

NUMERICAL AND COMPUTATIONAL ASPECTS OF COSMO-BASED ACTIVITY COEFFICIENT MODELS

Luiz F. K. Possani¹ and Rafael de P. Soares^{1*}

¹ Universidade Federal do Rio Grande do Sul, Departamento de Engenharia Química, Laboratório Virtual de Predição de Propriedades, Porto Alegre, RS, Brasil. ORCID: 0000-0002-4648-4659; E-mail: rafael@enq.ufrgs.br, ORCID: 0000-0002-0636-1363

(Submitted: November 10, 2017 ; Revised: May 21, 2018 ; Accepted: May 25, 2018)

Abstract - In the present work, some numerical and computational aspects of COSMO-based activity coefficient models were explored. The residual contribution in such models rely on the so called self-consistency equation. This equation does not have a closed-form solution and is usually solved by the successive substitution method. The performance of a classical Newton-Raphson method was tested in solving the self-consistency equation. The results obtained by the Newton implementation and by successive substitution agreed within the convergence tolerance. The CPU times for solving the model using both methods also were compared. Contradicting the usual experience, it was observed that the Newton method becomes slower than successive substitution when the number of components (or number of COSMO segments) in the mixture increases. An analysis of the number of floating point operations required showed the same, Newton's method will be faster only for small systems.

Keywords: COSMO-RS; COSMO-SAC; F-SAC; Newton; Quasi-Newton.

INTRODUCTION

Accurate prediction of mixture behavior and phase equilibrium properties are of great importance in the design, efficient operation, control and optimization of industrial plants. Currently the predictive activity coefficient models of greater success are UNIFAC (Fredenslund et al., 1975) and its variations, at least for engineering purposes. However, UNIFAC and its variants require a large amount of experimental data for the determination of their parameters.

In this sense, the *a priori* COSMO-based models, such as COSMO-RS (Klamt, 1995) and variations, are interesting alternatives for the prediction of activity coefficients in mixtures and phase equilibria calculations. For example in the work of Gerber and Soares (2013), the freely available version of Dortmund's modified UNIFAC Jakob et al. (2006) and COSMO-SAC models performances for predicting

experimental infinite dilution activity coefficient data were compared. The results showed that COSMO-SAC provided more reliable predictions for multi-functional or more complex molecules.

The increasing popularity of COSMO-based models led to continuous modifications proposed in the literature – *e.g.*, COSMO-SAC (Lin and Sandler, 2002), COSMO-RS(O1) (Grensemann and Gmehling, 2005), and F-SAC (Soares and Gerber, 2013; Soares et al., 2013). The COSMO (*C*onductor-like *S*creening *M*ODEl) theory was first developed by Klamt and Schuurmann (1993) in order to compute the effects of solvents and solvation energies. Based on COSMO methodology, Klamt (1995) developed the COSMO-RS (*R*ealistic *S*olvation) model. This model enabled the translation from an ideal state (complete solvated molecule in a perfect conductor) to the *real solution*, when in contact with other molecules.

* Corresponding author: Rafael de P. Soares - E-mail: rafael@enq.ufrgs.br

Each molecule is considered as a collection of surface segments and the chemical potential is calculated for the segments using the interaction energies between the segments. Then, the chemical potential for each molecule is computed by summing the segment energies.

Consider, for example, one possible contact for benzene and acetone molecules, as shown in Figure 1.

For each contact between the two molecules, the conductor is partially excluded. Eventually, the molecules in solution are completely put in contact with each other and the real solution state is achieved. It is evident that there are infinite possible arrangements for these contacts, therefore, a statistical thermodynamics treatment is necessary.

Using the COSMO-SAC (Lin and Sandler, 2002) nomenclature, the activity coefficient of a segment Γ_m , resulting from the statistical thermodynamics treatment, either in a pure liquid or mixture, is given by:

$$\ln \Gamma_m = -\ln \left[\sum_n p_n \Gamma_n e^{\frac{-\Delta W_{m,n}}{RT}} \right] \quad (1)$$

where p_n is the probability of finding a segment n either in a pure liquid or mixture; and $\Delta W_{m,n}$ is the interaction energy for the contact between segments m and n .

Solving Equation 1 for the activity coefficient represents a particular difference between COSMO-type and classical activity coefficient models,

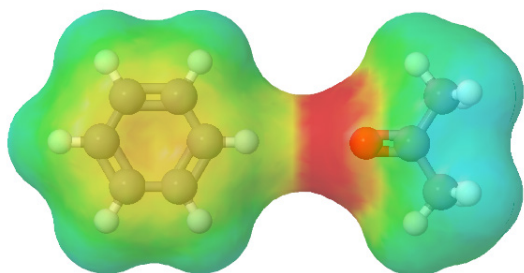


Figure 1. Representation of two molecules in contact according to COSMO-RS theory.

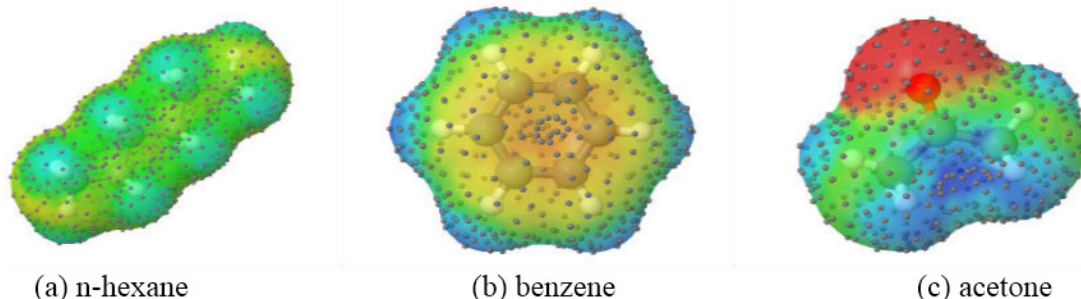


Figure 2. COSMO surface charge density segment centers (small balls) for some molecules.

such as Wilson, NRTL, UNIQUAC and UNIFAC variants. While the classical models are explicit equations, COSMO-based models relying on Equation 1 do not have a closed form expression. This equation is also known as the *self-consistency* equation (Klamt, 1995; Klamt et al., 1998; Lin and Sandler, 2002; Gmehling and Gmehling, 2005).

It is noteworthy that Equation 1 is actually a system of equations, consisting of one equation for each molecular segment m . Each molecule can have dozens of different segments, leading to a very large system of equations to be solved in mixtures of several different substances. Further, the actual computation of activity coefficients requires the solution of a system like Equation 1 for each pure compound and one for the mixture. In Figure 2 the actual segment centers (small balls) are shown for some typical molecules when using the JCOSMO package of Gerber and Soares (2010).

The freely available COSMO-SAC FORTRAN code of Virginia Tech (available at <http://www.design.che.vt.edu/VT-Databases.html>) represents all molecules with a fixed maximum number of segments (51 segments per molecule) with surface charge densities σ equally distributed in the range from -0.025 to 0.025 e/Å². However, as can be observed in Figure 2, the total number of segments obtained in the COSMO calculation (small balls) usually is much more than 51. This is actually treated by *averaging* and merging segments with similar surface charge densities as a single segment with equivalent area. As a side effect of this procedure, frequently there are segments with zero area (surface charge densities not observed in the molecule).

When the maximum number of segments is fixed at 51 segments per molecule within the limit of ± 0.025 e/Å², segments differing by less than $\Delta\sigma^{51} = 0.001$ e/Å² are not distinguished. If the maximum number of segments per molecule is reduced to only 11 segments, $\Delta\sigma^{11} = 0.005$ e/Å². Typically this information is omitted and a continuous histogram of area vs. apparent surface charge density σ is presented, as shown in Figure 3-(a). These histograms are known as σ -profiles.

However, the representation of σ -profiles in COSMO-type codes is actually discrete, as shown in

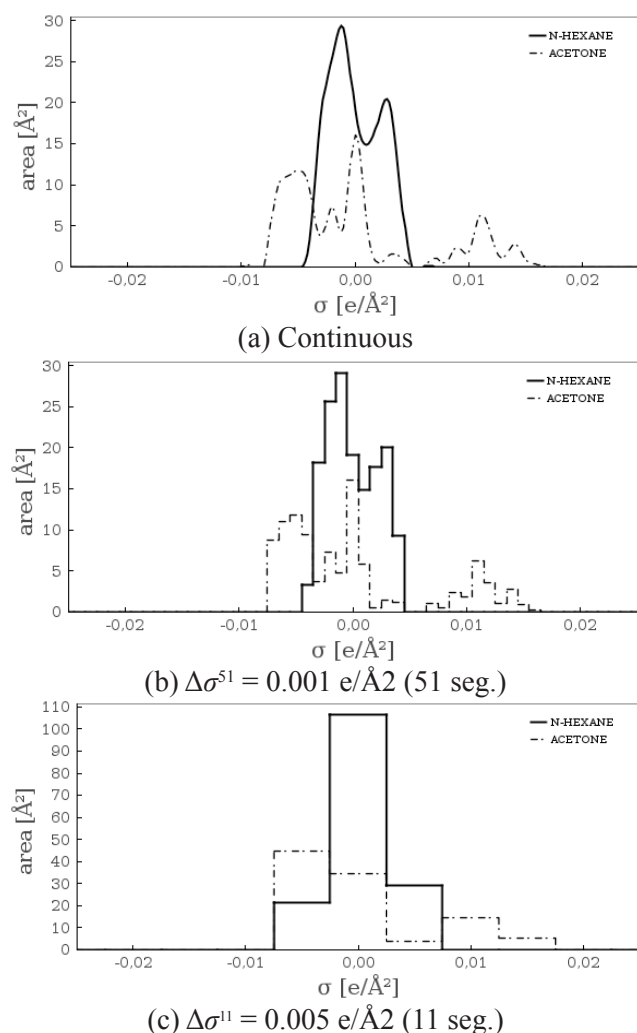


Figure 3. Typical continuous (smoothed) σ -profile (a); actual representation with $\Delta\sigma^{51}=0.001 e/\text{\AA}^2$ (b); and $\Delta\sigma^{11}=0.005 e/\text{\AA}^2$ (c).

Figure 3(b) and 3(c). As more segments per molecule are considered, less information from the COSMO computation is lost, but the CPU cost and memory usage for solving the model are increased.

In the COSMO-SAC and F-SAC implementations of our group, within the JCOSMO package (Gerber and Soares, 2010), the merging of segments with similar surface charge densities is optional and $\Delta\sigma$ can be configured. Further, segments with zero area are not taken into account. Thus, different molecules can end up with different numbers of segments. This makes the implementation more complex, but reduces the computational cost and memory usage.

In the works of Klamt (1995); Klamt et al. (1998); Lin and Sandler (2002), only a few of the computational issues regarding the methods for solving the self-consistency equation are explored. According to Klamt (1995) and; Klamt et al. (1998), the self-consistency equation must be solved by numerical iteration. The authors report the use of a successive substitution method.

Lin and Sandler (2002) also give no details on the numerical procedures for solving the self-consistency equation. One may suppose a successive substitution is used to solve Equation 1. Moreover, the COSMO-SAC FORTRAN code of Virginia Tech is implemented with a simple successive substitution method as well.

In the recent F-SAC model (Soares and Gerber, 2013; Soares et al., 2013), the numerical issues in solving the self-consistency are almost not explored as well. The F-SAC demonstration FORTRAN code, freely available at <https://github.com/lvpp> also implements the same successive substitution as in the code of Virginia Tech.

Therefore, in the present work the computational and numerical characteristics of COSMO-type models are further explored. The performance of a classical Newton-Raphson implementation (Press et al., 1992) is tested in the solution of Equation 1. Numerical results and computational costs of the Newton method and of the classical successive substitution are compared for different mixtures.

Further, one particular problem with the freely available COSMO-based codes is the difficulty to map from the formulations available in the literature to the actual implementations. Thus, in the appendix a more complete description of the COSMO-SAC (Lin and Sandler, 2002) and F-SAC (Soares and Gerber, 2013; Soares et al., 2013) formulations is given with equations and variables organized as in the actual freely available codes of Virginia Tech (http://www.design.che.vt.edu/VT_Databases.html) and the F-SAC demonstration FORTRAN code (<https://github.com/lvpp>) and as in the implementations of our group, within the JCOSMO package (Gerber and Soares, 2010).

SOLUTION OF THE SELF-CONSISTENCY EQUATION

Successive Substitution

In the work of Klamt (1995) a successive substitution implementation is used to solve the self-consistency equation, Equation 1. According to the author, the method starts with an unitary initial guess for all segments and updates it iteratively. To avoid numerical oscillations, the activity coefficient values are updated by the average of the previous and the new values. This procedure can slow down the convergence, but still leads to convergence within milliseconds in regular personal computers.

In the COSMO-SAC and F-SAC implementations of our group, a similar methodology is used. In order to apply a successive substitution method in Equation 1, one can eliminate the logarithm on both sides, as follows:

$$\Gamma_m = \left[\sum_n p_n \Gamma_n e^{\frac{-\Delta W_{m,n}}{RT}} \right]^{-1} \quad (2)$$

By doing this, the activity coefficient of one segment Γ_m can be updated as a function of the other

n segments. Figure 4 shows a simplified fluxogram of the algorithm, while a pseudo-code form is given in Algorithm 1.

As described in the fluxogram in Figure 4, the algorithm starts by setting the activity coefficient of all segments as 1.0 (initial guess). The initial guess was chosen as 1.0 for all segments because it corresponds

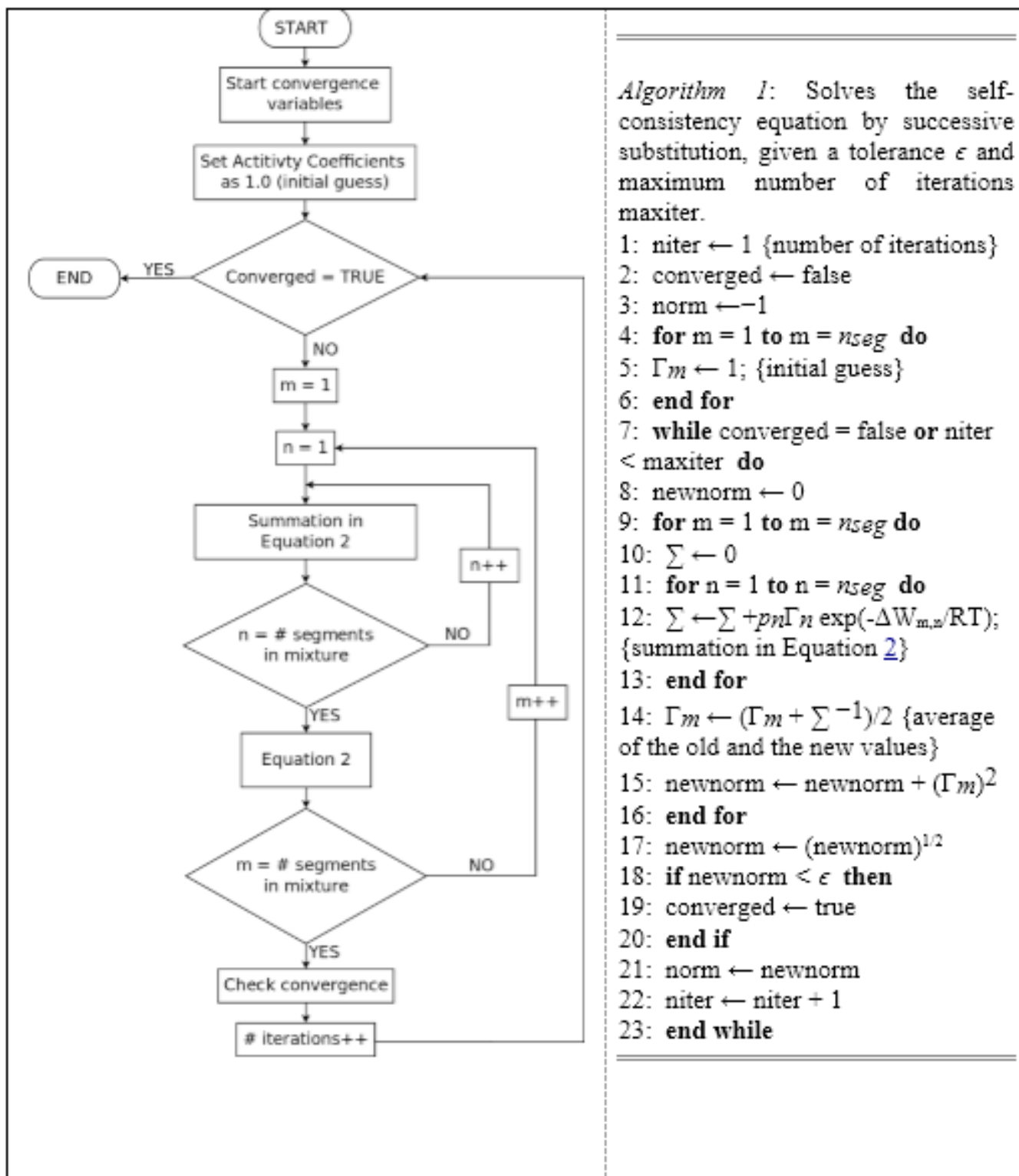


Figure 4. Successive substitution method fluxogram.

to the ideal solution case. Then, for each segment a new value of Γ_m is calculated using Equation 2. These updated values are always reused, and the method continues until the difference between the new and the current values of the Euclidian norm of the segments vector is smaller than a given tolerance.

From the Algorithm 1, the number of floating point operations (FLOP) for each iteration can be roughly estimated by simply counting the number of operations. It is noteworthy that, in our implementation, the matrix with elements given by $\exp(-\Delta W_{m,n}/RT)$ is computed only once for a given temperature and then stored. Thus, for each element of the sum in Equation 2 (inside the n-loop), only 2 multiplications and 1 addition are needed. This results in approximately $3 \cdot nseg$ FLOP per segment, where $nseg$ is the total number of segments in mixture.

For the m-loop, the number of FLOP per segment resulting from the n-loop must be first summed with 5 additional operations: 3 accounting for line 14 (the -1 exponent, the sum and the division by 2); and 2 for line 15 (the 2 exponent and the sum). Then, it should be multiplied by the number of segments, resulting in $3 \cdot nseg^2 + 5 \cdot nseg$ FLOP per segment.

For the total number of operations, the latter number of FLOP per segment needs to be summed with 4 more operations, due to the expressions in lines 17, 18 and 23. This results in an upper limit of $3 \cdot nseg^2 + 5 \cdot nseg + 4$ total FLOP per iteration.

Newton-Raphson

As noted in the previous sections, successive substitution is reported as the method of choice for most implementations when solving the self-consistency equation given by Equation 1. Naturally, it could be solved by another numerical method, for instance, Newton's method.

In the present paper, a classical Newton method is proposed as an alternative for solving the self-consistency equation. The Newton method, also called Newton-Raphson, is widely known and used for engineering purposes. For continuous functions, with continuous derivatives, and when given a good initial guess, usually this method converges very rapidly (Press et al., 1992).

The Newton method for a generalized vectorial system $F(x) = 0$ can be given by:

$$\bar{x}^{(k+1)} = \bar{x}^{(k)} - \frac{F(\bar{x}^{(k)})}{J(\bar{x}^{(k)})} \quad (3)$$

where k is the actual iteration step and $J(x^{(k)})$ is the Jacobian matrix defined by:

$$J(\bar{x}) = \begin{bmatrix} \frac{\partial F_1}{\partial x_1} & \dots & \frac{\partial F_1}{\partial x_M} \\ \vdots & \ddots & \vdots \\ \frac{\partial F_M}{\partial x_1} & \dots & \frac{\partial F_M}{\partial x_M} \end{bmatrix} \quad (4)$$

In order to apply Newton's method in Equation 1, one can eliminate the \ln on both sides and put it in a residual form, as follows:

$$F_m = \Gamma_m \left[\sum_n p_n \Gamma_n e^{\frac{-\Delta W_{m,n}}{RT}} \right] - 1 = 0 \quad (5)$$

For each element of the Jacobian matrix, Equation 5 needs to be differentiated with respect to the activity coefficient of a given segment Γ_j :

$$\frac{\partial F_m}{\partial \Gamma_j} = \Gamma_m A_{m,j} + \frac{\partial \Gamma_m}{\partial \Gamma_j} \sum_n A_{m,n} \Gamma_n \quad (6)$$

where $A_{m,n} = p_n \exp(-\Delta W_{m,n}/RT)$; and the partial derivative $\partial \Gamma_m / \partial \Gamma_j = 1$ when $m = j$ and $\partial \Gamma_m / \partial \Gamma_j = 0$ otherwise.

The Newton-Raphson method, as described by Equation 3 may, therefore, be applied to solve the vectorial system given by Equation 5 by using the Jacobian matrix with elements as in Equation 6. It is noteworthy that this Jacobian matrix will be dense, with all elements as potential non-zero elements.

In this paper, the classical Newton algorithm as given by the Numerical Recipes (Press et al., 1992) was implemented within the computer program JCOSMO described in Gerber and Soares (2010). Figure 5 shows the proposed algorithm fluxogram. The pseudo-code for this implementation is shown in Algorithm 2.

Basically, the method works as follows: first, the activity coefficient of all segments are set as 1.0 (initial guesses). Again, the initial guess was chosen as 1.0 for all segments because it corresponds to the ideal solution case. Then, the function (Equation 5) and the Jacobian (Equation 6) are calculated. Finally, the new activity coefficients for all segments are calculated in the matrix operation given by Equation 3. The method continues until the difference between the new and the current values of the Euclidian norm of the segments vector is smaller than a given tolerance.

Again, the number of FLOP for each iteration can be estimated by counting the number of operations necessary in the method given in Algorithm 2. For the n-loop, 2 multiplications (line 12) and 1 addition (line 13) are needed. This results in $3 \cdot nseg$ operations per segment.

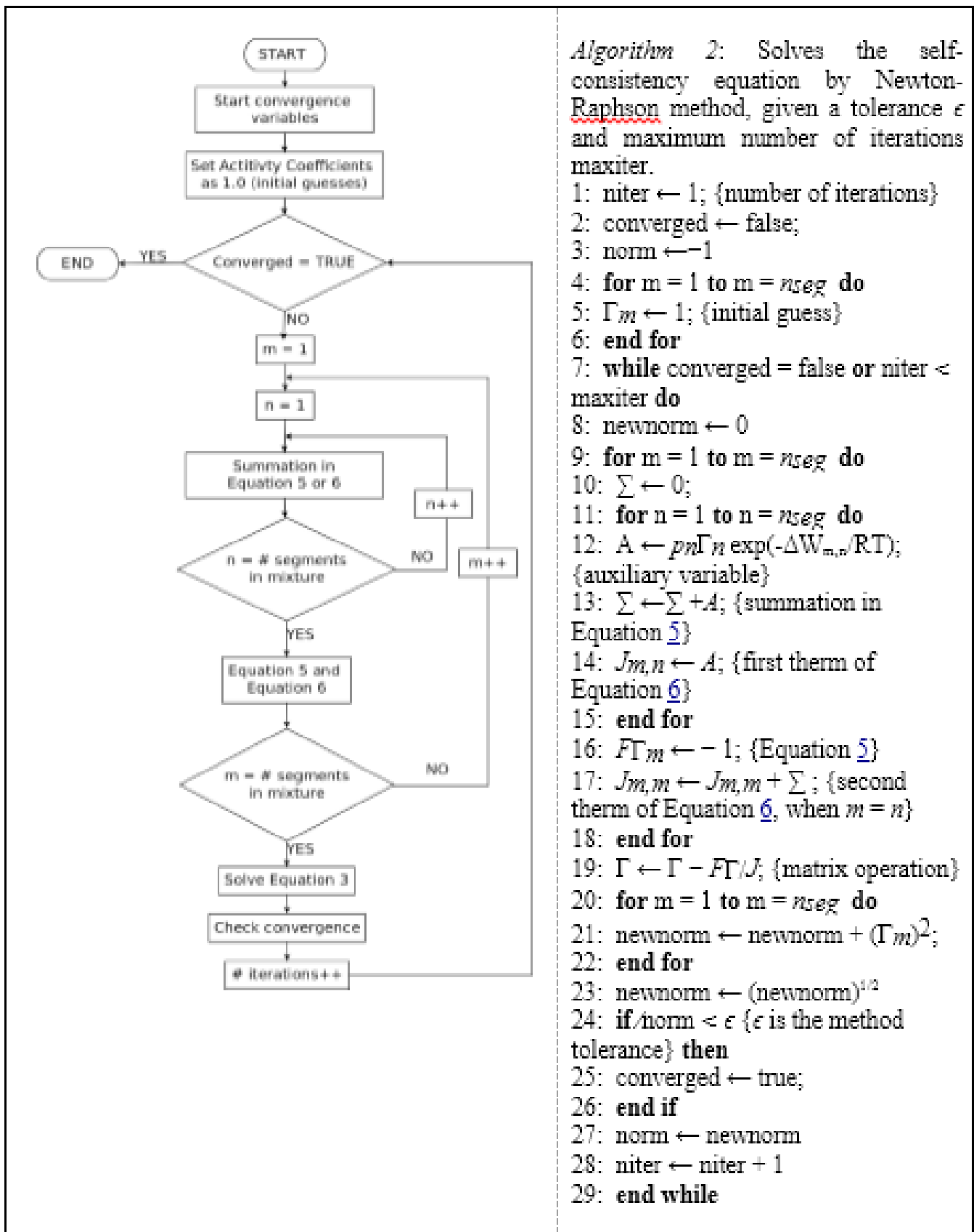


Figure 5. Newton method fluxogram implemented in this work.

For the m-loop, the $3 \cdot n_{seg}$ operations (n-loop) must be multiplied by n_{seg} and summed with the 3 additional operations: 1 multiplication and 1 subtraction in line 16; and 1 summation in line 17. This results in $3 \cdot n_{seg}^2 + 3 \cdot n_{seg}$ operations per iteration. The latter number must be summed with the FLOP necessary for solving the linear expression in Equation 3 (line 19), by LU decomposition and solution. In the Numerical Recipes (Press et al., 1992), the author estimates a total of $(2/3) \cdot n_{seg}^3$ floating point operations for the linear system solution (by LU), resulting in $(2/3) \cdot n_{seg}^3 + 3 \cdot n_{seg}^2 + 3 \cdot n_{seg}$ FLOP per iteration.

Finally, the number obtained before is summed with $2 \cdot n_{seg} + 4$ (convergence check, lines 21 – 28). This results in an overall FLOP of $(2/3) \cdot n_{seg}^3 + 3 \cdot n_{seg}^2 + 5 \cdot n_{seg} + 4$ per iteration.

RESULTS AND DISCUSSION

In this section it was verified whether the Newton and the successive substitution methods lead to the same results, within a given tolerance. Furthermore, the number of iterations and computational times necessary for converging with each method are compared. For both numerical methods, the activity coefficient of all segments started with the value of 1.0 (initial guesses). Since the objective is to compare the methods for solving the self-consistency equation, no combinatorial contribution was considered in the tests.

Table 1 shows the results for a mixture of acetone + chloroform, at $T = 298$ K, with mole fraction of acetone equal to 0.519. In the COSMO-SAC model implementation used in this work (Gerber and Soares, 2010), the molecules have 25 (acetone) and 20 (chloroform) segments. In the F-SAC formulation of Soares and Gerber (2013); Soares et al. (2013), these molecules have only 5 segments each. Activity coefficient calculations were also performed for several other systems, with different temperatures and compositions, but are not shown here.

It can be observed in Table 1 that the calculated activity coefficients for the tested system, using both successive substitution and Newton-Raphson, for each considered model, agreed with each other within a given tolerance of at least 10^{-7} . It is noteworthy the poor performance of COSMO-SAC when compared to the experimental data of Mueller and Kearns (1958). F-SAC, on the other

hand, performed better, mainly because it was previously calibrated (Soares et al., 2013) with experimental data for the considered mixture (acetone/chloroform). However, evaluating the performances of the COSMO-SAC and F-SAC models against experimental data is out of scope of the present work.

Similar to the results reported in Table 1, both methods (successive substitution and Newton-Raphson) produced essentially the same numerical results for all tests presented in this section. Nevertheless, no attempt was made to prove either the existence or the unicity of the solution of Equation 1 in this work. As expected, the Newton method required fewer iterations for convergence in all tests. However, contradicting the usual experience, higher computational times (not shown) were observed for the Newton method.

Further tests for other systems with different numbers of components (and then different number of segments) revealed that the Newton method becomes slower (in terms of CPU time) than successive substitution for systems with more than a certain number of segments. For all tests reported in this work, only the solution of Equation 1 for the mixture is taken into account, since the solution of Equation 1 for pure compounds should produce analogous responses (with respect to the number of segments). Such results can be observed in Figure 6 when using the COSMO-SAC

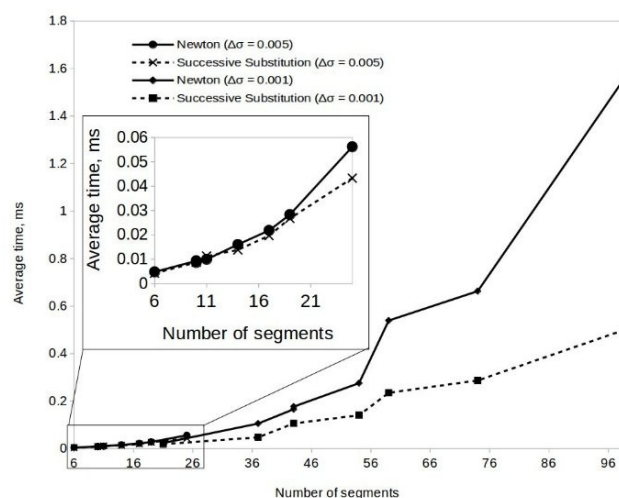


Figure 6. Successive substitution and Newton method average computational times for solving the “self-consistency” equation against the mixture number of segments.

Table 1. Newton and successive substitution (S. Subst.) results using COSMO-SAC and F-SAC for acetone (1)/chloroform (2), at $T = 298.15$ K, with mole fraction of acetone equal to 0.519, $\Delta\sigma^{S1} = 0.001$ e/Å² and a tolerance of $\epsilon = 10^{-8}$. Experimental data (Mueller and Kearns, 1958): $\ln\gamma_1 = -0.1820112$ and $\ln\gamma_2 = -0.2966690$.

	COSMO-SAC		F-SAC	
	Newton	S. Subst.	Newton	S. Subst.
$\ln\gamma_1$	0.0082563234	0.0082567143	-0.2025500789	-0.2025501341
$\ln\gamma_2$	0.0083324743	0.0083324329	-0.3083406786	-0.308340658
# iterations	4	23	5	32

implemented in the JCOSMO program (Gerber and Soares, 2010).

All the calculations were performed on a 4GB RAM Intel core i5 personal computer, running Ubuntu 14.04 LTS. The average computational times for solving the self-consistency equation by the successive substitution and Newton methods in Figure 6 were computed for the following equimolar mixtures at $T = 298$ K: cyclohexane + n-octane; cyclohexane + n-octane + benzene; cyclohexane + n-octane + benzene + toluene; cyclohexane + n-octane + benzene + toluene + chloroform; cyclohexane + n-octane + benzene + toluene + chloroform + acetone; chloroform + diethyl ether; cyclohexane + water; and benzene + toluene + water.

This set of mixtures covers a broad variety of deviations, ranging from negative deviations, nearly ideal mixtures ($\gamma_i \approx 1$), up to highly positive deviations (hydrocarbon/water). In Table 2 the numbers of COSMO segments considered for each molecule are listed for different values of $\Delta\sigma$.

As can be observed in Table 2, the simple cyclohexane + n-octane mixture presents 21 segments already, when using $\Delta\sigma^{51}$. However, as mentioned in the Introduction, the COSMO-SAC implementation of our group actually allows the maximum number of segments per molecule to be configured. Therefore, in order to create mixtures with less than 21 segments, the mixtures were also computed with $\Delta\sigma^{11}$ and the results are shown in Figure 6, in the range from 6 to 25 segments.

The results in Figure 6 for equimolar mixtures contradict the usual experience that the Newton method is faster than successive substitution. Actually, it is widely known that the rates of convergence are linear and quadratic for Successive Substitution and Newton methods, respectively (Press et al., 1992).

In fact, only a few (3 to 5) iterations were required by the Newton method in all tests, while more than 20 iterations, on average, were observed for successive substitution, indicating a higher convergence rate for Newton. The difference is in the *computational cost* of each iteration.

As described in the previous section, solving Equation 1 by successive substitution takes $3 \cdot nseg^2$

Table 2. Number of segments for each molecule used in the numerical comparison shown in Figure 6.

Molecule	$\Delta\sigma^{11} = 0.005e/\text{\AA}^2$	$\Delta\sigma^{51} = 0.001e/\text{\AA}^2$
Cyclohexane	3	10
N-octane	3	11
Benzene	4	16
Toluene	4	17
Chloroform	5	20
Acetone	6	25
Diethyl ether	5	23
Water	8	33

+ $5 \cdot nseg + 4$ FLOP per iteration, where $nseg$ is the mixture number of segments. Considering an average of 20 iterations for solving each system, successive substitution requires about $20 \cdot (3 \cdot nseg^2 + 5 \cdot nseg + 4)$ FLOP. On the other hand, considering an average of only 4 iterations for the Newton method, it takes about $4 \cdot [(2/3) \cdot nseg^3 + 3 \cdot nseg^2 + 5 \cdot nseg + 4]$ FLOP. Figure 7 shows these expected FLOP results.

It is observed that the results are qualitatively consistent with the numerical outcomes obtained in Figure 6. Even though the FLOP counting is only an approximation of the computational effort, it helps to understand the observed behavior. With this analysis, one can conclude that the Newton method, particularly for COSMO-based models, will usually be slower than a simple successive substitution method. Further, the substitution method will be even more efficient as the number of components (or segments) increases.

In order to investigate situations with even higher values of γ_i , similar tests were also performed for infinitely dilute systems (equimolar mixtures were considered in previous tests). The tested systems were cyclohexane, acetone, chloroform, cyclohexane + n-octane, cyclohexane + toluene, cyclohexane + n-octane + toluene, cyclohexane + n-octane + toluene + benzene, cyclohexane + n-octane + toluene + benzene + chloroform, cyclohexane + n-octane + toluene + benzene + chloroform + acetone, all infinitely dilute in water, at $T = 298$ K, with both $\Delta\sigma^{11}$ and $\Delta\sigma^{51}$.

These infinitely dilute mixtures produce a highly non-ideal response, with the natural logarithm of the activity coefficients ranging from 2 to 9. The computation times for these cases are shown in Figure 8.

As can be observed in Figure 8, the main difference from Figure 6 is *where* the computation time curves cross. It is expected that highly non-ideal systems

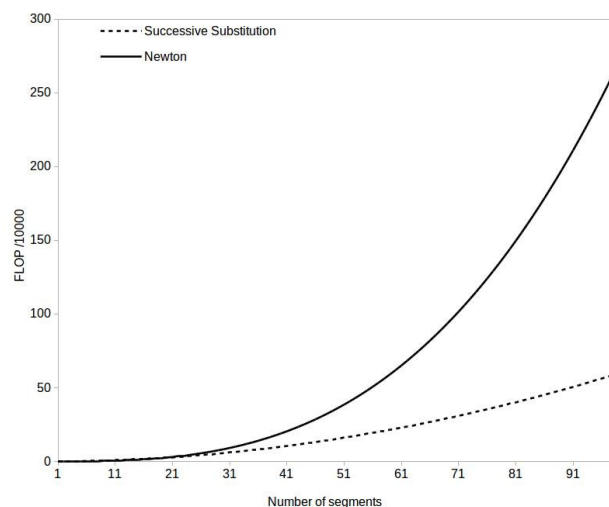


Figure 7. Number of FLOP required for solving the “self-consistency” equation *versus* the mixture number of segments.

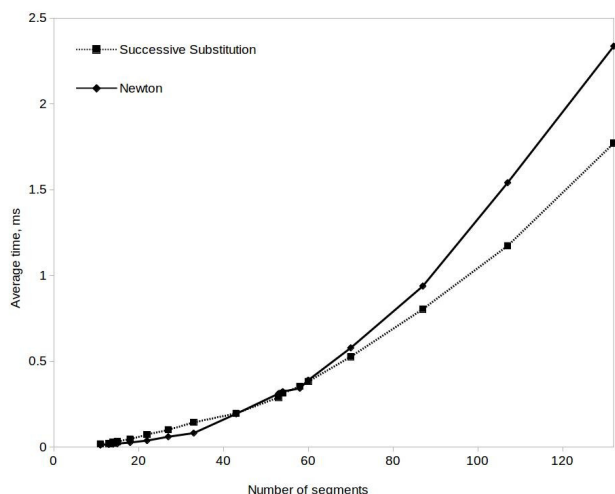


Figure 8. Successive substitution and Newton method average computational times for solving the “self-consistency” equation against the mixture number of segments for infinitely dilute systems in water.

would require more iterations to converge. In fact, while about 20 iterations were required to solve the problems in Figure 6 (equimolar mixtures), about 45 iterations were necessary in Figure 8 (infinitely dilute) for successive substitution. On the other hand, only about 5 iterations were required for converging all systems in Figure 8 when using the Newton’s method.

Further, as can be observed in the Formulation section, the FLOP numbers are directly dependent on the number of iterations required for each method. Therefore, one concludes that as the non-idealities become higher, the number of segments where the methods become equivalent will increase. However, the conclusion that the Newton method, particularly for COSMO-based models, will usually be slower than a simple successive substitution method as the number of components (or segments) increases is maintained and confirmed for highly non-ideal systems.

Such results can also be observed for the theoretical FLOP results, shown in Figure 9. These results use the same basis considered in Figure 7, but with the number of iterations updated to 45 and 5 for the successive substitution and the Newton method, respectively.

We have also investigated if different temperatures would lead to different computation times. Thus, simulations for higher temperatures, 373 K and 423 K, were also performed. The obtained results were essentially the same (not shown for the sake of conciseness).

One possible theoretical explanation for the observed anomalous behavior of the successive substitution and Newton methods when solving COSMO-based models is the particular form of Equations 2, 5 and 6. As can be observed in the Jacobian matrix, the second term on the right hand side of Equation 6 will only be different from zero in its diagonal. Nonetheless, this

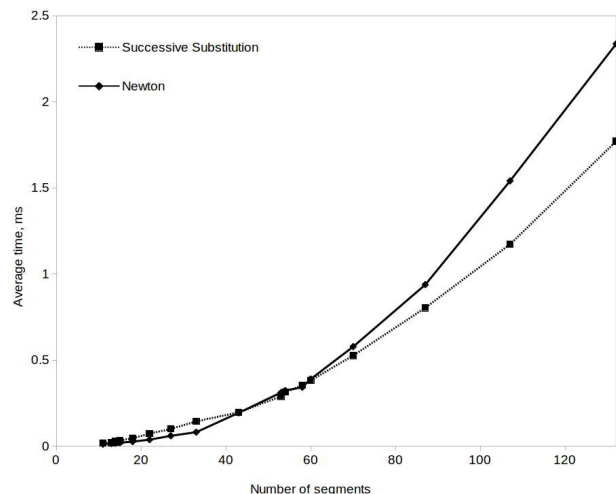


Figure 9. Number of FLOP required for solving the “self-consistency” equation versus the mixture number of segments for infinitely dilute systems in water.

matrix will always be dense. However, assuming that the summation $\sum_n A_{m,n} \cdot \Gamma_n \gg \Gamma_m \cdot A_{m,j}$ then the Jacobian matrix has diagonal elements much larger than the off-diagonal terms. Under these assumptions, one could simply approximate the Jacobian matrix by a diagonal matrix with elements given by:

$$\frac{\partial F_m}{\partial \Gamma_m} \approx \sum_n A_{m,n} \Gamma_n = \sum_n p_n \Gamma_n e^{\frac{-\Delta W_{m,n}}{RT}} \quad (7)$$

Thus, the inverse of this approximate diagonal Jacobian matrix is just the reciprocal of the elements on the diagonal:

$$\frac{\partial F_m}{\partial \Gamma_m}^{-1} \approx \left[\sum_n p_n \Gamma_n e^{\frac{-\Delta W_{m,n}}{RT}} \right]^{-1} \quad (8)$$

When applying Equation 8 in Equation 3 one gets:

$$\Gamma_m^{(k+1)} \approx \Gamma_m^{(k)} - \frac{\Gamma_m^{(k)} \left(\sum_n p_n \Gamma_n e^{\frac{-\Delta W_{m,n}}{RT}} \right)^{(k)} - 1}{\left(\sum_n p_n \Gamma_n e^{\frac{-\Delta W_{m,n}}{RT}} \right)^{(k)}} \quad (9)$$

which simplifies to:

$$\Gamma_m^{(k+1)} \approx \frac{1}{\left(\sum_n p_n \Gamma_n e^{\frac{-\Delta W_{m,n}}{RT}} \right)^{(k)}} \quad (10)$$

being essentially the equation used in the successive substitution method described in the section Successive Substitution.

Therefore, in this particular problem, the successive substitution method actually works as a *quasi*-Newton method with the Jacobian replaced by a diagonal matrix approximation. Then, the method has improved convergence rate properties without the cost of solving a dense linear system per iteration.

Features like symmetry can be exploited in the Newton-Raphson method. However, for our particular problem, unless it is a fortuitous case, the Jacobian matrix will never be symmetrical, given that pn and Γ_n are not symmetric vectors (see Equation 6). Also, because of the summation in the self-consistency equation (see Equation 5), one segment activity coefficient within the mixture depends on the other segments, therefore, the resulting Jacobian matrix will always be dense. A drop tolerance for elements in the Jacobian matrix could also be used; the effort per iteration would be reduced but the number of iterations will probably be higher. This was left for future work.

CONCLUSIONS

In this work, a Newton-Raphson method was tested in the solution of the so-called *self-consistency* equation of the residual contribution of COSMO-based models. The implementation was checked by comparing the activity coefficient results for several mixtures, using the Newton and the previously implemented successive substitution methods. The results agreed within a tolerance of 10^{-7} for the activity coefficients when a convergence tolerance of 10^{-8} was set.

As expected, the Newton-Raphson method showed a higher convergence rate for all our tests. However, it was observed that this method becomes slower (in terms of CPU time) than successive substitution when the number of components (or segments) in the mixture increases. This apparent contradiction is also verified by an analysis of the required floating point operations (FLOP). The Newton method FLOP cost depends on the cubic power of the number of segments, while the substitution method cost grows with the square of the number of segments. On the one hand the substitution method requires more iterations than the Newton method, but on the other hand this is not sufficient to counter-balance the extra effort per iteration.

ACKNOWLEDGEMENT

This work was partially supported by CNPq-Brazil under grants no. 454557/2014-0 and 304046/2016-7.

REFERENCES

Flôres, G. B., Staudt, P. B., and Soares, R. de P., Including dispersive interactions in the f-sac

- model. *Fluid Phase Equilib.* 426, 56–64 (2016). <https://doi.org/10.1016/j.fluid.2016.02.043>
- Fredenslund, A., Jones, R. L., and Prausnitz, J. M., Group-contribution estimation of activity coefficients in nonideal liquid mixtures. *AIChE Journal* 21 (6), 1086–1099 (1975). <https://doi.org/10.1002/aic.690210607>
- Gerber, R. P., and Soares, R. de P., Prediction of Infinite-Dilution Activity Coefficients Using UNIFAC and COSMO-SAC Variants. *Ind. & Eng. Chem. Res.* 49 (16), 7488–7496 (2010). <https://doi.org/10.1021/ie901947m>
- Gerber, R. P., and Soares, R. de P., Assessing the reliability of predictive activity coefficient models for molecules consisting of several functional groups. *Braz. J. Chem. Eng.* 30 (1), 1–11 (2013). <https://doi.org/10.1590/S0104-66322013000100002>
- Grensemann, H., and Gmehling, J., Performance of a Conductor-Like Screening Model for Real Solvents Model in Comparison to Classical Group Contribution Methods. *Ind. & Eng. Chem. Res.* 44 (5), 1610–1624 (2005). <https://doi.org/10.1021/ie049139z>
- Jakob, A., Grensemann, H., Lohmann, J., and Gmehling, J., Further Development of Modified UNIFAC (Dortmund): Revision and Extension 5. *Industrial & Engineering Chemistry Research* 45 (23), 7924–7933 (2006). <https://doi.org/10.1021/ie060355c>
- Klamt, A., Conductor-like Screening Model for Real Solvents: A New Approach to the Quantitative Calculation of Solvation Phenomena. *J. Phys. Chem.* 99 (7), 2224–2235 (1995). <https://doi.org/10.1021/j100007a062>
- Klamt, A., Jonas, V., Burger, T., and Lohrenz, J. C. W., Refinement and Parametrization of COSMO-RS. *J. Phys. Chem. A* 102 (26), 5074–5085 (1998). <https://doi.org/10.1021/jp980017s>
- Klamt, A., and Schuurmann, G., COSMO: a new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. *J. Chem. Soc., Perkin Trans. 2* (5), 799–805 (1993). <https://doi.org/10.1039/P29930000799>
- Lin, S.-T., and Sandler, S. I., A Priori Phase Equilibrium Prediction from a Segment Contribution Solvation Model. *Ind. & Eng. Chem. Res.* 41 (5), 899–913 (2002). <https://doi.org/10.1021/ie001047w>
- LVPP, 2015. F-SAC FORTRAN-2013. URL <http://www.enq.ufrgs.br/labs/lvpp/>.
- Mueller, C., and Kearns, E., Thermodynamic Studies of the System: Acetone and Chloroform. *J. Phys. Chem.* 62 (11), 1441–1445 (1958). <https://doi.org/10.1021/j150569a023>
- Possani, L. F. K., Flôres, G. B., Staudt, P. B., and Soares, R. de P., Simultaneous Correlation of Infinite Dilution Activity Coefficient, Vapor-Liquid, and Liquid-Liquid equilibrium data with F-SAC. *Fluid Phase Equilib.* 364, 31–41 (2014a). <https://doi.org/10.1016/j.fluid.2013.11.040>

Possani, L. F. K., Simões, R. L., Staudt, P. B., and Soares, R. de P., Mutual Solubilities of Hydrocarbon-Water Systems with F-SAC. *Fluid Phase Equilib.* 384, 121–133 (2014b).

Press, W. H., Teukolsky, S. A., Vetterling, W. T., and Flannery, B. P., *Numerical Recipes in C: The Art of Scientific Computing*. Press Syndicate of the University of Cambridge, 1992.

Soares, R. de P., and Gerber, R. P., Functional-Segment Activity Coefficient Model. 1. Model Formulation. *Ind. & Eng. Chem. Res.* 52 (32), 11159–11171 (2013). <https://doi.org/10.1021/ie400170a>

Soares, R. de P., Gerber, R. P., Possani, L. F. K., and Staudt, P. B., Functional-Segment Activity Coefficient Model. 2. Associating Mixtures. *Ind. & Eng. Chem. Res.* 52 (32), 11172–11181 (2013). <https://doi.org/10.1021/ie4013979>

Virginia Tech, 2015. COSMO-SAC-VT-2005. URL <http://www.design.che.vt.edu/VT-Databases.html>.

APPENDIX

COSMO-SAC formulation

In the COSMO-SAC model (Lin and Sandler, 2002), the activity coefficient for the liquid phase γ_i can be calculated as:

$$\ln \gamma_i = \ln \gamma_i^{\text{comb}} + \ln \gamma_i^{\text{res}} \quad (11)$$

where $\ln \gamma_i^{\text{comb}}$ and $\ln \gamma_i^{\text{res}}$ are, respectively, the combinatorial and the residual contributions.

The combinatorial contribution is given by the Staverman-Guggenheim formula, as described by Lin and Sandler (2002). While the combinatorial contribution is well known, the residual contribution is more complex. Here we provide a formulation particularly useful when trying to understand the freely available FORTRAN codes of Virginia Tech (2015) and LVPP (2015) and the implementations of our group, within the JCOSMO package (Gerber and Soares, 2010).

The residual contribution is calculated as the difference between the free energy to restore the charge around the solute molecule in solution, s , and to restore the charge in a pure liquid, i :

$$\ln \gamma_i^{\text{res}} = \sum_{m \in i} \frac{Q_m^i}{a_{\text{eff}}} (\ln \Gamma_m^s - \ln \Gamma_m^i) \quad (12)$$

where a_{eff} is the standard segment surface area, which is the same for all molecules and is one of the universal parameters in this model; $\ln \Gamma_m^s$ is the logarithm of the activity coefficient of a segment m in solution and $\ln \Gamma_m^i$ in pure liquid, given by the “self-consistency” equation:

$$\ln \Gamma_m^s = -\ln \left[\sum_{n \in s} p_n^s \Gamma_n^s e^{\frac{-\Delta W_{m,n}}{RT}} \right] \quad (13a)$$

$$\ln \Gamma_m^i = -\ln \left[\sum_{n \in i} p_n^i \Gamma_n^i e^{\frac{-\Delta W_{m,n}}{RT}} \right] \quad (13b)$$

The probability of finding a segment m in a mixture s (σ -profile of mixture) is:

$$p_n^s = \frac{x_i Q_m^i}{\sum_j x_j Q_j^i} \quad (14)$$

And the probability of finding a segment m in a pure liquid i (σ -profile of the pure compound) is:

$$p_m^i = \frac{Q_m^i}{Q^i} \quad (15)$$

where Q_m^i is the surface area of the segment m of molecule i ; and $Q^i = \sum_{m \in i} Q_m^i$ is the total cavity surface area of molecule i .

Finally, the interaction energy for each contact between segments m and n can be computed under different assumptions. Using the formulation of Lin and Sandler (2002), it is computed as a function of the segment charges σ_m and σ_n :

$$\Delta W_{m,n} = \left(\frac{\alpha'}{2} \right) (\sigma_m - \sigma_n)^2 + c_{\text{hb}} \max[0, \sigma_{\text{acc}} - \sigma_{\text{hb}}] \times \min[0, \sigma_{\text{don}} + \sigma_{\text{hb}}] \quad (16)$$

$$\alpha' = f_{\text{pol}} \left(\frac{0.3 a_{\text{eff}}^{3/2}}{\epsilon_0} \right) \quad (17)$$

where α' is the constant for the misfit energy; $\epsilon_0 = 2.395 \times 10^{-4}$ (e² mol)/(kcal Å) is the permittivity of vacuum; f_{pol} is the polarization factor; c_{hb} is a constant for hydrogen bonding (HB); σ_{hb} is the sigma-value cutoff for hydrogen bonding; σ_{acc} and σ_{don} are the larger and smaller values of σ_m and σ_n .

As stated before, the actual computation of activity coefficients is accomplished by solving Equation 13a one time for each pure compound and Equation 13b for the mixture.

F-SAC formulation

The F-SAC (Soares and Gerber, 2013; Soares et al., 2013) formulation is actually almost identical to the formulation present in the COSMO-SAC model.

The differences lie on the σ -profiles and the interaction energy, as described below.

In F-SAC, instead of using COSMO calculations to obtain the σ -profiles for the molecules, it is proposed that each molecule is subdivided into *functional groups* (group contribution theory) and each functional group has its own empirically calibrated σ -profile:

$$\{(\sigma_k^-, Q_k^-); (0, Q_k^0); (\sigma_k^+, Q_k^+)\} \quad (18)$$

where the σ -profile of each functional group is represented by three empirical parameters: Qk^+ , Qk^- and σk^+ . Qk^+ represents the functional group area of the positive segment; Qk^- is functional group area of the negative segment; and σk^+ is the charge density of the positive segment. With these definitions, the neutral area Qk^0 is given by the remaining area of the group surface area: $Qk^0 = Qk - Qk^+ - Qk^-$; and, by a charge balance to keep each group neutral, the group negative charge density can be computed as $\sigma k^- = -\sigma k^+ Qk^+ / Qk^-$.

The interaction energy for each contact between segments m and n was modified in Possani et al. (2014a,b) and is given by:

$$\Delta W_{m,n} = \theta_{m,n} \frac{\alpha'(\sigma_m + \sigma_n)^2}{2} - \frac{E_{m,n}^{\text{HB}}}{2} \quad (19a)$$

$$\theta_{m,n} = e^{-\beta_{m,n}(T-T_0)} \quad (19b)$$

$$E_{m,n}^{\text{HB}} = \theta_{m,n}^{\text{HB}} E_o^{\text{HB}} \quad (19c)$$

where T_0 is a reference temperature, taken as 323.15 K; $\beta_{m,n}$ is given by the following combining rule for each contact between the segments m, n :

$$\beta_{m,n} = \frac{\beta_m + \beta_n}{2} \quad (20)$$

with β_m, β_n being the electrostatic temperature dependence parameters for the group segments m, n , respectively; and E^{HB} accounts for hydrogen bond (HB) formation with its temperature dependence given by the following expression :

$$\theta_{m,n}^{\text{HB}} = e^{-\beta_{m,n}^{\text{HB}}(T-T_0)} \quad (21)$$

where the pairwise parameter $\beta_{m,n}^{\text{HB}}$ should be adjusted with temperature dependent experimental data for each HB acceptor-donor pair and E_o^{HB} is the HB formation energy when $T = T_0$.

Finally, extensions are easily implemented by improving the calculation of the interaction energy $\Delta W_{m,n}$. For instance, in the work of Flôres et al. (2016) a new contribution for dispersion forces was introduced.