1.54226\omega_i - 0.26992\omega_i^2, & \omega_i < 0.5, \\ 0.3796 + 1.485\omega_i - 0.1644\omega_i^2 + 0.01667\omega_i^3, & \omega_i \geq 0.5. \end{cases} \tag{D.3}$$

The residual internal energy, U, in the context of the Peng–Robinson EOS is expressed as follows.

$$U(T,V,N_1,\ldots,N_n) = N\frac{T\partial_T(a) - a}{2\sqrt{2}b}\ln\left[\frac{V+\delta_1 B}{V+\delta_2 B}\right] - NR(T-T_0) + \sum_{i=1}^{n} N_i\int_{T_0}^{T} c_{p,i}^{\mathrm{ig}}(\xi)\,d\xi + Nu_0, \tag{D.4}$$

where $\partial_T(a)$ is the temperature derivative of $a(T)$, T_0 is a reference temperature, α_{ik} are empirical constants, $\delta_1 = 1+\sqrt{2}$and$\delta_2 = 1-\sqrt{2}$. The residual entropy, S, is given as

$$S(T,V,N_1,\ldots,N_n) = NR\ln\left[\frac{V-B}{V}\right] + N\frac{\partial_T(a)}{2\sqrt{2}b}\ln\left[\frac{V+\delta_1 B}{V+\delta_2 B}\right] + R\sum_{i=1}^{n} N_i\ln\frac{VP_0}{N_i RT} + \sum_{i=1}^{n} N_i\int_{T_0}^{T}\frac{c_{p,i}^{\mathrm{ig}}(\xi)}{\xi}d\xi, \tag{D.5}$$

where $c_{p,i}^{\mathrm{ig}}(T)$ is the ideal gas heat capacity of component i and P_0 is a reference pressure. The heat capacity $c_{p,i}^{\mathrm{ig}}(T)$ can be written as:

$$c_{p,i}^{\mathrm{ig}}(T) = \sum_{k=0}^{3}\alpha_{ik}T^k. \tag{D.6}$$

Now, we can simplify the integral in (D.4) as

$$\int_{T_0}^{T} c_{p,i}^{\mathrm{ig}}(\xi)\,d\xi = \sum_{k=0}^{3}\alpha_{ik}\frac{T^{k+1}-T_0^{k+1}}{k+1},$$

and the integral in (D.5) as

$$\int_{T_0}^{T}\frac{c_{p,i}^{\mathrm{ig}}(\xi)}{\xi}d\xi, = \alpha_{i0}\ln\left(\frac{T}{T_0}\right) + \sum_{k=1}^{3}\alpha_{ik}\frac{T^k-T_0^k}{k}.$$

The coefficients $\alpha_0, \alpha_1, \alpha_2, \alpha_3$ for the fluids considered in this work are listed in Table D.16, while the parameters of the Peng–Robinson equation of state are summarized in Table D.17. It is important to note that the arguments of logarithmic terms must remain positive in Eqs. (D.4) and (D.5). If this condition is violated, the current step should be rejected or appropriately truncated to maintain physical consistency. The reference state is specified at $T_0 = 298.15\,\mathrm{K}$ and $P_0 = 1\,\mathrm{bar}$, where the molar internal energy is defined as

$$u_0 = u(T_0,P_0) = h(T_0,P_0) - RT_0 = -RT_0 = -2478.95687512\,\mathrm{J\,mol^{-1}}.$$

Table D.16
Correlation coefficients c_p^{ig} [11].

Component	α_0	α_1	α_2	α_3
C_1	19.25	5.213×10^{-2}	1.197×10^{-5}	-1.132×10^{-8}
H_2S	31.94	1.463×10^{-3}	2.432×10^{-5}	-1.176×10^{-8}
C_2	5.409	1.781×10^{-1}	-6.938×10^{-5}	8.713×10^{-9}
C_3H_6	3.710	2.345×10^{-1}	-1.160×10^{-4}	2.205×10^{-8}
C_3	−4.224	3.063×10^{-1}	-1.586×10^{-4}	3.215×10^{-8}
iC_4	−1.390	3.847×10^{-1}	-1.846×10^{-4}	2.895×10^{-8}
nC_4	9.487	3.313×10^{-1}	-1.108×10^{-4}	-2.822×10^{-8}
nC_5	−3.626	4.873×10^{-1}	-2.580×10^{-4}	5.305×10^{-8}
CO_2	19.80	7.344×10^{-2}	-5.602×10^{-5}	-1.715×10^{-8}

Table D.17
Parameters of Peng–Robinson EOS [11].

Component	T_{crit} [K]	P_{crit} [bar]	ω [-]
C_1	190.4	46.0	0.011
H_{2S}	373.2	89.4	0.081
C_2	305.4	48.8	0.099
C_3H_6	364.9	46.0	0.144
C_3	369.8	42.5	0.153
iC_4	408.2	36.5	0.183
nC_4	425.2	38.0	0.199
nC_5	469.7	33.7	0.251
CO_2	304.14	73.75	0.239

This definition ensures that the molar enthalpy of the ideal gas at the reference conditions is zero [11], i.e., $h(T_0,P_0) = 0$. Furthermore, the molar entropy of each pure component as an ideal gas is also set to zero at this state, $s_i^{\mathrm{ideal}}(T_0,P_0) = 0$.

The expressions for the Helmholtz free energy and the chemical potential are provided in the book by Michelsen and Møllerup [3]. For completeness, we reproduce them here. The residual part of the Helmholtz free energy is given by:

$$A(T,V,N_1,\ldots,N_n) = -NRT\ln\left[\frac{V-B}{V}\right] - N\frac{a(T)}{2\sqrt{2}b}\ln\left[\frac{V+\delta_1 B}{V+\delta_2 B}\right] \tag{D.7}$$

The residual part of chemical potential of the ith component can be calculated as the partial derivative of the Helmholtz free energy A with respect to N_i, keeping temperature T, volume V, and all other mole numbers constant:

$$\mu_i(T,V,N_1,\ldots,N_n) = \left(\frac{\partial A(T,V,N_1,\ldots,N_n)}{\partial N_i}\right)_{T,V,N_{j\neq i}} = RT\left(\frac{\partial F(T,V,N_1,\ldots,N_n)}{\partial N_i}\right)_{T,V,N_{j\neq i}}, \tag{D.8}$$

where the dimensionless Helmholtz energy is defined as $F := A/(RT)$ (see [3]).

The partial derivative of F with respect to N_i is given by:

$$\frac{\partial F}{\partial N_i} = -g - Ng_B B_i - \frac{D}{T}f_B B_i - \frac{f}{T}D_i, \tag{D.9}$$

where

$$f = \frac{1}{2\sqrt{2}B}\ln\left(\frac{V+\delta_1 B}{V+\delta_2 B}\right), \tag{D.10}$$

$$g = \ln\left(\frac{V-B}{V}\right), \tag{D.11}$$

$$g_B = -\frac{1}{V-B}, \tag{D.12}$$

$$f_V = \frac{1}{R(V+\delta_1 B)(V+\delta_2 B)}, \tag{D.13}$$

$$f_B = \frac{f+Vf_V}{B}, \tag{D.14}$$

$$D_i = 2\sum_{j=1}^{n} N_j a_{ij}, \tag{D.15}$$

$$b_{ij} = (1-l_{ij})(b_i+b_j)/2, \tag{D.16}$$

$$B_i = \frac{2\sum_{j=1}^{n} N_j b_{ij} - B}{N}. \tag{D.17}$$

Appendix E. Pseudocode for stability analysis

For completeness, we have included a concise pseudocode outline for the stability analysis in this section. Step 3 of the pseudocode is implemented using the `NLsolve.jl` solver.

Algorithm 2 Stability Analysis under UVN Specification

Require: Specified internal energy U_{spec}, volume V_{spec}, total composition $\boldsymbol{z}_{\text{spec}}$, and thermodynamic model

Estimate temperature T_{spec} consistent with $U_{\text{spec}}, V_{\text{spec}}, \boldsymbol{z}_{\text{spec}}$

Generate a set of trial concentrations $\{\boldsymbol{c}^{(j)}\}$ using a scaled simplex centered at the barycenter

for each trial concentration $\boldsymbol{c}^{(j)}$ **do**

Find the trial phase concentrations $\boldsymbol{c}'$, by solving the nonlinear system Eq. (5) at $T = T_{\text{spec}}$ using Newton's method, following the procedure in Section 3.1:

1. Set initial concentration guess $\boldsymbol{c}'^{,0}$, iteration counter $k \leftarrow 0$ and the solution $\boldsymbol{c}^\star \leftarrow \boldsymbol{c}'^{,0}$.
2. Set tolerance: $xtol \leftarrow 1\times10^{-8}$, $ftol \leftarrow 1\times10^{-8}, \epsilon \leftarrow \times10^{-10}$
3. **Repeat** for $k = 0, 1, 2, \ldots$ until convergence:

(a) Compute the Newton step $\Delta\boldsymbol{c}'^{,k}$ by solving:

$$\mathbf{J}(\boldsymbol{c}'^{,k})\Delta\boldsymbol{c}'^{,k} = -\mathbf{F}(\boldsymbol{c}'^{,k})$$

(b) Update the concentration vector using a suitable step length λ^k (Line search with third order Backtracking):

$$\boldsymbol{c}'^{,k+1} = \boldsymbol{c}'^{,k} + \lambda^k \Delta\boldsymbol{c}'^{,k}$$

(c) Terminate if, for every component i,

$$\frac{\left|F_i(\boldsymbol{c}'^{,k})\right|}{\left|F_i(\boldsymbol{c}'^{,0})\right| + \epsilon} < ftol, \qquad \frac{\left|c_i'^{,k+1} - c_i'^{,k}\right|}{\left|c_i'^{,k}\right| + \epsilon} < xtol,$$

Once converged, assign the final concentration vector $\boldsymbol{c}^\star \leftarrow \boldsymbol{c}'^{,k}$.

Compute internal energy density $u^* \leftarrow U(T_{\text{spec}}, 1.0, \boldsymbol{c}^\star)$.

Evaluate tangent plane distance D^* as per Eq. (2)

if $D^* \geq 0$ **then**

Mark system as phase-unstable for this trial

Store $\boldsymbol{c}^*$ and D^*

return Maximum D^*, corresponding $\boldsymbol{c}^*$, and list of all unstable points

Appendix F. Pseudocode for *ACQ*

The pseudocode for the *ACQ* formulation is briefly outlined in this section. For a robust implementation, the automatic addition and removal of phases, even during two-phase calculations, is critical, as detailed in Section 2.4. We have excluded this aspect from the current pseudocode to maintain focus on the core formulation. For details, we refer the reader to Castier [10]. Note that the objective function for line search can be either the sum of squares of the residual or, as in this work, a cubic polynomial approximation, following the approach discussed in detail in Numerical Recipes [33].

Data availability

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

Algorithm 3 Pseudocode for *ACQ*

Require: Total internal energy $U^\star$ [J], volume $V^\star$ [m^3], and mole numbers $\mathbf{N}^\star = [N_1^\star, \ldots, N_n^\star]$, the trial phase concentration vector $\mathbf{c}$ [mol/m^3] and the trial phase internal energy density u [J/m^3]

Generate initial guess using Algorithm 1: $x \leftarrow x^{(0)}$

Define gradient (see Appendix B) using AD (Note: AD gives exact derivative to machine precision.)

$$g(x) = \texttt{ForwardDiff.gradient}(\mathcal{L}, x)$$

Define Hessian (see Appendix C) using AD

$$H(x) = \texttt{ForwardDiff.hessian}(\mathcal{L}, x)$$

Initialize iteration count: $n_{\text{iters}} \leftarrow 0$

Set tolerance: $xtol \leftarrow 1\times10^{-8}$, $ftol \leftarrow 1\times10^{-8}$

while $n_{\text{iters}} <$ max_iters **do**

Compute Newton step δx by solving $H(x)\,\delta x = -g(x)$

$x_{\text{new}} \leftarrow x + \delta x$

Feasibility Check:

Let $x_{\text{trial}} = x + \delta x$.

If x_{trial} does not satisfy conditions Eq. (10) **or**

any logarithmic argument in Eq. (D.4) or Eq. (D.5) is not positive for x_{trial}:

Reduce step size for δx using line search with third-order backtracking (see [33]) and retry this step.

Else:

$x_{\text{new}} = x_{\text{trial}}$

Convergence Check:

grad_converged $\leftarrow \frac{\|g(x_{\text{new}})\|_\infty}{\|g(x^{(0)})\|_\infty + \epsilon} < ftol$

step_converged $\leftarrow \frac{\|x_{\text{new}} - x\|_\infty}{\|x\|_\infty + \epsilon} < xtol$

If grad_converged **or** step_converged

return x_{new}, n_{iters}

Update current solution: $x \leftarrow x_{\text{new}}$

Increment iteration count: $n_{\text{iters}} \leftarrow n_{\text{iters}} + 1$

return failure ▷ Maximum iterations reached without convergence

References

[1] M.L. Michelsen, The Isothermal Flash Problem. Part II. Phase-Split Calculation, Fluid Phase Equilibria, 1981.

[2] M.L. Michelsen, The Isothermal Flash Problem. Part I. Stability, Fluid Phase Equilibria, 1982.

[3] M.L. Michelsen, J.M. Mollerup, Thermodynamic Models : Fundamentals & Computational Aspects, Tie-Line Publications, 2007.

[4] A.R.J. Arendsen, G.F. Versteeg, Dynamic thermodynamics with internal energy, volume, and amount of moles as states: Application to Liquefied Gas Tank, Ind. Eng. Chem. Res. 48 (2009) 3167–3176.

[5] M. Castier, Dynamic simulation of fluids in vessels via entropy maximization, J. Ind. Eng. Chem. 16 (2010) 122–129.

[6] F.M. Goncalves, M. Castier, O.Q.F. Araújo, Dynamic simulation of flash drums using rigorous physical property calculations, Braz. J. Chem. Eng. 24 (2007) 277–286.

[7] M. Lu, L.D. Connell, The transient behaviour of CO2 flow with phase transition in injection wells during geological storage – Application to a case study, J. Pet. Sci. Eng. 124 (2014) 7–18.

[8] D. Müller, W. Marquardt, Dynamic multiple-phase flash simulation: Global stability analysis versus quick phase determination, Comput. Chem. Eng. 21 (1997) S817–S822.

[9] K.M. Brantferger, G.A. Pope, K. Sepehrnoori, Development of a thermodynamically consistent, fully implicit, equation-of-state, compositional steamflood simulator, in: SPE Symposium on Reservoir Simulation, SPE, Anaheim, California, 1991, pp. SPE–21253–MS.

[10] M. Castier, Solution of the isochoric–isoenergetic flash problem by direct entropy maximization, Fluid Phase Equilib. 276 (2009) 7–17.

[11] T. Smejkal, J. Mikyška, Phase stability testing and phase equilibrium calculation at specified internal energy, volume, and moles, Fluid Phase Equilib. 431 (2017) 82–96.

[12] D. Paterson, E.H. Stenby, W. Yan, Use of canonical variables to solve state function based flash problems, Fluid Phase Equilib. 571 (2023) 113795.
[13] D. Paterson, M.L. Michelsen, W. Yan, E.H. Stenby, Extension of modified RAND to multiphase flash specifications based on state functions other than (T, P), Fluid Phase Equilib. 458 (2018) 288–299.
[14] D. Paterson, W. Yan, M.L. Michelsen, E.H. Stenby, Multiphase isenthalpic flash: General approach and its adaptation to thermal recovery of heavy oil, AIChE J. 65 (2019) 281–293.
[15] D. Paterson, Flash Computation and EoS Modelling for Compositional Thermal Simulation of Flow in Porous Media. Springer Theses, Springer International Publishing, Cham, 2019.
[16] V. Lipovac, O. Duran, E. Keilegavlen, F. Radu, I. Berre, Unified flash calculations with isenthalpic and isochoric constraints, Fluid Phase Equilib. 578 (2024) 113991.
[17] M. Fathi, S. Hickel, Rapid multi-component phase-split calculations using volume functions and reduction methods, AIChE J. 67 (2021) e17174.
[18] M.L. Michelsen, State function based flash specifications, Fluid Phase Equilib. 158-160 (1999) 617–626.
[19] S. Saha, J.J. Carroll, The isoenergetic-isochoric flash, Fluid Phase Equilib. 138 (1997) 23–41.
[20] R. Bi, A. Firoozabadi, P.C. Myint, Phase-split computations in the internal energy, volume, and moles (UVN) space, Fluid Phase Equilib. 526 (2020) 112729.
[21] F. Medeiros, E.H. Stenby, W. Yan, State function-based flash specifications for open systems in the absence or presence of chemical reactions, AIChE J. 67 (2021) e17050.
[22] D.-Y. Peng, D.B. Robinson, A new two-constant equation of state, Ind. Eng. Chem. Fundam. 15 (1976) 59–64, Number: 1.
[23] D.V. Nichita, Robustness and efficiency of phase stability testing at VTN and UVN conditions, Fluid Phase Equilib. 564 (2023) 113624.
[24] J. Mikyška, A. Firoozabadi, Investigation of mixture stability at given volume, temperature, and number of moles, Fluid Phase Equilib. 321 (2012) 1–9.
[25] D.V. Nichita, A unified presentation of phase stability analysis including all major specifications, Fluid Phase Equilib. 578 (2024) 113990.
[26] N. Nagarajan, A. Cullick, A. Griewank, New strategy for phase equilibrium and critical point calculations by thermodynamic energy analysis. Part I. Stability analysis and flash, Fluid Phase Equilib. 62 (1991) 191–210.
[27] D.V. Nichita, J.-C. de Hemptinne, S. Gomez, Isochoric phase stability testing for hydrocarbon mixtures, Pet. Sci. Technol. 27 (2009) 2177–2191.
[28] D.V. Nichita, Fast and robust phase stability testing at isothermal-isochoric conditions, Fluid Phase Equilib. 447 (2017) 107–124.
[29] R. Bi, A. Zidane, A. Firoozabadi, Stability analysis in the internal energy, volume, and moles (UVN) space, Fluid Phase Equilib. 512 (2020) 112468.
[30] N. Jorge, Numerical Optimization, in: Springer Series in Operations Research and Financial Engineering, Springer New York, 2006.
[31] M. Benzi, G.H. Golub, J. Liesen, Numerical solution of saddle point problems, Acta Numer. 14 (2005) 1–137.
[32] J. Revels, M. Lubin, T. Papamarkou, Forward-mode automatic differentiation in Julia, 2016, arXiv:1607.07892 [cs].
[33] W.H. Press, S.A. Teukolsky, W.T. Vetterling, Numerical Recipes: The Art of Scientific Computing, third ed., Cambridge University Press, Cambridge, 2007.