1s. Independency of Gibbs-Duhem equation by the derivation variable

Let γ_1 and γ_2 two functions of the molar fractions (x_1, x_2) and temperature (T). Therefore, the following phrase is valid

$$x_1 * \frac{d(\ln \gamma_1)}{dx_1} + x_2 * \frac{d(\ln \gamma_2)}{dx_1} = 0 \Rightarrow x_1 * \frac{d(\ln \gamma_1)}{dx_2} + x_2 * \frac{d(\ln \gamma_2)}{dx_2} = 0. \quad (1s)$$

Proof:

By applying a change of variable in equation (1s) we obtain

$$x_1 * \frac{d(\ln \gamma_1)}{dx_1} + x_2 * \frac{d(\ln \gamma_2)}{dx_1} = x_1 * \frac{d(\ln \gamma_1)}{dx_2} * \frac{dx_2}{dx_1} + x_2 * \frac{d(\ln \gamma_2)}{dx_2} * \frac{dx_2}{dx_1}. \quad (2s)$$

Further, we know that x_1 and x_2 are correlated by the following equation

$$x_2 = 1 - x_1 \Rightarrow \frac{dx_2}{dx_1} = -1. \quad (3s)$$

Therefore, the right hand of equation (2s) becomes

$$x_1 * \frac{d(\ln \gamma_1)}{dx_2} * \frac{dx_2}{dx_1} + x_2 * \frac{d(\ln \gamma_2)}{dx_2} * \frac{dx_2}{dx_1} = -x_1 * \frac{d(\ln \gamma_1)}{dx_2} - x_2 * \frac{d(\ln \gamma_2)}{dx_2}, \quad (4s)$$

so

$$x_1 * \frac{d(\ln \gamma_1)}{dx_2} + x_2 * \frac{d(\ln \gamma_2)}{dx_2} = -\left(x_1 * \frac{d(\ln \gamma_1)}{dx_1} + x_2 * \frac{d(\ln \gamma_2)}{dx_1}\right) = -0. \quad (5s)$$

To conclude we have the (1s).

2s. Computational complexity

The analysis of the computational complexity of the characterization algorithm was executed on a Windows machine equipped with two CPU Intel Xeon Gold 5220 @ 2.20GHz, 192GB of RAM executing a Windows Server 2019 standard edition. Despite the machine allowed process multi-core processing, the library was executed using only one process and no parallelization was executed. As highlighted in the main paper, the characterization algorithm was test using mixtures from 2 to 5 components and polynomial order of the function from 1 to 5. Further information about the test executed can be found

in the main text of this paper. Figure 1s reports the computational time required for the characterization of the statistical function varying the amount of components in the mixture and the order of the selected polynomial. The figure reports the y-axis in logarithmic scale. From this figure it is possible to detect how the points of each curve are spread around a line. This means that, for both the variables, the complexity of the algorithm is exponential. Furthermore, the data have been used to fir the function $e^{a*NC*PO}$, where a is a fitting parameter, NC is the amount of components within the mixture and PO is the selected polynomial order. The fit over the experimental points reported in Figure 1s returned an R^2=0.99 with a parameter a=0.3694 confirming the exponential complexity of the proposed algorithm.

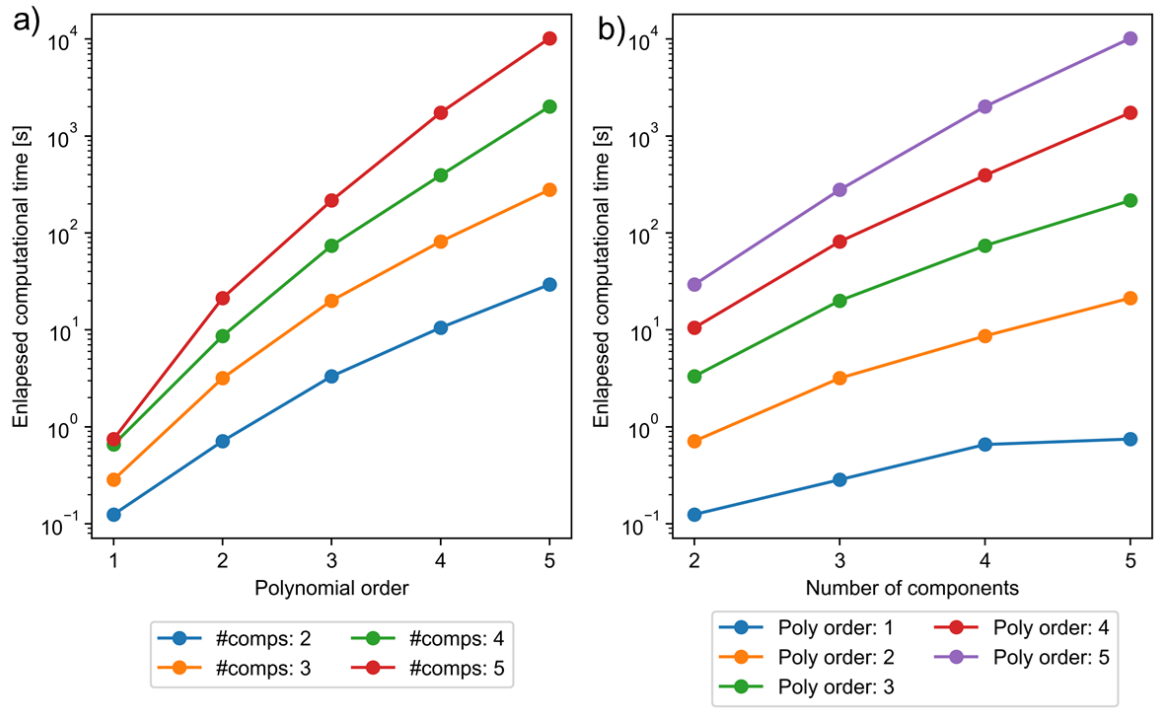

Figure 1s. Computational time analysis for the characterization function. The time is reported on the y-axis in logarithmic scale. a) Profile of the computation time varying the polynomial order parametric in the number of components. b) Profile of the computational time varying the number of components parametric in the polynomial order.

The analysis of the algorithm complexity was run only on the characterization part of the framework since this is the slowest step. After the characterization of the statistical function it is transformed in a python lambda function and the computational time of the function results to be negligible compared to the time reported in Figure 1s.

3s. The Aspen dataset

The Aspen experimental dataset was obtained using Aspen Plus V10 and non-random two liquids (NRTL) as model to compute the activity coefficients. The binary interaction parameters included in the NRTL model were obtained from Aspen internal dataset and they are reported in Table1s.

Table 1s. Binary interaction coefficients utilized to compute the Aspen dataset. The components names reported in the table are abbreviation. Following an abbreviation description is reported. THF: tetrahydrofuran, cyC6: cyclo-hexane, AcN: acetonitrile, Benz: benzene.

Component		A		B		C	D	E		F	
i	j	ij	ji	ij	ji	ij	ij	ij	ji	ij	ji
THF	cyC6	0	0	153.315	31.1346	0.3	0	0	0	0	0
THF	AcN	0	0	58.7341	145.02	0.3	0	0	0	0	0
cyC6	AcN	0	0	612.508	464.63	0.3	0	0	0	0	0
THF	Benz	0	0	308.159	-280.769	0.3	0	0	0	0	0
cyC6	Benz	0	0	-43.3406	182.755	0.3	0	0	0	0	0
AcN	Benz	0	0	128.25	220.003	0.3	0	0	0	0	0

The activity coefficients were calculated at the boiling point of the mixture. The mixtures were characterized by a molar fraction of overall impurities of 0.05, 0.1 and 0.2. The ratio between the acetonitrile and benzene was kept to 1. The concentration of tetrahydrofuran was varied to 0 to the maximum allowed from the impurity concentration with a variation of 0.05. The molar fraction of cyclohexane was calculated as the sum to 1 of the impurities concentration and tetrahydrofuran. All the mixtures used in the simulation did not show any phase separation.

4s. Function employed for the internal disturbance framework validation

The equations system reported in (6s) reports the model characterized for the validation case with internal disturbance.

$$
\begin{cases}
\ln(\gamma_1) = [C(1,8) - C(1,17) + C(3,3) - C(3,8) + C(3,17)] * x_1^2 x_3 + \\
+[B(1,8) - C(1,8) - C(1,9) * T + 2C(1,17) + C(1,18) * T - 3/2\, C(3,3) + 3/2\, C(3,8) - 3/2\, C(3,17)] * x_1 x_3 + \\
+B(1,6) * x_2 x_3 + B(1,8) * x_3^2 + C(1,6) * x_1 x_2 x_3 + C(1,8) * x_1 x_3^2 + C(1,9) * T * x_1 x_3 + C(1,12) * x_2^2 x_3 + \\
+C(1,14) * x_2 x_3^2 + C(1,15) * T * x_2 x_3 + C(1,17) * x_3^3 + C(1,18) * T * x_3^2 - C(1,18) * T * x_3 + \\
[1/2\, C(3,3) - C(1,17) - B(1,8) - 1/2\, C(3,8) + 1/2\, C(3,17)] * x_3 \\
\\
\ln(\gamma_2) = [C(2,14) - C(2,17) + C(3,12) - C(3,14) + C(3,17)] * x_2^2 x_3 + \\
+[B(2,8) - C(2,14) - C(2,15) * T + 2C(2,17) + C(2,18) * T - 3/2\, C(3,12) + 3/2\, C(3,14) - 3/2\, C(3,17)] * x_2 x_3 + \\
+B(2,3) * x_1 x_3 + B(2,8) * x_3^2 + C(2,3) * x_1^2 x_3 + C(2,6) * x_1 x_2 x_3 + C(2,8) * x_1 x_3^2 + C(2,9) * T * x_1 x_3 + \\
+C(2,14) * x_2 x_3^2 + C(2,15) * T * x_2 x_3 + C(2,17) * x_3^3 + C(2,18) * T * x_3^2 - C(2,18) * T * x_3 + \\
+[1/2\, C(3,12)\ - C(2,17) - B(2,8) - 1/2\, C\ (3,14) + 1/2\, C(3,17)] * x_3 \\
\\
\ln(\gamma_3) = B(3,8) * x_1 x_3 + B(3,8) * x_2 x_3 + B(3,8) * x_3^2 + C(3,3) * x_1^2 x_3 + C(3,6) * x_1 x_2 x_3 + C(3,8) * x_1 x_3^2 + \\
-C(3,8) * x_1 x_3 + C(3,12) * x_2^2 x_3 + C(3,14) * x_2 x_3^2 - C(3,14) * x_2 x_3 + 2C(3,17) * x_1 x_3 + 2C(3,17) * x_2 x_3 \\
+C(3,17) * x_3^3 + C(3,18) * T * x_1 x_3 + C(3,18) * T * x_2 x_3 + C(3,18) * T * x_3^2 + \\
-C(3,18) * T * x_3 + [C(3,17) + B(3,8)] * x_3
\end{cases}
\tag{6s}
$$