

Matrix Algebra and Matrix Differentiation Rules Applied to Excess Gibbs Energy Models

Charles R. A. Abreu

Department of Chemical Engineering
Universidade Federal do Rio de Janeiro
Rio de Janeiro, RJ 21949-900, Brazil

School of Chemical Engineering
State University of Campinas
Campinas, SP

E-mail: abreu@eq.ufrj.br

1. Introduction

Very often, chemical engineers make use of excess molar Gibbs energy (g^E) models to calculate thermodynamic properties and phase equilibrium conditions of mixtures. A g^E model expresses how intermolecular interactions affect the behavior of a real mixture when compared to an ideal mixture of the same components. Common calculations involve activity coefficients or excess enthalpies, which originate from first-order derivatives of g^E with respect to mole fractions or temperature, respectively. Other applications, such as phase stability analysis, even require higher-order derivatives.

Although multibody effects take place in real mixtures, it is often assumed that only pairwise contributions are important for the excess Gibbs energy. Thus, model parameters can be determined by fitting binary mixture data and then used to calculate properties of multicomponent mixtures. This simplification has been indispensable to the engineering practice of thermodynamics. As a mathematical consequence, g^E models generally contain only variables with one index (denoting individual components) or two indices (denoting pairs of components), which makes them amenable to matrix notation.

The benefits of applying matrix algebra to engineering problems have been discussed in the literature since the 1940's. {AMUNDSON1946, ACRIVOS1955, AMUNDSON1966, KEVREKIDIS1995, VARMA1997, RAMKRISHNA2004} Here, we highlight two aspects of the computer implementation of models expressed in matrix notation. First, as modern programming environments (e.g. Matlab, Scilab, Python, etc) are able to directly handle matrix operations, a single command-line can replace many lines which normally entail nested loops and references to individual array entries. From our experience, this significantly saves programming and debugging efforts by resulting in cleaner codes and reducing the chances of mistakes. Second, and more importantly, algorithmic aspects are detached from the model implementation. The back-end algorithms used to evaluate the matrix expressions can be extracted from some special library, optimized for a specific computer architecture. This can enable non-experts to take full advantage of modern machines with many parallel computing cores.

Despite the aforementioned advantages, it is not usual to find g^E models and their derivatives written in matrix notation. This is the main motivation of the present paper. Instead of merely writing existing derivative expressions in matrix notation, we obtain them by direct matrix differentiation. For this, we describe a simple procedure that is not complete, but sufficient for handling typical g^E models. We then tabulate matrix versions of first- and second-order derivatives (e.g., activity coefficients and their jacobian matrices) of well-known g^E models such as NRTL, UNIQUAC, and UNIFAC, as well as of newer COSMO-based models, intending to build a useful reference for those willing to implement them for either practical or educational purposes.

2. Methodology

2.1. Notation and Algebra

In this section, we describe the notation adopted in this paper and some algebraic rules associated with it. Except for the introduction of a new operator, the notation does not differ from that commonly used in linear algebra and matrix theory. Variables with one or two indices take part in operations like $\sum_i [a_i]$, $\sum_j [A_{ij}b_j]$, or $\sum_k [A_{ik}B_{kj}]$. Here, we regard a one-index variable as an entry of a column vector and a two-index variable an entry of a matrix. Table 1 contains rules for converting operations among indexed variables into matrix/vector operations. Boldface minor letters denote vectors and boldface capital letters denote matrices. Symbol **1** denotes a column vector of 1's and superscript "t" indicates transposition. The symbol \mathcal{D} will be defined shortly. In our notation, scalar functions behave as entrywise operators when applied to vectors or matrices (for instance, $\mathbf{a} = \ln \mathbf{b}$ means that $a_i = \ln b_i$ for all i). In Section 3.1, we will illustrate how to use Table 1 to rewrite g^E models in matrix notation.

With Table 1, all properties of ordinary matrix/vector addition, multiplication, and transposition are valid. Along the paper, we will make use of identities such as $\mathbf{A}(\mathbf{B}+\mathbf{C}) = \mathbf{AB}+\mathbf{AC}$, $(\mathbf{ABC})^t = \mathbf{C}^t\mathbf{B}^t\mathbf{A}^t$, etc. They can be found in books like Refs. {Amundson1966} and {Varma1997}. Entrywise matrix operations like the Hadamard product $\mathbf{A} \circ \mathbf{B}$ and entrywise exponentiation $\mathbf{A}^{[\alpha]}$ will be treated individually when necessary. Note that $\mathbf{A}^{[\alpha]}$ should not be confused with the ordinary matrix exponentiation \mathbf{A}^α (e.g. $\mathbf{A}^{[-1]}$ has each entry ij equal to $1/A_{ij}$ while \mathbf{A}^{-1} is the inverse matrix of \mathbf{A}). Another new symbol introduced in Table 1 is the superscript "s", which stands for symmetrization and is applicable to any square matrix.

A key concept of the present work is the diagonal operator \mathcal{D} introduced in Table 1. It builds a diagonal matrix from the elements of a vector, i.e.

$$\mathcal{D}_{ij}^\alpha(\mathbf{a}) = \begin{cases} a_i^\alpha & \text{if } i = j \\ 0 & \text{if } i \neq j \end{cases}.$$

By admitting an exponent α and resulting in an ordinary matrix, the operator \mathcal{D} can bring operations like entrywise vector multiplication ($a_i b_i$), division (a_i/b_i), and exponentiation (a_i^α) to the realm of linear algebra. It has some special properties. For instance, $\mathcal{D}^\alpha(\mathbf{1}) = \mathbf{I}$ for any exponent α and $\mathcal{D}^0(\mathbf{a}) = \mathbf{I}$ for any vector \mathbf{a} , where \mathbf{I} is the identity matrix. Being a diagonal matrix, $\mathcal{D}^\alpha(\mathbf{a})$ is symmetric and $\mathcal{D}^{-1}(\mathbf{a})$ is the inverse matrix of $\mathcal{D}(\mathbf{a})$. Table 2 contains a list of useful rules involving the diagonal operator, which we will apply in this paper for simplifying expressions in matrix notation.

2.2. Differentiation in Matrix Notation

The major contribution of this paper is the description of a simplified differentiation system directly applicable to functions in matrix notation. To favor understanding, we keep a close analogy with the familiar differentiation of scalar functions, which is the application of an operator ∂_x that obeys a set of basic rules such as those in Column A of Table 3. The act of differentiating a function is a successive application of these rules. Note that identities like $\partial_x(ay^\alpha) = \alpha ay^{\alpha-1}\partial_x y$ and $\partial_x(y/z) = (z\partial_x y - y\partial_x z)/z^2$ derive directly from them. Normally, one uses the same rules to deal with indexed variables, but here we look for new rules directly applicable to vectors and matrices. In this way we intend to perform, at the same time, differentiations involving all possible index

combinations (e.g., $\partial_{x_j} y_i$ for all i and j). We should remark that derivatives entailing more than two indices (e.g. $\partial_{x_k} Y_{ij}$ or $\partial_{x_{kl}} Y_{ij}$) are not naturally suitable for matrix notation. In Matrix Calculus, {MAGNUS1999} one deals with this problem by defining conventions for index condensation and by employing unusual operations like Kronecker products, vectorizations, and commutations. In this paper, however, we will only need derivatives of vector-valued functions with respect to vector variables ($\partial_{x_j} y_i$) and of matrix-valued functions with respect to scalar variables ($\partial_x Y_{ij}$). This enables us to avoid complexity by adopting an *ad hoc* strategy. Without departing from ordinary matrix algebra, we define differentiation rules that are sufficient for our needs. Nevertheless, we believe that the same rules can suffice in other situations and thus gain broader applicability.

First, we describe the differentiation of a vector-valued function with respect to a vector variable as the application of \mathbf{J}_x , called the Jacobian operator, which obeys a certain set of rules. If \mathbf{x} is an n -dimensional column vector and \mathbf{y} is a function resulting in an m -dimensional column vector, then $\mathbf{J}_x \mathbf{y}$ is a $m \times n$ matrix whose each entry ij is equal to $\partial_{x_j} y_i$. In this way, a basic set of differentiation rules for this case is shown in Column B of Table 3, where \mathbf{a} is a constant vector, $\mathbf{0}$ is a null matrix, \mathbf{A} is a constant matrix, and \mathbf{y} and \mathbf{z} are vector-valued functions of \mathbf{x} . Note that these rules are very similar to those regarding scalar functions, except that products are no longer commutative. If we regard a scalar as a single-entry vector, we can use these rules to differentiate scalar-valued functions with respect to vector variables (which makes \mathbf{J}_x equivalent to the transpose of the gradient operator ∇_x) and vice-versa. If both function and variable are scalars, then \mathbf{J}_x becomes equivalent to ∂_x . We can also derive other useful rules from those in Table 3. For example, if we make $\alpha = 1$ and $\mathbf{z} = \mathbf{z}\mathbf{1}$ in the Product Rule, where z is a scalar-valued function of \mathbf{x} , then we can use Table 2 to show that

$$\mathbf{J}_x(\mathbf{z}\mathbf{y}) = \mathbf{z}\mathbf{J}_x\mathbf{y} + \mathbf{y}\mathbf{J}_x\mathbf{z}. \quad (1)$$

Deducing from Table 2 that $\mathbf{z}^t \mathbf{y} = \mathbf{1}^t \mathbf{D}(\mathbf{z})\mathbf{y}$, we can also turn the Product Rule into

$$\mathbf{J}_x(\mathbf{z}^t \mathbf{y}) = \mathbf{z}^t \mathbf{J}_x \mathbf{y} + \mathbf{y}^t \mathbf{J}_x \mathbf{z}. \quad (2)$$

Finally, we should remark that Column B of Table 3 does not form a complete set. For instance, they do not allow us to evaluate $\mathbf{J}_x(\mathbf{Z}\mathbf{y})$, where \mathbf{Z} is an arbitrary matrix-valued function of \mathbf{x} . As already mentioned, however, the presented equations are enough for our purposes.

We now focus on the differentiation of matrix-valued functions with respect to scalar variables. In this case, a Jacobian operation $\mathbf{J}_x \mathbf{Y}$ results in a matrix of the same size as \mathbf{Y} whose each entry ij is equal to $\partial_x Y_{ij}$. For this, we introduce the set of rules in Column C of Table 3, where \mathbf{A} and \mathbf{B} are constant matrices while \mathbf{Y} and \mathbf{Z} are matrix-valued functions of x . Note that a similar form of the Product Rule applies for both ordinary and Hadamard products. Again, other rules are derivable from those in Column C. For example, we can derive a rule for $\mathbf{Y}^{[\alpha]}$ by applying the Hadamard Product Rule $\alpha-1$ times, which leads to

$$\mathbf{J}_x(\mathbf{Y}^{[\alpha]}) = \alpha \mathbf{Y}^{[\alpha-1]} \circ \mathbf{J}_x \mathbf{Y}. \quad (3)$$

By applying the rules of Column B of Table 3 to a g^E model in matrix notation, we can obtain a unique expression whose outcome is a vector containing the activity coefficients of all components of a mixture. By applying them again, we can obtain an expression for a Jacobian matrix of such activity coefficients with respect to all mole fractions. Moreover, we can apply rules of Column C

to obtain derivatives with respect to temperature, including properties like excess enthalpy and excess entropy.

2.3. Application to the Thermodynamics of Mixtures

Let $m = m(T, P, \mathbf{x})$ be a model for some molar property m of a mixture, where T is temperature, P is pressure, and \mathbf{x} represents the mole fractions of all components. Very often, we are interested in the partial molar property \bar{m}_i of every component i of the mixture. This is defined by $\bar{m}_i = \partial_{n_i}(Nm)$, where n_i is the number of moles of that component and $N = \mathbf{1}^t \mathbf{n}$ is the total number of moles of the mixture. {TESTER1997} If m is expressed in matrix notation, we can use Tables 1 to 3 to obtain an expression for the partial-molar vector $\bar{\mathbf{m}}$. First, we write $\bar{\mathbf{m}} = \mathbf{J}_n^t(Nm)$, where the transposition makes $\bar{\mathbf{m}}$ a column vector. Applying Eq. (1), we get $\bar{\mathbf{m}} = m\mathbf{J}_n^t N + N\mathbf{J}_n^t m$. Since $\mathbf{J}_n N = \mathbf{1}^t$ and $\mathbf{J}_n m = (\mathbf{J}_x m)(\mathbf{J}_n \mathbf{x})$, we have that $\bar{\mathbf{m}} = m\mathbf{1} + N(\mathbf{J}_n^t \mathbf{x})\mathbf{J}_x^t m$. As $\mathbf{x} = N^{-1}\mathbf{n}$, Eq. (1) and Table 3 make $\mathbf{J}_n \mathbf{x} = N^{-1}(\mathbf{I} - \mathbf{x}\mathbf{1}^t)$. Therefore,

$$\bar{\mathbf{m}} = m\mathbf{1} + (\mathbf{I} - \mathbf{x}\mathbf{1}^t)\mathbf{J}_x^t m. \quad (4)$$

This is identical to Equation (4) of Ref. {Castier1999}, but written in matrix notation. As discussed in such reference, some applications like phase stability analysis require derivatives like $N\partial_{n_j}\bar{m}_i = N\partial_{n_j}\partial_{n_i}(Nm)$. As the order of differentiation is irrelevant, the resulting matrix $N\mathbf{J}_n\bar{\mathbf{m}}$ is symmetric. Applying the chain rule again, we see that $N\mathbf{J}_n\bar{\mathbf{m}} = (\mathbf{J}_x\bar{\mathbf{m}})(\mathbf{I} - \mathbf{x}\mathbf{1}^t)$. Now this is identical to Equation (8) of Ref. {Castier1999}. Let us analyze some consequences of Eq. (4). First, notice that $\mathbf{1}^t \mathbf{x} = 1$ (that is, all mole fractions sum up to one) and that $(\mathbf{I} - \mathbf{x}\mathbf{1}^t)\mathbf{x} = \mathbf{0}$. Then, transposing Eq. (4) and multiplying it by \mathbf{x} , we have

$$\bar{\mathbf{m}}^t \mathbf{x} = m. \quad (5)$$

According to Table 1, this means that $m = \sum_i [x_i \bar{m}_i]$, which is a well-known feature of partial molar properties. {TESTER1997} In addition, by applying the operator \mathbf{J}_x to both sides of Eq. (4) and then using Eq. (2), we find that $\mathbf{J}_x \bar{\mathbf{m}} = (\mathbf{I} - \mathbf{x}\mathbf{1}^t)\mathbf{J}_x \mathbf{J}_x^t m$. Then, given that $\mathbf{x}^t(\mathbf{I} - \mathbf{x}\mathbf{1}^t) = \mathbf{0}^t$, we have

$$\mathbf{x}^t \mathbf{J}_x \bar{\mathbf{m}} = \mathbf{0}^t. \quad (6)$$

This is the matrix form of another well-known equation: the generalized Gibbs-Duhem relation at constant temperature and pressure. {TESTER1997} This approach can be useful in practice, especially in the determination of phase equilibrium conditions. For instance, m can be the departure molar Gibbs energy $g^d(T, P, \mathbf{x})$ obtained from an equation of state. In this case, $\bar{\mathbf{m}} = RT \ln \phi$, where ϕ contains the fugacity coefficients of all components. Thus, differentiation in matrix notation can be helpful in deriving expressions for ϕ corresponding to different mixing rules. In this paper, we consider the case when m is the excess molar Gibbs energy $g^E(T, \mathbf{x})$ of a liquid mixture. In this case, $\bar{\mathbf{m}}$ is related to the activity coefficients of all components by $\bar{\mathbf{m}} = RT \ln \gamma$. Thus, Eq. (4) becomes

$$RT \ln \gamma = \mathbf{1}g^E + (\mathbf{I} - \mathbf{x}\mathbf{1}^t)\mathbf{J}_x^t g^E. \quad (7)$$

Therefore, starting from a g^E model in matrix notation, we can directly obtain an expression for $\ln \gamma$. Then, by applying our procedure to such expression, we can derive a single equation for the Jacobian matrix $\mathbf{J}_x \ln \gamma$, which is often required in numerical methods for phase equilibrium calculations. Our procedure is also useful when derivatives with respect to temperature are needed. For instance, we often have to perform energy and entropy balances to determine energy

requirements in chemical plants or to design energy-effective processes. This involves calculations of molar enthalpies and molar entropies of flowing streams. These properties are related to g^E by means of the Gibbs-Helmholtz relation $h^E/T^2 = -J_T(g^E/T)$. {TESTER1997} Such relation, together with Eq. (4) and the fact that $J_x^t J_T = J_T J_x^t$, leads to

$$\frac{h^E}{RT^2} = -J_T \ln \gamma. \quad (8)$$

Expressions for h^E and \bar{h}^E can be used in parameter estimation since these properties are experimentally measurable. Moreover, $J_T \ln \gamma$ might be useful in numerical determinations of equilibrium temperature. Excess entropies come from $s^E = (h^E - g^E)/T$ and $\bar{s}^E = \bar{h}^E/T - R \ln \gamma$. By applying J_T to h^E , we can express the excess molar heat capacity of a mixture by

$$\frac{c_P^E}{R} = 2 \frac{h^E}{RT} - T^2 J_T^2 \left(\frac{g^E}{RT} \right), \quad (9)$$

where $J_T^2 = J_T J_T$. We can also obtain the Jacobian matrices $J_x \bar{h}^E$ and $J_x \bar{s}^E$, which come from $J_x J_T \ln \gamma$. Finally, we remark that our differentiation rules cannot manage second- or higher-order derivatives of $\ln \gamma$ and h^E with respect to the vector \mathbf{x} , but there is no limit for the order of differentiation with respect to temperature.

2.4. Additional Applicability

In addition to handling thermodynamic models, our methodology can be used to manage systems of equations involved in phase equilibrium and phase stability calculations or to write down material and energy balances in multicomponent/multistage operations such as distillation, extraction, and absorption. For instance, consider a system containing two liquid phases with compositions \mathbf{x}^I and \mathbf{x}^{II} . When equilibrium occurs, we have $x_i^{II} \gamma_i^{II} = x_i^I \gamma_i^I$ for all i , where $\gamma_i^\alpha = \gamma_i(T, \mathbf{x}^\alpha)$. {TESTER1997} Thus, from Table 1 we can write $\mathcal{D}(\gamma^{II}) \mathbf{x}^{II} = \mathcal{D}(\gamma^I) \mathbf{x}^I$. Premultiplying both sides by $\mathcal{D}^{-1}(\gamma^{II})$ and resorting to Table 2, we observe that $\mathbf{x}^{II} = \mathcal{D}(\mathbf{k}) \mathbf{x}^I$, where vector $\mathbf{k} = \mathcal{D}^{-1}(\gamma^{II}) \gamma^I$ contains the partition coefficients of all components. Equilibrium calculations may require derivatives of these partition coefficients. By the Product Rule as in Column B of Table 3, these are given by $J_z \mathbf{k} = \mathcal{D}(\mathbf{k})(J_z \ln \gamma^I - J_z \ln \gamma^{II})$, where \mathbf{z} can be replaced by T , \mathbf{x}^I , or \mathbf{x}^{II} .

Jacobian matrices are often used to solve non-linear systems of equation by the Newton-Raphson method. Sometimes, it is possible to encompass a system of equations in a single matrix expression like $\mathbf{f}(\mathbf{x}) = \mathbf{0}$. In this case, both \mathbf{x} and \mathbf{f} must have the same dimension and the solution is pursued by iterating $\mathbf{x} \leftarrow \mathbf{x} - (J_x \mathbf{f})^{-1} \mathbf{f}$. In some cases, the system of interest might contain dissimilar equations that cannot be assembled in a single expression. However, the might possibly be represented by a set of equations like $\mathbf{f}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, $\mathbf{g}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, and $\mathbf{h}(\mathbf{x}, \mathbf{y}) = \mathbf{0}$, where \mathbf{x} and \mathbf{y} are the vector variables we want to calculate. In this example, although the five vectors can have distinct dimensions, the sum of dimensions of the functions (\mathbf{f} , \mathbf{g} , and \mathbf{h}) must equal that of the variables (\mathbf{x} and \mathbf{y}). In this case, we can generalize the Newton-Raphson iteration by

$$\begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} \leftarrow \begin{pmatrix} \mathbf{x} \\ \mathbf{y} \end{pmatrix} + \begin{pmatrix} J_x \mathbf{f} & J_y \mathbf{f} \\ J_x \mathbf{g} & J_y \mathbf{g} \\ J_x \mathbf{h} & J_y \mathbf{h} \end{pmatrix}^{-1} \begin{pmatrix} \mathbf{f} \\ \mathbf{g} \\ \mathbf{h} \end{pmatrix}. \quad (10)$$

The equation above consists of block vectors and a block matrix, built by merging ordinary vectors and matrices in accordance with their relative positions inside the square brackets. With our

procedure, we can derive expressions for the individual matrices that form the block Jacobian matrix above. An example of application is the simultaneous solution of material balances, energy balances, and equilibrium relations in calculations of multicomponent separation processes.

3. Application to Excess Gibbs Energy Models

We now make use of the methodology described in Section 2 to manage g^E models in matrix notation. We selected the NRTL model{RENON1968} for a detailed description of the procedure. Subsequently, we obtain expressions for other models, namely UNIQUAC,{ABRAMS1975} UNIFAC,{FREDENSLUND1975} and Modified UNIFAC(Dortmund).{WEIDLICH1987A} Finally, we deal with a more recent class of models based on the Conductor-like Screening Model for Real Solvents (COSMO-RS) of Klamt and co-workers.{KLAMT1993, KLAMT1995, KLAMT2000}

3.1. Non-Random Two-Liquid (NRTL) Model

The Non-Random Two-Liquid (NRTL) model{RENON1968} is based on the local composition concept. It has three binary parameters for each pair of components: interaction parameters A_{ij} and A_{ji} , and non-randomness parameter $\alpha_{ij} = \alpha_{ji}$. By definition, $A_{ii} = 0$ and $\alpha_{ii} = 0$ for all i . The molar excess Gibbs energy of a mixture is given by

$$\frac{g^E}{RT} = \sum_{i=1}^n \frac{x_i \sum_{j=1}^n \Lambda_{ji} x_j}{\sum_{j=1}^n G_{ji} x_j}, \quad (11)$$

where

$$G_{ij} = \exp(-\alpha_{ij} A_{ij}/T) \text{ and } \Lambda_{ij} = A_{ij} G_{ij}/T. \quad (12)$$

To translate the model equations, we first express matrices \mathbf{G} and $\mathbf{\Lambda}$ as

$$\mathbf{G} = \exp(-T^{-1} \boldsymbol{\alpha} \circ \mathbf{A}) \text{ and} \quad (13)$$

$$\mathbf{\Lambda} = T^{-1} \mathbf{A} \circ \mathbf{G}. \quad (14)$$

Translation of Eq. (11) corresponds to a sequence of index eliminations. First, we define $a_i = \sum_j [G_{ji} x_j]$ and $b_i = \sum_j [\Lambda_{ji} x_j]$ for all i . We then use Table 1 to write $\mathbf{a} = \mathbf{G}^t \mathbf{x}$ and $\mathbf{b} = \mathbf{\Lambda}^t \mathbf{x}$. This eliminates index j from Eq. (11) by making $g^E/RT = \sum_i [x_i b_i / a_i]$. To eliminate index i , we define $c_i = b_i / a_i$ and $d_i = c_i x_i$, which translate to $\mathbf{c} = \boldsymbol{\Phi}^{-1}(\mathbf{a}) \mathbf{b}$ and $\mathbf{d} = \boldsymbol{\Phi}(\mathbf{c}) \mathbf{x}$, respectively. At this point, we have $g^E/RT = \sum_i [d_i] = \mathbf{1}^t \mathbf{d}$. Finally, back substitution leads to $g^E/RT = \mathbf{1}^t \boldsymbol{\Phi}^{-1}(\mathbf{G}^t \mathbf{x}) \mathbf{\Lambda}^t \mathbf{x}$. We can further simplify this equation via Table 2. For instance, we can write

$$\frac{g^E}{RT} = \mathbf{x}^t \mathbf{E} \mathbf{x}, \quad (15)$$

where $\mathbf{E} = \mathbf{\Lambda} \boldsymbol{\Phi}^{-1}(\mathbf{G}^t \mathbf{x})$. Let us now obtain an expression for the activity coefficients of all components. We start by rewriting Eq. (7) as

$$\ln \boldsymbol{\gamma} = \mathbf{J}_x^t \left(\frac{g^E}{RT} \right) + \mathbf{1} \left[\frac{g^E}{RT} - \mathbf{J}_x \left(\frac{g^E}{RT} \right) \mathbf{x} \right]. \quad (16)$$

Transposing $\mathbf{x}^t \mathbf{J}_x^t(g^E/RT)$ was possible because it results in a scalar value. Applying Eq. (2) to Eq. (15), it follows that $\mathbf{J}_x(g^E/RT) = \mathbf{x}^t \mathbf{J}_x(\mathbf{E}\mathbf{x}) + \mathbf{x}^t \mathbf{E}^t$. From the product rule in Table 3, we have $\mathbf{J}_x(\mathbf{E}\mathbf{x}) = \mathbf{L}\mathbf{D}^{-1}(\mathbf{G}^t\mathbf{x}) - \mathbf{L}\mathbf{D}(\mathbf{x})\mathbf{D}^{-2}(\mathbf{G}^t\mathbf{x})\mathbf{G}^t$. After substitution and application of Table 2, we obtain

$$\mathbf{J}_x\left(\frac{g^E}{RT}\right) = \mathbf{x}^t[\mathbf{E}^s - \mathbf{E}\mathbf{D}(\mathbf{x})\mathbf{L}^t], \quad (17)$$

where $\mathbf{L} = \mathbf{G}\mathbf{D}^{-1}(\mathbf{G}^t\mathbf{x})$. Now noticing that $\mathbf{L}^t\mathbf{x} = \mathbf{D}^{-1}(\mathbf{G}^t\mathbf{x})\mathbf{G}^t\mathbf{x} = \mathbf{1}$ (Cf. Table 2) and that $\mathbf{x}^t\mathbf{E}\mathbf{x} = \mathbf{x}^t\mathbf{E}^t\mathbf{x}$, we verify that the term inside square brackets in Eq. (16) vanishes for the NRTL model. Therefore, Eq. (16) simplifies to $\ln\gamma = \mathbf{J}_x(g^E/RT)$ in this case. Finally, by transposing Eq. (17), we obtain the desired expression for the activity coefficients, which is

$$\ln\gamma = [\mathbf{E}^s - \mathbf{L}\mathbf{D}(\mathbf{x})\mathbf{E}^t]\mathbf{x}. \quad (18)$$

Note that this expression is consistent with Eq. (5) since $\mathbf{x}^t\mathbf{L}\mathbf{D}(\mathbf{x}) = \mathbf{1}^t\mathbf{D}(\mathbf{x}) = \mathbf{x}^t$. Proceeding even further, we now apply the operator \mathbf{J}_x to the equation above to obtain $\mathbf{J}_x\ln\gamma = \mathbf{J}_x(\mathbf{E}\mathbf{x}) + \mathbf{J}_x(\mathbf{E}^t\mathbf{x}) - \mathbf{J}_x[\mathbf{L}\mathbf{D}(\mathbf{x})\mathbf{E}^t\mathbf{x}]$. After using Tables 2 and 3 to evaluate these derivatives and factoring out some terms of the resulting equation, we are able to express the Jacobian matrix $\mathbf{J}_x\ln\gamma$ for the NRTL model as

$$\mathbf{J}_x\ln\gamma = \{[\mathbf{E} - \mathbf{L}\mathbf{D}(\mathbf{E}^t\mathbf{x})][\mathbf{I} - \mathbf{D}(\mathbf{x})\mathbf{L}^t]\}^s. \quad (19)$$

This is a remarkably simple equation. Note that it satisfies the Gibbs-Duhem relation implied by Eq. (6). Let us now deal with temperature-derivatives of the g^E model. We start by applying the operator \mathbf{J}_T to Eq. (15) and using the Gibbs-Helmholtz relation to express h^E by

$$\frac{h^E}{RT} = -\mathbf{T}\mathbf{x}^t\dot{\mathbf{E}}\mathbf{x}, \quad (20)$$

where $\dot{\mathbf{E}} = \mathbf{J}_T\mathbf{E}$. From Column C of Table 3, we conclude that $\dot{\mathbf{E}} = [\dot{\mathbf{A}} - \mathbf{E}\mathbf{D}(\dot{\mathbf{G}}^t\mathbf{x})]\mathbf{D}^{-1}(\mathbf{G}^t\mathbf{x})$, with $\dot{\mathbf{G}} = \mathbf{J}_T\mathbf{G} = \mathbf{T}^{-1}\alpha\circ\mathbf{A}$ and $\dot{\mathbf{A}} = \mathbf{J}_T\mathbf{A} = \mathbf{T}^{-1}(\mathbf{A}\circ\dot{\mathbf{G}} - \mathbf{A})$. We can also apply \mathbf{J}_T to Eq. (18) and use Table 3 to derive

$$\mathbf{J}_T\ln\gamma = [\dot{\mathbf{E}}^s - \dot{\mathbf{L}}\mathbf{D}(\mathbf{x})\mathbf{E}^t - \mathbf{L}\mathbf{D}(\mathbf{x})\dot{\mathbf{E}}^t]\mathbf{x}, \quad (21)$$

where $\dot{\mathbf{L}} = \mathbf{J}_T\mathbf{L} = [\dot{\mathbf{G}} - \mathbf{L}\mathbf{D}(\dot{\mathbf{G}}^t\mathbf{x})]\mathbf{D}^{-1}(\mathbf{G}^t\mathbf{x})$. Partial molar excess properties (\mathbf{h}^E and \mathbf{s}^E) of all components of a mixture are obtained from $\mathbf{J}_T\ln\gamma$ (see Section 2.3). We can also express c_p^E by employing Eq. (9), which requires second-order derivatives of matrices \mathbf{G} and \mathbf{A} . To conclude, we summarize in Table 4 all matrix expressions derived for the NRTL model. Notice that, as several defined matrices are independent of mole fractions, we can compute them only once before a series of isothermal calculations are performed. We finally remark that the described procedure is actually easier than the standard method once one gets used to the rules of Tables 1 to 3. Moreover, it generates simple expressions that can be implemented with minimum chance of errors and thus minimum debugging effort.

3.2. Universal Quasi-Chemical (UNIQUAC) Model

The UNIQUAC model{ABRAMS1975} stems from Guggenheim's quasi-chemical theory and Wilson's local composition concept. It expresses g^E as a sum of a combinatorial (entropic) term g_C^E and a residual (energetic) term g_R^E . Each component i has two individual parameters, r_i and q_i , respectively related to its molecular volume and surface area. The combinatorial term is given by the Staverman-Guggenheim model, which can be written in matrix notation as

$$\frac{g_C^E}{RT} = \mathbf{x}^t [\mathcal{D}(\mathbf{m}) \ln \boldsymbol{\phi} + (z/2) \mathcal{D}(\mathbf{q}) \ln \boldsymbol{\theta}], \quad (22)$$

where $\mathbf{m} = \mathbf{1} - (z/2)\mathbf{q}$, $\boldsymbol{\phi} = \mathbf{r}(\mathbf{r}^t \mathbf{x})^{-1}$ and $\boldsymbol{\theta} = \mathbf{q}(\mathbf{q}^t \mathbf{x})^{-1}$. The coordination number z is usually taken as 10. In addition, each pair ij has two binary interaction parameters, A_{ij} and A_{ji} , with $A_{ii} = 0$ for all i . The residual term is then expressed as

$$\frac{g_R^E}{RT} = -\mathbf{x}^t \mathcal{D}(\mathbf{q}) [\ln \mathbf{q} - \ln \boldsymbol{\theta} - \ln(\boldsymbol{\Lambda}^t \mathbf{x})], \quad (23)$$

where $\boldsymbol{\Lambda} = \mathcal{D}(\mathbf{q}) \exp(-T^{-1} \mathbf{A})$. We apply the procedure detailed in the preceding section to Eqs. (22) and (23) and the results are summarized in Table 5.

3.3. Universal Functional-Group Activity Coefficient (UNIFAC) Model

Group-contribution models like UNIFAC,{FREDENSLUND1975} which consider molecules as collections of independent functional groups, are interesting alternatives to correlative models like NRTL and UNIQUAC. For a system with n_c components formed by n_g distinct groups, a $n_c \times n_g$ matrix \mathbf{v} is defined so that each v_{ij} is the number of repetitions of group j in a molecule of component i . Additional UNIFAC inputs consist of n_g -dimensional vectors \mathbf{R} and \mathbf{Q} , containing volume and size parameters of individual groups, and a $n_g \times n_g$ matrix \mathbf{A} containing group-group interaction parameters. The combinatorial part of the model is given by Eq. (22) with $\mathbf{r} = \mathbf{vR}$ and $\mathbf{q} = \mathbf{vQ}$. Its residual part is expressed in terms of activity coefficients by $\ln \gamma_R = \mathbf{v} \ln \gamma_G - (\mathbf{v} \circ \ln \boldsymbol{\Gamma}) \mathbf{1}$. In this equation, $\ln \gamma_G$ derives from Eq. (23), but now considering a mixture of free groups whose molar composition is proportional to $\mathbf{v}^t \mathbf{x}$. In addition, each row i of matrix $\boldsymbol{\Gamma}$ is equivalent to γ_G , but when pure substance i is also considered as a mixture of free groups. After some algebraic treatment, we obtain

$$\ln \gamma_R = \{ \boldsymbol{\Omega} \circ [\ln \boldsymbol{\Lambda} + (\boldsymbol{\Omega} \circ \boldsymbol{\Lambda}^{[-1]}) \mathbf{G}^t] \} \mathbf{1} - \mathcal{D}(\mathbf{q}) \ln \boldsymbol{\theta} - \boldsymbol{\Omega} \ln(\boldsymbol{\Lambda}^t \mathbf{x}) - \mathbf{L} \boldsymbol{\Omega}^t \mathbf{x}, \quad (24)$$

where $\mathbf{G} = \exp(-T^{-1} \mathbf{A})$, $\boldsymbol{\Omega} = \mathbf{v} \mathcal{D}(\mathbf{Q})$, $\boldsymbol{\Lambda} = \boldsymbol{\Omega} \mathbf{G}$, and $\mathbf{L} = \boldsymbol{\Lambda} \boldsymbol{\Omega}^{-1} (\boldsymbol{\Lambda}^t \mathbf{x})$. Compared to the corresponding expression in Table 5, the constant vector $\boldsymbol{\varepsilon}$ has been replaced by a more complex, temperature-dependent expression. This makes some UNIFAC expressions somewhat more involved than their UNIQUAC counterparts. Because the main diagonal of \mathbf{A} is null, the two models become equivalent when each molecule is a single functional group (that is, when $\mathbf{v} = \mathbf{I}$). In Table 6, we present all the matrix expressions obtained for the UNIFAC model.

Although UNIFAC is the most popular group-contribution g^E model, it has some shortcomings. For example, parameters used in vapor-liquid equilibrium calculations are not suitable for predicting liquid-liquid partition, infinity-dilution activity coefficients, or calorimetric properties such as h^E and c_p^E . A successful extension of UNIFAC is the UNIFAC-Dortmund model,{WEIDLICH1987A} which involves a small change in the combinatorial term and a greater number of interaction parameters estimated from experimental data. Therefore, the matrix version of the UNIFAC-Dortmund model is given by Table 5 with the following replacements:

$$\begin{aligned}
\mathbf{G} &= \exp[-(\mathbf{T}^{-1}\mathbf{A} + \mathbf{B} + \mathbf{TC})] \\
\dot{\mathbf{G}} &= (\mathbf{T}^{-2}\mathbf{A} - \mathbf{C}) \circ \mathbf{G} \\
\ddot{\mathbf{G}} &= (\mathbf{T}^{-2}\mathbf{A} - \mathbf{C}) \circ \dot{\mathbf{G}} - 2\mathbf{T}^{-3}\mathbf{A} \circ \mathbf{G} \\
g^E/RT &= \mathbf{x}^t[\ln\boldsymbol{\phi}' - \boldsymbol{\phi}' + \boldsymbol{\mathcal{D}}(\mathbf{q})(4\ln\boldsymbol{\theta} - 5\ln\boldsymbol{\phi}) + \boldsymbol{\varepsilon} - \boldsymbol{\Omega}\ln(\boldsymbol{\Lambda}^t\mathbf{x}) - \mathbf{q}] + 1 \\
\ln\gamma &= \ln\boldsymbol{\phi}' - \boldsymbol{\phi}' + \boldsymbol{\mathcal{D}}(\mathbf{q})(4\ln\boldsymbol{\theta} - 5\ln\boldsymbol{\phi}) + 5\boldsymbol{\phi}\mathbf{q}^t\mathbf{x} + \mathbf{m} + \boldsymbol{\varepsilon} - \boldsymbol{\Omega}\ln(\boldsymbol{\Lambda}^t\mathbf{x}) - \mathbf{L}\boldsymbol{\Omega}^t\mathbf{x} \\
\mathbf{J}_x\ln\gamma &= (\boldsymbol{\phi}' - \mathbf{1})\boldsymbol{\phi}'^t + (5\mathbf{q}\boldsymbol{\phi}^t - \mathbf{L}\boldsymbol{\Omega}^t)^s - 4\mathbf{q}\boldsymbol{\theta}^t - 5\boldsymbol{\phi}\mathbf{q}^t\mathbf{x}\boldsymbol{\phi}^t + \mathbf{L}\boldsymbol{\mathcal{D}}(\boldsymbol{\Omega}^t\mathbf{x})\mathbf{L}^t
\end{aligned}$$

where $\mathbf{r}' = \boldsymbol{\mathcal{D}}^{3/4}(\mathbf{r})\mathbf{1}$ and $\boldsymbol{\phi}' = \mathbf{r}'(\mathbf{r}'^t\mathbf{x})^{-1}$, while \mathbf{A} , \mathbf{B} , and \mathbf{C} are the interaction parameter matrices.

3.4. Conductor-like Screening Model for Real Solvents (COSMO-RS) and Related Models

Group-contribution models are often unable to discriminate isomers and to take into account the effect of the intramolecular environment on the behavior of individual groups. Moreover, one can only deal with groups whose parameters have been previously estimated from experimental data. To overcome these issues, a new class of models has recently emerged from the pioneering work of Klamt and coworkers.{KLAMT1993, KLAMT1995} In their approach, the behavior of a molecule in a solution results from a correction of its behavior when it is immersed in a continuous, perfectly conducting solvent. In this ideal condition, all charges that appear in the surface of a molecule are completely screened by opposite charges in the solvent. Hence, the model is known as the Conductor-like Screening Model for Real Solvents (COSMO-RS).{ KLAMT1995} It entails a small set of universal parameters estimated from experimental data, while needed information about specific molecules are obtained from *ab initio* calculations. With its natural ability to discriminate isomers and include intramolecular effects, this innovative approach is more widely applicable than group contribution methods. The original COSMO-RS model and parameters are implemented in commercial software, but open-access alternatives have been proposed in the literature, such as COSMO-SAC{LIN2002} and COSMO-RS(OI).{GRENNEMANN2005} Our aim here is to express in matrix notation the equations involved in these COSMO-based models, and then use our procedure to obtain Jacobian matrices and temperature derivatives. We start by describing the underlying physical picture of the model and presenting the entailed equations. After that, we apply the methodology of Section 2. Finally, we explain how to handle the required parameters.

As in UNIFAC, we distinguish each component i of the mixture by its molecular volume v_i and molecular surface area a_i . However, instead of viewing a molecule as a collection of functional groups, we regard its surface as a patchwork of segments of different types labeled with indices from 1 to n_t . Let us then define η_i as the total number of segments in the surface of a molecule of component i and P_{ij} as the fraction of segments of type j amongst them. In this way, a mixture of n_c components is characterized by three n_c -dimensional vectors \mathbf{v} , \mathbf{a} , and \mathbf{n} and one $n_c \times n_t$ matrix \mathbf{P} subject to the restriction $\mathbf{P}\mathbf{1} = \mathbf{1}$ (that is, the entries of each row of \mathbf{P} must sum up to one). Again, entropic and energetic effects are treated separately. To calculate the entropic part, we use Eq. (22) with $\mathbf{r} = v_0^{-1}\mathbf{v}$ and $\mathbf{q} = a_0^{-1}\mathbf{a}$, where v_0 is an arbitrary reference volume and a_0 is the surface area that a molecule spends to make z contacts with its neighbors. For the energetic part, we sum up the contributions of all surface segments present in the mixture (in the same way as we do for functional groups in the UNIFAC model). Therefore,

$$\ln\gamma_R = \boldsymbol{\mathcal{D}}(\boldsymbol{\eta})[\mathbf{P}\ln\boldsymbol{\gamma}_S - (\mathbf{P} \circ \ln\boldsymbol{\Gamma})\mathbf{1}], \quad (25)$$

where each entry j of the n_t -dimensional vector $\boldsymbol{\gamma}_S$ is the activity coefficient of the j -th segment type in the mixture while each entry ij of the $n_c \times n_t$ matrix $\boldsymbol{\Gamma}$ is the activity coefficient of the j -th segment type in pure substance i . The product $\boldsymbol{\mathcal{D}}(\boldsymbol{\eta})\mathbf{P}$ plays the role of matrix \mathbf{v} in UNIFAC. To obtain

expressions for γ_s and Γ , we assume that molecules interact with each other via surface contacts, with E_{jk} ($=E_{kj}$) being the energy involved in bringing a contact between segments of types j and k from the ideal state to the real state. In this way, an ensemble of paired segments can represent the energetics of the system. If we also assume that no segments remain unpaired (which is reasonable for a condensed phase) and that the pairs are formed randomly *a priori*, an exact analytical solution exists for a mixture with composition \mathbf{x}_s in terms of segment-type fractions. In matrix notation, such expression can be written as

$$\gamma_s = \mathcal{D}^{-1}[\mathbf{G}\mathcal{D}(\gamma_s)\mathbf{x}_s]\mathbf{1}, \quad (26)$$

where $\mathbf{G} = \exp(-\mathbf{T}^{-1}\mathbf{A})$, with each $A_{jk} = [E_{jk} - 0.5(E_{jj} + E_{kk})]/R$. Vector \mathbf{x}_s denotes the mixture composition in terms of segment types and can be calculated by $\mathbf{x}_s = \mathbf{P}^t\mathcal{D}(\rho)\mathbf{x}$, where $\rho = \boldsymbol{\eta}(\boldsymbol{\eta}^t\mathbf{x})^{-1}$. In the case of pure substance i , \mathbf{x}_s is equal to the i -th row of matrix \mathbf{P} , so that $\Gamma = [(\mathbf{P}\circ\Gamma)\mathbf{G}^t]^{[-1]}$. The described model constitutes an important advance in the analytical treatment of condensed-phase mixtures. {LIN2002} It is very flexible and relies on exact expressions, while other models rely on approximate ones. Notice that Eq. (26) cannot be solved explicitly for γ_s . However, it is possible to solve it recursively by starting from $\gamma_s = \mathcal{D}^{-1}(\mathbf{G}\mathbf{x}_s)\mathbf{1}$ and then updating it towards self-consistency. Because direct iteration of Eq. (26) usually diverges by oscillation, it is preferable to iterate the equivalent expression $\gamma_s = 0.5\{\gamma_s + \mathcal{D}^{-1}[\mathbf{G}\mathcal{D}(\gamma_s)\mathbf{x}_s]\mathbf{1}\}$. {KLAMT1995} In the case of Γ , we can initialize it as $\Gamma = (\mathbf{P}\mathbf{G}^t)^{[-1]}$ and then iterate $\Gamma = 0.5\{\Gamma + [(\mathbf{P}\circ\Gamma)\mathbf{G}^t]^{[-1]}\}$ until convergence. Of course, as Γ does not depend on \mathbf{x} , it can be computed only once in a series of isothermal calculations.

We can also use our methodology to obtain expressions for $\mathbf{J}_x\ln\gamma_r = \mathcal{D}(\boldsymbol{\eta})\mathbf{P}\mathbf{J}_x\ln\gamma_s$. For this, we apply the logarithm and the operator \mathbf{J}_x to Eq. (26), followed by adequate rules of Table 2 and Table 3. The term $\mathbf{J}_x\ln\gamma_s$ appears in both sides of the resulting equation, but we can isolate it by resorting to matrix inversion. This is also possible in the derivation of $\mathbf{J}_T\ln\gamma_s$. However, temperature derivatives of $\ln\gamma$ also depend on $\mathbf{J}_T\ln\Gamma$, which cannot be fully isolated in the same way. As a possible solution, one can obtain such matrix recursively, starting from $\mathbf{J}_T\ln\Gamma = \mathbf{0}$. We applied our methodology to obtain expressions for g^E , $\ln\gamma$, $\mathbf{J}_x\ln\gamma$, $\mathbf{J}_T\ln\gamma$, h^E , and c_p^E regarding the described model, all of which are presented in Table 7.

We now explain how to obtain the necessary parameters \mathbf{v} , \mathbf{a} , $\boldsymbol{\eta}$, \mathbf{P} , \mathbf{A} , and a_0 . First, the type of a surface segment is discerned by its charge density σ acquired when the molecule is immersed in a perfect conductor. Here, we consider the details of COSMO-SAC {LIN2002} for illustration. In this case, one considers 51 possible segment types by dividing the interval from $-0.025 \text{ e}/\text{\AA}^2$ to $0.025 \text{ e}/\text{\AA}^2$ in 50 parts. Thus, the representative charge density of a type j is given by $\sigma_j = -0.025 + 0.001 \times (j - 0.5)$, resulting in a 51-dimensional vector $\boldsymbol{\sigma}$. The σ -profile of a given component is obtained from a COSMO calculation, available in many quantum chemistry packages such as MOPAC, Gaussian, DMol3, Turbomole, GAMESS etc. Such calculation entails placing the molecule inside a cavity formed in a perfect conductor and then obtaining the charge density distribution that emerges in the cavity's surface due to the presence of the molecule. We note that the mentioned packages perform this procedure automatically. In COSMO-SAC, a subsequent averaging procedure provides σ -profiles considering segments with a standard area $a_{\text{eff}} = 7.5 \text{ \AA}^2$. More details can be found in Ref. {LIN2002}. After doing the COSMO calculation for one molecule of every component, matrix \mathbf{P} is defined so that P_{ij} is the probability of finding a segment with charge density σ_j in a molecule of component i . That is, each row i of \mathbf{P} is the sigma profile of the corresponding molecule. Vectors \mathbf{a} and \mathbf{v} contain, for every component, the area and volume of the cavity mentioned above, which are also provided by the quantum chemistry software. With equally-sized segments, we have $\boldsymbol{\eta} = a_{\text{eff}}^{-1}\mathbf{a}$ and, therefore, $\rho = \boldsymbol{\theta}$. In addition, for $z = 10$, a standard area $a_0 = 79.53 \text{ \AA}^2$ is considered. Now, the only additional parameter needed to evaluate all

equations of Table 7 is the matrix \mathbf{A} . In COSMO-based models, E_{ij} is the energy involved in removing the screening charges of two paired segments of types i and j so as to restore the real condition of the system. The model that generally provides matrix \mathbf{A} can be expressed in matrix notation as

$$\mathbf{A} = (0.5\alpha'/R)[(\mathbf{1}\boldsymbol{\sigma}^t)^s]^{[2]} + (c_{hb}/R) \min(\mathbf{0}, \boldsymbol{\Phi} \circ \boldsymbol{\Phi}^t), \quad (27)$$

where $\boldsymbol{\Phi} = [\max(\mathbf{0}, \boldsymbol{\sigma} - \sigma_{hb}\mathbf{1}) + \min(\mathbf{0}, \boldsymbol{\sigma} + \sigma_{hb}\mathbf{1})]\mathbf{1}^t$ and R is the ideal-gas constant, while α' , σ_{hb} , and c_{hb} are universal parameters obtained by fitting experimental data. The last term in Eq. (27) is responsible for identifying hydrogen bonds. In COSMO-SAC, parameter values are $\alpha' = 9034.97 \text{ (kcal/mol)} \times (\text{\AA}^4/\text{e}^2)$, $\sigma_{hb} = 0.0084 \text{ e/\AA}^2$, and $c_{hb} = 85580 \text{ (kcal/mol)} \times (\text{\AA}^4/\text{e}^2)$.

Table 1. Rules for translation from conventional notation to matrix notation.

Index Notation	Matrix Notation
$c = \sum_{i=1}^n a_i$	$\mathbf{c} = \mathbf{1}^t \mathbf{a}$
$c_i = \sum_{j=1}^n A_{ij} b_j$	$\mathbf{c} = \mathbf{A} \mathbf{b}$
$C_{ij} = \sum_{k=1}^n A_{ik} B_{kj}$	$\mathbf{C} = \mathbf{A} \mathbf{B}$
$c_i = a_i^\alpha b_i$	$\mathbf{c} = \mathcal{D}^\alpha(\mathbf{a}) \mathbf{b}$
$C_{ij} = a_i^\alpha B_{ij}$	$\mathbf{c} = \mathcal{D}^\alpha(\mathbf{a}) \mathbf{B}$
$C_{ij} = B_{ij} a_j^\alpha$	$\mathbf{c} = \mathbf{B} \mathcal{D}^\alpha(\mathbf{a})$
$C_{ij} = A_{ij} B_{ij}$	$\mathbf{C} = \mathbf{A} \circ \mathbf{B}$
$C_{ij} = A_{ij}^\alpha$	$\mathbf{C} = \mathbf{A}^{[\alpha]}$
$C_{ij} = A_{ij} + A_{ji}$	$\mathbf{C} = \mathbf{A}^s$

Table 2. Algebraic rules involving the Diagonal Operator \mathcal{D} introduced in Table 1.

Case	Identity
Arbitrary exponents	$[\mathcal{D}^\alpha(\mathbf{a})]^\mathbf{t} = \mathcal{D}^\alpha(\mathbf{a})$
	$\mathcal{D}^\alpha(\mathbf{a})\mathcal{D}^\beta(\mathbf{b}) = \mathcal{D}^\beta(\mathbf{b})\mathcal{D}^\alpha(\mathbf{a})$
	$\mathcal{D}^\alpha(\mathbf{a})\mathcal{D}^\beta(\mathbf{a}) = \mathcal{D}^{\alpha+\beta}(\mathbf{a})$
	$\mathcal{D}^\alpha(\mathbf{a})\mathcal{D}^\beta(\mathbf{b}) = \mathcal{D}^\beta[\mathcal{D}^{\alpha/\beta}(\mathbf{a})\mathbf{b}]$
	$\mathbf{a}^\alpha\mathcal{D}^\beta(\mathbf{b}) = \mathcal{D}^\beta(\mathbf{a}^{\alpha/\beta}\mathbf{b})$
	$\ln[\mathcal{D}^\alpha(\mathbf{a})\mathbf{b}] = \alpha\ln\mathbf{a} + \ln\mathbf{b}$
$\alpha = 1$	$\mathcal{D}(\mathbf{a})\mathbf{b} = \mathcal{D}(\mathbf{b})\mathbf{a}$
	$\mathcal{D}(\mathbf{a})\mathbf{1} = \mathbf{a}$
	$\mathbf{1}^\mathbf{t}\mathcal{D}(\mathbf{a}) = \mathbf{a}^\mathbf{t}$
	$\mathcal{D}(\mathbf{a}) + \mathcal{D}(\mathbf{b}) = \mathcal{D}(\mathbf{a} + \mathbf{b})$
$\alpha = -1$	$\mathcal{D}^{-1}(\mathbf{a})\mathbf{a} = \mathbf{1}$
	$\mathbf{a}^\mathbf{t}\mathcal{D}^{-1}(\mathbf{a}) = \mathbf{1}^\mathbf{t}$

Table 3. Basic differentiation rules.

Rule	A: Scalar Functions of Scalar Variables	B: Vector Functions of Vector Variables	C: Matrix Functions of Scalar Variables
Constant	$\partial_x a = 0$	$\mathbf{J}_x \mathbf{a} = \mathbf{0}$	$\mathbf{J}_x \mathbf{A} = \mathbf{0}$
Self	$\partial_x x = 1$	$\mathbf{J}_x \mathbf{x} = \mathbf{I}$	—
Proportional	$\partial_x (ay) = a \partial_x y$	$\mathbf{J}_x (\mathbf{A} \mathbf{y}) = \mathbf{A} \mathbf{J}_x \mathbf{y}$	$\mathbf{J}_x (\mathbf{A} \mathbf{Y} \mathbf{B}) = \mathbf{A} (\mathbf{J}_x \mathbf{Y}) \mathbf{B}$
Logarithm	$\partial_x \ln y = y^{-1} \partial_x y$	$\mathbf{J}_x \ln \mathbf{y} = \mathcal{D}^{-1}(\mathbf{y}) \mathbf{J}_x \mathbf{y}$	$\mathbf{J}_x \ln \mathbf{Y} = \mathbf{Y}^{[-1]} \circ \mathbf{J}_x \mathbf{Y}$
Exponential	$\partial_x \exp y = (\exp y) \partial_x y$	$\mathbf{J}_x \exp \mathbf{y} = \mathcal{D}(\exp \mathbf{y}) \mathbf{J}_x \mathbf{y}$	$\mathbf{J}_x \exp \mathbf{Y} = (\exp \mathbf{Y}) \circ \mathbf{J}_x \mathbf{Y}$
Sum Rule	$\partial_x (y + z) = \partial_x y + \partial_x z$	$\mathbf{J}_x (\mathbf{y} + \mathbf{z}) = \mathbf{J}_x \mathbf{y} + \mathbf{J}_x \mathbf{z}$	$\mathbf{J}_x (\mathbf{Y} + \mathbf{Z}) = \mathbf{J}_x \mathbf{Y} + \mathbf{J}_x \mathbf{Z}$
Product Rule	$\partial_x (zy) = z \partial_x y + y \partial_x z$	$\mathbf{J}_x [\mathcal{D}^\alpha(\mathbf{z}) \mathbf{y}] = \mathcal{D}^\alpha(\mathbf{z}) \mathbf{J}_x \mathbf{y} + \dots + \alpha \mathcal{D}^{\alpha-1}(\mathbf{z}) \mathcal{D}(\mathbf{y}) \mathbf{J}_x \mathbf{z}$	$\mathbf{J}_x (\mathbf{Z} \mathbf{Y}) = \mathbf{Z} \mathbf{J}_x \mathbf{Y} + (\mathbf{J}_x \mathbf{Z}) \mathbf{Y}$ or $\mathbf{J}_x (\mathbf{Z} \circ \mathbf{Y}) = \mathbf{Z} \circ \mathbf{J}_x \mathbf{Y} + (\mathbf{J}_x \mathbf{Z}) \circ \mathbf{Y}$
Chain Rule	$\partial_x y = (\partial_z y)(\partial_x z)$	$\mathbf{J}_x \mathbf{y} = (\mathbf{J}_z \mathbf{y})(\mathbf{J}_x \mathbf{z})$	$\mathbf{J}_x \mathbf{Y} = (\mathbf{J}_z \mathbf{Y})(\partial_x \mathbf{z})$
Transpose	—	—	$\mathbf{J}_x \mathbf{Y}^t = (\mathbf{J}_x \mathbf{Y})^t$
Diagonal	—	—	$\mathbf{J}_x [\mathcal{D}^\alpha(\mathbf{y})] = \alpha \mathcal{D}^{\alpha-1}(\mathbf{y}) \mathcal{D}(\mathbf{J}_x \mathbf{y})$

Table 4. Non-Random Two-Liquid (NRTL) Model.

Parameters	\mathbf{A} ($n_c \times n_c$ matrix with $A_{ii} = 0$) $\boldsymbol{\alpha}$ ($n_c \times n_c$ matrix with $\alpha_{ii} = 0$ for all i and $\boldsymbol{\alpha} = \boldsymbol{\alpha}^t$)
Excess Gibbs Energy	$\mathbf{G} = \exp(-T^{-1} \boldsymbol{\alpha} \circ \mathbf{A})$ $\boldsymbol{\Lambda} = T^{-1} \mathbf{A} \circ \mathbf{G}$ $\mathbf{E} = \boldsymbol{\Lambda} \boldsymbol{\Phi}^{-1}(\mathbf{G}^t \mathbf{x})$ $g^E/RT = \mathbf{x}^t \mathbf{E} \mathbf{x}$
Activity Coefficients	$\mathbf{L} = \mathbf{G} \boldsymbol{\Phi}^{-1}(\mathbf{G}^t \mathbf{x})$ $\ln \gamma = [\mathbf{E}^s - \mathbf{L} \boldsymbol{\Phi}(\mathbf{x}) \mathbf{E}^t] \mathbf{x}$
Jacobian Matrix of Activity Coefficients	$\mathbf{J}_x \ln \gamma = \{[\mathbf{E} - \mathbf{L} \boldsymbol{\Phi}(\mathbf{E}^t \mathbf{x})][\mathbf{I} - \boldsymbol{\Phi}(\mathbf{x}) \mathbf{L}^t]\}^s$
Excess Enthalpy	$\dot{\mathbf{G}} = T^{-1} \boldsymbol{\alpha} \circ \boldsymbol{\Lambda}$ $\dot{\boldsymbol{\Lambda}} = T^{-1}(\mathbf{A} \circ \dot{\mathbf{G}} - \boldsymbol{\Lambda})$ $\dot{\mathbf{E}} = [\dot{\boldsymbol{\Lambda}} - \mathbf{E} \boldsymbol{\Phi}(\dot{\mathbf{G}}^t \mathbf{x})] \boldsymbol{\Phi}^{-1}(\mathbf{G}^t \mathbf{x})$ $h^E/RT = -T \mathbf{x}^t \dot{\mathbf{E}} \mathbf{x}$
Temperature Derivatives of Activity Coefficients	$\dot{\mathbf{L}} = [\dot{\mathbf{G}} - \mathbf{L} \boldsymbol{\Phi}(\dot{\mathbf{G}}^t \mathbf{x})] \boldsymbol{\Phi}^{-1}(\mathbf{G}^t \mathbf{x})$ $\mathbf{J}_T \ln \gamma = [\dot{\mathbf{E}}^s - \dot{\mathbf{L}} \boldsymbol{\Phi}(\mathbf{x}) \mathbf{E}^t - \mathbf{L} \boldsymbol{\Phi}(\mathbf{x}) \dot{\mathbf{E}}^t] \mathbf{x}$
Excess Heat Capacity	$\ddot{\mathbf{G}} = T^{-1}(\boldsymbol{\alpha} \circ \dot{\boldsymbol{\Lambda}} - \dot{\mathbf{G}})$ $\ddot{\boldsymbol{\Lambda}} = T^{-1}(\mathbf{A} \circ \ddot{\mathbf{G}} - 2\dot{\boldsymbol{\Lambda}})$ $\ddot{\mathbf{E}} = [\ddot{\boldsymbol{\Lambda}} - 2\dot{\mathbf{E}} \boldsymbol{\Phi}(\dot{\mathbf{G}}^t \mathbf{x}) - \mathbf{E} \boldsymbol{\Phi}(\ddot{\mathbf{G}}^t \mathbf{x})] \boldsymbol{\Phi}^{-1}(\mathbf{G}^t \mathbf{x})$ $c_p^E/R = -T \mathbf{x}^t (2\dot{\mathbf{E}} + T\ddot{\mathbf{E}}) \mathbf{x}$

Table 5. Universal Quasi-Chemical (UNIQUAC) Model.

Parameters	\mathbf{r} and \mathbf{q} (n_c -dimensional vectors) \mathbf{A} ($n_c \times n_c$ matrix with all $A_{ii} = 0$)
Auxiliaries	$\mathbf{m} = \mathbf{1} - 5\mathbf{q}$, $\boldsymbol{\varepsilon} = \mathbf{q} + \mathcal{D}(\mathbf{q})\ln\mathbf{q}$
Excess Gibbs Energy	$\boldsymbol{\phi} = (\mathbf{r}^t\mathbf{x})^{-1}\mathbf{r}$ $\boldsymbol{\theta} = (\mathbf{q}^t\mathbf{x})^{-1}\mathbf{q}$ $\boldsymbol{\Lambda} = \mathcal{D}(\mathbf{q})\exp(-T^{-1}\mathbf{A})$ $g^E/RT = \mathbf{x}^t[\mathcal{D}(\mathbf{m})\ln\boldsymbol{\phi} + 4\mathcal{D}(\mathbf{q})\ln\boldsymbol{\theta} + \boldsymbol{\varepsilon} - \mathcal{D}(\mathbf{q})\ln(\boldsymbol{\Lambda}^t\mathbf{x}) - \mathbf{q}]$
Activity Coefficients	$\mathbf{L} = \boldsymbol{\Lambda}\mathcal{D}^{-1}(\boldsymbol{\Lambda}^t\mathbf{x})$ $\ln\boldsymbol{\gamma} = \mathcal{D}(\mathbf{m})\ln\boldsymbol{\phi} + 4\mathcal{D}(\mathbf{q})\ln\boldsymbol{\theta} - \boldsymbol{\phi}\mathbf{m}^t\mathbf{x} + \mathbf{m} + \boldsymbol{\varepsilon} - \mathcal{D}(\mathbf{q})\ln(\boldsymbol{\Lambda}^t\mathbf{x}) - \mathbf{L}\mathcal{D}(\mathbf{q})\mathbf{x}$
Jacobian Matrix of Activity Coefficients	$\mathbf{J}_x\ln\boldsymbol{\gamma} = -[\mathbf{m}\boldsymbol{\phi}^t + \mathbf{L}\mathcal{D}(\mathbf{q})]^s - 4\mathbf{q}\boldsymbol{\theta}^t + \boldsymbol{\phi}\mathbf{m}^t\mathbf{x}\boldsymbol{\phi}^t + \mathbf{L}\mathcal{D}(\mathbf{q})\mathcal{D}(\mathbf{x})\mathbf{L}^t$
Excess Enthalpy	$\dot{\mathbf{A}} = T^{-2}\mathbf{A} \circ \boldsymbol{\Lambda}$ $\mathbf{E} = \dot{\mathbf{A}}\mathcal{D}(\mathbf{q})\mathcal{D}^{-1}(\boldsymbol{\Lambda}^t\mathbf{x})$ $h^E/RT = T\mathbf{x}^t\mathbf{E}\mathbf{x}$
Temperature Derivatives of Activity Coefficients	$\mathbf{J}_T\ln\boldsymbol{\gamma} = \{\mathbf{L}\mathcal{D}(\mathbf{x})\mathbf{E}^t - \mathbf{E}^s\}\mathbf{x}$
Excess Heat Capacity	$\ddot{\mathbf{A}} = T^{-2}\mathbf{A} \circ \dot{\mathbf{A}} - 2T^{-1}\dot{\mathbf{A}}$ $\dot{\mathbf{E}} = [\ddot{\mathbf{A}}\mathcal{D}(\mathbf{q}) - \mathbf{E}\mathcal{D}(\dot{\mathbf{A}}^t\mathbf{x})]\mathcal{D}^{-1}(\boldsymbol{\Lambda}^t\mathbf{x})$ $c_p^E/R = T\mathbf{x}^t(2\mathbf{E} - T\dot{\mathbf{E}})\mathbf{x}$

Table 6. Universal Functional-Group Activity Coefficients (UNIFAC) Model.

Parameters	\mathbf{R} and \mathbf{Q} (n_g -dimensional vectors) \mathbf{v} ($n_c \times n_g$ matrix) \mathbf{A} ($n_g \times n_g$ matrix with all $A_{ii} = 0$)
Auxiliaries	$\mathbf{r} = \mathbf{vR}$ $\mathbf{q} = \mathbf{vQ}$ $\mathbf{m} = \mathbf{1} - 5\mathbf{q}$ $\mathbf{\Omega} = \mathbf{vD(Q)}$
Excess Gibbs Energy	$\mathbf{G} = \exp(-T^{-1}\mathbf{A})$ $\mathbf{\Lambda} = \mathbf{\Omega G}$ $\mathbf{U} = \mathbf{\Omega} \circ \mathbf{\Lambda}^{[-1]}$ $\mathbf{\epsilon} = [\mathbf{\Omega} \circ (\ln \mathbf{\Lambda} + \mathbf{U G}^t)] \mathbf{1}$ $\mathbf{\phi} = \mathbf{r}(\mathbf{r}^t \mathbf{x})^{-1}$ $\mathbf{\theta} = \mathbf{q}(\mathbf{q}^t \mathbf{x})^{-1}$ $g^E/RT = \mathbf{x}^t [\mathbf{D(m)} \ln \mathbf{\phi} + 4\mathbf{D(q)} \ln \mathbf{\theta} + \mathbf{\epsilon} - \mathbf{\Omega} \ln(\mathbf{\Lambda}^t \mathbf{x}) - \mathbf{q}]$
Activity Coefficients	$\mathbf{L} = \mathbf{\Lambda D}^{-1}(\mathbf{\Lambda}^t \mathbf{x})$ $\ln \gamma = \mathbf{D(m)} \ln \mathbf{\phi} + 4\mathbf{D(q)} \ln \mathbf{\theta} - \mathbf{\phi m}^t \mathbf{x} + \mathbf{m} + \mathbf{\epsilon} - \mathbf{\Omega} \ln(\mathbf{\Lambda}^t \mathbf{x}) - \mathbf{L \Omega}^t \mathbf{x}$
Jacobian Matrix of Activity Coefficients	$\mathbf{J_x} \ln \gamma = -(\mathbf{m \phi}^t + \mathbf{L \Omega}^t)^s - 4\mathbf{q \theta}^t + \mathbf{\phi m}^t \mathbf{x \phi}^t + \mathbf{L D(\Omega^t x) L}^t$
Excess Enthalpy	$\mathbf{\dot{G}} = T^{-2} \mathbf{A} \circ \mathbf{G}$ $\mathbf{V} = (\mathbf{\Omega \dot{G}}) \circ \mathbf{\Lambda}^{[-1]}$ $\mathbf{\dot{\epsilon}} = \{\mathbf{\Omega} \circ [\mathbf{U \dot{G}}^t - (\mathbf{U} \circ \mathbf{V}) \mathbf{G}^t + \mathbf{V}]\} \mathbf{1}$ $\mathbf{E} = \mathbf{\Omega \dot{G} D}^{-1}(\mathbf{\Lambda}^t \mathbf{x})$ $h^E/RT = T \mathbf{x}^t (\mathbf{E \Omega}^t \mathbf{x} - \mathbf{\dot{\epsilon}})$
Temperature Derivatives of Activity Coefficients	$\mathbf{J_T} \ln \gamma = \mathbf{\dot{\epsilon}} + [\mathbf{L D(\Omega^t x) E}^t - (\mathbf{E \Omega}^t)^s] \mathbf{x}$
Excess Heat Capacity	$\mathbf{\ddot{G}} = T^{-2} \mathbf{A} \circ \mathbf{\dot{G}} - 2T^{-1} \mathbf{\dot{G}}$ $\mathbf{W} = (\mathbf{\Omega \ddot{G}}) \circ \mathbf{\Lambda}^{[-1]} - \mathbf{U}^{[2]}$ $\mathbf{\ddot{\epsilon}} = \{\mathbf{\Omega} \circ [\mathbf{U \ddot{G}}^t - 2(\mathbf{U} \circ \mathbf{V}) \mathbf{\dot{G}}^t - 2(\mathbf{U} \circ \mathbf{W}) \mathbf{G}^t + \mathbf{W}]\} \mathbf{1}$ $\mathbf{\dot{E}} = \mathbf{\Omega \ddot{G} D}^{-1}(\mathbf{\Lambda}^t \mathbf{x}) - \mathbf{E D(E^t x)}$ $c_p^E/R = T \mathbf{x}^t [(2\mathbf{E} - T\mathbf{\dot{E}}) \mathbf{\Omega}^t \mathbf{x} - (2\mathbf{\dot{\epsilon}} + T\mathbf{\ddot{\epsilon}})]$

Table 7. Conductor-like Screening Model for Real Solvents (COSMO).

Parameters	\mathbf{r} , \mathbf{q} , and \mathbf{n} (n_c -dimensional vectors) \mathbf{P} ($n_c \times n_t$ matrix with $\mathbf{P}\mathbf{1} = \mathbf{1}$) \mathbf{A} ($n_t \times n_t$ matrix with all $A_{ii} = 0$)
Auxiliary	$\mathbf{m} = \mathbf{1} - 5\mathbf{q}$
Excess Gibbs Energy	$\mathbf{G} = \exp(-\mathbf{T}^{-1}\mathbf{A})$ $\mathbf{\Gamma} = 0.5\{\mathbf{\Gamma} + [(\mathbf{P}\circ\mathbf{\Gamma})\mathbf{G}^t]^{[-1]}\}$ recursive from $\mathbf{\Gamma} = (\mathbf{P}\mathbf{G}^t)^{[-1]}$ $\boldsymbol{\phi} = \mathbf{r}(\mathbf{r}^t\mathbf{x})^{-1}$ $\boldsymbol{\theta} = \mathbf{q}(\mathbf{q}^t\mathbf{x})^{-1}$ $\boldsymbol{\rho} = \mathbf{n}(\mathbf{n}^t\mathbf{x})^{-1}$ $\mathbf{x}_S = \mathbf{P}^t\boldsymbol{\mathcal{D}}(\boldsymbol{\rho})\mathbf{x}$ $\boldsymbol{\gamma}_S = 0.5\{\boldsymbol{\gamma}_S + \boldsymbol{\mathcal{D}}^{-1}[\mathbf{G}\boldsymbol{\mathcal{D}}(\boldsymbol{\gamma}_S)\mathbf{x}_S]\mathbf{1}\}$ recursive from $\boldsymbol{\gamma}_S = \boldsymbol{\mathcal{D}}^{-1}(\mathbf{G}\mathbf{x}_S)\mathbf{1}$ $g^E/RT = \mathbf{x}^t\{\boldsymbol{\mathcal{D}}(\mathbf{m})\ln\boldsymbol{\phi} + 5\boldsymbol{\mathcal{D}}(\mathbf{q})\ln\boldsymbol{\theta} + \boldsymbol{\mathcal{D}}(\mathbf{n})[\mathbf{P}\ln\boldsymbol{\gamma}_S - (\mathbf{P}\circ\ln\mathbf{\Gamma})\mathbf{1}]\}$
Activity Coefficients	$\ln\boldsymbol{\gamma} = \boldsymbol{\mathcal{D}}(\mathbf{m})\ln\boldsymbol{\phi} + 5\boldsymbol{\mathcal{D}}(\mathbf{q})\ln\boldsymbol{\theta} - \boldsymbol{\phi}\mathbf{m}^t\mathbf{x} + \mathbf{m} + \boldsymbol{\mathcal{D}}(\mathbf{n})[\mathbf{P}\ln\boldsymbol{\gamma}_S - (\mathbf{P}\circ\ln\mathbf{\Gamma})\mathbf{1}]$
Jacobian Matrix of Activity Coefficients	$\mathbf{E} = \boldsymbol{\mathcal{D}}^{-1}(\boldsymbol{\gamma}_S) + \mathbf{G}\boldsymbol{\mathcal{D}}(\boldsymbol{\gamma}_S)\boldsymbol{\mathcal{D}}(\mathbf{x}_S)$ $\mathbf{J}_x\ln\boldsymbol{\gamma} = -(\mathbf{m}\boldsymbol{\phi}^t)^s - 5\mathbf{q}\boldsymbol{\theta}^t + \boldsymbol{\phi}\mathbf{m}^t\mathbf{x}\boldsymbol{\phi}^t + \boldsymbol{\mathcal{D}}(\mathbf{n})\mathbf{P}\mathbf{E}^{-1}\mathbf{G}\boldsymbol{\mathcal{D}}(\boldsymbol{\gamma}_S)[\mathbf{x}_S\boldsymbol{\rho}^t - \mathbf{P}^t\boldsymbol{\mathcal{D}}(\boldsymbol{\rho})]$
Excess Enthalpy	$\dot{\mathbf{G}} = \mathbf{T}^{-2}\mathbf{A}\circ\mathbf{G}$ $\mathbf{U} = \mathbf{P}\circ\mathbf{\Gamma}$ $\mathbf{V} = -\mathbf{\Gamma}\circ(\mathbf{U}\dot{\mathbf{G}}^t)$ $\boldsymbol{\Omega} = 0.5\{\mathbf{V} - \mathbf{\Gamma}\circ[(\mathbf{U}\circ\boldsymbol{\Omega})\mathbf{G}^t]\}$ recursive from $\boldsymbol{\Omega} = \mathbf{V}$ $h^E/RT = \mathbf{T}\mathbf{x}^t\boldsymbol{\mathcal{D}}(\mathbf{n})[\mathbf{P}\boldsymbol{\alpha} + (\mathbf{P}\circ\boldsymbol{\Omega})\mathbf{1}]$
Temperature Derivatives of Activity Coefficients	$\boldsymbol{\alpha} = \mathbf{E}^{-1}\dot{\mathbf{G}}\boldsymbol{\mathcal{D}}(\boldsymbol{\gamma}_S)\mathbf{x}_S$ $\mathbf{J}_T\ln\boldsymbol{\gamma} = -\boldsymbol{\mathcal{D}}(\mathbf{n})[\mathbf{P}\boldsymbol{\alpha} + (\mathbf{P}\circ\boldsymbol{\Omega})\mathbf{1}]$
Excess Heat Capacity	$\ddot{\mathbf{G}} = \mathbf{T}^{-2}\mathbf{A}\circ\dot{\mathbf{G}} - 2\mathbf{T}^{-1}\dot{\mathbf{G}}$ $\mathbf{W} = \boldsymbol{\Omega}^{[2]} - \mathbf{\Gamma}\circ[\mathbf{U}\ddot{\mathbf{G}}^t + 2(\mathbf{U}\circ\boldsymbol{\Omega})\dot{\mathbf{G}}^t + (\mathbf{U}\circ\boldsymbol{\Omega}^{[2]})\mathbf{G}^t]$ $\dot{\boldsymbol{\Omega}} = 0.5\{\mathbf{W} - \mathbf{\Gamma}\circ[(\mathbf{U}\circ\dot{\boldsymbol{\Omega}})\mathbf{G}^t]\}$ recursive from $\dot{\boldsymbol{\Omega}} = \mathbf{W}$ $\dot{\boldsymbol{\alpha}} = \mathbf{E}^{-1}\{[\ddot{\mathbf{G}} - 2\dot{\mathbf{G}}\boldsymbol{\mathcal{D}}(\boldsymbol{\alpha}) + \mathbf{G}\boldsymbol{\mathcal{D}}^2(\boldsymbol{\alpha})]\boldsymbol{\mathcal{D}}(\boldsymbol{\gamma}_S)\mathbf{x}_S - \boldsymbol{\mathcal{D}}^{-1}(\boldsymbol{\gamma}_S)\boldsymbol{\mathcal{D}}(\boldsymbol{\alpha})\boldsymbol{\alpha}\}$ $c_p^E/R = \mathbf{T}\mathbf{x}^t\boldsymbol{\mathcal{D}}(\mathbf{n})\{\mathbf{P}(2\boldsymbol{\alpha} + \mathbf{T}\dot{\boldsymbol{\alpha}}) + [\mathbf{P}\circ(2\boldsymbol{\Omega} + \mathbf{T}\dot{\boldsymbol{\Omega}})]\mathbf{1}\}$