

# Free-Form Parametric Fitting of Van der Waals Binodal and Spinodal Curves with Bat Algorithm

Almudena Campuzano  
School of Chem., Bio. & Env. Eng. (CBEE)  
College of Engineering  
Oregon State University  
Corvallis, OR, USA  
School of Industrial Engineering  
University of Cantabria  
Santander, Spain  
campuzam@oregonstate.edu

Andrés Iglesias  
Dept. of Information Science  
Faculty of Sciences  
Toho University  
Funabashi, Japan  
Dept. Appl. Math. & Comp. Sci.  
University of Cantabria  
Santander, Spain  
iglesias@unican.es

Akemi Gálvez  
Dept. of Information Science  
Faculty of Sciences  
Toho University  
Funabashi, Japan  
Dept. Appl. Math. & Comp. Sci.  
University of Cantabria  
Santander, Spain  
galveza@unican.es

**Abstract**—This paper concerns the Van der Waals (VdW) equation of state, originally conceived to be a generalization of the ideal gas law. For practical use, it is often necessary to compute two characteristic curves of VdW, called binodal and spinodal curves. They are usually constructed through polynomial fitting from a collection of 2D points in the pressure-volume plane by using standard numerical procedures. However, the resulting models are still limited and can be further enhanced. In this paper, we carry out this task through least-squares approximation of sets of 2D points using free-form Bézier curves. This requires to perform data parameterization in addition to computing the poles of the curves. To this aim, we apply a powerful nature-inspired swarm intelligence method for continuous optimization called the bat algorithm. To test the performance of this new approach, it has been applied to real data of a gas. Our experimental results show that the method can reconstruct the characteristic curves with very good accuracy. In addition, the computing times are also very good, given the complexity of this problem. These remarkable features make this approach very promising in the field. Furthermore, it is actually ready to be applied to real-world instances of chemical components and mixtures.

**Index Terms**—equation of state, binodal curve, spinodal curve, data fitting, swarm intelligence, bat algorithm, Bézier curves

## I. INTRODUCTION

### A. Motivation

*Equations of state* (EoS) are of huge importance in physics, thermodynamics, and therefore engineering and the related fields [22]. Additionally, they state as a starting point for the development of constitutive equations. EoS are mathematical equations describing the relationship between the variables of temperature,  $T$ , volume,  $V$ , and pressure,  $P$ , for a pure component or a mixture (by applying the proper mixing rules). Therefore, they can be used to predict the behavior of thermodynamic systems under different conditions [18].

Different equations of state are used for different purposes and applied to different compounds with higher effectiveness than others (e.g., virial equation for polar molecules under

low pressures, Rackett equation for estimating the density of saturated liquids, and so on).

The simplest and most widely known EoS is undoubtedly the ideal gas law (universal gas equation). It is given by:

$$P.V = n.R.T = \frac{m}{M}.R.T \quad (1)$$

where  $m$  is the mass of the gas,  $M$  is the molar mass,  $n$  is the number of moles, and  $R = 0.082 \text{ L.atm.K}^{-1}.\text{mol}^{-1}$  is the universal gas constant. This equation is based on ideal assumptions, two of the most important ones being to neglect the inter-molecular forces between the gas molecules, and to consider the volume occupied by the molecules themselves negligible relative to the volume of the container. Still, it works properly for low pressures (i.e. around 1 atm) and high temperatures. However, it leads to inaccuracies when used under other conditions. In response, different corrections have been applied to this equation. A modification by Johannes D. Van der Waals was proposed in 1873, taking into account both molecular size and interaction forces between molecules, resulting in what is known nowadays as the *Van der Waals (VdW) equation*. It includes two parameters,  $a$  and  $b$ , characteristic of each substance. These parameters are real positive values and account for the inter-molecular forces and the molecular size, respectively [12]. The Van der Waals EoS is given by:

$$\left[ P + a \left( \frac{n}{V} \right)^2 \right] \left( \frac{V}{n} - b \right) = R.T \quad (2)$$

An isotherm curve can be represented graphically by fixing a value of the temperature  $T$  in (2) and then plotting  $P$  vs.  $V$  in the  $P$  vs.  $V$  plane (see our discussion in Sect. II). Analyzing the behavior of the substance for different isotherms, we can obtain a phase diagram, that represents graphically the different regions of coexistence and immiscibility of solid, liquid, and gas phases. The different areas enclosing the single-phase regions, are separated by phase boundaries (i.e., curves of non-analytic behavior where phase transitions occur). More specifically, the transition between gas and liquid phases for

Work funded by EU Horizon 2020 research and innovation programme (PDE-GIR project, MSCA grant agreement No 778035) and grant #TIN2017-89275-R of Spanish Agency of Research (AEI) and European Funds EFRD.

the VdW EoS can be further explained through the construction of two characteristic curves: the *binodal curve* and the *spinodal curve*. The former describes the conditions for two distinct phases to coexist, while the latter encloses a region of system's instability. Between both characteristic curves, a meta-stable region is defined.

Since the binodal and the spinodal curves cannot be calculated analytically, they are computed numerically by fitting a set of 2D points obtained for different isotherms. In particular, the sequence of points for the binodal curve is given by the leftmost roots, the critical point (the end point of the phase equilibrium curve, where liquid and gas phases are indistinguishable), and the rightmost roots of the isotherms. In turn, the spinodal curve is the curve connecting the sequence of points given by the local minima, the critical point, and the local maxima of the isotherms. Both sequences of points are determined by numerical procedures; then, polynomial data fitting is applied to obtain the characteristic curves (see [4], [6], [16] for details). This data fitting approach has several advantages, as polynomial functions are easy to plot and understand, and can be computed efficiently. However, it also has some drawbacks: for instance, it requires to provide an initial estimation of the degree of the approximating polynomial. This choice of the degree is critical for the good performance of the method. While high degree polynomials provide good numerical fitting, they tend to oscillate drastically and can lead to over-fitting. On the other hand, low degree polynomials are easier to work with, but present little flexibility and might lead to poor fitting. Hence, an adequate trade-off between both aspects is required. Unfortunately, it is a problem-dependent issue and generally difficult to determine. Finally, polynomials in canonical form can be difficult to apply to dynamic data fitting, as their coefficients lack a geometric meaning.

### B. Aims and Structure of this Paper

The limitations previously discussed can be solved by considering free-form parametric curves [8]. Among these curves, Bézier and B-splines are the most popular in computer-aided design and manufacturing (CAD/CAM) and other industrial settings [3], [16]. They are very flexible and can represent properly a large variety of continuous and smooth shapes [6]. In addition, their coefficients have a clear geometric meaning. Because of their simplicity, in this paper we focus on the particular case of Bézier curves. They are applied to perform least-squares approximation for the collection of 2D points associated with the binodal and spinodal curves. Since these curves are parametric, this task requires a proper data parameterization in addition to computing the poles of the curves. Unfortunately, this leads to a continuous nonlinear optimization problem that cannot be solved by standard mathematical optimization techniques. Thus, we use a metaheuristic algorithm for global optimization, called the bat algorithm. It is based on the echolocation behaviour of microbats. Our approach is then applied to real data of a gas. Our experimental results are very promising, as the method can reconstruct

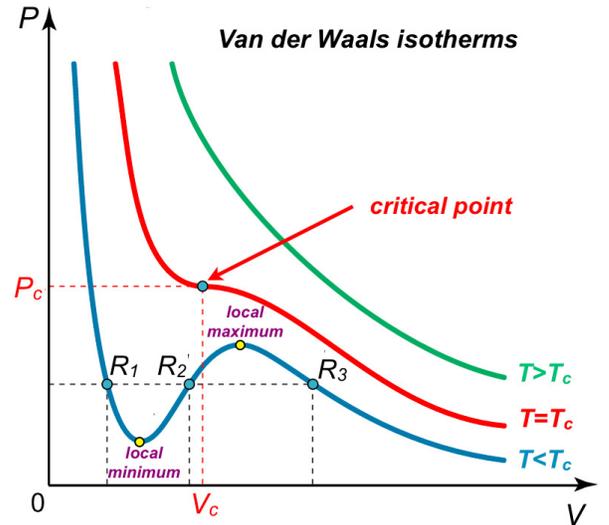


Fig. 1. Sketch of the isotherms for the VdW EoS and temperatures above (green), equal to (red), and below (blue) the critical temperature  $T_c$ .

the characteristic curves with high accuracy. In addition, the method is numerically stable and very fast.

This paper is organized as follows: Section II describes briefly the fundamentals of the problem to be solved, which is formulated as a continuous nonlinear optimization problem. Section III describes the bat algorithm, the evolutionary computation approach used in this work. The methodology proposed in this paper is described in detail in Section IV. Section V reports our experimental results. The paper closes with the main conclusions and some insight for future work in the field.

## II. PROBLEM TO BE SOLVED

### A. Background

In this paper, we consider the Van der Waals equation of state given by (2). From this equation, we can obtain an expression relating  $P$  and  $V$  for any value of  $T$ . Therefore, given a set of fixed temperatures,  $T_1, T_2, \dots, T_M$ , we can obtain a set of curves in the  $P$ - $V$  plane, called *isotherms*. Each temperature value has its associated isotherm. Fig. 1 shows an sketch of the shape of the different isotherms for any chemical component or mixture.

Without loss of generality, we can assume  $n = 1$  in Eq. (2) for simplicity. Multiplying it by  $V^2/P$  and rearranging terms, we get a cubic polynomial in  $V$ :

$$V^3 - \left(b + \frac{RT}{P}\right)V^2 + \frac{a}{P}V - \frac{ab}{P} = 0 \quad (3)$$

which has either one or three real roots. The first case occurs for temperatures  $T$  higher than the critical temperature,  $T_c$ , a value that is characteristic of each substance. A typical isotherm curve for this case is shown in green in Fig. 1. The second case, three real roots, happens for temperatures lower than  $T_c$ , when the isotherms oscillate up and down, as for

that in blue in Fig. 1. The isotherm corresponding to  $T = T_c$  (in red in Fig. 1), constitutes the boundary between these two cases. For this isotherm, the three real roots turn into a triple root, and merge into a single point, called the *critical point*. From now on, we will focus on the case  $T < T_c$ , where the isotherms have three real roots, labelled in increasing order as  $R_1$ ,  $R_2$  and  $R_3$ . For a liquid-gas system, the end roots,  $R_1$  and  $R_3$ , correspond to the liquid and the vapor phases, respectively. The second root,  $R_2$ , has not physical meaning; it is associated with an unstable molar volume and does not represent any real behavior.

Suppose now that we start with a temperature  $T < T_c$  and we increase it until we reach the critical value  $T_c$ . It can be observed that the molar volume of the saturated liquid, gets larger whereas the molar volume of the saturated vapor, gets smaller. Note that the saturated states correspond to the limit between the single-vapor or liquid phase, respectively, and two-phase liquid/vapor behavior. This means that increasing the temperature makes the end roots  $R_1$  and  $R_3$  move towards each other, until they merge into the critical point for  $T = T_c$ , in which the three roots become identical. This implies that at the critical point, the liquid and vapor molar volumes are equivalent, and therefore, liquid and vapor are indistinguishable. It can be proved that the critical values for the VdW EoS of a gas depend only on the parameters  $a$  and  $b$  as follows:

$$V_c = 3b, \quad P_c = \frac{a}{27b^2}, \quad T_c = \frac{8a}{27bR} \quad (4)$$

It is convenient to work with dimensionless variables by considering the reduced temperature, pressure and volume:

$$(T_r, P_r, V_r) = \left( \frac{T}{T_c}, \frac{P}{P_c}, \frac{V}{V_c} \right) \quad (5)$$

so, Eq. (3) becomes:

$$V_r^3 - \frac{1}{3} \left( 1 + \frac{8T_r}{P_r} \right) V_r^2 + \frac{3}{P_r} V_r - \frac{1}{P_r} = 0 \quad (6)$$

The isotherms for  $T < T_c$  in Fig. 1 show something surprising, as decreasing the volume from the right part makes the pressure rise, fall, and then rise again. This suggests that, for some molar volumes, compressing the fluid can cause its pressure to decrease. This means that the phase is unstable to density fluctuations. Thus, the stable states of such isotherms are divided in two parts: the first one, on the left, is characterized by relatively small molar volumes; it corresponds to the liquid states. The second part, on the right, is given by relatively large molar volumes; it corresponds to the gas states. Between these two parts, there is a set of unstable states.

This striking behavior violates the conditions of stability for thermodynamic equilibrium. A solution to fix this situation was proposed by famous scientist James Clerk-Maxwell in [15], and is now known as *Maxwell's equal area rule*. Basically, it states that the problem would be fixed if the oscillating part of the isotherm between the diluted liquid and the diluted gas was replaced by a horizontal line, called the *tie line*, that connects the dew point and the bubble point. According to

Maxwell's rule, the height of the tie line, should be taken such that it divides the isotherm into two equal regions.

### B. Data Points for the Binodal and Spinodal Curves

In order to obtain the list of data points defining the binodal curve, the following procedure is applied: first, we consider a set of increasing temperatures  $T_1 < T_2 < \dots < T_M < T_c$ . For each temperature  $T_j$ , ( $j = 1, \dots, M$ ) there exists a value of pressure, say  $P_j^*$ , that divides the isotherm into two equal areas, enclosed by the tie line (horizontal line) and the isotherm. Such value  $P_j^*$ , is computed by applying an iterative optimization procedure that calculates, compares and minimizes the difference between both areas. For this purpose, it starts from an initial guess  $\tilde{P}_j$  until convergence is reached. After this step, when the adequate pressure has been found, the corresponding roots of the isotherm, denoted as  $R_k^j$ , for ( $k = 1, 2, 3$ ), can be computed as the intersections of the isotherm curve for  $T_j$  and the tie line  $P = P_j^*$ . The collection of liquid and vapor roots for different isotherms, will define the binodal curve. Then, the list of data points for the binodal curve,  $\mathcal{B}$ , is given by:

$$\mathcal{B} = \left\{ \left\{ (R_1^j, P_j^*) \right\}_j, (1, 1), \left\{ (R_3^{M+1-j}, P_{M+1-j}^*) \right\}_{j=1, \dots, M} \right\} \quad (7)$$

Regarding the spinodal curve, it will be defined by the sequence of points consisting of the local minima of the isotherm curves,  $\mathbf{l}_j$ , the critical point, and the local maxima,  $\mathbf{L}_l$ . Note that in this paper vectors are denoted in bold.

These local optima are computed by solving  $dP/dV = 0$ , and checking the sign of the second derivative,  $d^2P/dV^2$  at the obtained solutions. If positive, the point is a minimum; otherwise, it is a maximum. Then, the list of data points for the spinodal curve,  $\mathcal{S}$ , is given by:

$$\mathcal{S} = \{ \{ \mathbf{l}_j \}_j, (1, 1), \{ \mathbf{L}_l \}_j \}_{j=1, \dots, M} \quad (8)$$

### C. Data Fitting of the Characteristic Curves

Once the data points for the binodal and spinodal curves are obtained, standard numerical routines for data fitting are applied to reconstruct such curves. Two different approaches can be applied to this problem: interpolation and approximation. Interpolation enforces the fitting function to pass through all the data points. This condition softens for approximation, where the function is only required to pass near the data points, according to a given metrics. Since data points for our problem are affected by noise, irregular sampling, and other disturbances, approximation is preferred, often in the form of least-squares optimization. In that case, we want to minimize the error functional  $\Xi$  given by the summed square of residuals, where the residual for the  $i$ -th data is defined as the difference between the observed data,  $\mu_i$ , and the fitted data,  $\hat{\mu}_i$ :

$$\Xi = \sum_{i=1}^{\chi} (\mu_i - \hat{\mu}_i)^2 \quad (9)$$

where  $\chi$  represents the number of data, and the fitted data are obtained from a certain fitting model function  $\varphi$ . Note that the

minimization process is performed on the free variables of  $\varphi$ . For the problem in this paper,  $\varphi$  is generally assumed to be a polynomial of a certain degree. As discussed in Sect. I-A, this choice has several advantages but also serious shortcomings, which can be avoided through free-form parametric curves. In this paper we consider the case of Bézier curves, as described in next section.

#### D. Data Fitting with Bézier Curves

We assume that the reader is familiar with the main concepts of free-form parametric curves [6], [8]. A *free-form parametric Bézier curve*  $\Phi(\tau)$  of degree  $\eta$  is defined as:

$$\Phi(\tau) = \sum_{j=0}^{\eta} \Delta_j \phi_j^\eta(\tau) \quad (10)$$

where  $\Delta_j$  are vector coefficients called *poles* (also *control points*),  $\phi_j^\eta(\tau)$  are the *Bernstein polynomials of index  $j$  and degree  $\eta$* , given by:

$$\phi_j^\eta(\tau) = \binom{\eta}{j} \tau^j (1-\tau)^{\eta-j}$$

with  $\binom{\eta}{j} = \frac{\eta!}{j!(\eta-j)!}$  and  $\tau$  is the *curve parameter*, defined on a finite interval  $[0, 1]$ . By convention,  $0! = 1$ .

Suppose that we are provided with a list of 2D data points  $\{\Theta_\mu\}_{\mu=1, \dots, \chi}$  in  $\mathbb{R}^2$ , such as those obtained for the binodal and spinodal curves, as described in Sect. II-B. Our goal is to compute the curve  $\Phi(\tau)$  approximating the data points better in the least-squares sense. To do so, we have to minimize the least-squares error,  $\Upsilon$ , defined as:

$$\Upsilon = \sum_{\mu=1}^{\chi} \left( \Theta_\mu - \sum_{j=0}^{\eta} \Delta_j \phi_j^\eta(\tau_\mu) \right)^2 \quad (11)$$

where we need a parameter value  $\tau_\mu$  to be associated with each data point  $\Theta_\mu$ ,  $\mu = 1, \dots, \chi$ . Considering the column vectors  $\Phi_j = (\phi_j^\eta(\tau_1), \dots, \phi_j^\eta(\tau_\chi))^T$ ,  $j = 0, \dots, \eta$ , where  $(\cdot)^T$  means transposition, and  $\Theta = (\Theta_1, \dots, \Theta_\chi)$ , Eq. (11) becomes the following system of equations (called the *normal equation*):

$$\begin{pmatrix} \Phi_0^T \cdot \Phi_0 & \dots & \Phi_\eta^T \cdot \Phi_0 \\ \vdots & \ddots & \vdots \\ \Phi_0^T \cdot \Phi_\eta & \dots & \Phi_\eta^T \cdot \Phi_\eta \end{pmatrix} \begin{pmatrix} \Delta_0 \\ \vdots \\ \Delta_\eta \end{pmatrix} = \begin{pmatrix} \Theta \cdot \Phi_0 \\ \vdots \\ \Theta \cdot \Phi_\eta \end{pmatrix} \quad (12)$$

which can be compacted as:

$$\mathcal{M} \mathcal{D} = \mathcal{R} \quad (13)$$

with  $\mathcal{M} = \left[ \sum_{j=1}^{\chi} \phi_l^\eta(\tau_j) \phi_i^\eta(\tau_j) \right]$ ,  $\mathcal{D} = (\Delta_0, \dots, \Delta_\eta)^T$ , and

$\mathcal{R} = \left[ \sum_{j=1}^{\chi} \Theta_j \phi_l^\eta(\tau_j) \right]$  for  $i, l = 0, \dots, \eta$ . If values are assigned to the  $\tau_i$ , Eq. (13) is a classical linear least-squares minimization that can readily be solved by standard numerical techniques. But if  $\tau_i$  are unknowns, the problem becomes more difficult. Since the blending functions  $\phi_j^\eta(\tau)$  are nonlinear in

$\tau$ , it is a nonlinear continuous optimization problem. It is also a multimodal problem, since there might be arguably more than one data parameterization vector leading to the optimal solution [16].

As shown, solving the parameterization problem leads to a difficult multimodal, multivariate, continuous, nonlinear optimization problem. This problem cannot be solved by applying classical mathematical optimization techniques [4]. In this paper, we address this optimization problem by using the bat algorithm, a powerful evolutionary computation method that has already been applied successfully to data fitting optimization in some previous works [9]–[11]. The algorithm is described in detail in next section.

### III. THE BAT ALGORITHM

The *bat algorithm* is a bio-inspired swarm intelligence algorithm originally proposed by Xin-She Yang in 2010 to solve continuous optimization problems [23], [24]. The algorithm is based on the echolocation behavior of microbats, which use a type of sonar called *echolocation*, with varying pulse rates of emission and loudness, to detect prey, avoid obstacles, and locate their roosting crevices in the dark. The idealization of the echolocation of microbats is as follows:

- 1) Bats use echolocation to sense distance and distinguish between food, prey and background barriers.
- 2) Each virtual bat flies randomly with a velocity  $\mathbf{v}_i$  at position (solution)  $\mathbf{x}_i$  with a fixed frequency  $f_{min}$ , varying wavelength  $\lambda$  and loudness  $A_0$  to search for prey. As it searches and finds its prey, it changes wavelength (or frequency) of their emitted pulses and adjust the rate of pulse emission  $r$ , depending on the proximity of the target.
- 3) It is assumed that the loudness will vary from an (initially large and positive) value  $A_0$  to a minimum constant value  $A_{min}$ .

Some additional assumptions are advisable for further efficiency. For instance, we assume that the frequency  $f$  evolves on a bounded interval  $[f_{min}, f_{max}]$ . This means that the wavelength  $\lambda$  is also bounded, because  $f$  and  $\lambda$  are related to each other by the fact that the product  $\lambda \cdot f$  is constant. For practical reasons, it is also convenient that the largest wavelength is chosen such that it is comparable to the size of the domain of interest (the search space for optimization problems). For simplicity, we can assume that  $f_{min} = 0$ , so  $f \in [0, f_{max}]$ . The rate of pulse can simply be in the range  $r \in [0, 1]$ , where 0 means no pulses at all, and 1 means the maximum rate of pulse emission.

With these idealized rules indicated above, the basic pseudo-code of the bat algorithm is shown in Algorithm 1. Basically, the algorithm considers an initial population of  $\mathcal{P}$  individuals (bats). Each bat, representing a potential solution of the optimization problem, has a location  $\mathbf{x}_i$  and velocity  $\mathbf{v}_i$ . The algorithm initializes these variables with random values within the search space. Then, the pulse frequency, pulse rate, and loudness are computed for each individual bat. Then, the swarm evolves in a discrete way over iterations, like time

---

**Require:** (Initial Parameters)

Population size:  $\mathcal{P}$  ; Loudness:  $\mathcal{A}$   
Maximum number of iterations:  $\mathcal{G}_{max}$   
Pulse rate:  $r$  ; Maximum frequency:  $f_{max}$   
Dimension of the problem:  $d$   
Objective function:  $\phi(\mathbf{x})$ , with  $\mathbf{x} = (x_1, \dots, x_d)^T$  ;  
Random number:  $\theta \in U(0, 1)$

- 1:  $g \leftarrow 0$
- 2: Initialize the bat population  $\mathbf{x}_i$  and  $\mathbf{v}_i$ , ( $i = 1, \dots, n$ )
- 3: Define pulse frequency  $f_i$  at  $\mathbf{x}_i$
- 4: Initialize pulse rates  $r_i$  and loudness  $\mathcal{A}_i$
- 5: **while**  $g < \mathcal{G}_{max}$  **do**
- 6:   **for**  $i = 1$  **to**  $\mathcal{P}$  **do**
- 7:     Generate new solutions by using eqns. (14)-(16)
- 8:     **if**  $\theta > r_i$  **then**
- 9:        $\mathbf{s}^{best} \leftarrow \mathbf{s}^g$    //select the best current solution
- 10:       $\mathbf{ls}^{best} \leftarrow \mathbf{ls}^g$    //generate a local solution around  $\mathbf{s}^{best}$
- 11:     **end if**
- 12:     Generate a new solution by local random walk
- 13:     **if**  $\theta < \mathcal{A}_i$  **and**  $\phi(\mathbf{x}_i) < \phi(\mathbf{x}^*)$  **then**
- 14:       Accept new solutions, increase  $r_i$  and decrease  $\mathcal{A}_i$
- 15:     **end if**
- 16:   **end for**
- 17:    $g \leftarrow g + 1$
- 18: **end while**
- 19: Rank the bats and find current best  $\mathbf{x}^*$
- 20: **return**  $\mathbf{x}^*$

---

**Algorithm 1:** Bat algorithm pseudocode

instances until the maximum number of iterations,  $\mathcal{G}_{max}$ , is reached. For each generation  $g$  and each bat, new frequency, location and velocity are computed according to the following evolution equations:

$$f_i^g = f_{min}^g + \beta(f_{max}^g - f_{min}^g) \quad (14)$$

$$\mathbf{v}_i^g = \mathbf{v}_i^{g-1} + [\mathbf{x}_i^{g-1} - \mathbf{x}^*] f_i^g \quad (15)$$

$$\mathbf{x}_i^g = \mathbf{x}_i^{g-1} + \mathbf{v}_i^g \quad (16)$$

where  $\beta \in [0, 1]$  follows the random uniform distribution, and  $\mathbf{x}^*$  represents the current global best location (solution), which is obtained through evaluation of the objective function at all bats and ranking of their fitness values. The superscript  $(.)^g$  is used to denote the current generation  $g$ . The best current solution and a local solution around it are probabilistically selected according to some given criteria. Then, search is intensified by a local random walk. For this local search, once a solution is selected among the current best solutions, it is perturbed locally through a random walk of the form:  $\mathbf{x}_{new} = \mathbf{x}_{old} + \epsilon \mathcal{A}^g$ , where  $\epsilon$  is a uniform random number on  $[-1, 1]$  and  $\mathcal{A}^g = \langle \mathcal{A}_i^g \rangle$ , is the average loudness of all the bats at generation  $g$ . If the new solution achieved is better than the previous best one, it is probabilistically accepted depending on the value of the loudness. In that case, the

algorithm increases the pulse rate and decreases the loudness. This process is repeated for the given number of iterations. In general, the loudness decreases once a new best solution is found, while the rate of pulse emission decreases. For simplicity, the following values are commonly used:  $\mathcal{A}_0 = 1$  and  $\mathcal{A}_{min} = 0$ , assuming that this latter value means that a bat has found the prey and temporarily stop emitting any sound. The evolution rules for loudness and pulse rate are as:  $\mathcal{A}_i^{g+1} = \alpha \mathcal{A}_i^g$  and  $r_i^{g+1} = r_i^0 [1 - \exp(-\gamma g)]$  where  $\alpha$  and  $\gamma$  are constants. Note that for any  $0 < \alpha < 1$  and any  $\gamma > 0$  we have:  $\mathcal{A}_i^g \rightarrow 0$ ,  $r_i^g \rightarrow r_i^0$  as  $g \rightarrow \infty$ . Generally, each bat should have different values for loudness and pulse emission rate, which can be achieved by randomization. To this aim, we can take an initial loudness  $\mathcal{A}_i^0 \in (0, 2)$  while the initial emission rate  $r_i^0$  can be any value in the interval  $[0, 1]$ . Loudness and emission rates will be updated only if the new solutions are improved, an indication that the bats are moving towards the optimal solution.

Bat algorithm has been successfully applied to several problems, such as image thresholding [1], curve reconstruction [11], optimal design of structures in civil engineering [14], robotics [19]–[21], fuel arrangement optimization [13], and many others. The interested reader is also referred to the general paper in [25] for a review of the bat algorithm, its variants and other interesting applications.

## IV. THE METHOD

### A. Overview of the Method

As explained in Sect. I-A, the VdW Eos in (2) depends on two parameters,  $a$  and  $b$ , specific for each chemical compound. The initial input of our problem consists of these parameters,  $a$  and  $b$ , and a list of temperatures  $T_1 < T_2 < \dots < T_M$  assumed to be under the critical temperature of the substance,  $T_c$ . Our method consists of the following steps:

- (1) Compute the critical values  $V_c, P_c, T_c$  using (4).
- (2) Compute the reduced variables  $V_c, P_c, T_c$  through (5).
- (3) Compute the isotherms from (2) at temperatures  $T_j$ .
- (4) For each isotherm of temperature  $T_j$ :
  - (4a) Consider an initial guess  $\tilde{P}_j$  and apply optimization for obtaining the value of  $P_j^*$ , according to Maxwell's rule.
  - (4b) Use  $P_j^*$  to compute the roots of (6), as described in Sect. II-B.
  - (4c) Obtain the local optima of (6), as described in Sect. II-B.
- (5) Perform data fitting with Bézier curves on  $\mathcal{B}$  and  $\mathcal{S}$  as follows:
  - (5a) Apply the bat algorithm to data parameterization for the lists  $\mathcal{B}$  and  $\mathcal{S}$  (see Sect. IV-B for details).
  - (5b) Apply least-squares optimization to compute the poles of the curve. The resulting system of equations can readily be solved by well-known numer-

ical procedures, such as the standard LU decomposition, singular value decomposition (SVD), and a modification of the LU decomposition for non-squared sparse problems (see [17] for details).

Step (5a) is the most critical part of the method and also the most remarkable contribution of this paper. It will be explained in detail in next section.

### B. Bat Algorithm for Data Fitting

In this section, the bat algorithm described in Sect. III is applied to data parameterization with Bézier curves. In order to succeed, there are some issues that we need to take care of:

1. *Bat encoding.* First of all, we need an adequate representation of the free variables of our optimization problem. Bats in our method, denoted as  $\mathcal{B}_k$ , are vectors of length  $M$  of real-valued numbers on the interval  $[0, 1]$  corresponding to a parameterization of data points:

$$\mathcal{B}_k = (\rho_1^k, \rho_2^k, \dots, \rho_M^k) \subset [0, 1]^M \quad (17)$$

All bats  $\{\mathcal{B}_k\}_k$  are initialized with uniformly distributed random numbers on the parametric domain. The  $\{\rho_i^k\}_i$  are also sorted in increasing order to replicate the ordered structure of data parameterization.

2. *Fitness function.* It corresponds to the evaluation of the least-squares function given by (11). But since this error function does not consider the number of data points, we also compute the RMSE (root-mean squared error), given by:

$$RMSE = \sqrt{\frac{\Upsilon}{\chi}} \quad (18)$$

3. *Curve parameters.* In this case, the only curve parameter is the curve degree,  $\eta$ , also determining its number of poles. In this paper, the optimal value is determined empirically. To this aim, we compute the RMSE for values of  $\eta$  ranging from the minimum  $\eta = 2$  to  $\eta = 9$ . The value minimizing the RMSE is finally selected.

4. *Bat algorithm parameters.* The bat algorithm depends of some control parameters that have to be properly tuned, as they determine the performance of the method at large extent. This task is challenging, because it is strongly problem-dependent. In this paper, our choice is based on a large collection of empirical results. The different parameters used in this work are arranged in rows in Table I. For each parameter, the table shows (in columns) its symbol, meaning, range of values, and the parameter value chosen in this paper. The most important parameters are the population size,  $\mathcal{P}$ , and the maximum number of iterations,  $\mathcal{G}_{max}$ . For the former, we set  $\mathcal{P} = 100$  in all examples of this paper. We also tried larger populations (up to 300 individuals) but found that our results do not change significantly. For the latter, bat algorithm is specially favorable in this regard with respect to other algorithms, which typically require a large number of iterations to converge. In our simulations, we found that  $\mathcal{G}_{max} = 500$  is enough to reach convergence in all our executions.

After selecting those parameters, the bat algorithm is run iteratively for the given number of iterations. Then, the bat

TABLE I  
BAT ALGORITHM PARAMETERS AND THEIR VALUES IN THIS PAPER.

Symbol	Meaning	Range	Used Value
$\mathcal{P}$	population size	50-300	100
$\mathcal{G}_{max}$	max. number of iterations	200-2,000	500
$\mathcal{A}^0$	initial loudness	(0, 2)	0.5
$\mathcal{A}_{min}$	minimum loudness	[0, 1]	0
$r^0$	initial pulse rate	[0, 1]	0.2
$f_{max}$	maximum frequency	[0, 10]	1.5
$\alpha$	multiplicative factor	(0, 1)	0.3
$\gamma$	exponential factor	[0, 1]	0.2

with the best fitness value for (18) is selected as the best solution to the problem.

## V. EXPERIMENTS AND RESULTS

### A. Application to a Real Example: Argon

Our method has been applied to the VdW EoS of a chemical component: the Argon, a noble gas with symbol *Ar* and atomic number 18. Argon is the third-most abundant gas in the Earth's atmosphere, and the most abundant noble gas in the Earth's crust. It is widely used in welding and in other high-temperature industrial processes. Its parameter values for the VdW EoS are [22]:  $a = 1.355$  and  $b = 0.03201$ . The critical temperature is  $T_c = 150.86$  K, with an uncertainty of 0.1 K according to [2], [7].

We applied the steps (1)-(3) of our workflow for the sets of temperatures:  $\{130, 133, 135, 137, 140, 142, 145, 147, 148, 149, T_c\}$  and  $\{128, 130, 133, 135, 137, 140, 142, 145, 147, 148, 149, 150.2, T_c\}$  for the binodal and the spinodal curves, respectively.

Then, step (4) is applied to obtain the lists  $\mathcal{B}$  and  $\mathcal{S}$  of data points for the binodal and the spinodal curves. The optimization process in step (4a) is carried out through standard polynomial linear fitting using the Vandermonde matrix. Then, the procedure described in Sect. IV-B is applied to perform data parameterization (step (5a) of our workflow). Finally, pole computation is achieved solving the resulting linear system of equations through SVD.

### B. Computational Results

To remove the stochastic effects and avoid premature convergence, 30 independent executions have been carried out for each value of parameter  $\eta$ . Then, the 10 worst simulations are rejected to prevent the spurious effects of unstable behaviors. Tables II and III report our computational results for the curve degree ranging from  $\eta = 2$  to  $\eta = 9$  (in rows). For each table, the following data are listed (in columns): the curve degree, the best RMSE (out of the 30 simulations), and the RMSE mean (for the 20 best simulations). We have also checked for over-fitting (column 4): we use the symbol  $\checkmark$  to indicate that over-fitting occurs, and symbol  $\times$  otherwise. Finally, we also analyzed the stability of the solutions (column 5), this concept being an indicator about whether or not several executions for the same values yield qualitatively different solutions. This is ranked from 1 (very low stability) to 5 (very high stability) using the diamond  $\blacklozenge$  symbol as many times as the value.

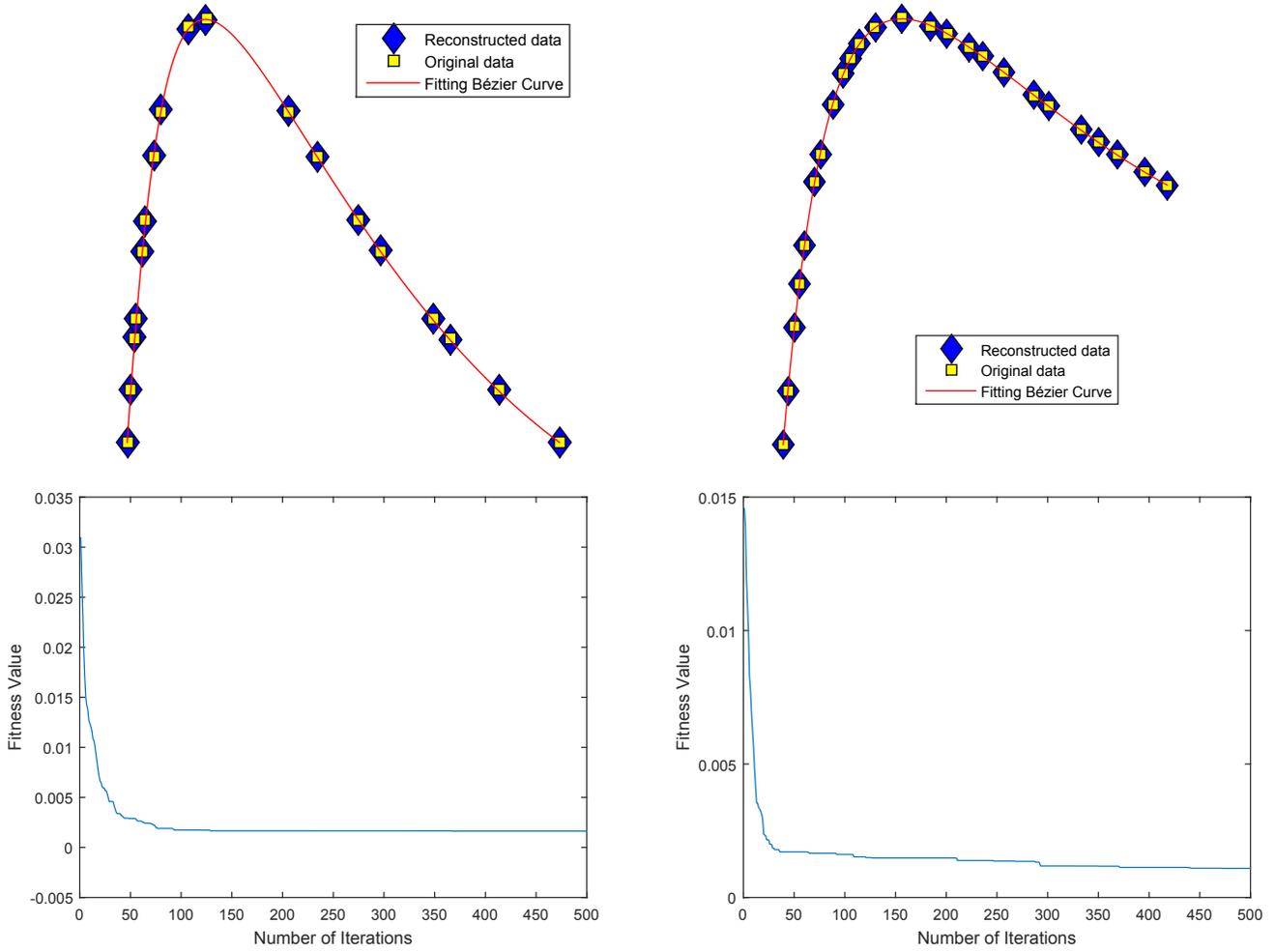


Fig. 2. Best Bézier fitting curve (top) and bat algorithm convergence diagram (bottom) for the binodal (left) and spinodal (right) curves.

TABLE II  
COMPUTATIONAL RESULTS FOR THE BINODAL CURVE.

Degree	RMSE (best)	RMSE (mean)	Over-fitting	Stability
$\eta = 2$	3.5349E-2	5.6509E-2	×	◆◆◆
$\eta = 3$	7.7394E-5	9.8801E-5	×	◆◆◆◆◆
$\eta = 4$	8.5572E-5	1.0833E-4	×	◆◆◆◆◆◆
$\eta = 5$	1.0365E-4	1.1776E-4	×	◆◆◆◆◆
$\eta = 6$	1.1927E-4	1.3328E-4	×	◆◆◆◆◆
$\eta = 7$	1.1624E-4	1.2684E-4	×	◆◆◆◆◆
$\eta = 8$	1.3014E-4	1.4915E-4	✓	◆◆◆◆◆
$\eta = 9$	1.2293E-4	1.3579E-4	✓	◆◆◆◆◆

TABLE III  
COMPUTATIONAL RESULTS FOR THE SPINODAL CURVE.

Degree	RMSE (best)	RMSE (mean)	Over-fitting	Stability
$\eta = 2$	4.9074E-2	7.1226E-2	×	◆◆
$\eta = 3$	9.8649E-5	1.0356E-4	×	◆◆◆◆◆
$\eta = 4$	1.1267E-4	1.2842E-4	×	◆◆◆◆◆◆
$\eta = 5$	1.2060E-4	1.2953E-4	×	◆◆◆◆◆
$\eta = 6$	1.4971E-4	1.8611E-4	✓	◆◆◆◆◆
$\eta = 7$	1.4750E-4	1.6975E-4	✓	◆◆◆◆◆
$\eta = 8$	1.6122E-4	2.0137E-4	✓	◆◆◆◆◆
$\eta = 9$	1.6678E-4	1.9152E-4	✓	◆◆◆◆◆

Some important observations can be obtained from our results in Tables II and III. As the reader can see, the method works pretty well, as very good fitting errors are achieved. The best and mean RMSE reach at least order  $10^{-4}$  for all curve degrees except  $\eta = 2$ . This can be explained by the fact that neither the binodal nor spinodal curves are really parabolas, so they cannot be fitted well with a quadratic curve. The best fitting curves are obtained for  $\eta = 3$ , meaning that four poles

seem to be best suited for optimal fitting. However, the value  $\eta = 4$  also yields very good results. Another observation is that RMSE degrees from  $\eta = 4$  to  $\eta = 9$  are generally of the same order, which is consistent with the fact the high-degree functions provide extra degrees of freedom (DOFs) and hence, they tend to fit the data points better. Of course, this occurs at the expense of the higher model complexity, so some kind of trade-off between the quality of the model and its complexity

might be required. Another problem is that DOFs might lead the model to over-fitting. We checked this issue through several trials and found that indeed, this happens for models of degree  $\eta \geq 8$  and  $\eta \geq 6$  for the binodal and the spinodal curves, respectively. This means that only low degree curves must be considered to apply the model as a predictor for other values of the temperature. This fact is also confirmed by the stability analysis: the most stable solutions occur for  $\eta = 3$  and  $\eta = 4$  for both binodal and spinodal curves.

Our good numerical results are also confirmed graphically. Figure 2 (top) shows the best fitting curves obtained for the binodal (left) and spinodal (right) curves. The original and the reconstructed data points are shown as yellow filled squares and blue filled diamonds, respectively. The figure also shows the best Bézier fitting curve, displayed as a red solid line. Note the excellent matching between original and reconstructed data points, and the very good fitting through the Bézier curve.

### C. Computing Times and Implementation Details

An important observation is that the CPU times are very competitive, as any single execution takes about 30~90 seconds. This is in contrast with other swarm intelligence methods, which can require several minutes for a single run to finish. It can be explained by the fact that this method does not require many iterations to converge. These excellent CPU times improve the applicability of our method are one of its most appealing features for practical settings.

About the implementation details, all computations in this paper have been performed on a 3.4 GHz. Intel Core i7 processor with 8 GB. of RAM. The source code has been implemented by the authors in the native programming language of the popular scientific program *Matlab*, version 2018b.

## VI. CONCLUSIONS AND FUTURE WORK

In this paper, we introduce a new method to construct the characteristic curves of the VdW EoS through data fitting using free-form Bézier curves. Given the parameters  $a$  and  $b$  of a chemical system, our method computes sets of data points for the binodal and spinodal curves; then, it applies the bat algorithm to data parameterization; finally, the poles of the curves are obtained by least-squares optimization with SVD. The method is applied to Argon, a noble gas for which the characteristic curves are computed with very good accuracy. Furthermore, the method is numerically stable and very fast, with CPU times in the range of tens of seconds for a single execution.

Our future work includes to improve the accuracy of our method even further, as well as to apply it to other interesting chemical components and mixtures. We also plan to extend this approach for the analysis of other equations of state described in the literature. Other interesting problems in the realm of chemical engineering and related fields will also be considered for this approach.

### ACKNOWLEDGMENT

The authors acknowledge the financial support from the project PDE-GIR of the European Union's Horizon

2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 778035, and from the the Spanish Ministry of Science, Innovation and Universities (Computer Science National Program) under grant #TIN2017-89275-R of the Agencia Estatal de Investigación and European Funds EFRD (AEI/FEDER, UE).

### REFERENCES

- [1] Alihodzic, A., Tuba, M.: Improved bat algorithm applied to multilevel image thresholding. *The Scientific World Journal*, **2014**, article ID 176718, 16 pages (2014).
- [2] Angus, S., Armstrong, B., Gosman, A.L., McCarty, R.D., Hust, J.G., Vasserman, A.A., Rabinovich, V.A.: *International Thermodynamic Tables of the Fluid State - 1 Argon*, Butterworths, London (1972).
- [3] Barnhill, R.E.: *Geometric Processing for Design and Manufacturing*. SIAM, Philadelphia (1992).
- [4] Dierckx, P.: *Curve and Surface Fitting with Splines*. Oxford University Press, Oxford (1993).
- [5] Engelbrecht, A.P.: *Fundamentals of Computational Swarm Intelligence*. John Wiley and Sons, Chichester, England (2005).
- [6] Farin, G.: *Curves and surfaces for CAGD (5th ed.)*. Morgan Kaufmann, San Francisco (2002).
- [7] Gosman, A.L., McCarty, R.D., Hust, J.G.: Thermodynamic properties of Argon from the triple point to 300 K at pressures to 1000 atmospheres. *Nat. Stand. Ref. Data Ser., Nat. Bur. Stand.*, NSRDS-NBS 27 (1969).
- [8] Hoschek, J., Lasser, D.: *Fundamentals of Computer Aided Geometric Design*. A.K. Peters, Wellesley, MA (1993).
- [9] Iglesias, A., Gálvez, A., Collantes, M.: Bat algorithm for curve parameterization in data fitting with polynomial Bézier curves. In: *Proc. of Cyberworlds, CW'2015*, IEEE Computer Society Press, Los Alamitos, CA, 107–114 (2015).
- [10] Iglesias, A., Gálvez, A., Collantes, M.: Global-support rational curve method for data approximation with bat algorithm. *IFIP Advances in Information and Communication Technology*, **458**, 191–205 (2015).
- [11] Iglesias, A., Gálvez, A., Collantes, M.: Multilayer embedded bat algorithm for B-spline curve reconstruction. *Integrated Computer-Aided Engineering*, **24**(4), 385–399 (2017).
- [12] Johnson, D. C.: *Advances in thermodynamics of the van der Waals Fluid*. Morgan & Claypool Publishers, Ames, (2014).
- [13] Kashi, S., Minucmehr, A., Poursalehi, N., Zolfaghari, A.: Bat algorithm for the fuel arrangement optimization of reactor core. *Annals of Nuclear Energy*, **64**, 144–151 (2014).
- [14] Kaveh, A., Zakian, P.: Enhanced bat algorithm for optimal design of skeletal structures. *Asian Journal Civil Engineering*, **15**(2), 179–212 (2014).
- [15] Maxwell, J.C.: On the dynamical evidence of the molecular constitution of bodies". *Nature*, **11**, 357–359 (1875).
- [16] Piegl, L., Tiller, W.: *The NURBS Book*, Springer Verlag, Berlin Heidelberg (1997). 146–165 (2004).
- [17] Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: *Numerical Recipes (second ed.)*. Cambridge University Press, Cambridge (1992).
- [18] Smith, J.M., Van Ness, H.C., Abbott, M.M.: *Introduction to chemical engineering thermodynamics*. McGraw-Hill, Boston (2005).
- [19] Suárez, P., Iglesias, A.: Bat algorithm for coordinated exploration in swarm robotics. *Advances in Intelligent Systems and Computing*, **514**, 134–144 (2017).
- [20] Suárez, P., Gálvez, A., Iglesias, A.: Autonomous coordinated navigation of virtual swarm bots in dynamic indoor environments by bat algorithm. Int. Conf. in Swarm Intelligence, ICSI 2017. *Lecture Notes in Computer Science*, **10386**, 176–184 (2017).
- [21] Suárez, P., Iglesias, A., Gálvez, A.: Make robots be bats: specializing robotic swarms to the bat algorithm. *Swarm and Evolutionary Computation*, **44**, 113–129 (2019).
- [22] Weast, R.C.: *Handbook of Chemistry and Physics (53rd Edition)*. Chemical Rubber Pub. (1972).
- [23] Yang, X.S.: A new metaheuristic bat-inspired algorithm. *Studies in Computational Intelligence*, Springer Berlin, **284**, 65–74 (2010).
- [24] Yang, X.S., Gandomi, A.H.: Bat algorithm: a novel approach for global engineering optimization. *Eng. Computations*, **29**(5) (2012), 464–483.
- [25] Yang, X.S.: Bat algorithm: literature review and applications. *Int. J. Bio-Inspired Computation*, **5**(3) (2013), 141–149.