A B A K Ó S
Instituto de Ciências Exatas e Informática
PUC Minas

Dimensionality Reduction through Spectral Analysis in Calculating Dew Points: A Bridge between Linear Algebra and Phase Equilibrium Problems*

Redução de Dimensionalidade via Análise Espectral no Cálculo de Pontos de Orvalho: Uma Ponte entre Álgebra Linear e Problemas de Equilíbrio de Fases

Loena Marins do Couto[1]
Gustavo Mendes Platt[2]

Abstract

Currently, several accurate mathematical models predict the relationship between properties of mixtures, such as equations of state and mixing rules. Used in phase equilibrium problems and oil reservoir simulations, these models play a fundamental role. A reduction theorem has recently been formulated, establishing conditions for reducing the dimensionality of the phase equilibrium equations of multicomponent mixtures, applying concepts of Linear Algebra (such as Spectral Theorem and Frobenius Norm) commonly seen in engineering courses. This study presents the main ideas, deducing a reduction formulation based on a cubic equation of state and classical mixing rules, highlighting the simplicity of the formulation and minimizing the approximation error. Numerical techniques, mathematical results, and algorithms are described to facilitate development. Additionally, a customized framework is presented to detail methodologies and algorithms. Numerical experiments illustrate the efficiency of the reduction technique in calculating pressures and isothermal dew point compositions, significantly reducing processing time. These advances have the potential to benefit industries and complex processes that rely on precise control of mixture properties, such as process optimization and oil reservoir simulation. In summary, dimensionality reduction stands out as an important contribution to mixture thermodynamics, providing efficiency and accuracy in complex engineering problems.

Keywords: Dew point. Thermodynamics. Spectral analysis. Frobenius norm.

---

[*] Submetido em 14/08/2024 - Aceito em 29/06/2025

[1] Professor, Computational Modeling, Polytechnic Institute, Rio de Janeiro State University, Brasil - loena@iprj.uerj.br.

[2] Professor, School of Chemistry and Food, Federal University of Rio Grande, Brasil - gmplatt@furg.br.

Resumo

Atualmente, diversos modelos matemáticos precisos preveem a relação entre propriedades de misturas, como equações de estado e regras de mistura. Utilizados em problemas de equilíbrio de fases e simulações de reservatórios de petróleo, esses modelos desempenham papel fundamental. Um teorema de redução foi formulado recentemente, estabelecendo condições para a redução da dimensionalidade das equações de equilíbrio de fases de misturas multicomponentes, aplicando conceitos de Álgebra Linear (como Teorema Espectral e Norma de Frobenius) comumente vistos em cursos de engenharia . Esse estudo apresenta as ideias principais, deduzindo uma formulação de redução com base em uma equação de estado cúbica e regras clássicas de mistura, destacando a simplicidade da formulação e minimizando o erro de aproximação. Técnicas numéricas, resultados matemáticos e algoritmos são descritos para facilitar o desenvolvimento. Além disso, é apresentado um framework customizado para detalhar metodologias e algoritmos. Experimentos numéricos ilustram a eficiência da técnica de redução no cálculo de pressões e composições isotérmicas de ponto de orvalho, reduzindo significativamente o tempo de processamento. Esses avanços têm potencial para beneficiar indústrias e processos complexos que dependem do controle preciso de propriedades de misturas, como otimização de processos de separação e simulação de reservatórios de petróleo. Em resumo, a redução de dimensionalidade destaca-se como uma contribuição importante à termodinâmica de misturas, proporcionando eficiência e precisão em problemas complexos de engenharia.

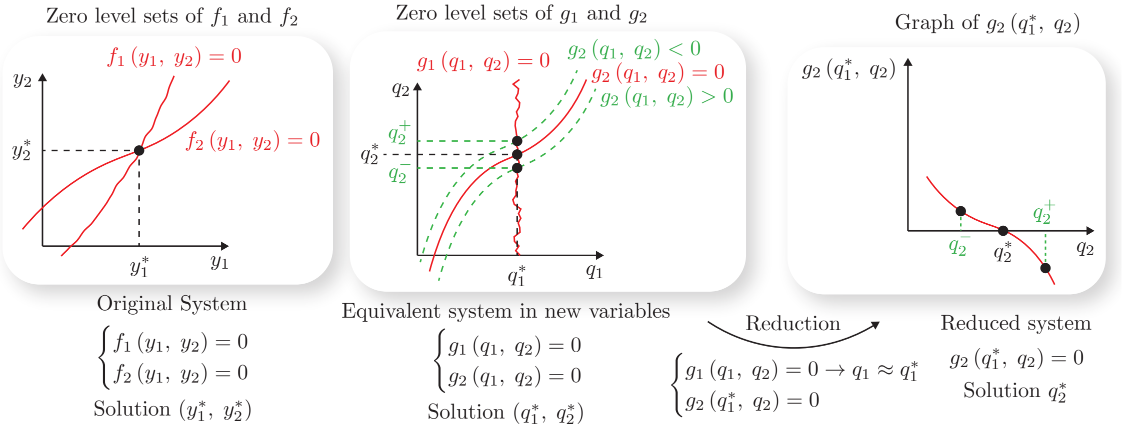
Palavras-chave: Ponto de orvalho. Termodinâmica. Análise espectral. Norma de Frobenius.

1 INTRODUCTION

This paper deals with the reduction of the system of equations that governs the phase equilibrium of non-reactive systems (absence of chemical reaction). Reductions in computation times can be extremely significant in multiphase flow problems (Matheis; Hickel, 2016) or in the simulation, design and optimization of separators, as, for instance, in flash operations (Watson *et al.*, 2017) and distillation columns. Particularly, some well-established chemical process simulation tools employ equations of state models in these unit operations (Mathias; Boston; Watanasiri, 1984).

Several reduction procedures applied to equilibrium problems have been proposed by Michelsen (1986), Nichita e Minescu (2004), Nichita (2006) and Nichita (2008). In terms of the phase equilibrium calculation, this consists of replacing the traditional variables (number of mols, molar fractions, partition coefficients) by a few functions of them. Therefore, the solution is no longer in the compositional space, but in the reduced space for the new variables introduced. Schematically, a reduction procedure can be summarized as detailed in Figure 1, using a system with two nonlinear equations. The original system exhibits two nonlinear equations, $f_1(y_1, y_2)$ and $f_2(y_1, y_2)$, to be zeroed. This system can be approximated by the nonlinear system $g_1(q_1, q_2)$ and $g_2(q_1, q_2)$. Considering that $q_1 \approx q_1^*$, the nonlinear system can be reduced to a single nonlinear equation $g_2(q_1^*, q_2) = 0$.

Figure 1 – A schematic representation of a reduction procedure in a nonlinear system



Source: The authors.

Hendriks (1988), using a formulation based on Gibbs free energy, proposes a reduction theorem to obtain approximations of the equilibrium equations. Here, following Nichita (2006), Nichita (2008), we present a simpler and more practical model using cubic state equations to formulate the thermodynamic equilibrium, together with Hendriks (1988) proposal of considering linear combinations of the composition as reduction parameters. We however disentangle the computation by highlighting the simple mathematical structure of the formulation, emphasizing the low rank approximation underlying the dimensionality reduction procedure. This is developed below and the mathematical structures is painstakingly explained. The reduced equilibrium equations system is formulated as a search for the zeros of a vector function in a space of smaller dimension. Furthermore, we present the numerical techniques and mathematical results on which the methods are based upon, and algorithms and implementation details are explained. Finally, we compare the results with the classical approach showing that this dimensionality reduction method is an accurate and faster algorithm.

From an educational point of view, Linear Algebra courses are usually placed in the freshman year of a 4 or 5-year undergraduate Chemical Engineering programs. On the other hand, phase equilibrium problems are subject of a second course in Engineering Thermodynamics, usually in the junior year. We consider that the methodology detailed here can be an useful tool for undergraduate students in Chemical Engineering, providing a clear relation between some important and non-trivial Linear Algebra concepts and its applications in Thermodynamics. As detailed by Ramkrishna e Amundson (2004), matrix theory is extremely important in Chemical Engineering field, for instance, in the simulation of distillation columns; on the other hand, the spectral theorem is a less explored subject in introductory Linear Algebra courses. In summary, from an educational perspective, our approach establishes a bridge between certain topics in Linear Algebra such as the Spectral Theorem, Frobenius Norm and rank reduction, which can be found in Strang (2000), and others related to Thermodynamics, such as phase equilibrium in multicomponent systems (Smith *et al.*, 2017).

As pointed out previously, it should be abundantly clear that the application of the spectral decomposition in the construction of phase envelopes was discussed, for instance, by Nichita (2008). In this work, our proposal is to present a detailed description of the mathematical and computational aspects involved in the implementation of such equilibrium problems, enabling the development of computational methods for them in undergraduate Chemical Engineering courses. Furthermore, we address the reduction technique providing a link to the low-rank approximations in the Frobenius norm. Finally, we present the use of a tailor-made computational framework, developed in Scilab 5.5 language (Enterprises, 2022), devoted to the demonstration of the computational gains (elapsed times) related to the use of the spectral decomposition. The application allows the user to choose the mixture to be studied, and the results are displayed in a graphical form. The program also permits the selection of the tolerance for some important variables – which will be discussed in the next sections, as follows: the quantity of non-null eigenvalues, and the minimum and maximum temperatures for the calculation of the dew points – using a graphical user interface (GUI). Furthermore, the computational code presents the non-null eigenvalues and the corresponding eigenvectors, as well as the elapsed time. The program can be easily adapted to other mixtures, depending on the critical properties, acentric factor and, obviously, the matrix of binary interaction parameters. This framework can be an useful tool in undergraduate courses in Thermodynamics, illustrating the main ideas of the spectral reduction in a simple and interactive way.

The organization of this work is as follows. Section 2 presents the vapor-liquid equilibrium equations and shows how the reduced equilibrium equations can be obtained by considering an optimal low rank approximation of the binary interaction matrix. The numerical methods employed to solve the reduced equations are also discussed in Section 2 while the algorithms and implementation details constitutes Section 3. Numerical simulations, using the developed computational framework, are presented in Section 4. Finally some conclusions are drawn in Section 5.

2 THEORETICAL BACKGROUND

In this Section, we describe the thermodynamic modelling of the phase equilibrium. We also discuss how to approximate the matrix of binary interaction parameters in the Frobenius norm. Finally, we present a brief description of the iterative scheme employed to solve the vapor-liquid equilibrium.

The main motivation of this Section is to provide a self-contained – but concise – structure of the implementation of the numerical techniques in an undergraduate setting. The description of some common subjects in phase equilibrium, such as the definition of the energy parameter, are strictly necessary, since some quantities will be the variables subject to reduction techniques. Furthermore, we present a scheme of the dependencies of the variables, as defined by the equations, in order to facilitate the understanding of the structure of the computational framework.

2.1 Vapor-liquid Equilibrium

Vapor-liquid equilibrium equations for a closed system with N components at temperature T, pressure P, and volume V, can be formulated as the equality of chemical potentials in both phases, together with mass conservation, for all components.

The chemical potential μ_i relates to the fugacity f_i of the component i of the mixture by the equation (see (Smith *et al.*, 2017))

$$d\mu_i = RTd\ln f_i \ , \ \ i = 1, \ldots, N \ , \tag{1}$$

where R is the universal gas constant. Therefore, by appropriate integration and simplifications, the vapor-liquid equilibrium equations can be rewritten as the equality of fugacities of the liquid and vapor phases

$$f_i^L = f_i^V, \tag{2}$$

with, for instance, f_i^L the fugacity of the i component in the liquid phase.

The fugacity coefficient of the component i is defined by the quotient

$$\varphi_i = \frac{f_i}{Pz_i}, \tag{3}$$

where the variable z_i represents the molar fraction of the i^{th} component. In a vapor-liquid mixture, instead of z_i, x_i is used for the composition in the liquid phase, and y_i for the composition in the vapor phase. Then, by Equation (3), the fugacities are written $f_i^L = \varphi_i^L P x_i$, $f_i^V = \varphi_i^V P y_i$. Thus the isofugacity equation, Equation (2), is rewritten as

$$\varphi_i^L x_i = \varphi_i^V y_i. \tag{4}$$

The coefficients of fugacity are obtained with an equation of state and here we use the Peng-Robinson equation (PR), written in its polynomial form as

$$Z^3 + (B-1)Z^2 + (A - 3B - 2B^2)Z + (B^3 + B^2 - AB) = 0, \tag{5}$$

with $Z = \frac{PV}{nRT}$ the compressibility factor, and n the total number of mols (Smith *et al.*, 2017). This equation is frequently used in chemical and oil industry simulations because of its relatively simple shape and good accuracy. Is this case, the logarithm of the coefficient of fugacity is given by (Smith *et al.*, 2017)

$$\begin{aligned} \ln \varphi_i &= \frac{B_i}{B}(Z-1) - \ln(Z-B) \\ &+ \frac{A}{2.282B}\left(\frac{B_i}{B} - 2\frac{\psi_i}{a_m}\right)\ln\left(\frac{Z+2.414B}{Z-0.414B}\right). \end{aligned} \tag{6}$$

Here, the factor ψ_i is

$$\psi_i = \sum_{j=1}^{N} a_{ij} z_j, \tag{7a}$$

and the compressibility factor, Z, assumes the value of one of the roots of Equation (5), the smallest in the case of the liquid phase, and the largest in the case of the vapor phase. The variables A, B, B_i, a_{ij}, and the energy parameter a_m are described below.

The energy parameter a_m and the covolume b_m of the mixture are given by the van der Waals–1 mixture rules, (Smith *et al.*, 2017)

$$a_m = \sum_{i=1}^{N}\sum_{j=1}^{N} a_{ij} z_i z_j \ , \ b_m = \sum_{i=1}^{N} b_i z_i \ , \tag{7b}$$

where, by the combination rules,

$$a_{ij} = \sqrt{a_i a_j}(1 - c_{ij}), \ i,j = 1,\ldots,N \ , \tag{7c}$$

following

$$a_i = \frac{\Omega_a (RT_{c_i})^2}{P_{c_i}} \alpha_i(T), \ \ b_i = \frac{\Omega_b R T_{c_i}}{P_{c_i}}, \tag{7d}$$

$\alpha_i(T) = [1 + S(1 - T_r^{0.5})]^2$, $S = S_0 + S_1\omega_i + S_2\omega_i^2$ and $T_r = \frac{T}{T_{c_i}}$. The coefficients c_{ij} are called the binary interaction parameters between components i and j of the mixture. The parameters a_i and b_i are the energy parameter and the covolume of the pure component i, respectively. The parameters for PR equation are given in Table 1.

Table 1 – Some of PR parameters

S_0	S_1	S_2	Ω_a	Ω_b
0.37464	1.54226	-0.26992	0.45724	0.07780

Source: Research data (Smith *et al.*, 2017).

The coefficients in PR equation, Equation (5), depend on A and B, with

$$A = \frac{a_m P}{(RT)^2}, \quad B = \frac{b_m P}{RT}, \tag{7e}$$

and the parameters B_i which show up in the equation for the coefficient of fugacity, Equation (6), are

$$B_i = \frac{b_i P}{RT}. \tag{7f}$$

Moreover, in order to compute the fugacity coefficients, say of the liquid phase, to use in Equation (4), the energy and covolume parameters should be computed with composition of the liquid phase, and the compressibility factor to be plugged in Equation (6) is the smallest root of Equation (5). Similarly, for the vapor phase, a_m and b_m should be computed with $z_i = y_i$ and the largest root of Equation (5) should be used in Equation (6).

Therefore, knowing the roots of Equation (5) it is possible at first to calculate the coefficients of fugacity of liquid and vapor with Equation (6). The thermodynamic equilibrium occurs if these coefficients produce equal fugacities as verified by resorting to Equation (4).

The computation of the equilibrium equations, Equation (4), is quite convolved. First, note that the main unknowns to be determined are the composition of the phases and the common pressure and temperature. Furthermore, the fugacity coefficients, Equation (6), depend directly on a_m, A, B, B_i, ψ_i, and Z. Also, by Equation (5), Z depends on A and B. Finally, A, B, B_i depend on a_m, b_m, P and T (Equations (7e) - (7f)). Table 2 exhibits the full dependencies, focusing on the fugacity coefficients which ultimately depend on $\mathbf{z}$, P and T, as can be seen by inspecting Equations (5) to (7).

Table 2 – Scheme of dependency: $\alpha \leftarrow (a, b, c)\ \ \beta$ means that variables α depend on variables β which is justified by Equations (a, b, c) or rows (a, b, c) of this table

Row		Equations		Rows		Rows		Rows	
1	φ_i	$\longleftarrow$ Equation (6)	a_m, A, B, B_i, ψ_i, Z	$\longleftarrow$ (2,5,6)	a_m, b_m, A, B, $\mathbf{z}$, P, T	$\longleftarrow$ (3,4)	a_m, b_m, $\mathbf{z}$, P, T	$\longleftarrow$ (7,8)	$\mathbf{z}$, P, T
2	Z	$\longleftarrow$ Equation (5)	A, B	$\longleftarrow$ (3,4)	a_m, b_m, P, T	$\longleftarrow$ (7,8)	$\mathbf{z}$, P, T		
3	A	$\longleftarrow$ Equation (7e)	a_m, P, T	$\longleftarrow$ (7)	$\mathbf{z}$, P, T				
4	B	$\longleftarrow$ Equation (7e)	b_m, P, T	$\longleftarrow$ (8)	$\mathbf{z}$, P, T				
5	B_i	$\longleftarrow$ Equation (7f)	P, T						
6	ψ_i	$\longleftarrow$ Equations (7a, 7c, 7d)	$\mathbf{z}$, T						
7	a_m	$\longleftarrow$ Equations (7b, 7d)	$\mathbf{z}$, T						
8	b_m	$\longleftarrow$ Equations (7b, 7d)	$\mathbf{z}$						

Source: The authors.

2.2 Approximation of the binary interaction matrix in the Frobenius norm

The dimensionality reduction of the equilibrium equations system is based on the principle of keeping the most relevant thermodynamic interactions between the components, as measured by the binary interaction parameters matrix, and parameterizing them using a smaller number of surrogate components.

Thus, one step in the dimensionality reduction method is to minimize the Frobenius norm of the binary interaction matrix approximation error, a problem theoretically characterized by the Eckart-Young-Mirsky theorem (Markovsky; Huffel, 2007). In this formulation, it is desired to obtain a low-rank approximation of the binary interaction parameter matrix C, according to the quoted theorem, and a discussion of this issue is given next. The matrix C, with entries $1 - c_{ij}$, is a square matrix of order N, where N is the number of components in the multicomponent system.

First, the binary interaction matrix is symmetric. Thus, by the spectral theorem for real symmetric matrices, there exist real eigenvalues, λ_i, and corresponding orthonormal (vertical) eigenvectors, $\mathbf{v}_i$, with $i = 1, \ldots, N$, such that

$$C = \sum_{i=1}^{N} \lambda_i \mathbf{v}_i \mathbf{v}_i^t. \tag{8}$$

Suppose that the eigenvalues are ordered so that

$$|\lambda_1| \geq |\lambda_2| \geq \ldots \geq |\lambda_N| \;\text{ and }\; |\lambda_i| \leq \varepsilon \;\text{ for }\; i = r+1, \ldots, N, \tag{9}$$

where ε is a prescribed tolerance. Then, define the reduced binary iteration matrix, C^*, by

$$C^* = \sum_{i=1}^{r} \lambda_i \mathbf{v}_i \mathbf{v}_i^t. \tag{10}$$

Using the bound on the eigenvalues (Equation (9)), this matrix approximates C in the Frobenius norm,

$$\|C - C^*\|_F^2 \;=\; \|\sum_{i=r+1}^{N} \lambda_i \mathbf{v}_i \mathbf{v}_i^t\|_F^2 = \sum_{i=r+1}^{N} \lambda_i^2 \leq (N-r)\varepsilon^2. \tag{11}$$

In fact, by the cited Eckart-Young-Mirsty theorem, C^* is the minimizer of the norm $\|C - R\|_F$, when R varies among the rank r matrices, *i.e.*,

$$\|C - C^*\|_F = \min_{R, \text{rank}(R) \leq r} \|C - R\|_F. \tag{12}$$

In other words, the best rank r approximation of C is obtained by diagonalizing C and constructing C^* as in Equation (10).

2.3 Reduction parameters

To obtain a system of equations for the thermodynamic equilibrium with a smaller number of unknowns, we follow Hendriks (1988) and introduce a new set of variables, $\mathbf{q} = (q_1, \cdots, q_N)^t$, representing surrogate variables of the composition of the phases, depending linearly on them, which can be written entry-wise or in vector-matrix notation, respectively, as

$$q_k = \sum_{i=1}^{N} m_{ik} z_i, \ \text{ or } \ \mathbf{q} = M^t \mathbf{z}, \tag{13}$$

where M is a $N \times N$ matrix. More specifically, we take

$$M = (m_1, \cdots, m_N) = (\sqrt{\mathbf{a}} \circ \mathbf{v}_1, \cdots, \sqrt{\mathbf{a}} \circ \mathbf{v}_N), \tag{14}$$

where $m_k = \sqrt{\mathbf{a}} \circ v_k$ is k-th column of matrix M, $\sqrt{\mathbf{a}}$ is the vector of the square roots of the energy parameters of the components, Equation (7d), the symbol $\circ$ is the pointwise Hadamard product between two matrices, and the columns of the matrix $V = (\mathbf{v}_1 \ldots \mathbf{v}_N)$ are the orthonormal basis of the binary interaction matrix, Equation (8). Componentwise,

$$q_k = (\sqrt{\mathbf{a}} \circ \mathbf{v}_k)^t \mathbf{z} = \sum_{i=1}^{N} \sqrt{a_i} v_{ik} z_i, \quad k = 1, \ldots, N. \tag{15}$$

When Equation (9) is satisfied, the problem can be approximated using only a few of the q variables and these entries of vector $\mathbf{q}$ are called *reduction parameters*, representing composition surrogate variables that enter a reduced thermodynamic equilibrium system.

2.4 Some parameters in reduced form

With the approximation of C by C^* we can rewrite a_m and ψ_i (Equations (7a), (7b)). The columns of matrix V present in Equation (14) are the eigenvectors of C. It is important to note that $\mathbf{v}_k$ does not depend on the composition.

So, from Equations (7b) and (7c), we write

$$\begin{aligned} a_m &= \sum_{i=1}^{N} \sum_{j=1}^{N} \sqrt{a_i a_j} (1 - c_{ij}) z_i z_j \\ &= \mathbf{z}^t (\sqrt{\mathbf{a}} \sqrt{\mathbf{a}}^t \circ C) \mathbf{z} \approx \mathbf{z}^t (\sqrt{\mathbf{a}} \sqrt{\mathbf{a}}^t \circ C^*) \mathbf{z}, \end{aligned} \tag{16}$$

using matrix notation, Hadamard product and approximation Equation (11).

From Equation (16), Equation (10), a property of Hadamard product and Equation (15),

$$\begin{aligned} a_m &\approx \mathbf{z}^t(\sqrt{\mathbf{a}}\sqrt{\mathbf{a}}^t) \circ C^*)\mathbf{z} = \mathbf{z}^t\left((\sqrt{\mathbf{a}}\sqrt{\mathbf{a}}^t) \circ \left(\sum_{k=1}^{r} \lambda_k \mathbf{v}_k \mathbf{v}_k^t\right)\right)\mathbf{z} \\ &= \sum_{k=1}^{r} \lambda_k \mathbf{z}^t(\sqrt{\mathbf{a}} \circ \mathbf{v}_k)(\sqrt{\mathbf{a}} \circ \mathbf{v}_k)^t \mathbf{z} = \sum_{k=1}^{r} \lambda_k q_k^2. \end{aligned} \tag{17}$$

Letting $\mathbf{\Psi}$ denote the vector with components ψ_i, Equation (7a), and using Equation (7c) and matrix notation, using the approximation guaranteed by Equation (11), proceeding in a similar way as before using Equation (10), and Equation (15) we get

$$\begin{aligned} \mathbf{\Psi} &= \left((\sqrt{\mathbf{a}}\sqrt{\mathbf{a}}^t) \circ C\right)\mathbf{z} \approx \left((\sqrt{\mathbf{a}}\sqrt{\mathbf{a}}^t) \circ C^*\right)\mathbf{z} \\ &= \sum_{k=1}^{r} \lambda_k (\sqrt{\mathbf{a}} \circ \mathbf{v}_k)(\sqrt{\mathbf{a}} \circ \mathbf{v}_k)^t \mathbf{z} = \sum_{k=1}^{r} \lambda_k \mathbf{m}_k q_k, \end{aligned} \tag{18}$$

recalling that $\mathbf{m}_k = \sqrt{\mathbf{a}} \circ \mathbf{v}_k$ is the k-th column of matrix M, Equation (14). Thus, in reduced form, we define

$$a_m(q) = \sum_{k=1}^{r} \lambda_k q_k^2; \quad \psi_i(q) = \sum_{k=1}^{r} \lambda_k m_{ik} q_k; \text{ for } i = 1, \ldots, N. \tag{19}$$

Note that the parameter b_m, Equation (7b), is not reduced since it does not depend on the matrix of binary interaction parameters.

2.5 The reduced equations for the dew point calculation

From Equations (7e) and (19), we verify that A depends on $\mathbf{q}$, P and T, $A = A(\mathbf{q}, P, T)$. Also, from Equation (7e), B depends on b_m, P and T, $B = B(b_m, P, T)$. Since A and B define the coefficients of the cubic state equation, Equation (5), one concludes that its roots depend on $\mathbf{q}$, b_m, P and T, $Z = Z(\mathbf{q}, b_m, P, T)$. Also B_i, Equation (7f), depends on P and T.

Therefore, since the fugacity coefficients depends on a_m, A, B, B_i, ψ_i, and $\mathbf{z}$ (see Table 2), one concludes that

$$\varphi_i = \varphi_i(\mathbf{q}, b_m, P, T), \tag{20}$$

and the reduced coefficient of fugacity is implicitly given by (Equation (6))

$$\begin{aligned} \ln \varphi_i(\mathbf{q}, b_m, P, T) &= \frac{B_i(P,T)}{B(b_m,P,T)}(Z(\mathbf{q}, b_m, P, T) - 1) - \ln(Z(\mathbf{q}, b_m, P, T) - B(b_m, P, T)) \\ &+ \frac{A(\mathbf{q}, P, T)}{2.282B(b_m, P, T)}\left(\frac{B_i(P,T)}{B(b_m,P,T)} - 2\frac{\psi_i(\mathbf{q})}{a_m(\mathbf{q})}\right) \ln\left(\frac{Z(\mathbf{q}, b_m, P, T) + 2.414B(b_m, P, T)}{Z(\mathbf{q}, b_m, P, T) - 0.414B(b_m, P, T)}\right). \end{aligned} \tag{21}$$

We assume a dew point problem, where the composition of the vapor phase is known and one wants to determine the liquid phase composition. Since we consider a dimensionality reduction in the liquid phase components, the liquid phase fugacities are written as functions of the reduced parameters, $\mathbf{q}^L$, b_m^L, P and T. From Equation (4) and from the previous explanation for the reduced case, $x_i = x_i(\mathbf{q}^L, \mathbf{y}, b_m^L, P, T)$, we get

$$x_i = \frac{\varphi_i^V(y_i, P, T) y_i}{\varphi_i^L(\mathbf{q}^L, b_m^L, P, T)}, \quad i = 1, \dots, N. \tag{22}$$

By multiplying these equations by m_{ik}, adding them over all components, and using Equation (15), we get

$$q_k^L = \sum_{i=1}^{N} m_{ik} x_i = \sum_{i=1}^{N} \frac{m_{ik} \varphi_i^V(y_i, P, T) y_i}{\varphi_i^L(\mathbf{q}^L, b_m^L, P, T)}, \quad k = 1, \dots, r. \tag{23}$$

These equations are the reduced equilibrium equations written for the surrogate composition variables (reduced variables), for a dew point calculation.

Similar equations and procedure would result if one is interested in computing bubble points; beginning at Equation (4), Equation (22) which would be modified to

$$y_i = \frac{\varphi_i^L(x_i, P, T) x_i}{\varphi_i^V(\mathbf{q}^V, b_m^V, P, T)}, \quad i = 1, \dots, N, \tag{24}$$

where $\mathbf{q}^V$ is the reduced variables for the vapor phase and b_m^V is the covolume of vapor phase.

For a dew point calculation under a specified temperature T, we have a system of $r + 2$ equations to $r + 2$ unknowns, $\mathbf{q}^L$, b_m and P

$$\begin{cases} q_k^L = \sum_{i=1}^{N} m_{ik} x_i, \quad k = 1, \dots, r \\ \sum_{i=1}^{N} x_i = 1, \\ b_m^L = \sum_{i=1}^{N} b_i x_i, \end{cases} \tag{25}$$

where x_i is as in Equation (22) and $\mathbf{y}$ and T are known.

More generally, in order to propose an algorithm for the determination of the composition of dew points define vector valued function $\mathbf{G} = (G_1, \cdots, G_{r+2})^t$, with entries

$$\begin{cases} G_k(\mathbf{q}^L, \mathbf{y}, b_m^L, P, T) = q_k^L - (\sum_{i=1}^{N} m_{ik} x_i), \quad k = 1, \cdots, r \;, \\ \\ G_{r+1}(\mathbf{q}^L, \mathbf{y}, b_m^L, P, T) = (\sum_{i=1}^{N} x_i) - 1, \\ \\ G_{r+2}(\mathbf{q}^L, \mathbf{y}, b_m^L, P, T) = b_m^L - \sum_{i=1}^{N} b_{c_i} x_i. \end{cases} \tag{26}$$

We can rewrite Equation (5) as

$$G_k(\mathbf{q}^L, \mathbf{y}, b_m^L, P, T) = 0, \text{ for } k = 1, \ldots, r+2, \tag{27}$$

or in vector form as

$$\mathbf{G}(\mathbf{q}^L, \mathbf{y}, b_m^L, P, T) = \mathbf{0}. \tag{28}$$

For the calculation of the dew point where $\mathbf{y} = \mathbf{y}^*$ and $T = T^*$ are known, define the function $\mathbf{F}$, given by

$$\mathbf{F}(\mathbf{q}^L, b_m^L, P) = \mathbf{G}(\mathbf{q}^L, \mathbf{y}^*, b_m^L, P, T^*), \tag{29}$$

replacing in the function $\mathbf{G}$, $\mathbf{y}$ and P with their known values.

Finally, the system to be solved for the dew point problem, with $r+2$ equations and $r+2$ unknowns, $\mathbf{q}^L$, b_m^L and P, is given in compact form by

$$\mathbf{F}(\mathbf{q}^L, b_m^L, P) = \mathbf{0}. \tag{30}$$

For a dew point calculation where $\mathbf{y} = \mathbf{y}^*$ and $P = P^*$ are known, we define $\mathbf{F}$ by $\mathbf{F}(\mathbf{q}^L, b_m^L, T) = \mathbf{G}(\mathbf{q}^L, \mathbf{y}^*, P^*, T)$, and the system of equations to be solved is $\mathbf{F}(\mathbf{q}^L, b_m^L, T) = 0$.

Equation (30) is a system of nonlinear algebraic equations that can be solved, for example, by the multivariate Newton-Raphson method as will be seen in the next sub-section.

2.6 The iterative scheme

To solve the reduced system of non-linear algebraic equations, Equation (30), is to find the roots of $\mathbf{F}$ defined by Equation (29). For that we can adapt iterative methods such as steepest descent, simulated annealing, differential evolution, among others (Nocedal; Wright, 1999). Here we use multivariate Newton-Raphson (MNR) method (Kelley, 1995), described below.

To obtain the iteration function of the MNR, use a Taylor expression,

$$\mathbf{F}(\mathbf{S}) = \mathbf{F}(\mathbf{S}^l) + J(\mathbf{S}^l)(\mathbf{S} - \mathbf{S}^l) + \mathbf{E} = \text{linear approximation} + \text{error}, \tag{31}$$

where $\mathbf{S} = (\mathbf{q}^L, b_m, P)$, $\mathbf{S}^l$ is the approximation vector in the l-th iteration, $J(\mathbf{S})$ is the Jacobian matrix of $\mathbf{F}(\mathbf{S})$, with entries

$$[J(\mathbf{S})]_{ij} = \left[\frac{\partial F_i(\mathbf{S})}{\partial x_j}\right], \text{ with } i, j = 1, \cdots, r+2, \tag{32}$$

and $\mathbf{E}$ is a vector that represents the error of the linear approximation.

In order to establish the iterative method, the approximation of the solution of the equation in the iteration $l+1$ given by $\mathbf{S}^{l+1}$ is defined by the vector that cancels the linear part of Equation 31, that is, the vector $\mathbf{S}^{l+1}$ such that

$$\mathbf{F}(\mathbf{S}^l) + J(\mathbf{S}^l)(\mathbf{S}^{l+1} - \mathbf{S}^l) = \mathbf{0}. \tag{33}$$

Pre-multiplying Equation (33) by the inverse of the Jacobian matrix, we obtain

$$\mathbf{S}^{l+1} = \mathbf{S}^l - J^{-1}(\mathbf{S}^l)\mathbf{F}(\mathbf{S}^l). \tag{34}$$

Since the inversion of matrices is an expensive operation, one can work with Equation (33) finding the approximation of the iteration $l+1$ solving a linear system to calculate $\Delta\mathbf{S}^l = \mathbf{S}^{l+1} - \mathbf{S}^l$

$$J(\mathbf{S}^l)\Delta\mathbf{S}^l = -\mathbf{F}(\mathbf{S}^l). \tag{35}$$

In this way, the new estimate will be

$$\mathbf{S}^{l+1} = \mathbf{S}^l + \Delta\mathbf{S}^l. \tag{36}$$

It is sometimes convenient to enter a step parameter, η, and define instead the new estimate by

$$\mathbf{S}^{l+1} = \mathbf{S}^l + \eta^l\Delta\mathbf{S}^l. \tag{37}$$

When the parameter is one, $\eta^l = 1$, we obtain Equation (36). Thus, given an initial estimate $\mathbf{S}^0 = ((\mathbf{q}^L)^0, b_m^0, P^0)$ for the solution of Equation (30), we can find a new approximate solution for Equation (30) using Equation (35), for the calculation of $\Delta\mathbf{S}^l$, and Equation (36) (or Equation (37)) for the calculation of $\mathbf{S}^{l+1}$. Given $\bar{\varepsilon}$, the stopping criteria is $\|\Delta\mathbf{S}^l\| < \bar{\varepsilon}$.

2.7 Initial estimates to solve the reduced system of equilibrium equations

Equation (13) is used with extreme values of $\mathbf{z}$ to get an initial estimate of $(\mathbf{q}^L)^0$. In fact, since $\mathbf{z} = \mathbf{x}$, with $0 \leq x_i \leq 1$, and $\sum_{i=1}^N x_i = 1$, we conclude that the surrogate reduced parameters $\mathbf{q}^L$ must be bounded by $(q_k^L)^m \leq (q_k^L) \leq (q_k^L)^M$ where

$$(q_k^L)^m = \min_{i=1,\cdots,N} m_{ik} \quad \text{and} \quad (q_k^L)^M = \max_{i=1,\cdots,N} m_{ik}. \tag{38a}$$

Then we take the initial estimate as the middle point,

$$(q_k^L)^0 = \frac{(q_k^L)^m + (q_k^L)^M}{2}, \tag{38b}$$

to define the vector entries of $(\mathbf{q}^L)^0$.

Now this depends on the knowledge of M, Equation (14), which in turn depends on $\sqrt{\mathbf{a}}$ and on the eigenvalues of the binary interaction matrix C. Recall that with the parameters c_{ij}, we can obtain the matrix C of inputs $1 - c_{ij}$. Diagonalizing this matrix and applying the spectral theorem we get its eigenvectors.

Thus, together with Equation (7d) for $\sqrt{\mathbf{a}}$, this allows the construction of Table 3 with all data needed to obtain an initial estimate $(\mathbf{q}^L)^0$. The value ε, from Equation (9) is in Table 3 since not all eigenvectors of C are needed to find the initial estimate of $(\mathbf{q}^L)^0$.

With the eigenvectors of C^*, and the other data of the Table 3, we can obtain an initial estimate for $\mathbf{q}^L$, since the vector $\mathbf{q}$ depends on the matrix M as shown in Equation (13), which in turn depends on the eigenvectors of the matrix C^* as shown in Equation (14).

Table 3 – Input values to get an initial estimate $\mathrm{S}^0 = ((\mathbf{q}^L)^0, (b_m^L)^0, P^0)$ for the isothermal dew point calculation

	known values	required choice	initial estimate
$(\mathbf{q}^L)^0$	Ω_a, Ω_b, S_0, S_1, S_2, R, ω_i, T_{c_i}, P_{c_i}, c_{ij}, N, ε	T^*	
$(b_m^L)^0$	Ω_b, R, T_{c_i}, P_{c_i}	$\mathbf{y}^*$	
P^0			P^0

Source: The authors.

The initial estimate of b_m^L is defined by the known fixed global composition of the vapor phase $\mathbf{z} = \mathbf{y}$, obtaining

$$(b_m^L)^0 = \sum_{i=1}^{N} b_i y_i^*, \tag{39}$$

or, in other words, $(b_m^L)^0 = b_m^V$ which is fixed by the vapor composition $\mathbf{y}^*$.

Then we have the initial vector $\mathrm{s}^0 = ((\mathbf{q}^L)^0, (b_m^L)^0)$.

3 REDUCTION ALGORITHMS

In this Section, we show the algorithms used to apply the reduction method in the dew point calculation, finding the zero of the function $\mathbf{F}$ described in the subsection 2.5. We write Algorithm 1 to find the eigenvectors of the reduced matrix C^* and with them calculate the vector $(\mathbf{q}^L)^0$, as shown in the subsection 2.7 and Algorithm 2 applies Newton method to the reduced equilibrium thermodynamic system, Equation (30).

Algorithm 1

Algorithm 1

```
Input: $\Omega_a$, $\Omega_b$, $S_0$, $S_1$, $S_2$, $R$, $\omega_i$, $T_{c_i}$, $P_{c_i}$, $c_{ij}$ $(i, j = 1 : N)$, $T^*$, $\mathbf{y}^*$, $P^0$, $\varepsilon$, $N$;
Construct the binary interaction matrix $C$ with entries $1 - c_{ij}$;
Obtain the $r$ eigenvalues, $\lambda_i$, and eigenvectors, $v_i$, of the binary interaction matrix $C$, with $| \lambda_i | \geq \varepsilon$;
Compute $\sqrt{\mathbf{a}}$ using Equation (7d);
Construct the first $r$ columns of matrix $M$, Equation (14);
Compute $(\mathbf{q}^L)^0$, Equation (38);
Compute $b_i$, using Equation (7d);
Compute $(b_m^L)^0$, using Equation (39);
Assemble $\mathbf{s}^0 = ((\mathbf{q}^L)^0, (b_m^L)^0)$ and $\mathbf{S}^0 = ((\mathbf{q}^L)^0, (b_m^L)^0, P^0)$;
Output: $\mathbf{S}^0$, $\mathbf{s}^0$, $b_i(i = 1 : N)$, $\lambda_i, v_i(i = 1 : r)$, $r$.
```

Source: The authors.

The algorithm to solve the reduced system has to compute values of φ_i^L and its derivative, ultimately as functions of $\mathbf{q}^L$, b_m^L, P.

Algorithm 2

Algorithm 2

```
Input: $\Omega_a$, $\Omega_b$, $S_0$, $S_1$, $S_2$, $R$, $\omega_i$, $T_{c_i}$, $P_{c_i}$, $b_i(i = 1 : N)$, $c_{ij}(i, j = 1 : N)$, $N$, $\mathbf{y}^*$, $\lambda_i, v_i(i = 1 : r)$, $r$, $\mathbf{s}^0$, $\mathbf{S}^0$; $T^*$, $P^0$, $\bar{\varepsilon}$, $\eta$;
Compute the fugacity coefficient of the vapor phase, using Equation (6);
Construct the first $r$ columns of matrix $M$, Equation (14);
Set $\Delta = 1/\bar{\varepsilon}(1, 0, \cdots, 0)$ (a vector with large norm);
while $\|\Delta\| \geq \bar{\varepsilon}$
    compute $\Delta$ satisfying (Equation (36)) $J(\mathbf{S}^l)\Delta = -\mathbf{F}(\mathbf{S}^l)$;
    set $\mathbf{S}^{l+1} = \mathbf{S}^l + \Delta$;
end while
Recover the liquid composition (Equation (22));
Output: Liquid composition, $\mathbf{x}$, and pressure, $P$.
```

Source: The authors.

To compute $\mathbf{F}(\mathbf{S}^l)$ and $J(\mathbf{S}^l)$, one needs to compute the values of φ_i^L and its derivatives ultimately as functions of $\mathbf{S}^l = ((\mathbf{q}^L)^l, (b_m^L)^l, P^l)$. The steps for this are given next.

Details of the 4^{th} step (a) of algorithm 2

Details of the 4^{th} step (a) of algorithm 2:

```
Compute $a_m((\mathbf{q}^L)^l)$ and $\varphi_i((\mathbf{q}^L)^l)$ using Equation (19);
Compute $A$, $B$, $B_i$ using Equations (7e), (7f);
Compute the roots of the PR equation of state, Equation (5), and choose the smallest root, $Z$;
Compute the fugacity coefficient of the liquid phase, $\varphi_i((\mathbf{q}^L)^l, (b_m^L)^l, P^l, T^*)$, using Equation (21);
Compute the derivatives of $\varphi_i$ to $((\mathbf{q}^L)^l, (b_m^L)^l, P^l, T^*)$;
Compute $\mathbf{F}(\mathbf{S}^l)$ and $J(\mathbf{S}^l)$.
```

Source: The authors.

Algorithm 3 is presented to compute a dew point curve, in the temperature *versus* pressure plane. It is attained by successively applying Algorithm 2 to compute the pressure and composition of the liquid phase for varying temperatures.

Algorithm 3

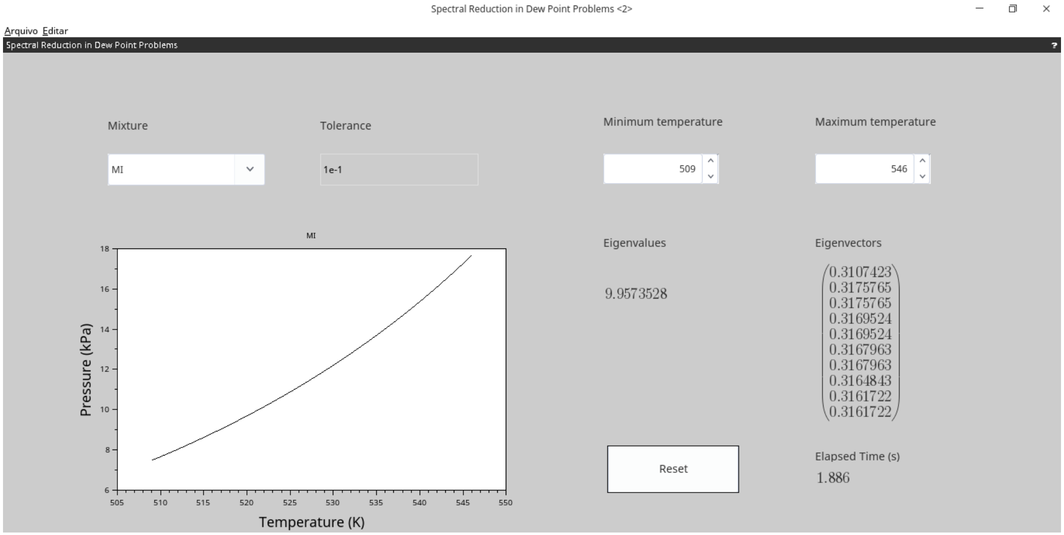
Algorithm 3

```
Input: $\Omega_a$, $\Omega_b$, $S_0$, $S_1$, $S_2$, $R$, $\omega_i$, $T_{c_i}$, $P_{c_i}$, $c_{ij}(i,j = 1:N)$, $T^*$, $\mathbf{y}^*$, $P^0$, $\varepsilon$, $N$, $b_i(i = 1:N)$, $\lambda_i, v_i(i = 1:r)$, $r$, $\mathbf{s}^0$, $\mathbf{S}^0$, $\bar{\varepsilon}$, $\eta$, $T_1$, $T_2$, $\Delta T$, $n$ (with $T_2 = T_1 + n\Delta T$);
for $j = 0:n$
    call Algorithm 1 with $T^* = T_j = T_1 + j\Delta T$;
    call Algorithm 2;
end for
Output: $\lambda_i, v_i(i = 1:r)$, $(T_j, P_j)$, $\mathbf{x}^j$ $(j = 0:n)$, time.
```

Source: The authors.

Figure 2 illustrates our framework with an example that uses only one non-null eigenvalue.

Figure 2 – Scilab code using spectral decomposition in dew point calculations



Spectral Reduction in Dew Point Problems <2>

Spectral Reduction in Dew Point Problems

Mixture: MI

Tolerance: 1e-1

Minimum temperature: 509

Maximum temperature: 546

Eigenvalues: 9.9573528

Eigenvectors:

$$\begin{pmatrix} 0.3107423 \\ 0.3175765 \\ 0.3175765 \\ 0.3169524 \\ 0.3169524 \\ 0.3167963 \\ 0.3167963 \\ 0.3164843 \\ 0.3161722 \\ 0.3161722 \end{pmatrix}$$

Elapsed Time (s): 1.886

Source: The authors.

4 RESULTS AND DISCUSSIONS

In this Section, we present the results obtained in the calculation of the dew curves, implemented in Scilab 5.5 language[3] and using an Intel Core™ i5-3337U CPU at 1.80GHz, with 4GB of RAM memory under 64-bit Windows operating system.

Using the developed framework described in Section 1, some comparisons were made with the classical method, which does not use a reduction (*i.e.*, solves the isofugacity equations). The mixtures used are denoted as MHA5 (referred as example 2 in the work of Mehra, Heidemann e Aziz (1983), also analyzed by Kaul e Thrasher (1996)) and MI (referred as Mixture 1 in the work of Gaganis e Varotsis (2013)), with 5 and 10 components respectively. Data for mixtures MHA5 and MI are presented in Tables 4 and 5, respectively. The mixture MHA5 uses some components present in the mixture MI, so we use the same values for critical pressure, P_{c_i}, critical temperature, T_{c_i}, and the acentric factor, ω_i, given in Gaganis e Varotsis (2013). The binary interaction parameters for the mixture MHA5 were obtained from Kaul e Thrasher (1996) and presented in Table 4. For the mixture MI, all binary interaction parameters are set to zero, except the ones relating methane with some other components, represented in Table 5 as c_{C_1-j} (Gaganis; Varotsis, 2013).

Table 4 – Data for mixture MHA5

Components	z	$T_{ci}(K)$	$P_{ci}(bar)$	ω_i	c_{C_2-j}	c_{C_3-j}	c_{nC_4-j}	c_{nC_5-j}	c_{nC_6-j}
C_2	0.39842	305.33	48.71	0.099	0.000				
C_3	0.29313	369.85	42.47	0.152	0.002	0.000			
nC_4	0.20006	425.25	37.92	0.199	0.005	0.001	0.000		
nC_5	0.07143	469.80	33.75	0.251	0.009	0.003	0.001	0.000	
nC_6	0.03696	507.90	30.35	0.299	0.012	0.005	0.002	0.000	0.000

Source: Research data (Mehra; Heidemann; Aziz, 1983; Kaul; Thrasher, 1996; Gaganis; Varotsis, 2013).

Table 5 – Data for mixture MI

Components	z	$T_{ci}(K)\}$	$P_{ci}(bar)$	ω_i	c_{C_1-j}
C_1	0.35	190.55	45.99	0.011	-
C_2	0.03	305.33	48.71	0.099	0.000
C_3	0.04	369.85	42.47	0.152	0.000
C_4	0.06	425.25	37.92	0.199	0.020
C_5	0.04	469.80	33.75	0.251	0.020
C_6	0.03	507.90	30.35	0.299	0.025
C_7	0.05	540.15	27.35	0.350	0.025
C_8	0.05	568.95	24.90	0.397	0.035
C_{10}	0.30	617.65	21.05	0.490	0.045
C_{14}	0.05	693.00	16.10	0.654	0.045

Source: Research data (Gaganis; Varotsis, 2013).

---

[3] Scilab 5.5 is distributed under CeCILL license.

A family of reduction methods is obtained by relaxing the level of approximation of C by C^*. Specifically, we let ε in Equation (9) to increase, resulting in r-reduction methods with $r = 3, 2$ and 1 eigenvalues. Clearly, a decrease in r implies a stronger reduction.

The eigenvalues and eigenvectors for the mixtures considered are listed in Tables 6 and 7. For the mixture MHA5 with $\varepsilon = 4 \times 10^{-4}$, we have three eigenvalues, λ_1, λ_2 and λ_3 and their respective eigenvectors $\mathbf{v}_1$, $\mathbf{v}_2$ and $\mathbf{v}_3$, represented in Table 6. With $\varepsilon = 1 \times 10^{-2}$, we have two eigenvalues, λ_1 and λ_2, and with $\varepsilon = 0.02$ only one eigenvalue remains. A similar analysis can be conducted for the mixture MI, using three tolerances: 1×10^{-6}, 0.03, and 0.08, as detailed in Table 7. The tolerance of the Newton method was set to $\bar{\varepsilon} = 1 \times 10^{-6}$ and we used $\eta = 1$ for both mixtures (see sub-section 2.6).

Table 6 – Mixture MHA5: eigenvalues of the binary interaction matrix with $\varepsilon = 4 \times 10^{-4}$ and corresponding eigenvectors

	1	2	3	4	5		$\boldsymbol{\lambda}$
$\mathbf{v}_1$	0.44613	0.44766	0.44784	0.44748	0.44694	λ_1	4.98400
$\mathbf{v}_2$	-0.71436	-0.25145	0.06023	0.36809	0.53603	λ_2	0.01539
$\mathbf{v}_3$	0.47079	-0.39961	-0.66228	0.25294	0.34067	λ_3	0.00056

Source: The authors.

Table 7 – Mixture MI: eigenvalues of the binary interaction matrix with $\varepsilon = 1 \times 10^{-6}$ and corresponding eigenvectors

	1	2	3	4	5	6
$\mathbf{v}_1$	0.31074	0.31757	0.31757	0.31695	0.31695	0.31679
$\mathbf{v}_2$	-0.80423	-0.18402	-0.18402	0.04364	0.04364	0.10056
$\mathbf{v}_3$	-0.50659	0.48693	0.48693	0.12512	0.12512	0.03467
	7	8	9	10		$\boldsymbol{\lambda}$
$\mathbf{v}_1$	0.31679	0.31648	0.31617	0.31617	λ_1	9.95735
$\mathbf{v}_2$	0.10056	0.21439	0.32823	0.32823	λ_2	0.07065
$\mathbf{v}_3$	0.03467	-0.14623	-0.32714	-0.32714	λ_3	-0.02800

Source: The authors.

To exemplify how eigenvalues and eigenvectors are selected according to a tolerance, we use the mixture MHA5, with the binary interaction parameters given in Table 4. Using the spectral decomposition, we obtain, through Equation (8)

$$
\begin{aligned}
C \;=\; & \begin{pmatrix} 1-0.000 & 1-0.002 & 1-0.005 & 1-0.009 & 1-0.012 \\ 1-0.002 & 1-0.000 & 1-0.001 & 1-0.003 & 1-0.005 \\ 1-0.005 & 1-0.001 & 1-0.000 & 1-0.001 & 1-0.002 \\ 1-0.009 & 1-0.003 & 1-0.001 & 1-0.000 & 1-0.000 \\ 1-0.012 & 1-0.005 & 1-0.002 & 1-0.000 & 1-0.000 \end{pmatrix} \\
= \; & -0.00024 \begin{pmatrix} -0.00230 \\ -0.20997 \\ 0.08910 \\ 0.74695 \\ -0.62451 \end{pmatrix} (-0.00230 \;\; -0.20997 \;\; 0.08910 \;\; 0.74695 \;\; -0.62451) \\
+ \; & 0.00027 \begin{pmatrix} 0.26267 \\ -0.72978 \\ 0.59097 \\ -0.20576 \\ 0.08261 \end{pmatrix} (0.26267 \;\; -0.72978 \;\; 0.59097 \;\; -0.20576 \;\; 0.08261) \\
+ \; & 0.00056 \begin{pmatrix} 0.47079 \\ -0.39961 \\ -0.66228 \\ 0.25294 \\ 0.34067 \end{pmatrix} (0.47079 \;\; -0.39961 \;\; -0.66228 \;\; 0.25294 \;\; 0.34067) \\
+ \; & 0.01539 \begin{pmatrix} -0.71436 \\ -0.25145 \\ 0.06023 \\ 0.36809 \\ 0.53603 \end{pmatrix} (-0.71436 \;\; -0.25145 \;\; 0.06023 \;\; 0.36809 \;\; 0.53603) \\
+ \; & 4.98400 \begin{pmatrix} 0.44613 \\ 0.44766 \\ 0.44784 \\ 0.44748 \\ 0.44694 \end{pmatrix} (0.44613 \;\; 0.44766 \;\; 0.44784 \;\; 0.44748 \;\; 0.44694). \qquad (40)
\end{aligned}
$$

Given a tolerance $\varepsilon = 4 \times 10^{-4}$, we select the three largest eigenvalues and their respective eigenvectors (see Equation (9)). Then, the matrix C is approximated by the reduced rank matrix (see Equation (10))

$$
\begin{aligned}
C^* &= 0.00056 \begin{pmatrix} 0.47079 \\ -0.39961 \\ -0.66228 \\ 0.25294 \\ 0.34067 \end{pmatrix} (0.47079 \ -0.39961 \ -0.66228 \ 0.25294 \ 0.34067) \\
&+ 0.01539 \begin{pmatrix} -0.71436 \\ -0.25145 \\ 0.06023 \\ 0.36809 \\ 0.53603 \end{pmatrix} (-0.71436 \ -0.25145 \ 0.06023 \ 0.36809 \ 0.53603) \\
&+ 4.98400 \begin{pmatrix} 0.44613 \\ 0.44766 \\ 0.44784 \\ 0.44748 \\ 0.44694 \end{pmatrix} (0.44613 \ 0.44766 \ 0.44784 \ 0.44748 \ 0.44694) \\
&= \begin{pmatrix} 1.0010702 & 0.9970882 & 0.9933694 & 0.9915972 & 0.9887815 \\ 0.9970882 & 1.0006583 & 1.0004416 & 0.9963989 & 0.9943479 \\ 0.9933694 & 1.0004416 & 1.0021064 & 0.9981939 & 0.9968189 \\ 0.9915972 & 0.9963989 & 0.9981939 & 1.0004314 & 1.0003027 \\ 0.9887815 & 0.9943479 & 0.9968189 & 1.0003027 & 1.0006526 \end{pmatrix} \qquad (41)
\end{aligned}
$$

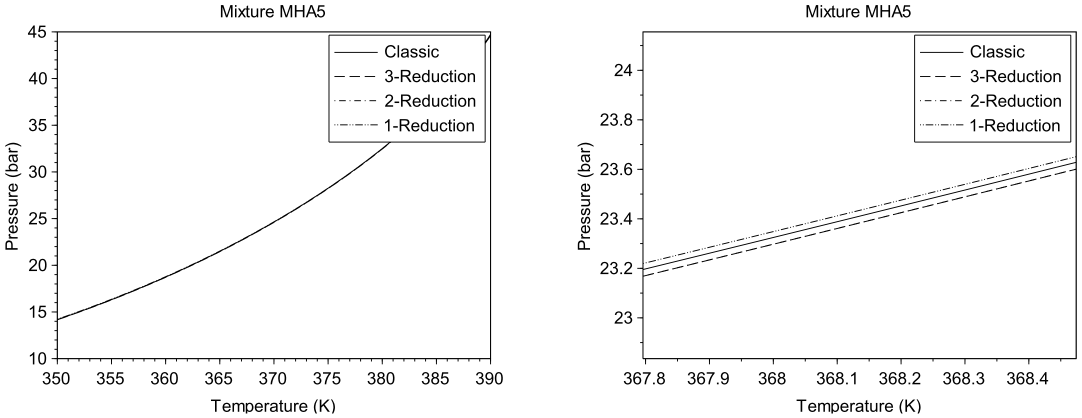
Calculating the Frobenius norm of the matrices C and C^* we have $||C||_F = 4.9840333$, $||C^*||_F = 4.9839563$ and $||C - C^*||_F = 0.0050498$. Thus, the error is given by $\frac{||C-C^*||_F}{||C||_F} = 0.0010132$. A similar analysis can be conducted with smaller rank matrices, that is, with two or one eigenvalue.

The numerical rank of the matrix C can be obtained by using a simple Scilab command: [U,S,V,rk] = svd(C,4e-4), where 4e-4 refers to the tolerance for the eigenvalues and rk is the numerical rank of the matrix. In this case, we obtain rk = 3, for C and C^*, as expected.

With less eigenvalues and eigenvectors, the thermodynamic equilibrium system, Equation (30), exhibits a smaller number of equations ($r + 2$).

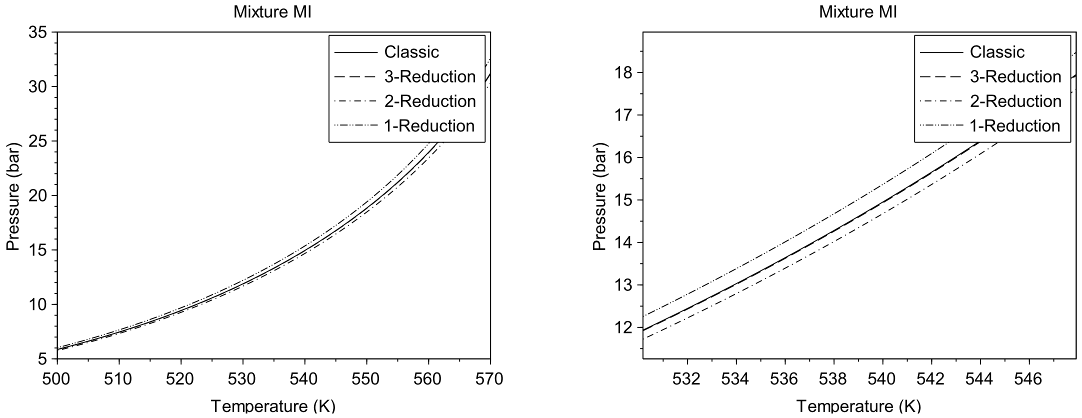
The temperature range (in Kelvin scale, K) for the mixture MHA5 is $T = [350; 390]$. Similarly, we use the range $T = [500; 570]$ for the mixture MI. The initial estimate of the pressure used is $P^0 = 10$ bar for the mixture MHA5 and $P^0 = 1$ bar for the mixture MI. The pressure values computed for each chosen temperature range generate dew curves that are shown in Figures 3 and 4, respectively for mixtures MHA5 and MI.

Figure 3 – a) Dew point curve for the mixture MHA5. b) Enlarged portion of the dew point curve for the mixture MHA5



Source: The authors.

Figure 4 – a) Dew point curve for the mixture MI. b) Enlarged portion of the dew point curve for the mixture MI



Source: The authors.

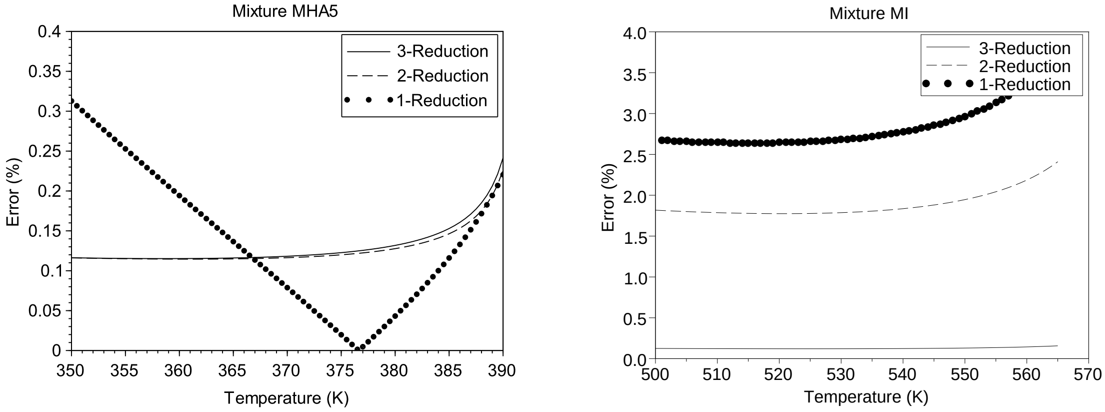
An enlarged portion of Figures 3(a) and 4(a) are given in Figures 3(b) and 4(b), where we observe the proximity of the dew point curves obtained by all r-reduction methods ($r = 3, 2, 1$) and the classical method.

The percentage error of the reduction is calculated according to the following expression:

$$\text{error} = \left| \frac{P_{\text{Reduction}} - P_{\text{Classic}}}{P_{\text{Classic}}} \right| \times 100. \tag{42}$$

Figure 5 shows that the percentage errors are quite small in all reduction cases considered in both mixtures. However, it is clear that for MHA5 mixture, the behavior of the percentage error, although small, is very different when only one eigenvalue is considered. Moreover, for the MI mixture, even though the error is small in all reduction methods, it is clear that the approximation is much better when 3 eigenvalues are considered, instead of 1 or 2.

Figure 5 – Percentage error in the calculation of the pressure using r-Reduction method with $r = 1, 2, 3$ for the: a) mixture MHA5, b) mixture MI



Source: The authors.

Table 8 presents the elapsed time for the reduction and classic methods, for both mixtures, for different quantities of eigenvalues. The table also presents the number of equations solved in each situation. Regarding the elapsed time, all reduction methods reduced it, as shown in Table 8. For 1-reduction method applied to the mixture MHA5, the elapsed time is less than 21 percent of the time required by the classic method. Considering the mixture MI, 1-reduction method takes approximately 11.8 percent of the time required by the classic method.

Table 8 – Elapsed time in seconds for mixtures MHA5 and MI

Mixture	Method	Number of eigenvalues r	Time (s)	Number of equations
MHA5	Classic	–	29.26	$N = 5$
	3-Reduction	3	11.44	$r + 2 = 5$
	2-Reduction	2	8.70	$r + 2 = 4$
	1-Reduction	1	6.13	$r + 2 = 3$
MI	Classic	–	178.57	$N = 10$
	3-Reduction	3	38.11	$r + 2 = 5$
	2-Reduction	2	28.42	$r + 2 = 4$
	1-Reduction	1	20.94	$r + 2 = 3$

Source: The authors.

Finally, it is possible to obtain the compositions of the incipient phases after the convergence of the method for the mixtures, using the last temperature for each example, $T = 390$ K and $T = 570$ K, respectively. Results for mixtures MHA5 and MI are presented in Tables 9 and 10, including the results obtained by the classic method, for comparison.

Table 9 – Results for liquid composition vector x for mixture MHA5, with $T = 390$ K

	x_1	x_2	x_3	x_4	x_5
Classic	0.25353	0.26684	0.25903	0.12881	0.09177
3-Reduction	0.25283	0.26659	0.25925	0.12914	0.09216
2-Reduction	0.25290	0.26659	0.25920	0.12914	0.09221
1-Reduction	0.25276	0.26653	0.25923	0.12922	0.09224

Source: The authors.

Table 10 – Results for liquid composition vector x for mixture MI, with $T = 570$ K

	x_1	x_2	x_3	x_4	x_5
Classic	0.0582087	0.0073516	0.0129007	0.0254645	0.0219082
3-Reduction	0.0581025	0.0073404	0.0128838	0.0254366	0.0218887
2-Reduction	0.0561642	0.0070663	0.0124159	0.0247125	0.0213145
1-Reduction	0.0626781	0.0077324	0.0135009	0.0264039	0.0225927
	x_6	x_7	x_8	x_9	x_{10}
Classic	0.0210672	0.0445856	0.0564696	0.5363523	0.2156916
3-Reduction	0.0210528	0.0445638	0.0564531	0.5364032	0.2158751
2-Reduction	0.0205973	0.0437057	0.0558614	0.5383216	0.2198381
1-Reduction	0.0215780	0.0454255	0.0570097	0.5329624	0.2094585

Source: The authors.

5 CONCLUSIONS

In this paper, dew point geometries are obtained for two test mixtures (referred as MHA5 and MI) using a dimensionality reduction algorithm on a PR state equation formulation, and a spectral approximation of the binary interaction coefficients matrix.

We scrutinized the formulation of the reduced equations based on a linear parameterization of the incipient phase using cubic equations of state and classical mixing rules. The formulation obtained represents a reduced isofugacity system of equations, as obtained previously by Nichita (2008).

The primary purpose of the reduction technique is to solve a smaller set of nonlinear algebraic equations when solving vapor-liquid thermodynamic equilibrium problems to diminish the elapsed time, without impairing the accuracy of the results. The results obtained with the reduction technique are compared with those produced by classical method (resolution of the isofugacity equations).

It is also seen that the method is quite efficient when a larger reduction in the rank of the matrix C is applied, considering a smaller number of eigenvalues and their respective eigenvectors, and this is also more advantageous when a mixture with a larger number of components is used.

In addition to allowing the dimensionality of the problems to be reduced, the reduction method guarantees the accuracy of the results. Even if several non-zero eigenvalues are neglected, the calculated dew pressures still remain close to those produced with the classical methodology.

These results – added to the computational framework developed in Scilab 5.5 language – can be an useful tool in advanced undergraduate courses in Thermodynamics, permitting the understanding of the computational gains of the spectral reduction (in terms of computation times), which can be a critical factor in simulation tools.

REFERENCES

ENTERPRISES, Scilab. Consortium Scilab. Digiteo. Scilab: Free and Open Source software for numerical computation (OS, Version 5.4. 1)(2012). 2022.

GAGANIS, Vassilis; VAROTSIS, Nikos. An improved BIP matrix decomposition method for reduced flash calculations. **Fluid Phase Equilibria**, Elsevier, v. 340, p. 63–76, 2013.

HENDRIKS, Eric M. Reduction theorem for phase equilibrium problems. **Industrial & engineering chemistry research**, ACS Publications, v. 27, n. 9, p. 1728–1732, 1988.

KAUL, Pradeep; THRASHER, R. L. A parameter-based approach for two-phase-equilibrium prediction with cubic equations of state. **SPE Reservoir Engineering**, SPE, v. 11, n. 04, p. 273–279, 1996.

KELLEY, Carl T. **Iterative methods for linear and nonlinear equations**. Society for Industrial and Applied Mathematics. SIAM, 1995.

MARKOVSKY, Ivan; HUFFEL, Sabine Van. Left vs right representations for solving weighted low-rank approximation problems. **Linear algebra and its applications**, Elsevier, v. 422, n. 2-3, p. 540–552, 2007.

MATHEIS, Jan; HICKEL, Stefan. Multi-component vapor-liquid equilibrium model for les and application to ECN Spray A. **arXiv: Fluid Dynamics**, 2016. Disponível em: <https://api.semanticscholar.org/CorpusID:2794471>.

MATHIAS, P. M.; BOSTON, J. F.; WATANASIRI, S. Effective utilization of equations of state for thermodynamic properties in process simulation. **AIChE journal; (United States)**, v. 30, 3 1984. Disponível em: <https://www.osti.gov/biblio/6680481>.

MEHRA, Rakesh K; HEIDEMANN, Robert A; AZIZ, Khalid. An accelerated successive substitution algorithm. **The canadian Journal of chemical Engineering**, Wiley Online Library, v. 61, n. 4, p. 590–596, 1983.

MICHELSEN, Michael L. Simplified flash calculations for cubic equations of state. **Industrial & Engineering Chemistry Process Design and Development**, v. 25, n. 1, p. 184–188, 1986.

NICHITA, Dan Vladimir. A reduction method for phase equilibrium calculations with cubic equations of state. **Brazilian Journal of Chemical Engineering**, SciELO Brasil, v. 23, p. 427–434, 2006.

NICHITA, Dan Vladimir. Phase envelope construction for mixtures with many components. **Energy & Fuels**, v. 22, n. 1, p. 488–495, 2008.

NICHITA, Dan Vladimir; MINESCU, Florea. Efficient phase equilibrium calculation in a reduced flash context. **The Canadian Journal of Chemical Engineering**, v. 82, n. 6, p. 1225–1238, 2004.

NOCEDAL, Jorge; WRIGHT, Stephen J. Numerical optimization. **Springer Science**, v. 35, n. 67-68, p. 7, 1999.

RAMKRISHNA, Doraiswami; AMUNDSON, Neal R. Mathematics in chemical engineering: A 50 year introspection. **AIChE Journal**, v. 50, n. 1, p. 7–23, 2004.

SMITH, Joseph Mauk *et al.* **Introduction to Chemical Engineering Thermodynamics**. 8. ed. Nubuque: McGraw-Hill Education, 2017.

STRANG, Gilbert. **Linear algebra and its applications**. 2000.

WATSON, Harry AJ *et al.* Reliable flash calculations: Part 1. nonsmooth inside-out algorithms. **Industrial & Engineering Chemistry Research**, ACS Publications, v. 56, n. 4, p. 960–973, 2017.