Received: 22 February 2024 | Revised: 11 April 2024 | Accepted: 12 April 2024

DOI: 10.1002/jcc.27389

SOFTWARE NOTE

Journal of COMPUTATIONAL CHEMISTRY WILEY

POMSimulator: An open-source tool for predicting the aqueous speciation and self–assembly mechanisms of polyoxometalates

Enric Petrus[1] | Jordi Buils[2,3] | Diego Garay-Ruiz[2] | Mireia Segado-Centellas[2,3] | Carles Bo[2,3]

[1]Department of Environmental Chemistry, EAWAG: Swiss Federal Institute of Aquatic Science and Technology, Dübendorf, Switzerland

[2]Institute of Chemical Research of Catalonia (ICIQ), Tarragona, Spain

[3]Departament de Química Física i Inorgànica, Universitat Rovira i Virgili, Tarragona, Spain

Correspondence
Carles Bo, Institute of Chemical Research of Catalonia (ICIQ), Av. Països Catalans 16, Tarragona, 43007, Spain.
Email: cbo@iciq.cat

Funding information
Centres de Recerca de Catalunya; Ministerio de Ciencia e Innovación; Institut Català d'Investigació Química; Spanish Ministry of Science and Innovation, Grant/Award Numbers: PID2020-112806RB-I00, CEX2019-000925-S; European Union NextGenerationEU/PRTR, Grant/Award Number: TED2021-132850B-I00

Abstract

Elucidating the speciation (in terms of concentration versus pH) and understanding the formation mechanisms of polyoxometalates remains a significant challenge, both in experimental and computational domains. POMSimulator is a new methodology that tackles this problem from a purely computational perspective. The methodology uses results from quantum mechanics based methods to automatically set up the chemical reaction network, and to build speciation models. As a result, it becomes possible to predict speciation and phase diagrams, as well as to derive new insights into the formation mechanisms of large molecular clusters. In this work we present the main features of the first open-source version of the software. Since the first report [Chem. Sci. 2020, 11, 8448-8456], POMSimulator has undergone several improvements to keep up with the growing challenges that were tackled. After four years of research, we recognize that the source code is sufficiently stable to share a polished and user-friendly version. The Python code, manual, examples, and install instructions can be found at https://github.com/petrusen/pomsimulator.

KEYWORDS
DFT, mechanisms, polyoxometalates, speciation

1 | INTRODUCTION

Polyoxometalates (POMs) are a distinguished family of molecular nanoclusters typically formed by a combination of transition metals (Mo, W, V, Nb, Ta) in high oxidation states and oxygen atoms. The chemistry of these clusters is remarkably old, as the first POM was discovered two centuries ago.[1] Despite this longevity, the field is still far from entirely explored, and many new compounds are yet being published every year. The application impact of POMs can be seen in multiple areas such as catalysis,[2,3] energy materials,[4] biochemistry,[5–8] medicine,[9,10] and semiconductor devices.[11,12]

Unlike other chemistry fields such as organic synthesis, where the reactivity patterns and synthetic rules are particularly well-defined, the formation routes of POMs are far less clear. Part of the problem is because of the complex reactivity in solution of these molecular clusters, which requires a precise and simultaneous control of pH, ionic strength, temperature, total metal concentration, and additional reducing agents if needed. In fact, the speciation of these oxo-clusters still draws considerable attention as it has a determinant role in the synthesis of novel materials.[13] Ultimately, the reaction crude is crystallized to yield the desired molecular clusters.[14] To cope with the intricate and manually intensive characterization of the crystallographic data, novel

automated protocols, such as POMFinder, are being developed.[15] Experiments have successfully shed light onto the nature of the species in solution.[16–19] Concerning the rationalization of this chemical phenomenon, two decades ago a theory based on the concept of a *dynamic library of building blocks* was developed to explain the formation of polyoxometalates.[20,21] At present, this concept has transcended the domain of polyoxometalates, and it has been recently proposed as a potential interface between physics and biology.[22,23]

In the past few decades, computational chemistry methods have also been applied to rationalize the formation of these molecular oxoclusters. The exploration of the potential energy surface has allowed the identification of some intermediates and transition-state structures in the formation of POMs.[24,25] Moreover, the use of *ab initio* molecular dynamics simulations has also contributed to the understanding of the reaction mechanisms.[26] On the other hand, the field of reaction mechanisms exploration is experiencing a change of paradigm because of the development of several automated frameworks.[27–32] In line with this emerging approach, we presented a computational method four years ago aimed at predicting the aqueous speciation and reaction mechanisms of POMs.[33] Since then, this methodology, that we have named POMSimulator, has been applied to five families of isopolyoxometalate clusters: molybdates, tungstates, vanadates, niobates, and tantalates.[34,35] We have also recently shown that it is possible to use the reaction network generated with POMSimulator to perform accurate microkinetic simulations,[36] thus opening the door to model recent kinetic experiments.[37–39] In this report we release an stable version of POMSimulator code. We quickly revisit its theoretical background, detail the most relevant technical aspects, and provide a recommendation guidance for its optimal usage.

2 | THEORETICAL BACKGROUND

POMSimulator is a methodology that addresses the aqueous speciation and self-assembly of POMs *in silico*, relying on three main theoretical pillars: quantum mechanics based methods, graph theory, and chemical equilibrium. By means of Density Functional Theory, the Gibbs free energies are computed for a set of the metal-oxo clusters at standard conditions (298 K and 1 atm). However, the design of the molecular set is not trivial. It must contain a combination of experimentally reported compounds and transient -hence undetected- intermediates. Because of the building block nature of POMs, the number of stable structural isomers (e.g., different ways of assembling the metal-oxo framework) is severely reduced.[40] Moreover, we employ the electrostatic potential to determine the most likely protonation sites of every species. For example, the results showed in this work regard a total of 45 compounds. Apart from the Gibbs free energies, our methodology also uses the molecular connectivity. This is determined using the Quantum Theory of Atoms in Molecules,[41] which represents a straightforward method to deduce chemical bonds from the bond critical points in the electronic density directly. So far, POMSimulator has exclusively used the Amsterdam Density Functional package[42] because of its computational efficiency with GGA functionals and its implementation of the QTAIM electron density analysis method. Nonetheless, geometry optimizations and Gibbs free energy calculations are ubiquitous in most quantum chemistry software, while molecular connectivity could also be determined by alternative methods (e.g., atom distance thresholds or other implementations of QTAIM). Therefore, we envision that open-source alternatives such as ORCA[43] and MOLCAS,[44] among others, may also be employed in the near future.

Once the essential data for all the metal-oxo clusters in the set has been gathered, the next step is to generate a reaction network that interconnects the multiple species. To address this task, we rely on the topological properties defined within Graph Theory. Molecules are represented as molecular graphs g_i, where atoms and bonds are transformed into nodes and edges, respectively. Moreover, atomic numbers are embedded as node attributes, and Gibbs free energies as graph attributes. Consequently, it becomes possible to evaluate the isomorphic property, which relates graphs that share a set or subset of nodes and edges. The application of this property boils down to the determination of all the potential chemical reactions for a given collection of compounds. For instance, unimolecular reactions involve the evaluation of two molecular graphs, whereas bimolecular reactions involve the consideration of three molecular graphs. To perform this evaluation systematically, we define a so-called Isomorphic Matrix which is expressed as a triangular matrix in Equation (1). The calculation of this matrix is carried out by the *generate_isomorphic_matrix* function, which allows generating and storing the Isomorphic Matrix as a separate csv file –thus avoiding recalculating this matrix multiple times. We take advantage of the fact that evaluating the isomorphism $g_i \rightarrow g_j$ versus $g_j \rightarrow g_i$ will essentially give the two directions of a same chemical reaction. The matrix has two possible Boolean values, based on whether a molecular graph pair is isomorphic (1, True) or not (0, False). For the sets of molecular graphs that are confirmed to be isomorphic, an atom count difference is employed to determine the reaction type (see Equation 2). The most relevant reactions in the context of nucleation of POMs are acid-base equilibria, condensation reactions, and addition reactions. It is noteworthy that as a raw approximation, it is possible to assume that all the molecular graphs are isomorphic (e.g., take matrix 1 as it is) and only filter the reactions through stoichiometry (more details in Section 3.1).

$$\text{Isomorphic Matrix} = \begin{array}{c} \begin{matrix} g_1 & g_2 & g_3 & g_4 & g_5 \end{matrix} \\ \begin{pmatrix} 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 \\ 1 & 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 & 1 \end{pmatrix} \end{array} \begin{matrix} g_1 \\ g_2 \\ g_3 \\ g_4 \\ g_5 \end{matrix} \tag{1}$$

$$\Delta(g_i, g_j) = (M_i - M_j, O_i - O_j, H_i - H_j) \tag{2}$$

The third pillar of POMSimulator is Chemical Equilibrium Theory. To calculate the concentration of the molecular-oxo clusters at a given pH using the chemical equilibrium reactions, it is necessary to solve an

speciation model. This model essentially contains chemical reactions plus the mass balance Equation (3). Each reaction is expressed as in Equation (4), where concentrations are the dependent variables, and the ΔG_r is the independent variable.

$$\sum_{i=1}^{N} \nu_{M,i}[X]_i - c_0 = 0 \tag{3}$$

$$[C]^{\nu_C}[D]^{\nu_D} e^{-\Delta G_r/(RT)} - [A]^{\nu_A}[B]^{\nu_B} = 0 \tag{4}$$

Because the dependent terms are expressed as products, the subsequent system of equations becomes non-linear thus increasing the complexity. However, there is a fundamental problem because the number of reactions typically exceeds the number of compounds. Therefore, the system is overdetermined as the number of equations is greater than the number of variables. This evidence hinders the idea of constructing a solvable speciation model with all the chemical reaction network. To address this issue, we set up all the possible combinations of chemical reactions, each of them leading to a different speciation model. This approach ensures that the problem becomes solvable, but at the cost of creating a computationally-demanding task. Because the reaction sampling in the speciation models is crucial, we implemented *two* assumptions to filter out chemically unsound models:

1. **Assumption I**: Every speciation model must contain all the acid-base equilibria. Thus, different speciation models will nonetheless share the same set of protonation/deprotonation reactions. This assumption is made based on the paramount role of pH in the aqueous reactivity of POMs. Because the speciation of these molecular-oxo clusters is very sensible to small changes in pH, it becomes mandatory to account for this effect in every model. As a consequence, the chemical reactions included in the speciation models can be divided in acid-base (which are constant in all the models) and nucleation reactions (which change from one model to another).
2. **Assumption II**: The nucleation reactions of every speciation model must consider all the nuclearities of the molecular set. In other words, the condensation and addition reactions must account for the formation of every nuclearity (i.e., POMs with the same number of metal and oxygen atoms) to avoid biasing the formation of one type of clusters in front of others. Otherwise, we could end up in a situation where one cluster is not formed, not because of being thermodynamically unstable, but because there is no reaction that leads to its formation in the speciation model.

To facilitate the comparison and evaluation of the simulated results, we express the speciation data as formation constants, in line with the convention used in experiments.[45] The formation reaction is defined in Equation 5, and it accounts for how many moles of reference molecules, $[MO_4]^{2-}$, and protons are needed to form one mole of product. It is noteworthy that the reference compound is not restricted to the monomer. For instance, in isopolyoxoniobates the Lindqvist anion, $[Nb_6O_{19}]^{8-}$, is taken as reference. Formation constants are typically reported at a fixed value of ionic strength (see Equation 6).

$$p\,[MO_4]^{2-} + q\,H^+ \rightleftarrows [H_xM_pO_m]^{z-} + n\,H_2O \tag{5}$$

$$K_f = \frac{(a_{[H_xM_pO_m]^{z-}}) \cdot (a_{H_2O})^n}{(a_{[MO_4]^{2-}})^p \cdot (a_{H^+})^q} \tag{6}$$

The aqueous speciation of POMs also depends on the ionic strength. Large values of I favor more charged species, and viceversa. To consider the *effective concentration* of the species, aka activity, we rely on a modified version of the classic Davies equation, which was proposed in the *Geochem* software to expand its range of applicability.[46,47] This modification consisted in splitting the activity equation in two conditions, as shown in Equation (7). Essentially, the function would remain the same but it would only change two parameters: a_0 and B_0. For instance, if $I \leq 0.5$ M, a_0 would be equal to 1/B, whereas B_0 would be equal to the product of 0.2, B_0 and I. Instead, if $I > 0.5M$, a_0 would be equal to $(3+|z|)\cdot 10^{-8}$, and B_0 to a constant value of 0.041. In this manner, the second term would not increase exponentially if large values of I were employed.

$$\begin{aligned} \log_{10}\gamma_{\pm}^{GC} &= -A \cdot z_j^2 \cdot \frac{\sqrt{I}}{1 + B \cdot a_0 \cdot \sqrt{I}} \cdot B_0 \cdot I \\ I \leq 0.5M \quad a_0 &= \frac{1}{B} \qquad B_0 = 0.2 \cdot A \cdot z^2 \\ I > 0.5M \quad a_0 &= (3+|z|)\cdot 10^{-8} \qquad B_0 = 0.041 \end{aligned} \tag{7}$$

The first report about POMSimulator was published in 2020, and during these 4 years we have found a systematic yet intriguing feature. It was discovered *a posteriori* that the formation constants obtained through the simulations were overestimated respect to the values reported in the experiments. That being said, it is rather well-known that the reaction energies derived from DFT yield equilibrium constants that are typically deviated from the reported values.[48] This is specially common for acid-dissociation constants, where there is a broad literature available.[49] This deviation is commonly attributed to the poor modeling of the solvation energy of the proton, which is present in both the K_a and K_f equations. Therefore, the theoretical formation constants calculated with POMSimulator need to be linearly scaled using experimental values. In our previous studies, we have shown that this scaling step is very accurate (R^2>0.9990 and Root Mean Squared Error, RMSE<1.0) and it unlocks the quantitative prediction of unreported metal-oxo clusters. Even so, the intrinsic dependence on experimental data is a clear limitation in terms of applicability to other polyoxometalate systems. In our previous studies, we have noticed that the slope parameter appears to be constant for the five isopolyoxometalate families studied so far. Thus, Equation (8) shows a phenomenological expression for a-hypothetical-universal scaling of the DFT constants. Although this feature still demands further investigation, it has the potential of making POMSimulator independent of available experimental data.

$$logK_f^{Exp} \approx 0.3 \cdot logK_f^{DFT} + b \quad (8)$$

3 | TECHNICAL DETAILS

In this section we will explain the most relevant technical aspects of the software package. To do so, we employ as example a reduced set of isopolyoxotungstates that we previously published.[34] In this manner we can complement the explanations with chemical plots. Besides, the employed molecular set is also available in the release source code. The main simulation file of POMSimulator is named as *simulation_tungstates.py*. This file uses the functionalities that are implemented in the internal modules. Below we will detail the two most important dependencies which correspond to the *graph_module* and *msce_module*.

3.1 | Chemical reaction network

This part of the methodology depends entirely on the internal library *graph_module*. It relies on standard Python dependencies, such as itertools, multiprocessing and Numpy.[50] Moreover, we employ the NetworkX library to call the Graph Theory functionalities.[51] The chemical information parsed from the Amsterdam Density Functional output files[42] is passed to the *Molecular_Graph*() function which ultimately converts all the molecules to molecular graphs. This step must be carefully monitored because the bond connectivity derived from QTAIM can contain bond artifacts. That is why we have implemented a functionality that creates .mol files so that the user can double-check the bond connectivity. Although this part may seem trivial, it plays a crucial role ensuring that the isomorphism property is successfully employed.

The next step is to map the isomorphism property for all the molecular graphs. However, this is a computationally demanding step that may create a considerable bottleneck. Therefore, we have enabled the possibility of not calling the isomorphism function, and approximate the isomorphic matrix as the triangular matrix shown in Equation (1). Nonetheless, a more rigorous approach involves the computation the isomorphic matrix with the function *Molecular_Graphs_to_Isomorphic_Matrix*(). To alleviate the computational cost of this step, we have vectorized the mapping of the matrix. In this manner, it can run in parallel according to the number of cores assigned in the input. Still, there will be batches that will be solved faster than others, but if needed the user can tweak function parameters to optimize the performance.

Once the isomorphic matrix has been calculated, it is used as a grid for enumerating all the possible chemical reactions, calling the function *Isomorphism_to_ChemicalReactions*(). The user must define which are the reaction types of interest for the particular system. So far, the package includes eleven reaction types which are summarized in Table 1. As mentioned before, acid/base reactions are essential for describing the pH dependent behaviour of polyoxometalates. Also, hydration reactions must be considered because metal-oxo monomers tend to expand their coordination sphere, from tetrahedral to trigonal bipyramid or octahedral, by coordinating water ligands. Then, condensation reactions are the main driving force for the growth of POMs. There are several types of condensation reactions because the nucleation pattern differs from one cluster to another. For instance, the most common pattern corresponds to the *Cw1* which would have this reaction associated:

$$HMO_4 + HMO_4 \rightarrow M_2O_7 + H_2O \quad (9)$$

In this particular case, a $[M_mO_{3m+1}]^z$ growth pattern is followed. However, there are other cases where POMs can undergo distinct pathways, such as $[M_mO_{3m}]^z$. For example, the *Cw4* reaction type which would have this reaction pattern associated:

$$H_2MO_4 + H_2MO_4(H_2O)_2 \rightarrow M_2O_6 + 4H_2O \quad (10)$$

We also contemplate the addition reaction type, which is similar to the condensation but no water molecules are released. Another important reaction is hydroxylation, which was needed when modeling the speciation of alkaline-clusters such as isopolyoxoniobates and-tantalates. The last reaction type implemented in the present version is acid hydrolysis, which is a frequent process in acid-clusters such as isopolyoxomolybdates, -tungstates, and -vanadates. It is worth highlighting that so far the methodology only contemplates two main families of transformations: proton/water acid equilibria, and nucleation reactions. Redox reactions are not implemented yet, and further modifications would have to be performed to adapt the core functions to electron transfer processes.

The POMSimulator packages offers a script to plot the chemical reaction network: *plot_reac_map*. Figure 1 shows the type of plot that can be obtained with this functionality, depicting all the different nuclearities in the CRN as nodes (in black) and the reactions as lines, which can be colored according the reaction energy.

TABLE 1 List of the 11 chemical reaction types implemented in POMSimulator.

Reaction type	Abbreviation	Description
Protonation	P	$R_1 + H_5O_2^+ \rightarrow P + H_4O_2$
1 Water Hydration	H2Ow1	$R_1 + H_2O \rightarrow P$
2 Water Hydration	H2Ow2	$R_1 + 2H_2O \rightarrow P$
1 Water Condensation	Cw1	$R_1 + R_2 \rightarrow P + H_2O$
2 Water Condensation	Cw2	$R_1 + R_2 \rightarrow P + 2H_2O$
3 Water Condensation	Cw3	$R_1 + R_2 \rightarrow P + 3H_2O$
4 Water Condensation	Cw4	$R_1 + R_2 \rightarrow P + 4H_2O$
10 Water Condensation	Cw10	$R_1 + R_2 \rightarrow P + 10H_2O$
Addition	A	$R_1 + R_2 \rightarrow P$
Hydroxylation	HO	$R_1 + H_6O_3 \rightarrow P + H_5O_2^+$
Acid Hydrolysis	H3O	$R_1 + H_3O \rightarrow P$

3.2 | Formation constants

One of the main functionalities of POMSimulator is the determination of the formation constants of *all* metal oxides in the molecular set.

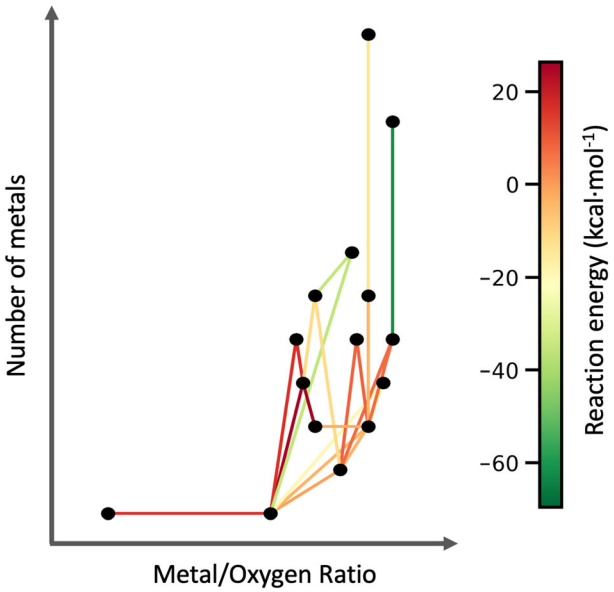

FIGURE 1 Chemical reaction network of tungstates. Compounds (black nodes) are positioned according to the number of tungstens and oxygens. Reaction energies are colored according to whether it is spontaneous (green) or not (red).

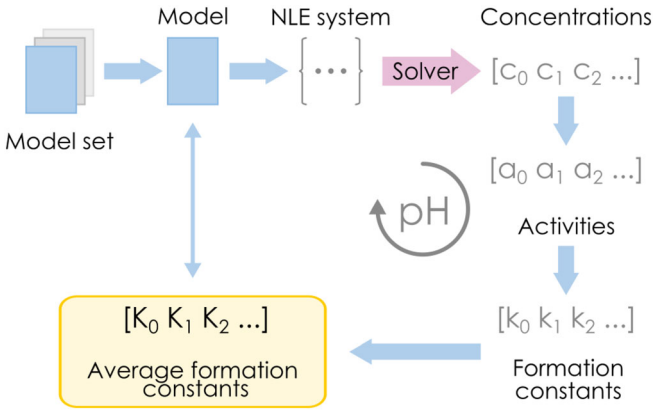

FIGURE 2 Schematic depiction of formation constant determination in POMSimulator.

This is a major advantage against experimental characterizations, which are often limited only to selected compounds in the reaction network. These constants are essential for all further applications of the method, as they are employed to assess the quality of speciation models and therefore to select the best one, as well as for the generation of speciation and phase diagrams.

As introduced in previous section, formation constants will be always referred to a single species selected as reference (Equation 5), whose corresponding K_f value will be zero. However, the reactions that we have defined and are present in the model are not be referred to this species, but do instead correspond to elementary processes in the overall self-assembly (e.g., a dimerization of two W_6 species to form a W_{12} compound). Thus, to retrieve the desired K_f values applying Equation (5) for each species, we should know the concentrations of all the compounds in the reaction network, by solving the speciation model.

Thus, it is necessary to solve a multi-species multi-equilibria problem. Each speciation model, as previously defined, produces a system of non-linear equations (NLE) involving the concentrations of all species in the molecular set. Therefore, for $\mathbf{N_m}$ speciation models, it becomes necessary to solve $\mathbf{N_m}$ different sets of NLEs, sheerly increasing the computational complexity of POMSimulator. The procedure, explained in the following paragraphs, is outlined in Figure 2, and is carried out through the function *Speciation_from_Equilibrium* in the *msce_module*.

If we define the equilibrium constant both in terms of a quotient of activities (Equation 5) and in terms of the reaction energy (Equation 11), we can define a non-linear equation (Equation 12). For simplicity and from now on, although as shown in Equation (5) activities a_X are used, we will represent terms as if they were concentrations $[X]$.

$$K_f = e^{-\Delta G^0/RT} \tag{11}$$

$$\frac{[H_x M_p O_m^{z-}][H_2O]^n}{[MO_4^{2-}]^p[H^+]^q} - e^{-\Delta G^0/RT} = 0 \tag{12}$$

For all reactions in Table 1, the expression corresponding to Equation (12) is encoded as a text string inside the *DataBase* module of the program. These expressions are defined in terms of three (monomolecular) or four (bimolecular) parameters, them being the concentration of the *product*, the reaction energy in kcal mol^{-1}, and the concentration(s) of the *reactant*(s). Therefore, for a given speciation model, the corresponding NLEs will be retrieved, setting up the required indices for each species in the concentration vector as well as the reaction energy. Also, an additional equation corresponding to the mass balance of the metal atoms (Equation 3) is always included in the system. Then, the equations are transformed from a string to a Python object: while in the original version of the code this was done at runtime, a precompilation of the corresponding object allowed us to achieve a 15-to-20-fold speed-up of the code which greatly increased its applicability to more complex systems. While even further optimization, by for example linearizing the equilibrium constant equations taking logarithms, would be highly desirable, the requirement for a mass balance equation hinders this approach.

The Scipy[52] library is then used to solve the system of NLEs, through the *root* function of the package, using an hybrid solver based on the Powell method,[53] and obtaining values for the concentrations of all the species. In all cases, a vector of zeros is used as the initial guess for the solver. Additionally, the accuracy of each solution resulting from Powell's algorithm is double-checked by applying an alternative Least Squares (LS) solver, and then comparing both results through a linear regression. Only if the normalized RMSE of this regression is below a user-specified threshold (between 0.1 and 1.0) the solution is accepted. This process has to be repeated along a range of different pH values, due to the strong dependency that the assembly of POMs has on pH. Once we have gathered concentrations for every compound at every pH point, we may determine their formation constants in terms of the reference species. As the computed K_f

values are likely to vary depending on pH, the average value across the pH range is computed, assigning a single set of formation constants for every speciation model.

As mentioned, applying this procedure for a large set of speciation models can imply a quite large computational cost. One key optimization from the original versions of the code was the implementation of parallelization, solving different speciation models in different CPU cores. Moreover, we have also implemented *batches* of models, whose size can be tweaked by the user depending on the available computational resources. For the tungsten set (18,432 models) provided with this software release, up to 0.60 models/(s core) were solved in a Intel Core i9-12900K, resulting in a total computation time of around 22 min for the complete dataset if 24 cores of the processor are used. The aforementioned equation precompilation is essential to achieve those times: if the string-based equations are directly evaluated, as it was done in the original versions of the program, the solution rate reduces to 0.04 models/(s core), thus requiring 5.5 h in the same hardware. Given that the number of speciation models grows exponentially with the complexity of the reaction network, the 15-fold speed-up is essential to be able to realistically apply POMSimulator to other systems.

3.3 | Speciation and phase diagrams

Once we have computed the formation constants of all species for each speciation model, we are now able to generate speciation and phase diagrams. We need to select the *best model*. To do so, as stated in Section 3, we perform linear regressions between the theoretical constants and the experimental reference values. Previous results have shown that there is an accurate linear correlation between these pairs of values across the whole set of models. From there, we can obtain the equation to scale the theoretical constants to better match the experimental results. Moreover, we can sort the models according to the RMSE values of these linear regressions. The speciation model with the lowest RMSE value will be selected as the *best model*, as it provides the closest match with experiments. Therefore, this model will be used to generate both speciation diagrams and phase speciation diagrams.

Once we have selected the formation constants from the best model and scaled them accordingly, we will establish the formation equilibrium for each species except for the reference compound. We then need to calculate the stoichiometric coefficients for each formation reaction, and set up the corresponding expression for the formation constant. From the equilibrium expression in Equation (6), we will now simplify the notation, denoting the reference compound as *Ref* and all other metal oxides as X_i: see Equations (13), (14) and (15).

$$P_i \cdot [Ref] + Q_i \cdot [H_3O^+] \rightleftharpoons [X]_i + n_i \cdot [H_2O] \tag{13}$$

$$K_{fi} = \frac{[X]_i \cdot [H_2O]^{n_i}}{[Ref]^{P_i} \cdot [H_3O^+]^{Q_i}} \tag{14}$$

$$[X]_i \cdot [H_2O]^{n_i} - K_{f,i} \cdot [Ref]^{P_i} \cdot [H_3O^+]^{Q_i} = 0 \tag{15}$$

At this point we will have N species in the molecular set and (N-1) equations, due to the reference species having a formation constant equal to zero.

Thus, we will set up the last equation of the NLE system, that corresponds to the mass balance (see Equation 3). If we fix a pH value and consequently proton concentration, we can solve the system of NLEs (Equation 16). For this reason, we establish a pH range, and solve the NLE system throughout the whole set of pH values. This results in an array of concentrations, containing the concentration of all species at all pH values. The corresponding function is named *Speciation_from_Formation_singlemetal*, belonging to *msce_module*.

As with formation constants, we employ the hybrid solver implemented in Scipy to solve Equation (16), with an initial guess of zeros for all concentrations.

$$\begin{cases} [X]_0 \cdot [H_2O]^{n_0} - K_{f,0} \cdot [Ref]^{P_0} \cdot [H_3O^+]^{Q_0} = 0 \\ [X]_1 \cdot [H_2O]^{n_1} - K_{f,1} \cdot [Ref]^{P_1} \cdot [H_3O^+]^{Q_1} = 0 \\ \vdots \\ [X]_i \cdot [H_2O]^{n_i} - K_{f,i} \cdot [Ref]^{P_i} \cdot [H_3O^+]^{Q_i} = 0 \\ \sum [X]_i + [Ref] - C_0 = 0 \end{cases} \tag{16}$$

From the array of concentrations, we can readily compute the molar percentage of the metal $\%M_{Xi} = 100 * \nu_{M,Xi}[X]_i / C_0$. In this way, we have a more direct comparison with experimental results, as well as a better representation of the abundance of larger oxo-clusters containing many metal atoms. An example of a speciation diagram $\%M$ vs *pH* is showcased in Figure 3.

To generate the phase diagrams, we follow the same methodology as per the speciation, repeating the process along a range of initial metal concentration C_0. In this way, we determine many different

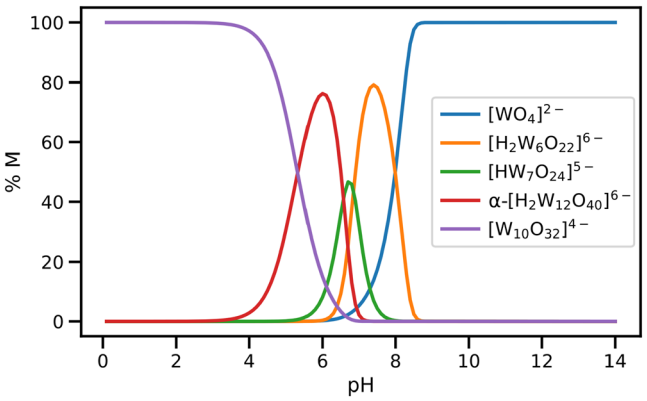

FIGURE 3 Speciation diagram for the tungsten system, at 298.15 K, 1 atm and 0.25 M ionic strength of NaCl. Each compound is colored according to the legend.

speciation diagrams, exploring how the speciation changes depending on the initial amount of metal. Then, for each of these diagrams, we can reduce the dimensionality by selecting the species with the largest molar fraction at every pH value. In the end, we get the most relevant species at each (C_0, *pH*) pair. Next, we may color each pixel in the map according to the dominant species, retrieving the phase diagram. An example for the tungsten system is collected in Figure 4.

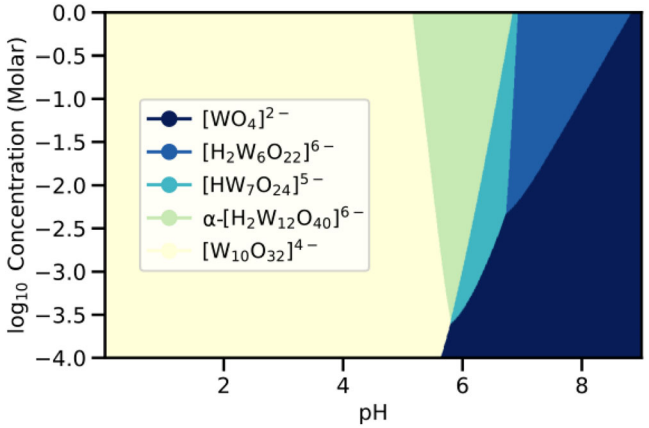

FIGURE 4 Phase diagram for the tungsten system at 298.15 K, 1 atm and 0.25 M ionic strength of NaCl. Each compound is colored according to the legend.

4 | INSPECTING THE INPUT FILE

POMSimulator offers some options that can be modified according to the needs of each particular system. However, this flexibility can also become an inconvenient when striving to determine which are the ideal settings for achieving an optimal accuracy/time ratio. Along this section, we will detail some of the most important parameters to guide the user.

```
### HARDWARE-RELATED PARAMETERS
cores = 2
batch_size = 5
```

CODE SNIPPET 1 Hardware and paralellization-related parameters in POMSimulator.

Code Snippet 1 contains basic parameters regarding the parallelization of the code, namely the total number of cores and the number of speciation models per batch, which will be used when computing formation constants. It is worth noting that the parallelization has only been implemented using the cores of one same CPU; as opposed to using cores from different CPUs.

Code Snippet 2 contains a set of key parameters related to the chemical assumptions taken in POMSimulator to reduce the number of reactions in the CRN and the final number of speciation models.

```
###    CHEMICAL PARAMETERS
use_isomorphisms = True
energy_threshold = 80
proton_numb = 2
reference = ['P', 'H2Ow1','H2Ow2',
             'Cw1','Cw2','Cw3',
             'Cw4', 'A','HO','H3O']
""" These parameters are not meant to
    be routinely modified """
conditions_dict = {
"proton_numb":proton_numb,
"restrain_addition":1,
"restrain_condensation":1,
"include_dimerization":True,
"force_stoich":[11],
"adjust_protons_hydration":True}
```

CODE SNIPPET 2 Main chemical parameters and assumptions in POMSimulator.

1. *use_isomorphisms* flags the utilization of isomorphisms to build the CRN. When set to True, it is necessary to have previously run the *compute_isomorphism* script in the Utilities folder. In this manner, the isomorphic matrix will be already available. Else, the program will use only stoichiometric criteria to define reactions.
2. *energy_threshold* sets a maximum value for reaction energies, disregarding processes that are too high in energy. It has been observed[35] that some endergonic reactions can become exergonic if we consider the simultaneous effect of the protonation reactions: thus, some caution shall be taken when tweaking this parameter.
3. *proton_number* limits the maximum difference in the number of protons between the two reactants in a nucleation reaction. This restriction is based on the fact that two species with different number of protons are unlikely to be found at the same value of pH, and therefore are unlikely to interact.
4. *reference* establishes the reaction types that are to be mapped in the isomorphic matrix. Protonation, condensation, and addition reactions should always be selected, since they are the basis of POMs reactivity. Only reactions appearing both in this list and in the molecular set will be included in the CRN.

Then, *conditions_dict* contains additional constraints (e.g., chemical assumptions) on the reactions considered in the network, which strongly depend on the system under study.

1. *restrain_addition*. Limits the maximum size of the building blocks employed in addition reactions. If set to 1, only additions involving at least one monomer will be accepted.
2. *restrain_condensation*. Same as the previous one, but for condensation reactions.
3. *include_dimerization*. Allows addition and condensation reactions between two equal building blocks, independently of previous restrictions.

4. *force_stoich*. Forces the acceptance of reactions leading to a specific stoichiometry, independently of energy, protonation or any other threshold.
5. *adjust_protons_hydration*. If True, takes into account the degree of hydration of POMs when handling the protonation threshold.

The parameters in Code Snippet 3 rule the calculation of formation constants.

```
### FORMATION CONSTANT PARAMETERS
I, C0 = 0.25, 0.005
min_pH, max_pH, grid = 0, 35, 70
ref_compound = 'W01O04-0H'
```

CODE SNIPPET 3 Parameters related to formation constant calculation in POMSimulator.

1. *I*. Ionic strength of the solution, selected according to the requirements of each particular system. It is worth reminding that ionic strength values above 1 mol $\cdot L^{-1}$ may become unstable due to the limitation of the Davies-modified activity equation.
2. C_0. Initial molar concentration of the metal.
3. *min_pH*, *max_pH* and *grid*. pH range employed for the determination of formation constants. Due to the aforementioned mismatch between experimental and computed constants, large pH ranges shall be used at this step: from previous experience, values between 0 and 35 for acidic POMs (W, Mo) and between 0 and 70 for amphoterous (V) and alkaline (Nb, Ta) POMs provide a good coverage. The total number of values is marked by *grid*: the larger this value, the more accurate but more time-demanding the determination will be.
4. *ref_compound* is the label for the selected reference compound for formation constants (see Equation 6). If there is experimental data available, we recommend to take the same reference compound so that the data can be compared afterwards.

To favor that the results generated by POMSimulator are reproducible, when the main simulation file is run it generates a text file with the values of all the parameters specified in the input and detailed in Code Snippets 1 to 3.

5 | CONCLUSIONS AND OUTLOOK

We have presented the first open–source release of the software package POMSimulator. This first version allows simulating the aqueous speciation of isopolyoxometalates from first-principle calculations. By releasing a public version of the code, we aim at providing a tool for complementing the discovery of novel polyoxometalates. Moreover, having an accessible version of the code means that other researchers can modify the source code based on their particular needs.

At present, POMSimulator is being actively developed, and we envision to incorporate further functionalities in the near future. Some of these directions are:

1. Support of heteropolyoxometalates. So far the code only works with isopolyoxometalates. To enable the exploration of more complex clusters, such as the well-known phosphomolybdate anion, some modifications are needed in the source code, such as an additional mass balance equation for the addendum atom.
2. Statistical treatment. As mentioned throughout the manuscript, the number of speciation models scales factorially with the number of reactions and the complexity of the CRN. Consequently, the computational cost associated to the resolution of the entire set of models can be remarkably high. An alternative approach that we are currently exploring relies on the stochastic sampling of models, together with a statistical analysis to classify the different types of models according to their speciation behavior.
3. Redox reactivity. The current version of the code does not consider any redox reaction (Table 1), even though POMs' metal framework is prone to suffer changes in the oxidation states. The addition of redox reactions would require fundamental changes in the definition of the equations participating in the speciation models (e.g., Nernst equation, electroneutrality).

ACKNOWLEDGMENTS

We acknowledge the Spanish Ministry of Science and Innovation MCIN/AEI/10.13039/501100011033 (PID2020-112806RB-I00 and CEX2019-000925-S), the European Union NextGenerationEU/PRTR (TED2021-132850B-I00), the ICIQ Foundation and the CERCA program of the Generalitat de Catalunya for funding.

CONFLICT OF INTEREST STATEMENT

The code is licensed under a GNU Affero General Public License v3.0.

DATA AVAILABILITY STATEMENT

The release code of POMSimulator 1.0 can be accessed and referenced via Zenodo.[54] The data that support the findings of this study are openly available in ioChem-BD at http://dx.doi.org/10.19061/iochem-bd-1-201. The molecular geometries of all oxo-clusters were fully optimized using a DFT method, and employing ADF package (SCM ADF version 2019.1).[42] We have used the standard functional PBE,[55,56] with the relativistic corrections related to the scalar-relativistic zero-order regular approximation (ZORA),[57,58] using a TZP basis set level. The effect of solvent was introduced by means of the continuous solvent model COSMO with Klamt radii for water.[59] Stationary points were characterized with analytic frequency calculations. Ground state free energies were computed at 298.15 K and 1 atm, using the ideal gas-rigid rotor-harmonic oscillator (IGRRHO) model. Dataset collections are both available in the Github package and in the ioChem-BD repository[60] via the Reference 61.

REFERENCES

[1] J. J. Berzelius, *Ann. Phys.* **1826**, *82*, 369.
[2] J. C. Raabe, M. J. Poller, D. VoSS, J. Albert, *ChemSusChem* **2023**, *16*, e202300072.
[3] Z. Li, Y. Huang, H. Li, F. Zhang, Y. Ren, W. Shi, Q. Liu, X. Wang, *J. Am. Chem. Soc.* **2024**, *146*, 450.
[4] J. J. Chen, L. Vilà-Nadal, A. Solá-Daura, G. Chisholm, T. Minato, C. Busche, T. Zhao, A. Y. Kandasamy, B. Ganin, R. M. Smith, I. Colliard, J. J. Carbó, J. M. Poblet, M. Nyman, L. Cronin, *J. Am. Chem. Soc.* **2022**, *144*, 8951.
[5] M. Aureliano, N. I. Gumerova, A. Rompel, *Metals* **2023**, *13*, 6.
[6] H. Soria-Carrera, E. Atrián-Blasco, R. Martín-Rapún, S. G. Mitchell, *Chem. Sci.* **2023**, *14*, 10.
[7] L. Zhang, P. He, H. Chen, Q. Liu, L. Li, X. Wang, J. Li, *Nano Res.* **2024**, *17*, 262.
[8] A. Barba-Bon, N. I. Gumerova, E. Tanuhadi, M. Ashjari, Y. Chen, A. Rompel, W. M. Nau, *Adv. Mater.* **2024**, *36*, e2309219.
[9] Z. Khoshkhan, M. Mirzaei, A. Amiri, N. Lotfian, J. T. Mague, *Inorg. Chem.* **2024**, *63*, 2877.
[10] N. Song, M. Lu, J. Liu, M. Lin, P. Shangguan, J. Wang, B. Shi, J. Zhao, *Angew. Chem. Int. Ed.* **2024**, *63*, e202319700.
[11] C. Busche, L. Vilà-Nadal, J. Yan, H. N. Miras, D. L. Long, V. P. Georgiev, A. Asenov, R. H. Pedersen, N. Gadegaard, M. M. Mirza, D. J. Paul, J. M. Poblet, L. Cronin, *Nature* **2014**, *515*, 545.
[12] J. Ding, G. Yan, F. Adamu-Lema, Y. Dou, Y. Chen, B. Ding, V. P. Georgiev, A. Asenov, *J. Nanoelectron. Optoelectron.* **2021**, *16*, 884.
[13] N. I. Gumerova, A. Rompel, *Sci. Adv.* **2023**, *9*, eadi0814.
[14] A. Misra, K. Kozma, C. Streb, M. Nyman, *Angew. Chem., Int. Ed.* **2020**, *59*, 596.
[15] A. S. Anker, E. T. S. Kjær, M. Juelsholt, K. M. Ø. Jensen, *J. Appl. Crystallogr.* **2024**, *57*, 34.
[16] E. F. Wilson, H. N. Miras, M. H. Rosnes, L. Cronin, *Angew. Chem., Int. Ed.* **2011**, *50*, 3720.
[17] M. Piot, B. Abécassis, D. Brouri, C. Troufflard, A. Proust, G. Izzet, *Proc. Nat. Acad. Sci.* **2018**, *115*, 8895.
[18] M. Nyman, *Coord. Chem. Rev.* **2017**, *352*, 461.
[19] R. L. Meyer, R. Love, W. W. Brennessel, E. M. Matson, *Chem. Commun.* **2020**, *56*, 8607.
[20] A. Müller, P. Kögerler, *Coord. Chem. Rev.* **1999**, *182*, 3.
[21] A. Müller, P. Gouzerh, *Chem. Soc. Rev.* **2012**, *41*, 7431.
[22] A. Sharma, D. Czégel, M. Lachmann, C. P. Kempes, S. I. Walker, L. Cronin, *Nature* **2023**, *622*, 321.
[23] K. Y. Monakhov, *Nat. Sci.* **2024**, e20230020.
[24] L. Vilá-Nadal, A. Rodríguez-Fortea, L. K. Yan, E. F. Wilson, L. Cronin, J. M. Poblet, *Angew. Chem., Int. Ed.* **2009**, *48*, 5452.
[25] Z. L. Lang, W. Guan, L. K. Yan, S. Z. Wen, Z. M. Su, L. Z. Hao, *Dalton Trans.* **2012**, *41*, 11361.
[26] L. Vilà-Nadal, E. F. Wilson, H. N. Miras, A. Rodríguez-Fortea, L. Cronin, J. M. Poblet, *Inorg. Chem.* **2011**, *50*, 7811.
[27] S. Maeda, K. Ohno, K. Morokuma, *Phys. Chem. Chem. Phys.* **2013**, *15*, 3683.
[28] P. M. Zimmerman, *J. Comput. Chem.* **2013**, *34*, 1385.
[29] J. A. Varela, S. A. Vázquez, E. Martínez-Núñez, *Chem. Sci.* **2017**, *8*, 3843.
[30] Y. Guan, V. M. Ingman, B. J. Rooks, S. E. Wheeler, *J. Chem. Theory Comput.* **2018**, *14*, 5249.
[31] T. A. Young, J. J. Silcock, A. J. Sterling, F. Duarte, *Angew. Chem., Int. Ed.* **2021**, *60*, 4266.
[32] J. P. Unsleber, S. A. Grimmel, M. Reiher, *J. Chem. Theory Comput.* **2022**, *18*, 5393.
[33] E. Petrus, M. Segado-Centellas, C. Bo, *Chem. Sci.* **2020**, *11*, 8448.
[34] E. Petrus, C. Bo, *J. Phys. Chem. A* **2021**, *125*, 5212.
[35] E. Petrus, M. Segado-Centellas, C. Bo, *Inorg. Chem.* **2022**, *61*, 13708.
[36] E. Petrus, D. Garay-Ruiz, M. Reiher, C. Bo, *J. Am. Chem. Soc.* **2023**, *145*, 18920.
[37] H. N. Miras, C. Mathis, W. Xuan, D. L. Long, R. Pow, L. Cronin, P. Natl, *Acad. Sci.* **2020**, *117*, 10699.
[38] D. Lockey, C. Mathis, H. N. Miras, L. Cronin, *Matter* **2022**, *5*, 302.
[39] K. Li, S. Zhang, K. L. Zhu, L. P. Cui, L. Yang, J. J. Chen, *J. Am. Chem. Soc.* **2023**, *145*, 24889.
[40] D.-L. Long, R. Tsunashima, L. Cronin, *Angew Chem Int Ed* **2010**, *49*, 1736.
[41] R. Bader, *Atoms in Molecules – a Quantum Theory*, Vol. 360, Oxford University Press, Oxford **1994**.
[42] G. te Velde, F. M. Bickelhaupt, E. J. Baerends, C. Fonseca Guerra, S. J. A. van Gisbergen, J. G. Snijders, T. Ziegler, *J. Comput. Chem.* **2001**, *22*, 931.
[43] F. Neese, *WIREs Comput. Mol. Sci.* **2022**, *12*, e1606.
[44] F. Aquilante, J. Autschbach, R. K. Carlson, L. F. Chibotaru, M. G. Delcey, L. De Vico, I. Fdez, N. F. Galván, L. M. Frutos, L. Gagliardi, M. Garavelli, A. Giussani, C. E. Hoyer, G. L. Manni, H. Lischka, D. Ma, P. Å. Malmqvist, T. Müller, A. Nenov, M. Olivucci, T. B. Pedersen, D. Peng, F. Plasser, B. Pritchard, M. Reiher, I. Rivalta, I. Schapiro, J. Segarra-Martí, M. Stenrup, D. G. Truhlar, L. Ungur, A. Valentini, S. Vancoillie, V. Veryazov, V. P. Vysotskiy, O. Weingart, F. Zapata, R. Lindh, *J. Comput. Chem.* **2016**, *37*, 506.
[45] N. I. Gumerova, A. Rompel, *Chem. Soc. Rev.* **2020**, *49*, 7568.
[46] G. Sposito, S. V. Mattigod, Geochem: A Computer Program for the Calculation of Chemical Equilibria in Soil Solution and Other Natural Water Systems. **1980**.
[47] D. R. Parker, L. W. Zelazny, T. B. Kinraide, *Soil Sci. Soc. Am. J.* **1987**, *51*, 488.
[48] P. Pracht, R. Wilcken, A. Udvarhelyi, S. Rodde, S. Grimme, *J. Comput. Aided. Mol. Des.* **2018**, *32*, 1139.
[49] P. G. Seybold, G. C. Shields, *WIREs Comp. Mol. Sci.* **2015**, *5*, 290.
[50] C. R. Harris, K. J. Millman, S. J. van der Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N. J. Smith, R. Kern, M. Picus, S. Hoyer, M. H. van Kerkwijk, M. Brett, A. Haldane, J. Fernández del Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, *Nature* **2020**, *585*, 357.
[51] A. A. Hagberg, D. A. Schult, P. J. Swart, Exploring network structure, dynamics, and function using NetworkX, in *Proceedings of the 7th Python in Science Conference* (SciPy2008) (Eds: G. Varoquaux, T. Vaught, J. Millman), Pasadena, CA, USA **2008**, p. 11.
[52] P. Virtanen, R. Gommers, T. E. Oliphant, M. Haberland, T. Reddy, D. Cournapeau, E. Burovski, P. Peterson, W. Weckesser, J. Bright, S. J. van der Walt, M. Brett, J. Wilson, K. J. Millman, N. Mayorov, A. R. J. Nelson, E. Jones, R. Kern, E. Larson, C. J. Carey, İ. Polat, Y. Feng, E. W. Moore, J. VanderPlas, D. Laxalde, J. Perktold, R. Cimrman, I. Henriksen, E. A. Quintero, C. R. Harris, A. M. Archibald, A. H. Ribeiro, F. Pedregosa, P. van Mulbregt, *Nat. Methods* **2020**, *17*, 261.
[53] M. J. D. Powell, *Comput. J.* **1964**, *7*, 155.
[54] E. Petrus, J. Buils, D. Garay-Ruiz, M. Segado-Centellas, C. Bo, petrusen/pomsimulator: Release 1.0.0, **2024**. https://doi.org/10.5281/zenodo.10689769
[55] J. P. Perdew, *Phys. Rev. B* **1986**, *33*, 8822.
[56] J. P. Perdew, *Phys. Rev. B* **1986**, *34*, 7406.
[57] E. Van Lenthe, E. J. Baerends, J. G. Snijders, *J. Chem. Phys* **1993**, *99*, 4597.
[58] E. Van Lenthe, E. J. Baerends, *J. Comput. Chem.* **2003**, *24*, 1142.
[59] A. J. Klamt, *J. Phys. Chem.* **1995**, *99*, 2224.
[60] M. Álvarez-Moreno, C. De Graaf, N. López, F. Maseras, J. M. Poblet, C. Bo, *J. Chem. Inf. Model.* **2015**, *55*, 95.
[61] E. Petrus, ioChem Data Collection. **2021** https://doi.org/10.19061/iochem-bd-1-201

How to cite this article: E. Petrus, J. Buils, D. Garay-Ruiz, M. Segado-Centellas, C. Bo, *J. Comput. Chem.* **2024**, *45*(26), 2242. https://doi.org/10.1002/jcc.27389