Differential equations for critical curves of fluid mixtures

Ulrich K. Deiters

Institute of Physical Chemistry, University of Cologne, Greinstr. 4–6, 50939 Köln, Germany

Ian H. Bell

Applied Chemicals and Materials Div., National Institute of Science and Technology, Boulder, CO 80305, USA

ABSTRACT

The critical conditions of multicomponent fluid mixtures are expressed in concentration coordinates (isochoric thermodynamics formalism) and then used to construct differential equations for critical curves. These are then integrated with adaptive numerical integration methods. This method can be used with all Helmholtz energy-explicit equations of state, including models that can return unphysical results when applied to thermodynamic states within a two-phase region. In combination with the "parametric marching" technique, the new approach is able to follow critical curves of arbitrary shape.

The approach makes use of the formalism of directional derivatives. Some methods for the numerical differentiation with machine precision are proposed.

The supplemental information provides an implementation of this approach for the GERG-2004 and Peng–Robinson models in the Python language.

1. INTRODUCTION

For pure fluids there exists a single critical point, which marks the end of the vapor pressure curve and at which the saturated liquid and the saturated vapor coalesce. For binary fluid mixtures the locus of critical states is a 1-dimensional object, e.g., a curve in pTx-space (pressure, temperature, mole fraction). In the commonly used phase diagrams—two-dimensional projections onto the pT, Tx or px planes—critical curves separate regions where the fluid is in a single-phase state from regions where two-phase equilibria are possible.

The matter is complicated by the fact that, for mixtures, not only vapor–liquid phase separations, but also liquid–liquid separations can occur, and that a single phase diagram may contain several critical curves.

The practical importance of critical curves is not only due to the fact that they indicate the single-phase regions of fluid mixtures, but also that one can from them qualitatively construct the complete phase equilibrium (as far as fluid–fluid equilibria are concerned; solid–fluid equilibria are a different matter). The patterns of critical curves therefore are the basis of phase diagram classification schemes, e.g., that of van Konynenburg and Scott [1] or the more refined scheme of Bolz et al. [2].

For mixtures of three and more components, the loci of the critical states are surfaces or hypersurfaces, respectively, which are difficult to visualize. It is then usual to assume constraints, e.g., to keep some mole fractions or mole fraction ratios constant, which bring the dimension of the critical locus down to 1. In this sense it is possible to discuss critical curves of multicomponent mixtures, and we explicitly state that the methods for the calculation of critical curves of this work are applicable to multicomponent mixtures.

The calculation of critical curves of fluid mixtures has, of course, a long history. The critical conditions for a binary mixture, expressed with the molar Gibbs energy (as a function of temperature T, pressure p, and mole fraction x_1) are

$$\begin{aligned} \left(\frac{\partial^2 G_\mathrm{m}}{\partial x_1^2}\right)_{p,T} &= 0 \\ \text{and } \left(\frac{\partial^3 G_\mathrm{m}}{\partial x_1^3}\right)_{p,T} &= 0\ . \end{aligned} \tag{1}$$

These equations, however, are not well suited for work based on volume-explicit equations of state. It is possible, fortunately, to construct alternative criteria in terms of derivatives of the molar Helmholtz energy

(again for binary mixtures),

$$
\begin{aligned}
D_2 &= \begin{vmatrix} \frac{\partial^2 A_\mathrm{m}}{\partial x_1^2} & \frac{\partial^2 A_\mathrm{m}}{\partial x_1 \, \partial V_\mathrm{m}} \\ \frac{\partial^2 A_\mathrm{m}}{\partial x_1 \, \partial V_\mathrm{m}} & \frac{\partial^2 A_\mathrm{m}}{\partial V_\mathrm{m}^2} \end{vmatrix} \\
D_3 &= \begin{vmatrix} \frac{\partial^2 A_\mathrm{m}}{\partial x_1^2} & \frac{\partial^2 A_\mathrm{m}}{\partial x_1 \, \partial V_\mathrm{m}} \\ \frac{\partial D_2}{\partial x_1} & \frac{\partial D_2}{\partial V_\mathrm{m}} \end{vmatrix} ,
\end{aligned}
\tag{2}
$$

for instance by application of Jacobian determinants. These criteria were known at the beginning of the 20th Century already. One of the earliest attempts to compute critical curves from them was that of van Laar from 1905 [3], when only mechanical calculation devices were available.

These criteria can be extended to multicomponent mixtures, and they form the basis of most of the published algorithms for the calculation of critical curves. Exceptions are the method of Heidemann and Khalil [4], which uses derivatives of the Helmholtz energy with respect to amounts of substance, but in the end leads to criteria containing determinants, and the method of Quiñones-Cisneros [5, 6], which analyzes the eigenvalues of the Hessian matrix of an appropriate thermodynamic potential. In principle, the calculation of critical curves is therefore a thoroughly studied and solved problem.

In practice, however, this problem still deserves attention, for the convergence even of sophisticated modern root finders for the systems of equations (1) or equivalent equations expressed in terms of Helmholtz energies is notoriously bad. Methods based on eigenvalue analysis converge better, but still leave much to be desired. Therefore much effort went into the development of (usually sequential) iteration schemes that can overcome these difficulties. We mention here the works of Deiters and Schneider [7], Hicks and Young [8], Heidemann and Khalil [4], Boshkov et al. [9, 10], van Pelt et al. [11], Sadus [12], and Bell and Jäger [13] as well as works cited therein. The algorithms proposed in these works are, in principle, able to locate all critical curves of a given fluid mixture.

But still some problems remain. The search strategy by the most of the aforementioned algorithms is to vary two state variables, e.g., density and temperature, independently over their physically meaningful ranges and to watch for changes of sign of the criteria, Eq. (2). It is then unavoidable that the equation of state is invoked for metastable or unstable states. Usually this is not a problem: The algorithms "recognize" unstable states and move away from them towards the sought-for critical state. But there exist equations of state which exhibit an erratic behavior within their two-phase region, for instance the GERG-2008 equation [14] (Its erratic behavior is clearly revealed in the "criticality plots" of Bell and Jaeger [13]). This equation cannot be used reliably with classical algorithms for critical-curve calculation, unless one can guarantee that the search path of the algorithm does not cross "forbidden ground"—and therefore the course of a critical curve needs to be known before one can compute it!

The sometimes erratic behavior of the GERG-2008 equation affects not only calculations of critical curves, but also of phase envelopes. For the latter, however, we recently proposed a robust algorithm based on differential equations in the framework of "isochoric thermodynamics" [15]. It suggests itself to develop differential equations for the critical curves of fluid mixtures.

2. THEORY

2.1. Conventions. In the following sections we shall use a shorthand notation for partial derivatives of a molar thermodynamic property X_m with respect to some variables y and z,

$$
X_{(iyjz)} \equiv \left(\frac{\partial^{i+j} X_\mathrm{m}}{\partial y^i \, \partial z^j} \right) ,
$$

where the not mentioned *natural* variables are kept constant, e.g., $G_{(x)} = (\partial G_\mathrm{m}/\partial x_1)_{p,T}$.

We shall use expressions like

$$\left.\frac{\mathrm{d}X}{\mathrm{d}y}\right|_C$$

to indicate the derivative of X with respect to y along a path described by the condition C. An example is the change of the concentration of component i with respect to temperature *along the critical curve*, $\mathrm{d}\rho_i/\mathrm{d}T|_\mathrm{c}$.

Finally, derivatives like

$$\frac{\partial X(\vec{x})}{\partial \vec{y}}$$

are directional derivatives, i.e., derivatives where the components of the argument vector $\vec{x}$ are varied in a direction specified by the vector $\vec{y}$. The reader is referred to Appendix B. for definitions.

2.2. Classical Approach. The classical derivation of differential equations for critical curves [6, Ch. 5.6.1], which we show here for the readers' convenience, starts from Eq. (1) and generates the total differential of the first condition, $G_{(2x)} = 0$,

$$\mathrm{d}G_{(2x)} = -S_{(2x)}\,\mathrm{d}T + V_{(2x)}\,\mathrm{d}p + G_{(3x)}\,\mathrm{d}x_1 \stackrel{!}{=} 0\ . \tag{3}$$

The natural variables are those of G_m in this case, namely T, p, and x_1. Now the last term of Eq. (3) is zero because of the second critical condition. Division by $\mathrm{d}T$ and rearrangement yields a first differential equation for the critical curve of a binary mixture,

$$\left.\frac{\mathrm{d}p}{\mathrm{d}T}\right|_\mathrm{c} = \frac{S_{(2x)}}{V_{(2x)}}\ . \tag{4}$$

The calculation of the pT projection of the critical curve can thus be treated as an initial-value problem; the critical curve can be obtained by numerically integrating Eq. (4), starting at the critical point of one of the two pure components.

Forming the total differential of the second critical condition in a similar fashion yields

$$\mathrm{d}G_{(3x)} = -S_{(3x)}\,\mathrm{d}T + V_{(3x)}\,\mathrm{d}p + G_{(4x)}\,\mathrm{d}x_1 \stackrel{!}{=} 0\ , \tag{5}$$

which can be rearranged to

$$\left.\frac{\mathrm{d}T}{\mathrm{d}x_1}\right|_\mathrm{c} = \frac{G_{(4x)}}{S_{(3x)} - V_{(3x)}\left.\frac{\mathrm{d}p}{\mathrm{d}T}\right|_\mathrm{c}} = \frac{G_{(4x)}}{S_{(3x)} - \frac{V_{(3x)}S_{(2x)}}{V_{(2x)}}}\ . \tag{6}$$

Numerical integration of this equation gives the temperature along the critical curve as a function of the mole fraction.

The calculation of critical curves by integrating the differential equations (4) and (6) might seem easy at a first glance, but the devil is in the details: The derivatives $S_{(2x)}$, $V_{(2x)}$, and $G_{(4x)}$ are derivatives *at constant pressure*, whereas equations of state are usually functions of volume or density. It is possible to convert

between these two kinds of derivatives by means of Jacobian determinants, but the resulting expressions get impressively complicated even for binary mixtures:

$$
\begin{aligned}
G_{(2x)} &= A_{(2x)} + A_{(Vx)}V_{(x)} \text{ with } V_{(x)} = -\frac{A_{(Vx)}}{A_{(2V)}} \\
S_{(2x)} &= -A_{(2xT)} - A_{(VxT)}V_{(x)} - A_{(Vx)}V_{(xT)} \text{ with } V_{(xT)} = -\frac{A_{(VxT)}A_{(2V)} - A_{(2VT)}A_{(Vx)}}{A_{(2V)}^2} \\
V_{(2x)} &= -\frac{1}{A_{(2V)}}\left[A_{(V2x)} + 2A_{(2Vx)}V_{(x)} + A_{(3V)}V_{(x)}^2\right] \\
G_{(3x)} &= A_{(3x)} + 3A_{(V2x)}V_{(x)} + 3A_{(2Vx)}V_{(x)}^2 + A_{(3V)}V_{(x)}^3 \\
S_{(3x)} &= -A_{(3xT)} - 3A_{(V2xT)}V_{(x)} - 3A_{(V2x)}V_{(xT)} - 3A_{(2VxT)}V_{(x)}^2 - 6A_{(2Vx)}V_{(x)}V_{(xT)} \\
&\quad - A_{(3VT)}V_{(x)}^3 - 3A_{(3V)}V_{(x)}^2V_{(xT)} \\
G_{(4x)} &= A_{(4x)} + 4A_{(V3x)}V_{(x)} + 6A_{(2V2x)}V_{(x)}^2 + 4A_{(3Vx)}V_{(x)}^3 + A_{(4V)}V_{(x)}^4 - 3A_{(2V)}V_{(2x)}^2
\end{aligned}
\tag{7}
$$

An extension to multicomponent mixtures is possible in principle, but rather tedious, as it involves the evaluation of Jacobian determinants with more than two rows.

Eq. (4) can be used to prove that $S_{(2x)} = 0$ or $V_{(2x)} = 0$ indicate pressure extrema or temperature extrema, respectively, along critical curves. But, to our knowledge, Eqs. (4) and (6) have never been used to compute critical curves of fluid mixtures.

2.3. Isochoric Thermodynamics. "Isochoric thermodynamics" is a formalism that does not use mole fractions x_i as input variables, but amounts of substance in a fixed volume (hence the name) or molar concentrations ρ_i. The conversion between the classical mole fraction-based formalism and the new one is achieved by [16]

$$
\begin{aligned}
\rho_i \equiv \frac{x_i}{V_\mathrm{m}} = x_i\rho \quad \rho = \sum_{i=1}^{N}\rho_i \\
V_\mathrm{m} = \frac{1}{\rho} \quad x_i = \frac{\rho_i}{\rho}\ .
\end{aligned}
\tag{8}
$$

It is important to note that the concentrations ρ_i are mutually independent, whereas one of the mole fractions depends on the other ones.

The main thermodynamic potential of isochoric thermodynamics is the Helmholtz energy density Ψ, which is defined as

$$
\Psi \equiv \frac{A}{V} = A_\mathrm{m}\rho\ . \tag{9}
$$

The natural variables of Ψ are the concentrations ρ_i and the temperature T. The chemical potentials are given by

$$
\begin{aligned}
\mu_i &= \left(\frac{\partial\Psi}{\partial\rho_i}\right)_{T,\rho_{j\neq i}}, i = 1,\ldots,N \\
\text{vector notation: } \vec{\mu} &= \nabla\Psi\ ,
\end{aligned}
\tag{10}
$$

and the pressure by

$$p = -\Psi + \sum_{i=1}^{N} \left(\frac{\partial \Psi}{\partial \rho_i} \right)_{T,\rho_{j\neq i}} \rho_i \tag{11}$$
$$\text{vector notation: } p = -\Psi + \vec{\mu} \cdot \vec{\rho} \,,$$

where the "$\cdot$" operator indicates the inner vector product (dot product). These equations can also be written in terms of residual properties, which may be more convenient for practical applications [15].

In the formalism of isochoric thermodynamics, the phase stability of a fluid mixture depends on the local curvature of the $\Psi(\vec{\rho})$ surface, which is described by the Hessian matrix,

$$\mathbf{\Psi} = \mathrm{H}\,\Psi = \begin{pmatrix} \Psi_{11} & \dots & \Psi_{1N} \\ \vdots & \ddots & \vdots \\ \Psi_{N1} & \dots & \Psi_{NN} \end{pmatrix} \quad \text{with } \Psi_{ij} \equiv \left(\frac{\partial^2 \Psi}{\partial \rho_i \, \partial \rho_j} \right)_T \,. \tag{12}$$

$\mathbf{\Psi}$ is symmetric and, for a stable phase, positive definite. This means that any perturbation with an arbitrary concentration vector $\vec{c}$ increases Ψ by $\vec{c} \cdot \mathbf{\Psi} \vec{c} > 0$, i.e., the $\Psi(\vec{\rho})$ surface has a convex curvature. This is equivalent to stating that all eigenvalues of the matrix are positive.

At a critical point, the $\Psi(\vec{\rho})$ surface becomes "flat" in one direction. This means that the lowermost eigenvalue, λ_1, vanishes; the associated eigenvector $\vec{u}_1$ points along this direction. The first criterion for a critical point is therefore

$$\lambda_1 = 0 \,. \tag{13}$$

We note in passing that this criterion is *not* equivalent to $\det \mathbf{\Psi} = 0$. A fluid mixture loses its stability when the *lowermost* eigenvalue of $\mathbf{\Psi}$ becomes zero. The determinant of $\mathbf{\Psi}$, on the other hand, is equal to the product of all eigenvalues and therefore becomes zero if *any* eigenvalue vanishes. Using determinants in the criteria of critical states introduces unwanted roots that impede the rootfinder algorithms and need to be sorted out afterwards.

The second criterion of a critical state is that, along a path aligned with $\vec{u}_1$, λ_1 must have a local minimum; the λ_1 curve "touches" zero at the critical point. This criterion can be expressed as

$$\frac{\mathrm{d}\lambda_1(\vec{\rho})}{\mathrm{d}\sigma_1} = 0 \quad \text{with } \vec{\rho} = \vec{\rho}^{\,\mathrm{c}} + \sigma_1 \vec{u}_1 \,, \tag{14}$$

where σ_1 is a scalar. This way of formulating the critical conditions goes back to Quiñones-Cisneros [5], who used eigenvalues and -vectors of the Hessian matrix of the molar Helmholtz energy as a function of molar volume and mole fractions, $A_\mathrm{m}(V_\mathrm{m}, \vec{x})$. In contrast to this, Eqs. (13) and (14) are based on the Hessian matrix of $\Psi(\vec{\rho})$, because using a uniform set of variables ρ_i avoids issues with having variables of different dimensionality in a matrix.

The approach is illustrated in Fig. 1, which shows isolines of the lowermost eigenvalue λ_1 for a simple binary system ((methane + propane) at 250 K, computed using the Peng–Robinson equation of state [17] with parameters given elsewhere [15]). Isolines with a negative λ_1 are unstable; the critical points lies on the $\lambda_1 = 0$ isoline. Here the eigenvector $\vec{u}_1$ is aligned with this isoline. The other eigenvector (for a binary mixture there are merely two), $\vec{u}_2$, is perpendicular to $\vec{u}_1$. The $\lambda_1(\vec{\rho})$ surface. Any displacement from the critical point in the direction of $\vec{u}_1$ increases λ_1. There critical point is therefore a local minimum according to Eq. (14).

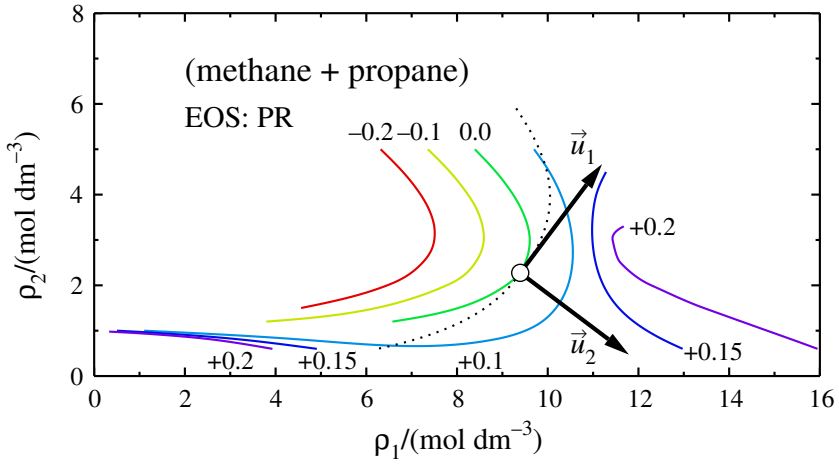


Fig. 1: Concentration–concentration contour diagram of the $\lambda_1(\vec{\rho})$ surface of the (methane + propane) system at 250 K (model: Peng–Robin equaiton of state). —— λ_1 isoline (parameter: λ_1/(J dm^6 mol^{-1})), ∘ binary critical point, ····· eigenvector path through the critical point.

In analogy with the previous section, Eq. (13) can be turned into a differential equation by forming the total differential and postulating that it is zero along a critical curve,

$$\mathrm{d}\lambda_1 = \sum_{i=1}^{N} \left(\frac{\partial \lambda_1}{\partial \rho_i}\right)_{\rho_{j\neq i},T} \mathrm{d}\rho_i + \left(\frac{\partial \lambda_1}{\partial T}\right)_{\vec{\rho}} \mathrm{d}T = (\nabla\lambda_1)\cdot \mathrm{d}\vec{\rho} + \left(\frac{\partial \lambda_1}{\partial T}\right)_{\vec{\rho}} \mathrm{d}T \stackrel{!}{=} 0\,. \tag{15}$$

Division by $\mathrm{d}T$ then yields the first differential equation,

$$(\nabla\lambda_1)\cdot \left.\frac{\mathrm{d}\vec{\rho}}{\mathrm{d}T}\right|_{\mathrm{c}} = -\left(\frac{\partial \lambda_1}{\partial T}\right)_{\vec{\rho}}\,. \tag{16}$$

In order to obtain a differential equation from Eq. (14) we need to recall that, along the path indicated by the eigenvector, the variation of the concentration vector is $\mathrm{d}\vec{\rho} = \vec{u}_1\,\mathrm{d}\sigma$. Substituting this into Eq. (15), assuming constant temperature, gives

$$\mathrm{d}\lambda_1 = (\nabla\lambda_1)\cdot\vec{u}_1\,\mathrm{d}\sigma_1\,. \tag{17}$$

Division by $\mathrm{d}\sigma_1$ and using Eq. (14) then leads to

$$(\nabla\lambda_1)\cdot\vec{u}_1 = 0\,, \tag{18}$$

i.e., the concentration gradient of λ_1 and the critical eigenvector are orthogonal. Forming the total differential of this expression yields

$$\sum_i \left(\frac{\partial(\nabla\lambda_1)\cdot\vec{u}_1}{\partial\rho_i}\right)\mathrm{d}\rho_i + \left(\frac{\partial(\nabla\lambda_1)\cdot\vec{u}_1}{\partial T}\right)_{\vec{\rho}}\mathrm{d}T = 0\,; \tag{19}$$

dividing this by $\mathrm{d}T$ gives another differential equation for the critical curve,

$$\nabla\left((\nabla\lambda_1)\cdot\vec{u}_1\right)\cdot\left.\frac{\mathrm{d}\vec{\rho}}{\mathrm{d}T}\right|_{\mathrm{c}} = -\left(\frac{\partial(\nabla\lambda_1)\cdot\vec{u}_1}{\partial T}\right)_{\vec{\rho}} . \tag{20}$$

Eqs. (16) and (20) are two linear equations for the concentration derivatives along the critical curve, $\mathrm{d}\vec{\rho}/\mathrm{d}T|_{\mathrm{c}}$. The complete system of linear equations is

$$\begin{pmatrix} \left(\frac{\partial\lambda_1}{\partial\rho_1}\right) & \left(\frac{\partial\lambda_1}{\partial\rho_2}\right) & \cdots & \left(\frac{\partial\lambda_1}{\partial\rho_N}\right) \\ \left(\frac{\partial(\nabla\lambda_1)\cdot\vec{u}_1}{\partial\rho_1}\right) & \left(\frac{\partial(\nabla\lambda_1)\cdot\vec{u}_1}{\partial\rho_2}\right) & \cdots & \left(\frac{\partial(\nabla\lambda_1)\cdot\vec{u}_1}{\partial\rho_1}\right) \\ a_{11} & a_{12} & \cdots & a_{1N} \\ \vdots & \vdots & \ddots & \vdots \end{pmatrix} \left.\frac{\mathrm{d}\vec{\rho}}{\mathrm{d}T}\right|_{\mathrm{c}} = \begin{pmatrix} \left(\frac{\partial\lambda_1}{\partial T}\right) \\ \left(\frac{\partial(\nabla\lambda_1)\cdot\vec{u}_1}{\partial T}\right) \\ 0 \\ \vdots \end{pmatrix} . \tag{21}$$

The a_{ij} are stoichiometric coefficients; they are needed for ternary and higher-order mixtures in order to specify which concentrations, mole fractions, or mole fractions ratios are to be kept constant. Their calculation has been described on a previous publication on the computation of vapour–liquid equilibria [18].

Derivatives Expressed with ρ_i Coordinates. There is more than one way to proceed from here. One can, for instance, use Eq. (20) directly. Then the computation of $\nabla((\nabla\lambda_1)\cdot\vec{u}_1)$ is best accomplished via the product rule,

$$\begin{gathered} \nabla((\nabla\lambda_1)\cdot\vec{u}_1) = (\mathrm{H}\,\lambda_1)\vec{u}_1 + (\nabla\otimes\vec{u}_1)\,\nabla\lambda_1 \\ \text{with } \nabla\otimes\vec{u}_1 = \begin{pmatrix} \left(\frac{\partial u_{1,1}}{\partial\rho_1}\right) & \cdots & \left(\frac{\partial u_{1,N}}{\partial\rho_1}\right) \\ \vdots & \ddots & \vdots \\ \left(\frac{\partial u_{1,1}}{\partial\rho_N}\right) & \cdots & \left(\frac{\partial u_{1,N}}{\partial\rho_N}\right) \end{pmatrix} \end{gathered} \tag{22}$$

where the $u_{1,i}$ are the elements of the critical eigenvector. This equation reduces the number of second-order derivatives, which is advantageous when the differentiations are performed numerically.

The latter is not difficult, although Eq. (22) might look complicated: One varies the concentrations by increments $\pm\delta\rho_i$, each time determines the lowermost eigenvalue and the associated eigenvector, and uses the results to approximately compute $\nabla\lambda_1$, $\mathrm{H}\,\lambda_1$, and $\nabla\otimes\vec{u}_1$.

Our test program based on this approach obtained critical curves correctly, but proved rather slow.

Derivatives Expressed with Transformed Coordinates. An alternative formulation of the criteria for critical points uses transformed concentration coordinates which are aligned with the eigenvectors of the system,

$$\vec{\rho} = \vec{\rho}^{\,\mathrm{c}} + \sum_{k=1}^{N}\vec{u}_k\sigma_k = \vec{\rho}^{\,\mathrm{c}} + \boldsymbol{U}\vec{\sigma}\,, \tag{23}$$

where $\boldsymbol{U}$ represents a matrix whose row vectors are the eigenvectors $\vec{u}_k, k = 1,\ldots,N$. These are shown in Fig. 1 for a binary system; here σ_1 and σ_2 represent displacements from the critical along $\vec{u}_1$ and $\vec{u}_2$, respectively.

As $\boldsymbol{U}$ is orthogonal (all eigenvectors are mutually orthogonal), $\vec{\sigma}$ can be obtained from

$$\vec{\sigma} = \boldsymbol{U}^{\mathrm{T}}(\vec{\rho} - \vec{\rho}^{\mathrm{c}}) \; . \tag{24}$$

The Hessian of Ψ is diagonal in $\vec{\sigma}$ space, hence we can write

$$\boldsymbol{\Psi} = \begin{pmatrix} \frac{\partial^2\Psi}{\partial\sigma_1^2} & \cdots & \frac{\partial^2\Psi}{\partial\sigma_1\,\partial\sigma_N} \\ \vdots & \ddots & \vdots \\ \frac{\partial^2\Psi}{\partial\sigma_N\,\partial\sigma_1} & \cdots & \frac{\partial^2\Psi}{\partial\sigma_N^2} \end{pmatrix} = \begin{pmatrix} \lambda_1 & & 0 \\ & \ddots & \\ 0 & & \lambda_N \end{pmatrix} \; , \tag{25}$$

so that the eigenvalues are given by

$$\lambda_k = \left(\frac{\partial^2\Psi}{\partial\sigma_k^2} \right) \; . \tag{26}$$

This equation does not make the eigenvalue calculation superfluous, because the eigenvectors need to be known in order to define the σ_k. But it simplifies the calculation of derivatives of λ_1. For example, Eq. (18) thus becomes

$$(\nabla_{\vec{\sigma}}\lambda_1)\cdot\vec{u}_1 = \begin{pmatrix} \left(\frac{\partial\lambda_1}{\partial\sigma_1}\right) \\ \vdots \\ \left(\frac{\partial\lambda_N}{\partial\sigma_N}\right) \end{pmatrix} \cdot \begin{pmatrix} 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} = \left(\frac{\partial\lambda_1}{\partial\sigma_1}\right)_{\sigma_{j>1}} = 0 \; , \tag{27}$$

where "$\nabla_{\vec{\sigma}}$" denotes a gradient operator in $\vec{\sigma}$ coordinates. This equation is equivalent to Eq. (14). The algebraic equations for a critical point of an N-component mixture can therefore be summarized as

$$\boxed{\begin{aligned} \lambda_1 &= \left(\frac{\partial^2\Psi}{\partial\sigma_1^2}\right)_{\sigma_{j>1}} = 0 \\ \left(\frac{\partial\lambda_1}{\partial\sigma_1}\right)_{\sigma_{j>1}} &= \left(\frac{\partial^3\Psi}{\partial\sigma_1^3}\right)_{\sigma_{j>1}} = 0 \; . \end{aligned}} \tag{28}$$

This representation of the critical conditions is particularly useful for machine computation, as it does not require the evaluation of complicated determinants.

Similar expressions, albeit in $V_{\mathrm{m}}x$ coordinates, were obtained by Quiñones-Cisneros. Eq. (28) closely resembles Eq. (1), but is far more powerful, as it can be applied to multicomponent mixtures.

Expanding the first of these equations into a series in analogy to Eq. (15) yields

$$(\nabla_{\vec{\sigma}}\lambda_1)\cdot\left.\frac{\mathrm{d}\vec{\sigma}}{\mathrm{d}T}\right|_{\mathrm{c}} = -\left(\frac{\partial\lambda_1}{\partial T}\right)_{\vec{\sigma}} \; . \tag{29}$$

In order to obtain an expression for the $\vec{\rho}$ derivatives, we insert the matrix product $\boldsymbol{U}\boldsymbol{U}^T = 1$ on the left-hand side and observe the commutation rule for matrix–vector products, $\vec{v}\boldsymbol{M} = \boldsymbol{M}^{\mathrm{T}}\vec{v}$, which leads to

$$(\nabla_{\vec{\sigma}}\lambda_1)\boldsymbol{U}\boldsymbol{U}^{\mathrm{T}}\left.\frac{\mathrm{d}\vec{\sigma}}{\mathrm{d}T}\right|_{\mathrm{c}} = \left[(\nabla_{\vec{\sigma}}\lambda_1)\boldsymbol{U}\right]\left.\frac{\mathrm{d}\vec{\rho}}{\mathrm{d}T}\right|_{\mathrm{c}} = \left[\boldsymbol{U}^{\mathrm{T}}(\nabla_{\vec{\sigma}}\lambda_1)\right]\left.\frac{\mathrm{d}\vec{\rho}}{\mathrm{d}T}\right|_{\mathrm{c}} \; . \tag{30}$$

Hence the first differential equation for critical curves, Eq. (16), becomes

$$\boxed{\left[\boldsymbol{U}^{\mathrm{T}}(\nabla_{\vec{\sigma}}\lambda_1)\right]\left.\frac{\mathrm{d}\vec{\rho}}{\mathrm{d}T}\right|_{\mathrm{c}} = -\frac{\partial'}{\partial T}\left(\frac{\partial^2\Psi}{\partial\sigma_1^2}\right)} \tag{31}$$

with

$$\nabla_{\vec{\sigma}}\lambda_1 = \begin{pmatrix} \frac{\partial'}{\partial\sigma_1}\left(\frac{\partial^2\Psi}{\partial\sigma_1^2}\right) \\ \vdots \\ \frac{\partial'}{\partial\sigma_N}\left(\frac{\partial^2\Psi}{\partial\sigma_1^2}\right) \end{pmatrix} . \tag{32}$$

The second critical condition in Eq. (28) can be treated in a similar fashion; the result is

$$\boxed{\nabla'_{\vec{\sigma}}\left(\nabla_{\vec{\sigma}}\lambda_1\right)\cdot\vec{u}_1)\cdot\left.\frac{\mathrm{d}\vec{\sigma}}{\mathrm{d}T}\right|_{\mathrm{c}} = \left[\boldsymbol{U}^{\mathrm{T}}\left(\nabla'_{\vec{\sigma}}((\nabla_{\vec{\sigma}}\lambda_1)\cdot\vec{u}_1)\right)\right]\cdot\left.\frac{\mathrm{d}\vec{\rho}}{\mathrm{d}T}\right|_{\mathrm{c}} = -\frac{\partial'}{\partial T}\left(\frac{\partial^3\Psi}{\partial\sigma_1^3}\right)} \tag{33}$$

with

$$\nabla'_{\vec{\sigma}}\left((\nabla_{\vec{\sigma}}\lambda_1)\cdot\vec{u}_1\right) = \begin{pmatrix} \frac{\partial'}{\partial\sigma_1}\left(\frac{\partial^3\Psi}{\partial\sigma_1^3}\right) \\ \vdots \\ \frac{\partial'}{\partial\sigma_N}\left(\frac{\partial^3\Psi}{\partial\sigma_1^3}\right) \end{pmatrix} . \tag{34}$$

Here the primes on the outer differentiation operators serve as a reminder that the eigenvectors are not kept constant, but change during these differentiations. In contrast to Eq. (20), Eq. (34) does not require the computation of the full Hessian matrix of λ_1, which saves computer time.

2.4. Local Stability. As mentioned above, the relevant thermodynamic potential of a stable thermodynamic state has a convex shape. For example, the molar Gibbs energy of a binary mixture at constant pressure and temperature must have a positive curvature with respect to the mole fraction, $G_{(2x)} > 0$ (cf. Eq. (1)). This curvature vanishes at a critical point. For overall stability it is then necessary that the next non-vanishing even-order derivative is positive, i.e., $G_{(4x)} > 0$.

This condition can also be formulated with the Helmholtz energy:

$$\begin{aligned} G_{(4x)} =& A_{(4x)} + 4A_{(3xV)}V_{(x)} + 6A_{(2x2V)}V_{(x)}^2 + 4A_{(x3V)}V_{(x)}^3 + A_{(4V)}V_{(x)}^4 \\ &- \frac{3}{A_{(2V)}}\left[A_{(2xV)} + 2A_{(x2V)}V_{(x)} + A_{(3V)}V_{(x)}^2\right]^2 \end{aligned} \tag{35}$$

An extension to multicomponent mixtures by using determinants in analogy to Eq. (2) is possible. But the resulting expressions are extremely complicated even for simple equations of state, and their evaluation tends to suffer from rounding errors.

In contrast to this, the formalism of isochoric thermodynamics simply yields

$$\left.\frac{\mathrm{d}^2\lambda_1}{\mathrm{d}\sigma_1^2}\right|_{\mathrm{c}} > 0 \tag{36}$$

for the condition of stability. The pitfall, however, is that the derivative is not simply computed along the critical eigenvector; instead, the differentiation must account for the rotation of the $\vec{\sigma}$-based coordinate system as one moves away from the critical point. Practically, this means that one has to follow a (curved) eigenvector (see Fig. 1) path through the critical point. A possible procedure for this is presented in Section 3.6..

3. PRACTICAL COMPUTATION

3.1. Outline of the Algorithm. The central part of the computer program is a loop in which the marching variable—the temperature or another suitable variable—is incremented. For each value, the integration subroutine computes the derivatives of thermodynamic properties along the critical curve (cf. Algorithm 1) and uses them to determine the next point of the curve.

1. Before the calculation of a critical curve for a multicomponent mixture ($N \geq 3$) is started, it is necessary to specify which concentrations, mole fractions, or mole fraction ratios are to be kept constant during the calculation, and then to set the constraint coefficients in Eq. (21) accordingly. This step is not needed for binary mixtures.
2. In order to solve the system of ordinary differential equations (ODE)—Eq. (31), Eq. (33), and eventually constraint equations—as an initial-value problem, it is necessary to specify an initial critical state $(\vec{\rho}, T)$. For vapor–liquid critical curves, the critical points of the pure components are convenient initial states, as discussed in Section 3.4. (also see Section 4.4. for an initialization strategy for ternary mixtures); the initialization of a liquid–liquid critical curve is demonstrated in Section 4.3..
3. The step function of the integration subroutine (cf. Algorithm 1) must first determine the eigenvector matrix $\boldsymbol{U}$ at the given state point, then a set of higher-order derivatives as required by Eqs. (31–34). Ways to compute these derivatives are described in Section 3.2.. If one of the concentrations at the local state is zero, some derivatives become undefined and need to be replaced by limiting values as explained in Section 3.4..

   For the solution of the eigenvalue problem a method is needed that finds (at least) the lowermost eigenvalue and all eigenvectors. For binary mixtures it is advisable to solve the secular equation; for multicomponent mixtures we used Jacobi's rotation method [19]. Section 3.3. contains more recommendations on this topic.
4. The step function is invoked by the integration subroutine to calculate a critical point at the next temperature. The examples presented in this work were computed with the adaptive integration method of Cash and Karp [20].
5. As in the course of the integration round-off errors tend to accumulate, the result should be corrected ("polished") after each integration step by running a rootfinder, for instance a Newton–Raphson subroutine, on the system of algebraic equations Eq. (28). As the integration usually yields results that are correct to five or more decimal digits, one or two iterations steps are sufficient in most cases, and convergence is practically certain.

3.2. The Calculation of Derivatives. The calculation of third- and fourth-order derivatives of $\Psi(\vec{\rho}, T)$ might seem a formidable task, but can be accomplished very efficiently by numerical differentiation. It is advisable to split the Helmholtz energy density into an ideal and a residual part, and to apply the numerical differentiation to the residual part only, making use of directional derivatives (cf. Appendix B.). In particular,

we have

$$\left(\frac{\partial^2\Psi}{\partial\sigma_1^2}\right) = +RT\sum_{i=1}^{N}\frac{u_{1,i}^2}{\rho_i} + \frac{\partial^2\Psi^{\mathrm{r}}}{\partial\vec{u}_1^2}$$
$$\left(\frac{\partial^3\Psi}{\partial\sigma_1^3}\right) = -RT\sum_{i=1}^{N}\frac{u_{1,i}^3}{\rho_i^2} + \frac{\partial^3\Psi^{\mathrm{r}}}{\partial\vec{u}_1^3} \tag{37}$$
$$\left(\frac{\partial^4\Psi}{\partial\sigma_1^4}\right) = +RT\sum_{i=1}^{N}\frac{2u_{1,i}^4}{\rho_i^3} + \frac{\partial^4\Psi^{\mathrm{r}}}{\partial\vec{u}_1^4}\ .$$

There are efficient and robust ways to compute the "pure derivatives" $\partial^k\Psi/\partial\vec{u}_1^k$ very precisely by complex differentiation (see Appendices A.2. and A.3.). The algorithm outlined in the previous section also needs some mixed derivatives, but these can be obtained from pure derivatives with simple first-order difference formulas (cf. Appendix A.1.), namely

$$\frac{\partial'}{\partial\sigma_j}\left(\frac{\partial^k\Psi^{\mathrm{r}}(\vec{\rho},T)}{\partial\sigma_1^k}\right) \approx \left[\left(\frac{\partial^k\Psi^{\mathrm{r}}(\vec{\rho}^{\,+},T)}{\partial\sigma_1^k}\right) - \left(\frac{\partial^k\Psi^{\mathrm{r}}(\vec{\rho}-,T)}{\partial\sigma_1^k}\right)\right]\frac{1}{2\,\delta\rho}\ ,\ k=2,3$$
$$\text{with}\quad \vec{\rho}^{\,-} = \begin{pmatrix}\rho_1\\ \vdots\\ \rho_j-\delta\rho\\ \vdots\\ \rho_N\end{pmatrix}\quad\text{and}\quad \vec{\rho}^{\,+} = \begin{pmatrix}\rho_1\\ \vdots\\ \rho_j+\delta\rho\\ \vdots\\ \rho_N\end{pmatrix} \tag{38}$$

and

$$\frac{\partial'}{\partial T}\left(\frac{\partial^k\Psi^{\mathrm{r}}(\vec{\rho},T)}{\partial\sigma_1^k}\right) \approx \left[\left(\frac{\partial^k\Psi^{\mathrm{r}}(\vec{\rho},T+\delta T)}{\partial\sigma_1^k}\right) - \left(\frac{\partial^k\Psi^{\mathrm{r}}(\vec{\rho},T-\delta T)}{\partial\sigma_1^k}\right)\right]\frac{1}{2\delta T}\ ,\ k=2,3\ , \tag{39}$$

where $\delta\rho$ and δT are properly chosen positive increments.

The derivatives on the right-hand side must be computed using their proper $\vec{\sigma}$ coordinates. It is therefore necessary to compute the eigenvectors at the states $(\vec{\rho}^{\,\pm},T)$ or $(\vec{\rho},T\pm\delta T)$, respectively; it would not be correct to use the critical eigenvector here (cf. Algorithm 1).

Evidently, a centered finite-difference formula cannot give very precise results, but as the results of the ODE integration are improved by an algebraic solver, small deviations can be tolerated.

3.3. Eigenvalues and Eigenvectors. For the determination of critical points and particularly for the determination of the transformed concentrations it is necessary to compute all eigenvectors and at least the lowermost eigenvalue of given $\mathbf{\Psi}$ matrices. A general-purpose method that was found to work reliably is Jacobi's rotation method [19]. Within finite-difference differentiations (as described in the previous section), only the lowermost eigenvalue and the associated eigenvector are required, and estimates are available for both. Then a computationally less expensive calculation method can be used, e.g., Rayleigh quotient iteration [21].

For small numbers of components, particularly for binary mixtures, it is advantageous to solve the secular equation,

$$\begin{vmatrix}\Psi_{11}-\lambda_k & \Psi_{12}\\ \Psi_{12} & \Psi_{22}-\lambda_k\end{vmatrix} = 0 \tag{40}$$

and then to obtain the eigenvectors from the system of linear equations

$$\begin{aligned}(\Psi_{11}-\lambda_k)u_{k,1}+\Psi_{12}u_{k,2}&=0\\ \Psi_{12}u_{k,1}+(\Psi_{22}-\lambda_k)u_{k,2}&=0\ .\end{aligned} \tag{41}$$

Here one of the $u_{k,i}$ can be chosen freely ($\neq 0$).

A pitfall of all eigenvalue algorithms is that the direction of the eigenvectors is undefined, i.e., that both $\vec{u}_k$ and $-\vec{u}_k$ are solutions. When eigenvectors are calculated within numerical differentiations (Eqs. (38) and (39)), it is important to "align" the eigenvectors: If $\vec{u}_k$ is an eigenvector of $\mathbf{\Psi}(\vec{\rho},T)$ and $\vec{u}_k^+$ an eigenvector of $\mathbf{\Psi}(\vec{\rho}^+,T)$, it is necessary to ensure $\vec{u}_k\cdot\vec{u}_k^+>0$ (cf. Algorithm 1, function Eigen1).

3.4. Pure Fluids and Dilute Mixtures. If one or more concentrations approach zero, some of the equations in the previous sections cannot be evaluated straightforwardly anymore, because they contain powers of ρ_i in the denominators of the ideal-gas contributions. The problem occurs, for instance, when one wants to start the calculation of a critical curve at a pure-fluid critical point. It turns out, however, that the affected equations can be extended to cover the dilute case.

Derivatives of Ψ, Eq. (37). The critical eigenvector $\vec{u}_1$ is a solution of the eigenvalue equation

$$\begin{pmatrix}\Psi_{11} & \dots & \Psi_{1N}\\ \vdots & \ddots & \vdots\\ \Psi_{N1} & \dots & \Psi_{NN}\end{pmatrix}\vec{u}_1=\lambda_1\vec{u}_1 \quad \text{with } \lambda_1=0\ . \tag{42}$$

This is equivalent to demanding that $\vec{u}_1$ is orthogonal to all row vectors of $\mathbf{\Psi}$,

$$\vec{\Psi}_j\cdot\vec{u}_1=0,\ j=1,\dots,N\ . \tag{43}$$

If now one concentration, e.g., ρ_j, approaches zero, the corresponding main-diagonal element Ψ_{jj} from Eq. (42) diverges as RT/ρ_j. But the product $\vec{\Psi}_j\cdot\vec{u}_1$ can only remain finite if the vector element $u_{1,j}$ approaches zero at least as ρ_j^1. Hence the ideal gas terms in Eq. (37), which contain terms of the form $u_{1,i}^{n+1}/\rho_i^n$, do not diverge: The terms depending on the vanishing concentration safely approach zero. Consequently, the RT/ρ_i terms in Eq. (37) need to be computed only if $\rho_i>0$, but can be neglected otherwise.

Determining the Eigenvectors, Eqs. (31–33). If a concentration approaches zero, $\rho_j\to 0$, the corresponding main diagonal element of the Hessian of $\Psi(\vec{\rho})$ diverges, and then standard numerical methods for the determination of the eigenvalues and -vectors are likely to fail. This problem can be overcome in two steps:

1. First we consider a $(N-1)$-component mixture where component j has been omitted and all other concentrations are positive. We solve the eigenvalue problem for this mixture.

2. Then we construct the vectors and matrices for the full N-component system (We use here—rather loosely—the infinity symbol to indicate a diverging term):

$$\vec{\rho} = \begin{pmatrix} \rho_1 \\ \vdots \\ 0 \\ \vdots \\ \rho_N \end{pmatrix} \qquad \boldsymbol{\Psi} = \begin{pmatrix} \Psi_{11} & \dots & \Psi_{1j} & \dots & \Psi_{1N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Psi_{j1} & \dots & \infty & \dots & \Psi_{jN} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ \Psi_{N1} & \dots & \Psi_{Nj} & \dots & \Psi_{NN} \end{pmatrix} \tag{44}$$

$$\vec{\lambda} = \begin{pmatrix} \lambda_1 \\ \vdots \\ \lambda_{N-1} \\ \infty \end{pmatrix} \qquad \boldsymbol{U} = \begin{pmatrix} u_{1,1} & \dots & 0 & \dots & u_{1,N} \\ \vdots & \ddots & \vdots & \ddots & \vdots \\ u_{N-1,1} & \dots & 0 & \dots & u_{N-1,N} \\ 0 & \dots & 1 & \dots & 0 \end{pmatrix} .$$

Now all N eigenvectors are available, so that the σ coordinate system can be constructed. The diverging eigenvalue λ_N is of no concern, as only λ_1 is relevant.

If properly implemented, the algorithm described above never invokes any diverging functions.

Numerical Differentiation w.r.t. σ_i, Eq. (38). If one of the concentrations is zero, a numerical differentiation according to Eq. (38) can cause negative concentrations to appear. With some (not all!) equations of state, this can cause trouble. An alternative differentiation formula for dilute mixtures is then the forward differentiation equation

$$\frac{\partial'}{\partial\sigma_j}\left(\frac{\partial^k\Psi^{\mathrm{r}}(\vec{\rho},T)}{\partial\sigma_1^k}\right) \approx \left[4\left(\frac{\partial^k\Psi^{\mathrm{r}}(\vec{\rho}^{\,+},T)}{\partial\sigma_1^k}\right) - \left(\frac{\partial^k\Psi^{\mathrm{r}}(\vec{\rho}^{\,++},T)}{\partial\sigma_1^k}\right) - 3\left(\frac{\partial^k\Psi^{\mathrm{r}}(\vec{\rho},T)}{\partial\sigma_1^k}\right)\right]\frac{1}{2\,\delta\rho}$$
$$\text{with } \vec{\rho}^{\,++} = \begin{pmatrix} \rho_1 \\ \vdots \\ \rho_j + 2\,\delta\rho \\ \vdots \\ \rho_N \end{pmatrix} . \tag{45}$$

Here negative concentrations are safely avoided.

3.5. Parametric Marching. The differential equations presented above allow the calculation of critical curves as functions of temperature. Many mixtures, however, have critical curves with temperature extrema. Therefore the temperature is not always a good "marching variable". The same is true for pressure, the density, or mole fractions; there are even mixtures where, in a px projection, the critical curve exhibits a loop.

For mixtures with non-monotonic critical curves, parametric marching is an option. The technique has been explained elsewhere [22], therefore we merely summarize the working equations here.

The idea is to define a new integration variable, t, which is related to the arc length of the critical curve and therefore increases monotonically along the curve. The set of differential equations describing the critical

curve then becomes

$$\left.\frac{\mathrm{d}\vec{\rho}}{\mathrm{d}t}\right|_{\mathrm{c}} = \left.\frac{\mathrm{d}\vec{\rho}}{\mathrm{d}T}\right|_{\mathrm{c}} \left.\frac{\mathrm{d}T}{\mathrm{d}t}\right|_{\mathrm{c}}$$
$$\text{with } \left.\frac{\mathrm{d}T}{\mathrm{d}t}\right|_{\mathrm{c}} = \pm\frac{1}{\left\| \left.\frac{\mathrm{d}\vec{\rho}}{\mathrm{d}T}\right|_{\mathrm{c}} \right\|}, \tag{46}$$

where $\mathrm{d}\vec{\rho}/\mathrm{d}T|_{\mathrm{c}}$ is obtained from Eq. (21); $\mathrm{d}T/\mathrm{d}t|_{\mathrm{c}}$ is reciprocally proportional to the Euclidean norm, the length of this vector. The sign must be chosen in such a way that this vector changes continuously along the critical curve: A possible criterion is that, if the critical curve is known at the location t_0 in the vicinity of t, the scalar product

$$\left.\frac{\mathrm{d}\vec{\rho}(t_0)}{\mathrm{d}t}\right|_{\mathrm{c}} \cdot \left.\frac{\mathrm{d}\vec{\rho}(t)}{\mathrm{d}t}\right|_{\mathrm{c}} > 0 \tag{47}$$

should be positive. This guarantees that the curve will not change its direction abruptly.

3.6. Local Stability. For the determination of the local stability of a critical state it is necessary move away from the critical point in the direction of the eigenvector $\vec{u}_1$. This direction changes during this movement; the eigenvector path is therefore usually curved.

It can be easily constructed, however, by using a Runge–Kutta four-point formula, as shown in Algorithm 2.

3.7. Program Code The method we propose in this work invokes a significant amount of unfamiliar mathematics and thermodynamics, so in order to educate the reader we have compiled an example in the Python language for tracing the critical curve of the (methane + propane) system. The example is reduced to its essence and does not include features like the parametric tracing required to address non-monotonic critical curves. It includes the use of the GERG-2004 model [23] (along with the pure-fluid EOS of that model), and the Peng-Robinson model [17] (analytic derivatives provided by Deiters [18]).

The program code is supplied as Supporting Information. The resulting critical curves are presented in Section 4.1..

4. APPLICATION

4.1. (Methane + Propane). The (methane + propane) system is a so-called simple fluid mixture—Class I according to the classification of van Konynenburg and Scott [1], 1^{P} according to the rational nomenclature of Bolz et al. [2]—with a single vapor–liquid critical curve connecting the critical points of the pure fluids, as shown in Fig. 2.

The diagrams show critical curves obtained with the GERG-2004 model [23] and the Peng–Robinson equaiton of state [17], demonstrating that both models yield similar critical curves, and that tracing from one pure fluid starting point is successful in constructing the critical line for a binary mixture. We note in passing that the GERG-2018 model [14] yields practically identical results. At the resolution of the diagram, differences between the vapour pressure curves obtained with the GERG-2004 and PR equations are not discernible.

The remarkable fact about this phase diagram is that it has been computed for the GERG-2004 model, a rather complicated fundamental equation comprising several equations for pure fluids, among them some

reference equations. Because of its structure, conventional algorithms for computing critical curves usually fail for this model. The reason for this is that the underlying pure-fluid reference equations exhibit unphysical behavior in their two-phase regions; a critical-curve algorithm scanning (T, ρ) space for zeros of the algebraic criteria Eq. (28)) is likely to invoke the GERG-2004 model for thermodynamic conditions at which it does not return meaningful results, thus confusing the algorithm.

In contrast to this, the method proposed here invokes the fundamental equation only in the vicinity of computable states, and therefore is safe to be used with the GERG models or similar ones. Bell and Jäger [13] used a similar approach, but for a fixed composition.

The calculation of critical curves by solving differential equations requires an initial state from which the integration can proceed. For vapor–liquid critical curves, as in this case, the critical points of the pure components are natural choices. It does not matter whether the integration is started at the critical point of methane or propane. As the other component has a zero concentration there, it is necessary to modify the calculation of the derivatives as explained in Section 3.4..

4.2. (Tetrafluoromethane + Butane). Fig. 3 shows the pT phase diagram of the (CF_4 + butane) system (CF_4 = tetrafluoromethane, R14). The diagram belongs to Class III (rational nomenclature: $1^{\mathrm{C}}1^{\mathrm{Z}}$. The calculations were made with the Peng–Robinson (PR) equation of state [17] and the usual Soave mixing rules. The parameters for this mixture can be found in a previous publication [22]. This systems exhibits two separate critical curves, and it is evidently necessary to start the ODE integration at both pure-component critical points. The phase diagram was computed with parametric marching.

The critical curve originating at the critical point of CF_4 first runs to higher temperatures and pressures, then passes through a cusp (a cusp in the pT projection only), and finally disappears at negative temperatures. The critical cusp marks the onset of unstability; the downward branch is thermodynamically unstable. There is a critical endpoint to the left of the cusp (not shown here), where the critical curve passes through an *llg* three-phase surface and becomes metastable.

4.3. (Methane + Tetrafluoromethane). This system belongs to Class II (rational nomenclature: 1^{P}l, which means that it exhibits an uninterrupted *lg* critical curve and, in addition, a separate *ll* critical curve at lower temperatures (Fig. 4). In reality, this curve terminates in an upper critical endpoint located between the vapor pressure curves (not shown here); the calculation can follow this curve beyond the endpoint into the metastable and, after a critical cusp, unstable region.

Class II systems pose a problem for the method proposed here, because ODE integrations started at the pure-component critical points yield only the *lg* critical curve, but not the *ll* critical curve. There is, however, a workaround, at least for equations of state with a well-defined behavior at high densities (e.g., equations containing a hard-core repulsion term, in particular cubic equations of state):

Binary fluid mixtures are equivalent—at least qualitatively—to ternary lattice gases, where the third component represents "holes", i.e., empty space [24]. A ternary lattice gas can have three boundary critical points, namely the critical points of the binary subsystems (component 1 + holes) in the absence of component 2, (component 2 + holes) in the absence of component 1, and (component 1 + component 2) in the absence of holes. The first two critical points are equivalent to the pure-fluid critical points, whereas the last one corresponds to the end of the *ll* critical curve. The situation is shown in Fig. 5: For a Class II system, one critical curve connects the two pure-component critical points, and the other (ll) curve runs from the third critical point (located on the diagonal boundary) to a critical cusp, where it becomes unstable; for a Class III system, one critical curve connects a pure-component critical point and the third critical point, and the other critical curve runs from the other pure-component critical point to the critical cusp.

The important property of this third critical point is that it is topologically equivalent to a pure-component critical point—and therefore just as easy to calculate (merely two unknown properties to determine, direction of the critical eigenvector fixed). We solve the critical conditions, Eq. (28) for x_1 and T while setting the total density ρ to the maximal value that the equation of state allows[1]. Then we start the ODE integration at this critical state to obtain the complete *ll* critical curve.

It should be noted that, for the (methane + CF_4) system, the composition along the *ll* critical curve is almost constant. A conventional search strategy where the mole fraction is used as "marching variable" is therefore likely to miss the *ll* critical curve.

4.4. (Methane + Propane + Decane). Fig. 6 contains the pT phase diagram of the binary mixture (methane + propane), this time computed with the Peng–Robinson equation of state. In addition, the diagram contains a critical curve of the ternary system (methane + propane + decane) for a constant decane mole fraction $x_3 = 0.0032$. This constraint is realized by setting the stoichiometric coefficients in Eq. (21) to $1 - x_3$ for decane or $-x_3$ otherwise, in this case $\vec{a}_1 = (-0.0032, -0.0032, 0.9978)$. It is interesting to observe that the presence of a very small amount of the heavy component causes a topological change of the *lg* critical curve: This curve passes through an upper critical endpoint (not shown in the diagram) and then becomes unstable—a behavior typical for Class V binary mixtures (rational nomenclature: 2^{P}. There is also a low-temperature critical curve, which is always unstable.

Initial states for the calculation of the ternary critical curves were found by computing pieces of the critical curves of the binary boundary systems (methane + decane) and (propane + decane) for decane mole fractions $x_3 = 0. . .0.0032$.

The initial slope of the stable ternary critical curve close to the propane critical point is positive. This detail would probably not be discovered by conventional search strategies, but is immediately evident if parametric marching is used.

5. CONCLUSION

We propose a method for computing critical curves of mixtures—binary as well as multicomponent—by solving *differential equations*. In contrast to conventional methods, where *algebraic equations* representing the critical conditions must be solved, the new method does not suffer from convergence problems.

This is particularly important when the thermodynamic model underlying the calculation is an engineering fundamental equation like the GERG-2018 equation [14, 23]. Conventional methods scanning the density space for critical points by means of checking the algebraic conditions of criticality are likely to fail, because they tend to invoke the fundamental equation at states at which it gives unphysical results. Integrating the differential equation, however, invokes the fundamental equation only at states for which it works well.

With conventional methods, which scan density–composition spaces to detect changes of sign of the algebraic conditions of criticality, it is usually necessary to sort the computed critical points before the critical curves become apparent. The method proposed here generates mixture critical points along the critical curves, thus avoiding the need for a sorting step.

The new method supports parametric marching and can therefore follow critical curves of arbitrary shape.

[1]For numerical reasons we use 90% of the covolume of cubic equations of state.

SYMBOLS

A	Helmholtz energy
G	Gibbs energy
$\Im$	imaginary part of a (multi)complex number
N	number of components of a mixture
p	pressure
R	gas constant
$\Re$	real part of a (multi)complex number
S	entropy
t	parametric-marching variable, arc length of the critical curve
T	temperature
$\vec{u}_k$	kth eigenvector, $\vec{u}_k = (u_{k,1}, \ldots, u_{k,N})$
$\boldsymbol{U}$	matrix of eigenvectors
V	volume
$\vec{x}$	vector of mole fractions, $\vec{x} = (x_1, \ldots, x_N)$
λ_k	kth eigenvalue; λ_1: lowermost eigenvalue
$\vec{\mu}$	vector of chemical potentials, $\vec{\mu} = (\mu_1, \ldots, \mu_N)$
$\vec{\rho}$	vector of molar concentrations, $\vec{\rho} = (\rho_1, \ldots, \rho_N)$
$\vec{\sigma}$	vector of transformed concentrations, $\vec{\sigma} = (\sigma_1, \ldots, \sigma_N)$
Ψ	Helmholtz energy density
$\boldsymbol{\Psi}$	Hessian matrix of Ψ, $\boldsymbol{\Psi} = (\{\Psi_{ij}\})$

Sub- and Superscripts

c	critical state
m	molar property

ASSOCIATED CONTENT

Supporting Information Python code for tracing the critical curve of the (methane + propane) system using the GERG-2004 or Peng–Robinson model; numerical results (CSV)

Python code for tracing the critical curve of the (tetrafluoromethane + butane) system using the GERG-2004 model

REFERENCES

(1) van Konynenburg, P. H.; Scott, R. L. Critical lines and phase equilibria in binary van der Waals mixtures. *Philos. Trans. R. Soc. A* **1980**, *298*, 495–540.

(2) Bolz, A.; Deiters, U. K.; Peters, C. J.; de Loos, Th. W. Nomenclature for phase diagrams with particular reference to vapour–liquid and liquid–liquid equilibria. *Pure Appl. Chem.* **1998**, *70*, 2233–2257.

(3) van Laar, J. J. On the shape of the plaitpoint curves for mixtures of normal substances (2nd communication). *Proc. K. Ned. Akad. Wet.* **1905**, *8*, 33–48.

(4) Heidemann, R. A.; Khalil, A. M. The calculation of critical points. *AIChE J.* **1980**, *26*, 769–779.

(5) Quiñones-Cisneros, S. E. Barotropic phenomena in complex phase behaviour. *Phys. Chem. Chem. Phys.* **2004**, *6*, 2307–2313.

(6) Deiters, U. K.; Kraska, Th. *High-Pressure Fluid Phase Equilibria—Phenomenology and Computation*, volume 2 of *Supercritical Fluid Science and Technology (E. Kiran, ed.)*. Elsevier, Amsterdam, 2012.

(7) Deiters, U.; Schneider, G. M. Fluid mixtures at high pressures. Computer calculations of the phase equilibria and the critical phenomena in fluid binary mixtures from the Redlich–Kwong equation of state. *Ber. Bunsenges. Phys. Chem.* **1976**, *80*, 1316–1321.

(8) Hicks, C. P.; Young, C. L. Theoretical prediction of phase behaviour at high temperatures and pressures for non-polar mixtures. *J. Chem. Soc. Faraday Trans. II* **1977**, *73*, 597–612.

(9) Mazur, V. A.; Boshkov, L. Z.; Fedorov, V. B. Phase equilibria in two-component Lennard-Jones systems (in Russian). *Dokl. Akad. Nauk SSSR* **1985**, *282*, 137–140.

(10) Boshkov, L. Z.; Mazur, V. A. Phase equilibria and critical lines of binary mixtures of Lennard-Jones molecules (in Russian). *Zh. Fiz. Khim.* **1986**, *60*, 29–33.

(11) van Pelt, A.; Deiters, U. K.; Peters, C. J.; de Swaan Arons, J. Global phase behaviour based on the simplified-perturbed-hard-chain equation of state. *J. Chem. Phys.* **1995**, *102*, 3361–3375.

(12) Sadus, R. J. *High Pressure Behaviour of Multicomponent Fluid Mixtures*. Elsevier, Amsterdam, 1992.

(13) Bell, I. H.; Jäger, A. Calculation of critical points from Helmholtz-energy-explicit mixture models. *Fluid Phase Equilib.* **2017**, *433*, 159–173.

(14) Kunz, O.; Wagner, W. The GERG-2008 wide-range equation of state for natural gases and other mixtures: an expansion of GERG-2004. *J. Chem. Eng. Data* **2012**, *57*, 3032–3091.

(15) Bell, I. H.; Deiters, U. K. On the construction of binary mixture $p - x$ and $T - x$ diagrams from isochoric thermodynamics. *AIChE J.* **2018**, *64*, 2745–2757.

(16) Quiñones-Cisneros, S. E.; Deiters, U. K. An efficient algorithm for the calculation of phase envelopes of fluid mixtures. *Fluid Phase Equilib.* **2012**, *329*, 22–31.

(17) Peng, D. Y.; Robinson, D. B. A new two-constant equation of state. *Ind. Eng. Chem. Fundam.* **1976**, *15*, 59–64.

(18) Deiters, U. K. Differential equations for the calculation of fluid phase equilibria. *Fluid Phase Equilib.* **2016**, *428*, 164–173.

(19) Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. *Numerical Recipes in C*. Cambridge University Press, Cambridge, 2002.

(20) Cash, J. R.; Karp, A. H. A variable-order Runge–Kutta method for initial-value problems with rapidly varying right-hand sides. *ACM Trans. Math. Software* **1990**, *16*, 201–222.

(21) Golub, G. H.; van Loan, Ch. F. *Matrix Computations*, 3rd edition. John Hopkins University Press, Baltimore, 1996.

(22) Deiters, U. K.; Bell, I. H. Calculations of phase envelopes of fluid mixtures through parametric marching. *AIChE J.* **2019**, *65*, e16730:1–13.

(23) Kunz, O.; Klimeck, R.; Wagner, W.; Jaeschke, M. *The GERG-2004 Wide-Range Reference Equation of State for Natural Gases*, volume 15 of *GERG (Groupe Européen de Recherches Gazières) Technical Monographs*. VDI-Verlag, Düsseldorf, 2007.

(24) Meijer, P. H. E.; Napiórkowski, M. The three-state lattice gas as model for binary gas–liquid systems. *J. Chem. Phys.* **1987**, *86*, 5771–5777.

(25) Leal, A. autodiff (automatic differentiation for C++). https://github.com/autodiff/autodiff, 2018–2019.

(26) Jordan-Engeln, G.; Reutter, F. *Numerische Mathematik für Ingenieure*. Bibliographisches Institut, Mannheim, 1985.

(27) Lyness, J. N.; Moler, C. B. Numerical differentiation of analytic functions. *SIAM J. Numer. Anal.* **1967**, *4*, 202–210.

(28) Squire, W.; Trapp, G. Using complex variables to estimate derivatives of real functions. *SIAM Rev.* **1998**, *40*, 110–112.

(29) Higham, N. Differentiation with(out) a difference. *SIAM News* **2018**, *51(5)*, 2.

(30) Lantoine, G.; Russell, R. P.; Dargent, Th. Using multicomplex variables for automatic computation of high-order derivatives. *ACM Trans. Math. Softw.* **2012**, *38*, no. 16.

(31) Verheyleweghen, A. Computation of higher-order derivatives using the multi-complex step method. Technical report, NTNU, 2014.

(32) Bell, I. H. mcx (multicomplex algebra library). https://github.com/ianhbell/mcx, 2019.

(33) https://en.wikipedia.org/wiki/Directional_derivative. visited 2018-02-21.

Algorithm 1: Calculation of concentration derivatives along a critical curve, $\mathrm{d}\vec{\rho}/\mathrm{d}T\big|_{\mathrm{c}}$.

function $\mathrm{ODE}(\vec{\rho}, T)$**:**

Input: vector of concentrations $\vec{\rho}$ and temperature T at a critical state
Result: vector of derivatives $\mathrm{d}\vec{\rho}/\mathrm{d}T\big|_{\mathrm{c}}$
vector $\vec{\lambda}, \vec{u}_1^{\pm}$ [eigenvalues, eigenvectors]
matrix $\boldsymbol{U}$ [matrix of eigenvectors]
$\mathrm{EigenProblemSolver}(\mathrm{H}\,\Psi(\vec{\rho}, T), \vec{\lambda}, \boldsymbol{U})$ [full eigen problem solver]
find smallest concentration, ρ_k
if $u_{1,k} < 0$ **then** $\vec{u}_1 := -\vec{u}_1$
vector g, h [$\nabla_{\vec{\sigma}}\lambda_1$ and $\nabla'_{\vec{\sigma}}((\nabla_{\vec{\sigma}}\lambda_1)\cdot\vec{u}_1)$, Eqs. (32), (34), (38)]
forall $j := 1, \ldots, n$ **do**
$\vec{\rho}^{+} := \vec{\rho},\ \rho_j^{+} := \rho_j^{+} + \delta\rho$
$\mathrm{Eigen1}(\vec{\rho}^{+}, T, \vec{u}_1, \lambda_1^{+}, \vec{u}_1^{+})$
$f^{+} := (\partial^3\Psi(\vec{\rho}^{+}, T)/\partial\sigma_1^3)$
$\vec{\rho}^{-} := \vec{\rho},\ \rho_j^{-} := \rho_j^{-} - \delta\rho$
if *all* $\rho_i \geq 0$ **then**
$\mathrm{Eigen1}(\vec{\rho}^{-}, T, \vec{u}_1, \lambda_1^{-}, \vec{u}_1^{-})$
$f^{-} := (\partial^3\Psi(\vec{\rho}^{-}, T)/\partial\sigma_1^3)$
$g_j := (\lambda_1^{+} - \lambda_1^{-})/(2\,\delta\rho)$, $h_j := (f^{+} - f^{-})/(2\,\delta\rho)$
else
$f^{-} := (\partial^3\Psi(\vec{\rho}, T)/\partial\sigma_1^3)$
$\vec{\rho}^{++} := \vec{\rho},\ \rho_j^{++} := \rho_j^{++} + 2\,\delta\rho$
$\mathrm{Eigen1}(\vec{\rho}^{++}, T, \vec{u}_1, \lambda_1^{++}, \vec{u}_1^{++})$
$f^{++} := (\partial^3\Psi(\vec{\rho}^{++}, T)/\partial\sigma_1^3)$
$g_j := (4\lambda_1^{+} - \lambda_1^{++})/(2\,\delta\rho)$, $h_j := (4f_1^{+} - f_1^{++} - 3f_1^{-})/(2\,\delta\rho)$
end
end
vector $\vec{b}$ [$\partial/\partial T(\partial^k\Psi/\partial\sigma_1^k)$, Eq. (31), (39)]
$T^{+} := T + \delta T$
$\mathrm{Eigen1}(\vec{\rho}^{+}, T, \vec{u}_1, \lambda_1^{+}, \vec{u}_1^{+})$
$f^{+} := (\partial^3\Psi(\vec{\rho}, T^{+})/\partial\sigma_1^3)$
$T^{-} := T - \delta T$
$\mathrm{Eigen1}(\vec{\rho}^{-}, T, \vec{u}_1, \lambda_1^{-}, \vec{u}_1^{-})$
$f^{-} := (\partial^3\Psi(\vec{\rho}, T^{-})/\partial\sigma_1^3)$
$b_1 := -(\lambda_1^{+} - \lambda_1^{-})/(2\,\delta T)$, $b_2 := -(f^{+} - f^{-})/(2\,\delta T)$, $b_{i>2} := 0$
matrix $\boldsymbol{a}$, $\vec{a}_1 := \boldsymbol{U}^{\mathrm{T}}\vec{g},\ \vec{a}_2 := \boldsymbol{U}^{\mathrm{T}}\vec{h}, \vec{a}_{i>2} = \mathrm{constraints}$ [Eq. (21)]
vector $\vec{y} := \mathrm{LinearSystemSolver}(\boldsymbol{a}, \vec{b})$
return $\vec{y}$

end

function $\mathrm{Eigen1}(\vec{\rho}, T, \vec{d}, \lambda_1, \vec{u}_1)$**:**

Input: vector of concentrations $\vec{\rho}$, temperature T, alignment vector $\vec{d}$
Result: lowermost eigenvalue λ_1, aligned associated eigenvector $\vec{u}_1$
matrix $\boldsymbol{\Psi} = \mathrm{H}\,\Psi(\vec{\rho}, T)$ [Hessian of Ψ]
$\mathrm{EigenProblemSolver1}(\boldsymbol{\Psi}, \lambda_1, \vec{u}_1)$ [lowermost eigenvalue and associated eigenvector]
if $\vec{u}_1 \cdot \vec{d} < 0$ **then** $\vec{u}_1 := -\vec{u}_1$

end

Algorithm 2: Test for local stability of a critical state.

function $\mathrm{LocalStability}(\vec{\rho}, T)$**:**

Input: vector of concentrations $\vec{\rho}$, temperature T

Result: 0 = stable/metastable, 1 = unstable

vector $\vec{\rho}^{+}, \vec{\rho}^{-}, \vec{\lambda}, \vec{\lambda}^{+}, \vec{\lambda}^{-}$

matrix $\boldsymbol{U}, \boldsymbol{U}^{+}, \boldsymbol{U}^{-}$

$\mathrm{EigenProblemSolver}(\mathrm{H}\,\Psi(\vec{\rho}, T), \vec{\lambda}, \boldsymbol{U})$

$\vec{\rho}^{+} := \mathrm{EigenVectorPath}(\vec{\rho}, T, \vec{u}_1, \delta\rho)$

$\mathrm{EigenProblemSolver}(\mathrm{H}\,\Psi(\vec{\rho}^{+}, T), \vec{\lambda}^{+}, \boldsymbol{U}^{+})$

$\vec{\rho}^{-} := \mathrm{EigenVectorPath}(\vec{\rho}, T, \vec{u}_1, -\delta\rho)$

$\mathrm{EigenProblemSolver}(\mathrm{H}\,\Psi(\vec{\rho}^{-}, T), \vec{\lambda}^{-}, \boldsymbol{U}^{-})$

if $\lambda_1^{+} < \lambda_1$ *or* $\lambda_1^{-} < \lambda_1$ **then** return 1

else return 0

end

function $\mathrm{EigenVectorPath}(\vec{\rho}, T, \vec{u}_1, \delta\rho)$**:**

Input: vector of concentrations $\vec{\rho}$, temperature T, eigenvector $\vec{u}_1$ at a critical state, increment $\delta\rho$

Result: new vector of concentration $\vec{\rho}$ after moving by $\delta\rho$ along the eigenvector path

vector $\vec{\lambda}, \vec{\rho}^{\pm}, \vec{d}$

$\vec{d} := \frac{1}{6}\vec{u}_1$

$\vec{\rho}^{\pm} := \vec{\rho} + \delta\rho\,\vec{u}_1$

$\mathrm{Eigen1}(\vec{\rho}^{\pm}, T, \vec{u}_1, \lambda_1, \vec{u}_1^{\pm})$ [get eigenvalue and -vector at new location]

$\vec{d} := \vec{d} + \frac{2}{6}\vec{u}_1^{\pm}$

$\vec{\rho}^{\pm} := \vec{\rho} + \delta\rho\,\vec{u}_1^{\pm}$

$\mathrm{Eigen1}(\vec{\rho}^{\pm}, T, \vec{u}_1, \lambda_1, \vec{u}_1^{\pm})$

$\vec{d} := \vec{d} + \frac{2}{6}\vec{u}_1^{\pm}$

$\vec{\rho}^{\pm} := \vec{\rho} + 2\,\delta\rho\,\vec{u}_1^{\pm}$

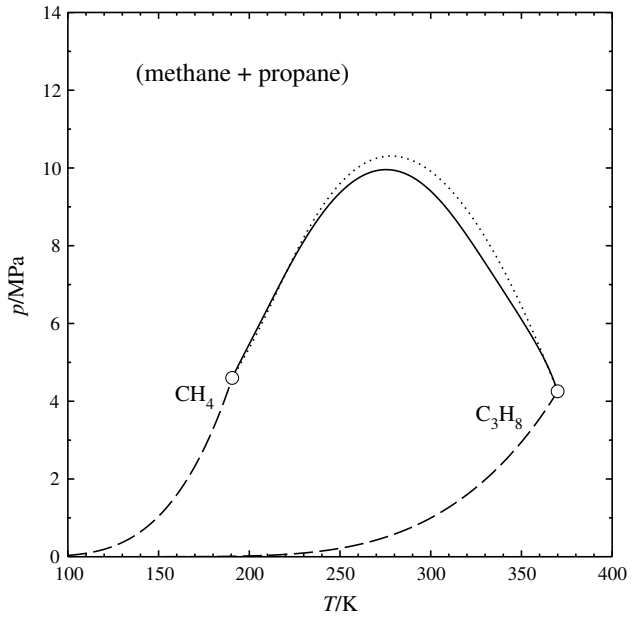
$\mathrm{Eigen1}(\vec{\rho}^{\pm}, T, \vec{u}_1, \lambda_1, \vec{u}_1^{\pm})$

$\vec{d} := \vec{d} + \frac{1}{6}\vec{u}_1^{\pm}$

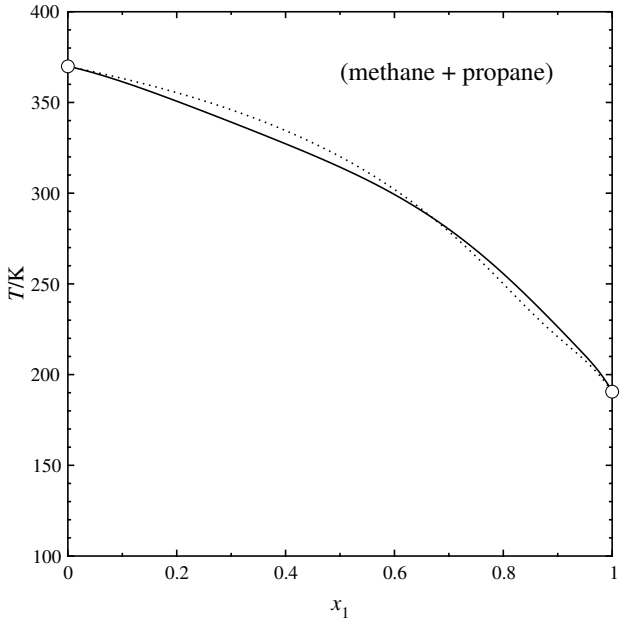
$\vec{\rho}^{\pm} := \vec{\rho} + 2\,\delta\rho\,\vec{u}_1$

return $\vec{\rho}^{\pm}$

end



(a) pressure vs. temperature projection



(b) temperature vs. methane mole fraction projection

Fig. 2: Phase diagram of the (methane + propane) system. —— critical curve (GERG-2004), ····· critical curve (Peng–Robinson), – – vapor pressure curve, ○ pure-component critical point.

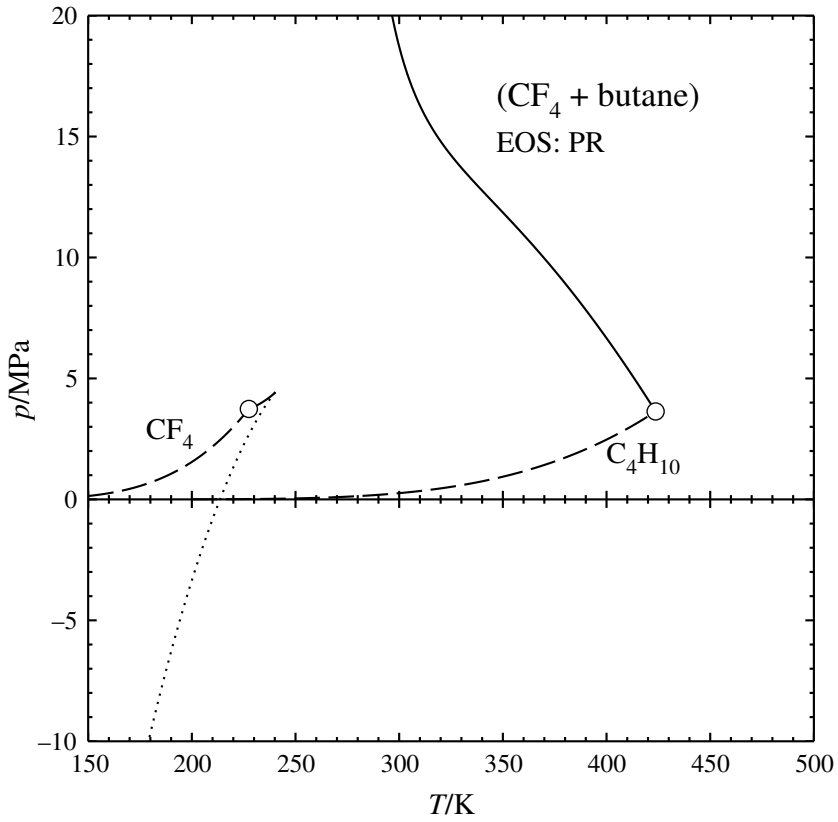


Fig. 3: pT phase diagram of the (tetrafluoromethane + butane) system, computed with the Peng–Robinson equation of state. − − vapor pressure curve, —— stable critical curve, ····· unstable critical curve, ○ pure-component critical point.

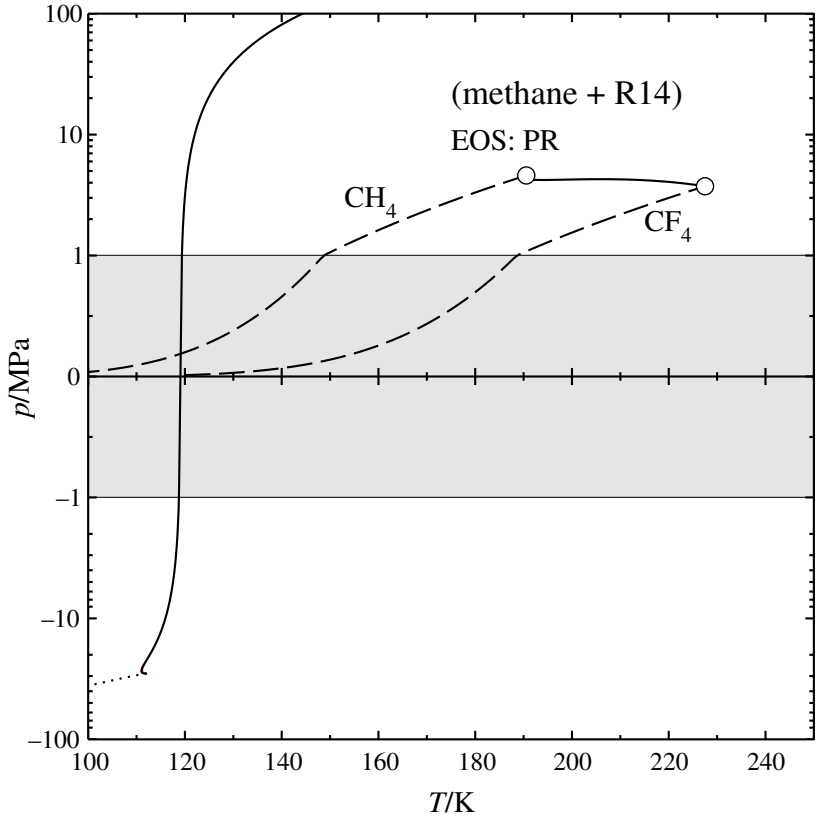
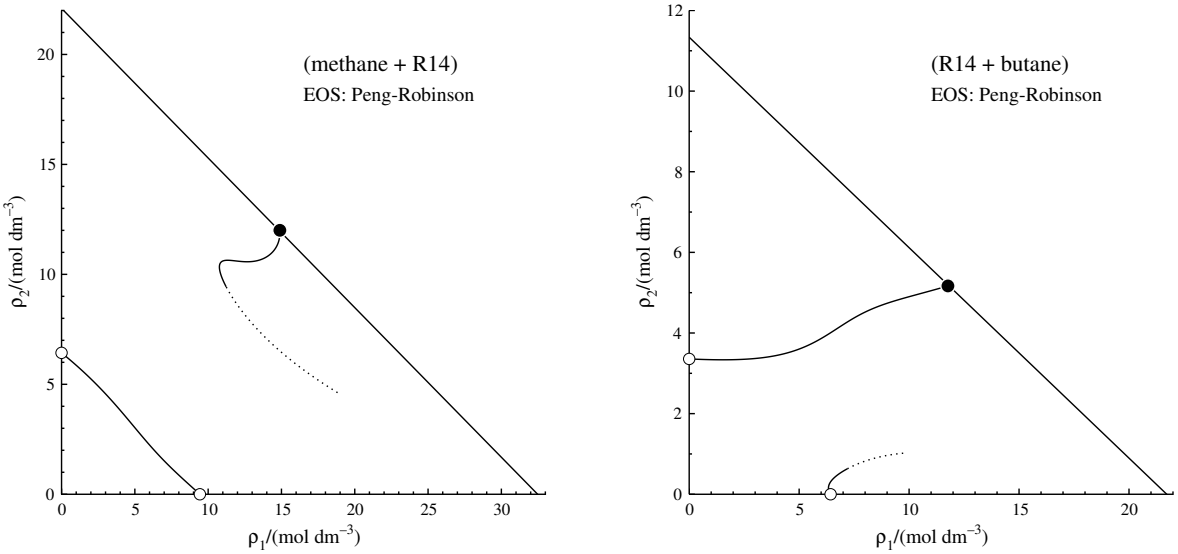


Fig. 4: pT phase diagram of the (methane + tetrafluoromethane) system, computed with the Peng–Robinson equation of state. – – vapor pressure curve, —— stable critical curve, ····· unstable critical curve, ○ pure-component critical point. The ordinate uses "symmetric logarithmic scaling" with a linear scaling in the range $-1 \ldots +1$ MPa.



(a) (methane + tetrafluoromethane)

(b) (tetrafluoromethane + butane)

Fig. 5: Meijer diagrams (concentration–concentration plots) showing the critical curves of a Class II and a Class III system, calculated with the Peng–Robinson equation of state. —— stable, ····· unstable critical curve, ○ pure-component critical point, • compressed mixture critical point. The diagonal boundaries represent maximal fluid densities based on 115% of the covolumes.

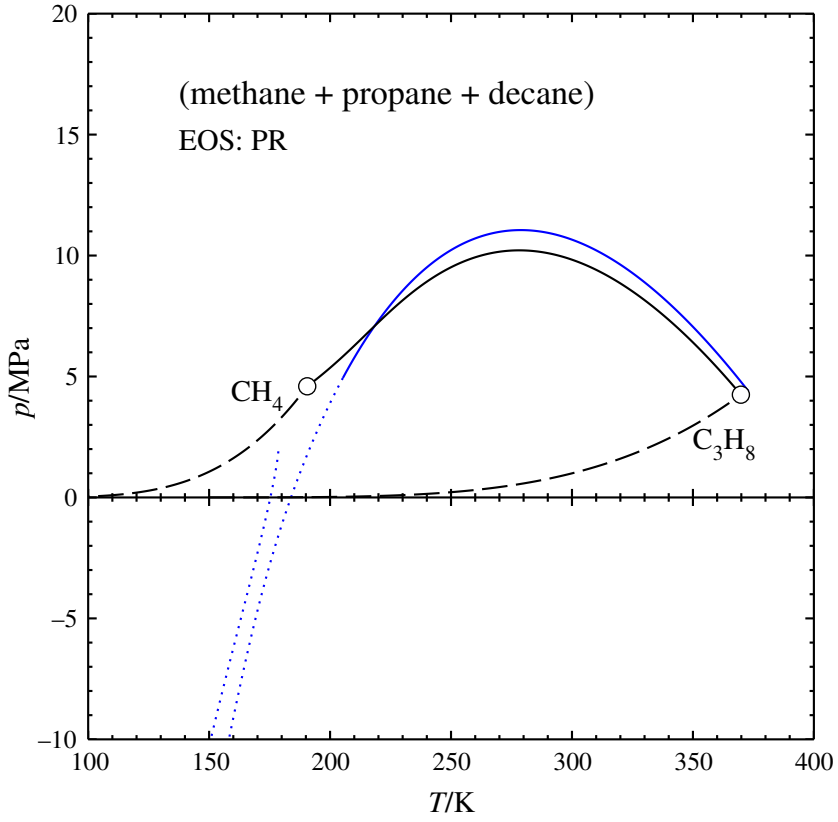


Fig. 6: Critical curves of the ternary system (methane + propane + decane), calculated with the Peng–Robinson equation of state. —— stable critical curve (black: binary system (methane + propane), blue: ternary system, $x_3 = 0.0032$), ····· unstable critical curve, – – vapor pressure curve, ○ pure-component critical point.

A. NUMERICAL DIFFERENTIATION

The calculation of derivatives—often of higher order—of functions is a recurring task in thermodynamics. This can—and usually should—be done by analytical differentiation ("calculus"). But as most functions, with the exception of polynomials, tend to get more complicated upon differentiation, analytical differentiation is not always a practical option.

Of course analytical differentiation even of complicated functions can be performed by means of symbolic-algebra programs. Many of these, however, are expensive and not available to everybody. Furthermore, they may not be applicable if the function to be differentiated connot be written explicitly, for instance because it contains iterations.

In this context implicit analytic differentiation methods, like the *autodiff* software [25], should also be mentioned. The latter is restricted to C++, however, at least at present.

It is therefore advantageous to consider numerical differentiation methods.

A.1. Finite-Difference Methods. These methods assume that the function to be differentiated, $f(x)$, can be expanded into a Taylor series at the location x. Then the value of the function at $x + h$ is

$$\begin{aligned} f(x+h) &= f(x) + f'(x)h + \frac{1}{2}f''(x)h^2 + \frac{1}{6}f'''(x)h^3 + \dots \\ &= \sum_{k=0}^{\infty} \frac{1}{k!} f^{(k)}(x) h^k \ , \end{aligned} \tag{48}$$

where the primes indicate the order of the derivative. When the difference of the function values at the locations $x + h$ and $x - h$ is calculated, the terms containing even powers of h cancel:

$$f(x+h) - f(x-h) = 2f'(x)h + \frac{1}{3}f'''(x)h^3 + \dots \ . \tag{49}$$

Rearrangement gives the first-order derivative,

$$f'(x) = \frac{1}{2h}\left(f(x+h) - f(x-h)\right) + O(h^2) \ . \tag{50}$$

Adding instead of subtracting the function values at $x \pm h$ yields

$$f(x+h) + f(x-h) = 2f(x) + f''(x)h^2 + \dots \tag{51}$$

and, after rearrangement, a formula for the second-order derivative,

$$f''(x) = \frac{1}{h^2}\left(f(x+h) - 2f(x) + f(x-h)\right) + O(h^2) \ . \tag{52}$$

These are already very useful results: The equations give good approximations for the first- and second-order derivatives, and their errors decreases with the square of the size of the increment h. Unfortunately, however, it is not possible to make h very small, for then the difference of the function values becomes unprecise because of the cancellation of digits. It is therefore necessary to weigh the cancellation error against the series truncation error $O(h^2)$. With "double precision" arithmetics, as it is presently defined by the standard IEEE 754, an increment size of $h \approx 10^{-5}x$ is usually a good compromise.

An improvement can be achieved by computing additional function values and to use these to eliminate more terms of the Taylor series. For example, the difference of the function values at the locations $x \pm 2h$ is

$$f(x+2h) - f(x-2h) = 4f'(x)h + \frac{8}{3}f'''(x)h^3 + \dots \ . \tag{53}$$

Combining this equation with Eq. (49) makes the cancellation of the third-order term possible,

$$8\left(f(x+h) - f(x-h)\right) - \left(f(x+2h) - f(x-2h)\right) = 12f'(x) + \dots \ , \tag{54}$$

which leads to

$$f'(x) = \frac{1}{12}\left(-f(x+2h) + 8f(x+h) - 8f(x-h) + f(x-2h)\right) + O(h^4) \tag{55}$$

as a better approximation.

Romberg proposed an iterative scheme for obtaining derivatives of arbitrary order [26]. The method requires the evaluation of the function at the locations $x \pm \alpha h$, $x \pm \alpha^2 h$, etc., where α is an expansion factor usually chosen between 1.4 and 2.

Ridders' method combines Romberg's scheme with a self-checking feature that terminates the iteration when no further reduction of the numerical uncertainty can be achieved [19].

An inherent disadvantage of all finite-difference methods for numerical differentiation is the cancellation of digits during the computation of the differences. The problem gets worse with increasing order of the differentiation. Romberg's or Ridders' methods alleviate the problem and can yield good results for first- and perhaps even second-order derivatives, but become increasingly unprecise for higher orders, unless the user employs a computing precision better than "double precision".

A.2. Complex Integration. The numerical problems of the finite-difference methods can be completely avoided by using methods based on complex-number algebra. Their basis is Cauchy's integral theorem, which relates the value of a function $f(z)$ or any of its to derivatives at the location z_0 to a path integral,

$$\frac{\mathrm{d}f^{(n)}(a)}{\mathrm{d}z^n} = \frac{n!}{2\pi}\oint_C \frac{f(z+z_0)}{z^n}\,\mathrm{d}z \ , \tag{56}$$

where C denotes a closed path in the complex plane that encloses z_0. The simplest choice is a circle with the radius r centered at $z = z_0$. Then Cauchy's formula can be written as

$$\frac{\mathrm{d}f^{(n)}(z_0)}{\mathrm{d}z^n} = \frac{n!}{2\pi}\int_0^{2\pi} \frac{f(z+z_0)}{z^n}\,\mathrm{d}\phi \quad \text{with} \quad z \equiv re^{i\phi} \ . \tag{57}$$

The function $f(z)$ must evidently take a complex argument and return a complex value. But this does not constitute a practical problem, as modern computer languages like C++ can handle complex algebra. Particular for C++, a complex version of an existing subroutine for computing, e.g., the residual Helmholtz energy density $\Psi^{\mathrm{r}}(\vec{\rho}, T)$ can be provided by merely providing a template declaration.

The integration is usually accomplished numerically by means of the trapezium rule. This is usually known as a rather inferior method, but for periodic functions it is very efficient: With increasing number of evaluation nodes N, the error of the integral decreases faster than any power of $2\pi/N$.

The advantage of using an *integral* formula to obtain derivatives is that numerical integration—quite in contrast to numerical differentiation by means of finite-difference formulas—does not suffer from cancellation errors. If the radius of the integration path is chosen large enough (so large that $f(z+z_0)$ shows a significant variation), derivatives can be computed practically with machine precision.

Even if we extend thermodynamic functions like the Helmholtz energy density $\Psi(\vec{\rho}, T)$ or $\Psi(\vec{\sigma}, T)$ to complex arguments and values, we must demand that these functions have real values and real derivatives when evaluated at real locations. For such functions the method of Lyness and Moler [27] is an interesting option. This method constructs a series of approximations $y^{(m)}$ for the nth-order derivative,

$$y^{(m)} = y^{(m-1)} + \frac{n!}{r^n} M_m \left[-\Re\left\{f(z_0)\right\} + \frac{1}{h} \sum_{k=0}^{mn-1} \Re\left\{ f\left(z_0 + r e^{2\pi h k i} \right) \right\} \right], \; m = 1, 2, \ldots \quad \text{with } y^{(0)} = 0 \text{ and } h = \frac{1}{mn} \; . \tag{58}$$

The M_m are so-called Möbius numbers, which are defined as

$$M_m = \begin{cases} +1 & \text{if } m = 1 \text{ or if } m \text{ is a product of different prime numbers} \\ -1 & \text{if } m \text{ is a prime number} \\ 0 & \text{otherwise} \end{cases} \tag{59}$$
$$= \{+1, -1, -1, 0, -1, +1, -1, 0, 0, +1, -1, 0, -1, +1, +1, 0, -1, 0, -1, 0, \ldots\} \; .$$

Convergence is usually quite rapid; the calculation can safely be terminated when the difference of two successive approximations falls below a predefined threshold.

A.3. Finite Differences Using (Multi)Complex Algebra. Even when extended to complex algebra, the functions needed for this work have the property that they return real values for real arguments. For such functions, the algorithm of Squire and Trapp [28, 29] is a remarkably simple and efficient way to compute first-order derivatives:

$$f'(x) = \frac{1}{h} \Im\left\{f(x + hi)\right\} \quad \text{with} \quad h \approx 10^{-100} \tag{60}$$

This is, in principle, a finite-difference method. But as the function argument is incremented along the imaginary axis and $\Im\left\{f(x)\right\} = 0$ by definition, there is no loss of precision even if a very small increment is used.

It is advisable, however, to test this algorithm on the computer on which it is to be used, as it depends on the correct handling of limiting values by the complex-function library.

The method of Squire and Trapp can be extended to higher-order derivatives—including mixed derivatives—by using multicomplex algebra. Just as a complex number comprises two real numbers ($z = x + iy$), a level-2 multicomplex number (also called bicomplex number) comprises two complex numbers,

$$z^{(2)} = z_0^{(1)} + i^{(2)} z_1^{(1)} \; , \tag{61}$$

where the $z_{0,1}^{(1)}$ are complex numbers and $i^{(2)}$ is the complex unit of the second level. Similarly, a level-3 multicomplex number comprises two level-2 complex numbers and the imaginary unit of the third level, and so on. Complex numbers can be regarded as level-1 multicomplex numbers, and $i \equiv i^{(1)}$. Evidently, an nth level multicomplex number comprises 2^n real components.

An nth order derivative is then obtained from

$$\frac{\mathrm{d}^n f(x)}{\mathrm{d}x^n} = \frac{1}{h^n} \Im^n \left\{ f(x + i^{(1)} h + i^{(2)} h + \ldots + i^{(n)} h) \right\} \; , \tag{62}$$

where $\Im^n$ is an operator that returns the last, "most imaginary" real component of an nth-level multicomplex number [30, 31].

Without giving a formal proof we wish to point out that multicomplex differentiation can be used to obtain mixed derivatives of functions of more than one variable, e.g.,

$$\left(\frac{\partial^2 f(x,y)}{\partial x\,\partial y}\right) = \frac{1}{h^2}\Im^2\left\{f(x+i^{(1)}h, y+i^{(2)}h)\right\}\ . \tag{63}$$

For the readers' convenience we have made Python and C++ libraries for multicomplex differentiation available [32].

B. DIRECTIONAL DERIVATIVES

B.1. Definitions. This article makes use of "directional derivatives", i.e. derivatives of thermodynamic functions—mostly the Helmholtz energy density— along a path specified by a direction vector. Let $f(\vec{x})$ denote a (scalar) function of a vector $\vec{x} = (x_1, \ldots, x_N)$, and $\vec{u} = (u_1, \ldots, u_N)$ a direction vector. Then the directional derivative of $f(\vec{x})$ is defined by [33]

$$\frac{\partial f(\vec{x})}{\partial \vec{u}} \equiv \lim_{\xi \to 0} \frac{1}{\xi} \left(f(\vec{x} + \xi \vec{u}) - f(\vec{x}) \right) \ . \tag{64}$$

$\vec{u}$ does not have to be a unit vector[2]; we merely demand that its length is not zero. ξ is a scalar diplacement parameter.

The total differential of f is

$$\mathrm{d}f = \sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i} \right)_{x_{j \neq i}} \mathrm{d}x_i \ . \tag{65}$$

If we now set

$$\mathrm{d}x = \vec{u} \, \mathrm{d}\xi \tag{66}$$

the total differential of f becomes

$$\mathrm{d}f = \sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i} \right)_{x_{j \neq i}} u_i \, \mathrm{d}\xi \ , \tag{67}$$

and then division by $\mathrm{d}\xi$ yields the directional derivative

$$\frac{\partial f}{\partial \vec{u}} = \sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i} \right)_{x_{j \neq i}} u_i = \vec{u} \cdot \nabla f \ . \tag{68}$$

If $\vec{u}$ is a unit vector aligned with one of the coordinate axes, the directional derivative coincides with the associated partial derivative,

$$\vec{u} = \begin{pmatrix} 0 \\ \vdots \\ u_k = 1 \\ \vdots \\ 0 \end{pmatrix} \quad \Rightarrow \quad \frac{\partial f}{\partial \vec{u}} = \left(\frac{\partial f}{\partial x_k} \right)_{x_{i \neq k}} \ , \tag{69}$$

and an all-1 vector yields

$$\vec{u} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \quad \Rightarrow \quad \frac{\partial f}{\partial \vec{u}} = \sum_{i=1}^{N} \left(\frac{\partial f}{\partial x_i} \right)_{x_{j \neq i}} \ . \tag{70}$$

[2]The mathematical literature is not uniform about this.

In the particular case that there is an complete *orthonormal* set of direction vectors $\vec{u}_k, k = 1, \ldots, N$, so that a variation along one vector does not imply a variation along any of the others, the directional derivatives can be regarded as partial derivatives of the $\vec{u}$ space,

$$\frac{\partial f(\vec{x})}{\partial \vec{u}_k} = \left(\frac{\partial f(\vec{\xi})}{\partial \xi_k} \right)_{\xi_{i \neq k}} , \; k = 1, \ldots, N \; . \tag{71}$$

The ξ_i are coordinates of the $\vec{u}$ space.

B.2. Directional Derivatives of Ψ. According to Eq. (68), the directional derivative along an arbitrary direction vector $\vec{u}$ at constant temperature, using a concentration displacement $\mathrm{d}\vec{\rho} = \vec{u}\,\mathrm{d}\sigma$, is

$$\frac{\partial \Psi}{\partial \vec{u}} = \sum_{i=1}^{N} \mu_i u_i = \vec{\mu} \cdot \vec{u} \; . \tag{72}$$

If $\vec{u}$ is a unit vector of the $\vec{\rho}$ space, the directional derivative is the associated chemical potential (cf. Eq. (69)),

$$\vec{u} = \begin{pmatrix} 0 \\ \vdots \\ u_k = 1 \\ \vdots \\ 0 \end{pmatrix} \quad \Rightarrow \quad \frac{\partial \Psi}{\partial \vec{u}} = \mu_k \; . \tag{73}$$

According to Eq. (70), setting all elements of $\vec{u}$ to 1 gives

$$\vec{u} = \begin{pmatrix} 1 \\ \vdots \\ 1 \end{pmatrix} \quad \Rightarrow \quad \frac{\partial \Psi}{\partial \vec{u}} = \sum_{i=1}^{N} \mu_i = p + \Psi \; , \tag{74}$$

which provides a useful way to obtain the pressure from Ψ, which requires merely one differentiation instead of N as in Eq. (11).

We now use the kth eigenvector of the Hessian matrix $\mathbf{\Psi}$ as directional vector, remembering at the same time that the Helmholtz energy density Ψ of a fluid mixture can be written as the sum of an intrinsic part (which contains thermochemical data which are not relevant for this work), an ideal-mixing term, and a residual term,

$$\Psi = \Psi^{\ominus} + RT \sum_{i=1}^{N} \rho_i \ln \rho_i + \Psi^{\mathrm{r}} \; . \tag{75}$$

Then Eq. (72) becomes

$$\frac{\partial \Psi}{\partial \vec{u}_k} = \vec{\mu}^{\mathrm{r}} \cdot \vec{u}_k + RT \sum_{i=1}^{N} \left(\ln \rho_i + 1 \right) u_{k,i} \; . \tag{76}$$

As the eigenvectors $\vec{u}_k$ form an orthonormal basis, the path derivative can be regarded as a partial derivative with respect to σ_k, the displacement in $\vec{u}_k$-direction, and we may write

$$\left(\frac{\partial\Psi}{\partial\sigma_k}\right)_{\sigma_{j\neq k}} = \frac{\partial\Psi}{\partial\vec{u}_k} = \vec{\mu}^{\mathrm{r}}\cdot\vec{u}_k + RT\sum_{i=1}^{N}\left(\ln\rho_i + 1\right)u_{k,i}\ . \tag{77}$$

Next we form the total differential of this derivative,

$$\mathrm{d}\left(\frac{\partial\Psi}{\partial\sigma_k}\right) = \sum_{i=1}^{N}\left(\frac{\partial^2\Psi}{\partial\sigma_k\,\partial\rho_i}\right)\mathrm{d}\rho_i = \sum_{i=1}^{N}\left(\frac{\partial^2\Psi}{\partial\sigma_k\,\partial\rho_i}\right)u_{k,i}\mathrm{d}\sigma_k\ . \tag{78}$$

After division by $\mathrm{d}\sigma_k$ as well as splitting Ψ into ideal and residual parts we obtain the second-order directional derivative,

$$\left(\frac{\partial^2\Psi}{\partial\sigma_k^2}\right) = \frac{\partial(\partial\Psi/\partial\vec{u}_k)}{\partial\vec{u}_k} = \sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial\mu_j}{\partial\rho_i}\right)u_{k,j} = \underbrace{\sum_{i=1}^{N}\sum_{j=1}^{N}\left(\frac{\partial\mu_j^{\mathrm{r}}}{\partial\rho_i}\right)u_{k,j}}_{\left(\partial^2\Psi^{\mathrm{r}}/\partial\sigma_k^2\right)} + RT\sum_{i=1}^{N}\frac{u_{k,i}^2}{\rho_i}\ . \tag{79}$$

Repeating the directional differentiation along $\vec{u}_k$ gives expressions for the third- and fourth-order derivatives,

$$\left(\frac{\partial^3\Psi}{\partial\sigma_k^3}\right) = \left(\frac{\partial^3\Psi^{\mathrm{r}}}{\partial\sigma_k^3}\right) - RT\sum_{i=1}^{N}\frac{u_{k,i}^3}{\rho_i^2} \tag{80}$$

and

$$\left(\frac{\partial^4\Psi}{\partial\sigma_4^3}\right) = \left(\frac{\partial^4\Psi^{\mathrm{r}}}{\partial\sigma_k^4}\right) + RT\sum_{i=1}^{N}\frac{2u_{k,i}^4}{\rho_i^3}\ . \tag{81}$$

B.3. Mixed Derivatives. Directional derivatives can used to obtain mixed higher-order derivatives from pure derivatives. Forming the total differential of Eq. (68) yields

$$\mathrm{d}\frac{\partial f}{\partial\vec{u}} = \sum_{j=1}^{N}\frac{\partial}{\partial x_j}\left(\sum_{i=1}^{N}\left(\frac{\partial f}{\partial x_i}\right)u_i\right)\mathrm{d}x_j\ . \tag{82}$$

Using Eq. (66) again and dividing by $\mathrm{d}\xi$ then gives

$$\frac{\partial^2 f}{\partial\vec{u}^2} = \sum_{j=1}^{N}\sum_{i=1}^{N}\left(\frac{\partial^2 f}{\partial x_i\,\partial x_j}\right)u_i u_j\ . \tag{83}$$

For simplicity we now assume $N = 2$, but the following equations can easily be generalized to arbitary values of N. We can then set

$$\begin{aligned}
\vec{u} &= \begin{pmatrix}1\\0\end{pmatrix} \quad\Rightarrow\quad f_{10} = \frac{\partial f}{\partial\vec{u}} = \left(\frac{\partial^2 f}{\partial x_1^2}\right)\\
\vec{u} &= \begin{pmatrix}0\\1\end{pmatrix} \quad\Rightarrow\quad f_{01} = \frac{\partial f}{\partial\vec{u}} = \left(\frac{\partial^2 f}{\partial x_2^2}\right)\\
\vec{u} &= \begin{pmatrix}1\\1\end{pmatrix} \quad\Rightarrow\quad f_{11} = \frac{\partial f}{\partial\vec{u}} = \left(\frac{\partial^2 f}{\partial x_1^2}\right) + 2\left(\frac{\partial^2 f}{\partial x_1\,\partial x_2}\right) + \left(\frac{\partial^2 f}{\partial x_2^2}\right)\ .
\end{aligned} \tag{84}$$

Consequently

$$\left(\frac{\partial^2 f}{\partial x_1 \, \partial x_2}\right) = \frac{1}{2}\left(f_{11} - f_{10} - f_{01}\right) \tag{85}$$

is an expression for the mixed derivative.

It should be noted that Eq. (85) is not a finite-difference formula like the equations in Appendix A.1. and therefore does not necessarily suffer from cancellation errors.

This way to calculate mixed derivatives is advantageous if complex integration methods are used (Appendix A.2.), for the generalization of such methods to multidimensional functions is rather complicated.