

PyES – An open-source software for the computation of solution and precipitation equilibria

Lorenzo Castellino^a, Eugenio Alladio^a, Stefano Bertinetti^a, Gabriele Lando^b, Concetta De Stefano^b, Salvador Blasco^c, Enrique García-España^c, Sofia Gama^d, Silvia Berto^{a, **}, Demetrio Milea^{b, *}

^a Dipartimento di Chimica, Università di Torino, via P. Giuria 7, 10125, Torino, Italy

^b Dipartimento di Scienze Chimiche, Biologiche, Farmaceutiche ed Ambientali, CHIBIOFARAM, Università degli Studi di Messina, Viale Ferdinando Stagno d'Alcontres 31, 98166, Messina, Italy

^c Institute of Molecular Sciences, University of Valencia, c/Catedrático José Beltrán Martínez 2, 46980, Paterna, Valencia, Spain

^d Centro de Ciências e Tecnologias Nucleares, Instituto Superior Técnico, Universidade de Lisboa, Estrada Nacional 10 (km 139.7), 2695-066, Bobadela LRS, Portugal

ARTICLE INFO

PyES and this work are dedicated to the memory of Prof. Silvio Sammartano, for his invaluable contribution to the development of computer programs for solution equilibria.

Keywords:

Species distribution
Titrations
Stability constants
Solution equilibria
Precipitation equilibria
Python

ABSTRACT

A new, open-source, practical, modern and multi-platform Python application for concentrations calculation (PyES - Python Equilibrium Species) was developed by re-writing and implementing the ES4, a freeware computer program originally written using the BASIC programming language. Currently PyES has two working modes, i.e., titration simulation and species distribution, and it can handle both precipitation and solution equilibria. Noteworthy, PyES is actually the only open-source software able to perform calculations at variable ionic strength, taking into account the dependence of the stability constants on it, and to take into account the error propagation to estimate the uncertainties in the calculated concentrations using those of the formation constants, solubility products and components concentrations. Various tests were performed to verify the reliability of PyES with very satisfying results. Moreover, PyES is user friendly and compatible with existing operative systems. The data analysis results can be visualized in a graphical presentation and can be easily exported as .xlsx or .csv files. PyES and the corresponding source code are available for download at <https://github.com/Kastakin/PyES>.

1. Introduction

The need for modern chemists for IT tools to allow and facilitate their research activities is very well known. From simple spreadsheets editors such as Microsoft Excel, to complex computational routines specifically written for the problems in hand, every chemists must resort to some computer tools in their career. Therefore, simpler tools able to aid in routine operations are favored over complex scripts that require some coding knowledge to be written. The main issue is that, often, such tools are non-existent or lackluster, and their use creates a virtual barrier between what the users want and what they can effectively do.

This has been observed firsthand within the COST ACTION 18202, NECTAR - "Network for Equilibria and Chemical Thermodynamics Advanced Research", a European group of researchers focused on studies concerning thermodynamic equilibria. For example, through an

internal survey, it was possible to have an insight into the software commonly used by these researchers to calculate stability constants by the elaboration of data coming from different experimental techniques (e.g., potentiometry, spectrophotometry, NMR), as well as the most relevant issues they observe using it (Fig. 1): the Hyperquad suite (including, among others, Hyperquad, HypSpec, HypNMR) [1,2], SUPERQUAD [3], PSEQUAD [4], BSTAC [5], OPIUM [44].

Beside software required to process experimental data, another essential tool is that for the computation of the species concentration at equilibrium in solution and, eventually, for the drawing of the so-called *speciation diagrams*. In fact, equilibrium analysis and chemical speciation studies, though originally considered as a niche of analytical chemistry, are nowadays accounted as fundamental in many other fields of basic and applied research, such as bio-, medicinal and food chemistry, geochemistry and environmental science, to cite some. Among the most

* Corresponding author.

** Corresponding author.

E-mail addresses: silvia.berto@unito.it (S. Berto), dmilea@unime.it (D. Milea).

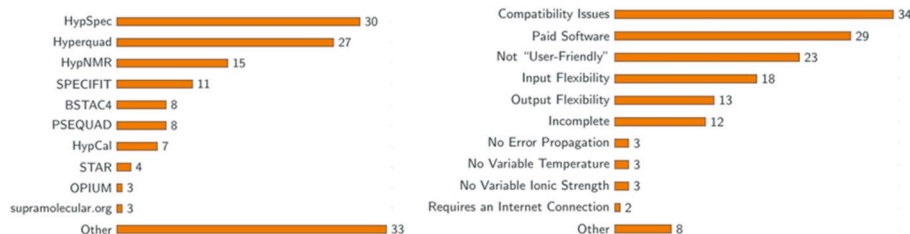


Fig. 1. Results of the survey conducted within NECTAR (COST ACTION CA18202) participants. Software commonly used (left) and issues found when using it (right).

used software, worth mentioning are: Hyperquad Simulation and Speciation (HySS) program [7], ES4ECI (ES4) [8], Medusa [45], PHREEQC [46], MINEQL+ [47].

Unfortunately, most of these programs show many limitations, and/or are not in an active development state anymore (some have already lost compatibility with current operating systems, a tendency that is not going to get better with time without any intervention). Moreover, the general dissatisfaction with currently available solutions suggests the need for a new open-source software, written by chemists for chemists, compatible with modern systems and available free of charge. This was exactly the purpose of the development of PyES, an open-source software for the computation of solution and precipitation equilibria and the simulation of titration curves.

For its peculiarities, the ES4 software [8] was used as baseline material for the writing of the new PyES program. ES4 is an open-source software for the simulation of titration curves and the calculation of concentration of species in solution at equilibrium, written in BASIC, by Prof. Silvio Sammartano from the University of Messina and his co-workers, between the 70s and the 80s. The software has two working modes: titration simulation and species distribution. While it is still being used by researchers that have been trained on it, it is difficult to recommend to new users due to the absence of a graphical user interface and the requirement of input of data in textual form, with a quite convoluted grammar prone to user errors. However, it still features quite interesting functionalities, currently not developed together by any other available open-source software. In particular, it can: i) estimate and take into account the variation of the ionic strength in the system, either as a function of pX ($pX = -\log [X]$, where X is the independent species, usually H^+) or during the titration process; ii) consider the error propagation to evaluate the uncertainties on the estimated concentrations of the species. Thus, using its primary logic as a starting point, the aim of this work was the creation of a new software, called PyES (Python Equilibrium Species), still keeping the above-mentioned peculiarities of the parent ES4 code, but compatible with modern operating systems, with a graphical user interface and some further improvements over the original software, to allow fast and reliable computation of the species concentration in a chemical system at thermodynamic equilibrium and to simulate titration curves. The intention was to provide a freely downloadable software that can then be used without the need for any internet connection, with its source code publicly available. Furthermore, PyES also shows a series of novelties in the field. It is the unique open-source software able to: i) simultaneously consider the effect of the ionic strength on the equilibrium (by adjusting the stability constants accordingly); ii) compute results in terms of their concentration and uncertainties for both in solution and precipitate species. Finally, and from a technical standpoint, PyES has been envisioned with a modular structure at its core, so having the possibility of further addition of extra functionalities in the future as, for example, the refinement of stability constants from analytical data. All this without the need for a complete overhaul of its codebase.

2. Materials and methods

2.1. Algorithm

The calculation of the concentration of species in solution at equilibrium can be described by an algorithm. The steps to be considered to reach the final result in a fast and reliable manner are summarized in the flow diagram in Scheme 1.

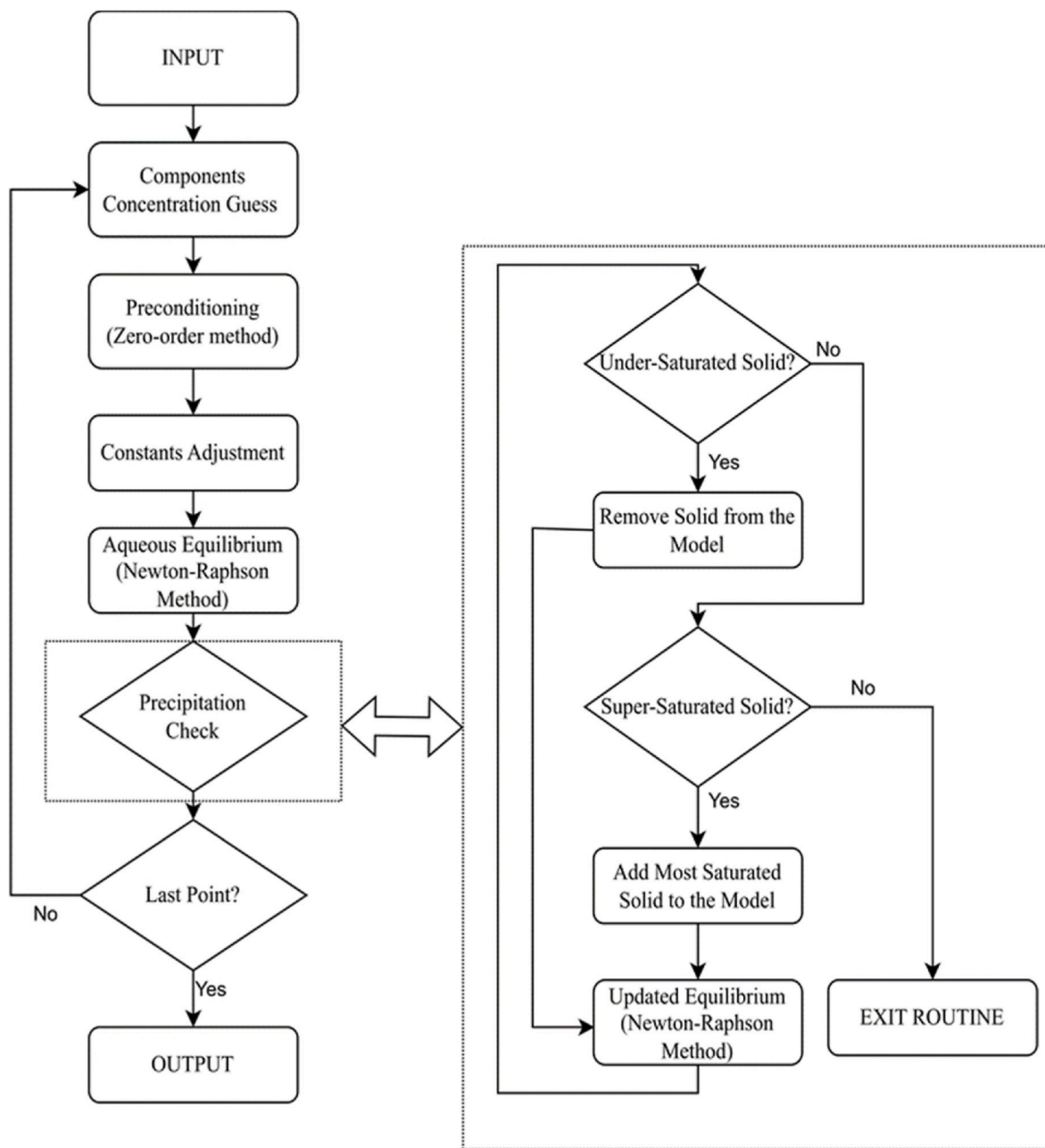
If the free concentration in solution of a given species or component is known (e.g., $[H^+]$, used as independent variable), that of other components and soluble species considered in the chemical model (whose stability constants are added as input) can be obtained from the analytical (total) concentrations of the components (added as input as well) by solving their mass balance equations for each value of the independent variable by an iterative method, such as Newton-Raphson. To enhance the stability of the algorithm, a preconditioning routine was implemented. It is applied to the initial guesses to obtain concentration values less far from the correct ones and that can be fed into the Newton-Raphson algorithm. Once the optimized values (by the Newton-Raphson algorithm) are obtained, the saturation conditions are verified for solid species, if present in the model (corresponding solubility products are added as input). If the system is in super-saturated conditions, the most saturated solid species is added to the model and the species concentrations are refined again. Once the concentration of the species is assessed, the uncertainty can be estimated. This process is iterative and will continue for each value of the independent variable until the convergence criteria are satisfied. The details of each step are reported in the next paragraphs.

2.2. Mathematical approach

The most common way to define a chemical species (and the corresponding equilibrium) in a solution in an easily programmable way is using Morel's table formalism, which has been popularized by the French scientist F. Morel [12]. According to Morel, any system in which a N_s number of chemical species S_i are present in solution, and eventually a N_p number of precipitable P_l species, can be represented as a linear combination of N_c number of C_j components. As an example, Table 1 shows Morel's table for a generic metal-ligand-proton system.

Each S_i species (N_s rows, one per species) is described by the stoichiometric coefficient of each C_j component (N_c columns, one per component). Therefore, the resulting matrix is defined by the $p_{i,j}$ elements corresponding to the value of the stoichiometric coefficient that the component C_j has in the species S_i . Morel's formalism alone gives a pure qualitative description of the composition of the system of interest. To be able to compute the concentration of every species at equilibrium, it is necessary to express its relationship with the concentration of each of its components by the overall stability constants (commonly denoted as β) and the solubility products (denoted as K_s).

In their generic form (for a metal-ligand-proton system):



Scheme 1. Flowchart of the logic implemented in PyES.

Table 1

Morel's table for a metal-ligand-proton system.

Species (S_i)	Components (C_j)		
	M^{m+}	L^{z-}	H^+
M^{m+}	1	0	0
L^{z-}	0	1	0
H^+	0	0	1
OH^-	0	0	-1
$M_p L_q H_r^{pm + r-qz}$	p	q	r

$$\beta = \frac{[M_p L_q H_r]}{[M]^p [L]^q [H]^r} \quad (1)$$

$$K_s = [M]^p [L]^q [H]^r \quad (2)$$

Moreover, each component C_j of the system must obey the Mass

Balance Equation (MBE):

$$[C_j]_T = \sum_i^{N_s} p_{ij} [S_i] + \sum_l^{N_p} b p_{lj} [P_l] \quad (3)$$

where $[C_j]_T$ represents the total concentration for each of the j^{th} component, that is the sum of the free concentration of the soluble $[S_i]$ and/or precipitate $[P_l]$ species containing that component, adjusted for its stoichiometric coefficient (p_{ij} or $b p_{lj}$ if referencing the i^{th} species in solution or the l^{th} in the precipitate, respectively). Please note that, in this formalism, the free concentration of the component $[C_j]$ is included in the sum of soluble species. Since the free concentration of a species in solution is related to the stability constant by:

$$[S_i] = \beta_i \prod_k^{N_C} [C_k]^{p_{ik}} \quad (4)$$

Eq. (3) becomes:

$$[C_j]_T = \sum_i^{N_S} p_{ij} \beta_i \prod_k^{N_C} [C_k]^{p_{ik}} + \sum_l^{N_P} b p_{lj} [P_l] \quad (5)$$

Unlike the case of the species in solution, it is not possible to use Eq. (4) to calculate the amount of solid species present at equilibrium conditions. The relation used to obtain this quantity is derived from the solubility product of the solid species, as defined in Eq. (2). At equilibrium, precipitate formation will be observed if:

$$\prod_j^{N_C} [C_j]^{b p_{lj}} > K_{S_l} \quad (6a)$$

i.e.,

$$\frac{\prod_j^{N_C} [C_j]^{b p_{lj}}}{K_{S_l}} \equiv S_{I_l} > 1 \quad (6b)$$

where S_{I_l} is the saturation index of the precipitate l .

Since all the species concentrations can be directly or indirectly obtained from the free concentrations of the components, the problem is reduced to finding which values are mathematical roots of the system of $N_C + N_P$ nonlinear equations, defined as:

$$Y_j = [C_j]_T - \sum_i^{N_S} p_{ij} \beta_i \prod_k^{N_C} [C_k]^{p_{ik}} - \sum_l^{N_P} b p_{lj} [P_l] \quad (7a)$$

$$Y_l = 1 - \frac{\prod_j^{N_C} [C_j]^{b p_{lj}}}{K_{S_l}} \quad (7b)$$

or, in the matrix form:

$$Y(\vec{X}) = 0 \quad (8)$$

where the vector \vec{X} includes both the $[C_j]$ and $[P_l]$ unknowns.

As mentioned before, an iterative Newton-Raphson method is used to solve this system. Being an iterative procedure, starting from an initial guess of \vec{X} , a better guess is obtained at the k^{th} iteration as:

$$\vec{X}_{k+1} = \vec{X}_k + \Delta \vec{X}_k \quad (9)$$

The applied shift $\Delta \vec{X}_k$ can be obtained by solving the system of equations:

$$-J_k \Delta \vec{X}_k = \vec{Y}_k \quad (10)$$

where the vector \vec{Y}_k is obtained as defined in Eqs. (7a) and (7b) and J_k is the Jacobian Matrix, a square matrix with $N_C + N_P$ dimension, and whose elements a_{ij} are equal to:

$$a_{ij} = \frac{\partial f_i}{\partial x_j} \quad (11)$$

in which f_i represents Eqs. (7a) and (7b), and x_j is the j^{th} unknown element of the vector \vec{X} .

The analytical form of these derivatives is known, so it is possible to calculate the value of a_{ij} at each iteration k .

$$a_{ij}^k = \sum_z^{N_S} p_{z,i} p_{z,j} \frac{[S_z]^k}{[C_j]^k} \quad i, j = 1, \dots, N_C \quad (12a)$$

$$a_{ij}^k = b p_{ij} \quad i = 1, \dots, N_C; j = N_C + 1, \dots, N_C + N_P \quad (12b)$$

$$a_{ij}^k = -b p_{ij} \frac{S_{I_l}^k}{[C_j]^k} \quad i = N_C, \dots, N_C + N_P; j = 1, \dots, N_C \quad (12c)$$

$$a_{ij}^k = 0 \quad i, j = N_C + 1, \dots, N_C + N_P \quad (12d)$$

Eq. (12a) gives the block of the Jacobian Matrix relative to the soluble species, while Eqs. (12b)-(12d) are related to the presence of precipitable species.

After k iterations, the solution is considered converged (and, consequently, the desired solution to the system of equations found) if the following equations are both satisfied:

$$\sum_j^{N_C} \left(\frac{Y_j}{[C_j]_T} \right)^2 \leq \varepsilon_1 \quad (13)$$

$$|Y_j| \leq \varepsilon_2 \quad j = N_C + 1, \dots, N_C + N_P \quad (14)$$

with $\varepsilon_1 = 1 \times 10^{-16}$ and $\varepsilon_2 = 1 \times 10^{-9}$.

To find the final desired solution we adopted Bethke's methodology [13] to define the final composition of the solid phase assemblage. In detail, no precipitation equilibrium is considered at the beginning. In these conditions, the system of Eq. (8) consists purely of the N_C nonlinear equations in the form of Eq. (7a). The result obtained can be considered a pre-equilibrium condition in which the saturation indexes of the solid species can be checked. A two-steps check is then performed: if any solid presents a $S_I < 1$ and positive concentration, it means it is dissolved and therefore removed from the model; on the contrary, if a solid exhibits $S_I > 1$ it is included in the model. In both cases, using the Newton-Raphson method, a new equilibrium condition is obtained, the final concentrations are updated, and another check is performed. This routine is repeated as long as no solid is in super-saturated conditions.

What we presented, so far, is generally valid for a single-point calculation, but some minor changes have to be introduced in the algorithm to effectively simulate titrations or species distributions for discrete values of one of the components. In the first case, the total concentration of the components must be adjusted at each point, to account for the added titrant volume:

$$[C_j]_T^n = \frac{[C_j]_T^0 \cdot v_0 + [C_j]_T^t \cdot v_n}{v_0 + v_n} \quad (15)$$

where $[C_j]_T^n$, $[C_j]_T^0$, and $[C_j]_T^t$ are the total concentration of the j^{th} component at the n^{th} point of the titration, at the beginning of the titration and in the titrant, respectively; v_0 and v_n are the initial volume and that of the titrant added at the n^{th} point.

2.3. Method robustness

One of the main limitations of the Newton-Raphson technique applied to solve systems of nonlinear equations lies in the necessity of initial guesses that are close enough to the desired roots to avoid divergence problems [14]. For this reason, three strategies have been implemented.

2.3.1. Strictly positive concentration

Working under the assumption that the species concentrations can be obtained from those of the components, it is clear that the latter can be close to zero but never actually be zero. To enforce this behavior we adopted a technique proposed by Bethke [13]: at each Newton-Raphson iteration, the concentration of components is updated, using a slightly modified Eq. (9):

$$\vec{X}_{k+1} = \vec{X}_k + \delta_{relax} \cdot \Delta \vec{X}_k \quad (16)$$

where:

$$\frac{1}{\delta_{relax}} = \max \left(1, -\frac{\Delta \bar{X}_k}{0.5 \bar{X}_k} \right) \quad (17)$$

2.3.2. Extrapolation from previous points

Subsequent points in a simulated titration are “equispaced”: the amount of titrant added is constant at each point throughout the titration. Similarly, for a species distribution, the concentration of the independent component varies linearly within the investigated range. Therefore, it is possible to extrapolate a better guess of component concentrations from the solution of previous points. For the titration case, the equation used is the following:

$$l_{j,n} = l_{j,n-1} + \left(\frac{l_{j,n-1} - l_{j,n-2}}{v_{n-1} - v_{n-2}} \right)^2 \left(\frac{v_{n-2} - v_{n-3}}{l_{j,n-2} - l_{j,n-3}} \right) (v_n - v_{n-1}) \quad (18)$$

while, in the case of the species distribution, the following equation is used instead:

$$l_{j,n} = l_{j,n-1} + \frac{(l_{j,n-1} - l_{j,n-2})^2}{l_{j,n-2} - l_{j,n-3}} \quad (19)$$

In both equations, the $l_{j,n}$ terms represent the \log_{10} of the concentrations at the n^{th} point. The main difference between the two lies in the fact that, for the titration, the effect of dilution is kept into account.

From Eqs. (18) and (19) it comes out that this procedure can be applied from the fourth point on. As such, the concentration at the first point is conventionally set as 1/1000 of the total concentrations defined as input by the user. For the following two points (i.e., the second and the third), the results of the previous points are used as the best initial guess.

2.3.3. Preconditioning

The starting guesses, in particular those of the first points, which usually are of lower quality, may not be good enough to assure convergence of the Newton-Raphson method alone. To further enhance the stability of the algorithm we implemented a preconditioning routine, applied to the initial guesses obtained from the extrapolation process, before feeding them into the Newton-Raphson algorithm, called “Positive Continuous Fraction Method” (PCFM). Originally proposed in a simpler form by Carayrou et al. [14] and improved by Marinoni et al. [15], PCFM is a zero-order method, that is an iterative routine, like the Newton-Raphson method, which does not require the computation of derivatives at each iteration, resulting in a less computationally intense process, that rapidly reaches a guess in the neighborhood of the searched solution. The algorithm requires the definition of two new quantities for each component – reactant sum Sum_j^{react} and product sum Sum_j^{prod} – whose definition is different and depends on the sign of the total concentration for the component in question. If $[C_j]_T \geq 0$:

$$Sum_j^{react} = \sum_{p_{i,j} > 0} p_{i,j} [S_i] \quad (20a)$$

$$Sum_j^{prod} = [C_j]_T + \sum_{p_{i,j} < 0} |p_{i,j}| [S_i] \quad (20b)$$

while, if $[C_j]_T < 0$:

$$Sum_j^{react} = |[C_j]_T| + \sum_{p_{i,j} > 0} p_{i,j} [S_i] \quad (21a)$$

$$Sum_j^{prod} = \sum_{p_{i,j} < 0} |p_{i,j}| [S_i] \quad (21b)$$

Accordingly, the MBE (Eq. (3)) can be rewritten utilizing these new quantities, so that, at equilibrium:

$$Sum_j^{prod} = Sum_j^{react} \quad (22)$$

As such, it is possible to iteratively obtain a better approximation of

the free concentration of the j^{th} component as:

$$[C_j]^{k+1} = [C_j]^k \left(\frac{Sum_j^{prod,k}}{Sum_j^{react,k}} \right)^{\frac{1}{p_{0j}}} \quad (23)$$

where p_{0j} is the smallest strictly positive stoichiometric coefficient that the j^{th} component presents in Morel's Table. The convergence condition used to terminate the iterative procedure is defined as:

$$\frac{|Sum_j^{react} - Sum_j^{prod}|}{Sum_j^{react} + Sum_j^{prod}} \leq \varepsilon_3 \quad j = 1, \dots, N_C \quad (24)$$

where $\varepsilon_3 = 1 \times 10^{-9}$ for the first point and $\varepsilon_3 = 0.25$ for each subsequent point.

If this procedure had to be applied for all the components simultaneously, it would introduce unfavorable fluctuations in successive iterations, possibly causing divergence problems. However, it is possible to attenuate the effect of the applied shift, trading off a minor risk of diverging with a slower process overall:

$$[C_j]^{k+1} = \theta_j [C_j]^k \left(\frac{Sum_j^{prod,k}}{Sum_j^{react,k}} \right)^{\frac{1}{p_{0j}}} + (\theta_j - 1) [C_j]^k \quad (25)$$

In the previous equation, θ can assume values between 0 and 1. For values close to 1, no attenuation is applied, and Eq. (25) tends to Eq. (23). Instead, the closer to 0 θ is, the more intense the attenuation will be, resulting in a slower convergence. Ideally, this routine would return concentration values close to those at equilibrium, as fast as possible, while minimizing the risk of diverging due to the aforementioned fluctuations. To achieve this, we implemented the same solution proposed by Marinoni et al. [15], consisting in a dynamic update of the value θ , using the equations:

$$Sum_j^{react} > Sum_j^{prod} \quad \theta_j = 0.9 - 0.8 \frac{Sum_j^{prod}}{Sum_j^{react}} \quad (26a)$$

$$Sum_j^{prod} > Sum_j^{react} \quad \theta_j = 0.9 - 0.8 \frac{Sum_j^{react}}{Sum_j^{prod}} \quad (26b)$$

To further speedup the routine, θ is updated only if, at the k^{th} step:

$$\theta_j^k > \theta_j^{k-1} \quad (27)$$

This condition ensures that, if the method is taking too long to reach convergence (i.e., the applied attenuations are too strong), the next iterations will apply more aggressive shifts, speeding up the procedure.

2.4. Ionic strength dependence

Stability constants reported in main collections and databases [16–22], as well as single papers, are mainly stoichiometric constants, i.e., they are expressed in terms of concentrations instead of activities. For a generic equilibrium



the stoichiometric stability constant is given by:

$$\beta = \frac{[C]^c [D]^d}{[A]^a [B]^b} \quad (29)$$

Being the activity of a generic species S given by

$$a_S = \gamma_S [S] \quad (30)$$

in which γ_S is its activity coefficient, it turns out that the stoichiometric constant β is related to the thermodynamic constant β^T by:

$$\beta^T = \frac{a_C^c a_D^d}{a_A^a a_B^b} = \frac{\gamma_C^c \gamma_D^d}{\gamma_A^a \gamma_B^b} \frac{[C]^c [D]^d}{[A]^a [B]^b} = \frac{\gamma_C^c \gamma_D^d}{\gamma_A^a \gamma_B^b} \beta \quad (31)$$

As such, since the activity coefficients depend on the system conditions (e.g., ionic strength, temperature, pressure), the same holds for the stoichiometric stability constants (from now on generally denoted as stability constants).

Several models (for some details see, e.g., Ref. [23]) exist to take into account the dependence of activity coefficients on system conditions, with particular reference to ionic strength and temperature (since most reactions occur at ambient pressure, dependence on the latter assumes relevance in just particular cases like, e.g., in geochemical studies). However, the most common theories and approaches (e.g., Davies [24], Bromley [25], Pitzer [26,27], Specific Ion Interaction Theory – SIT [23, 28]) can be considered as evolutions of the (Extended) Debye-Hückel equation for the dependence of activity coefficients on ionic strength (I):

$$\log \gamma = -z^2 A \frac{\sqrt{I}}{1 + B\sqrt{I}} + L(I) \quad (32)$$

in which the $L(I)$ term differs from one equation to another. The Extended Debye-Hückel equation (from now on denoted as EDH) used in PyES and already implemented in ES4 is given by:

$$\log \gamma = -z^2 A \frac{\sqrt{I}}{1 + B\sqrt{I}} + CI + DI^3 + EI^2 \quad (33)$$

where A and B are the so-called Debye-Hückel parameters ($A = 0.51$ and $B = 1.5$ at $T = 298.15$ K), and C , D and E are empirical parameters (the E term can generally be neglected for $I \leq 1.0$ mol dm⁻³). This equation can be applied to calculate a stability constant (or a solubility product) at a desired ionic strength, if this constant is known at another I value (including that at infinite dilution, i.e., β^T when $I = 0$):

$$\log \beta^{I_2} = \log \beta^{I_1} - z^* A \left(\frac{\sqrt{I_2}}{1 + B\sqrt{I_2}} - \frac{\sqrt{I_1}}{1 + B\sqrt{I_1}} \right) + C(I_2 - I_1) + D \left(I_2^3 - I_1^3 \right) + E(I_2^2 - I_1^2) \quad (34)$$

where 1 and 2 subindices refer to two different ionic strength values. PyES, as its precursor ES4, allows the input of A and B terms in common for all species of the model, and the input of C , D and, eventually, E for each single species. However, based on the vast experience of some of the developers of ES4 and their coworkers, between the 80's and 90's it has been proposed a model [29], with the relative equation, for the ionic strength dependence of formation constants, in which the three terms are split into two parameters per term, depending on both the stoichiometric coefficients (p) and the charges (z) of species involved in the considered equilibrium:

$$C = c_0 p^* + c_1 z^* \quad (35a)$$

$$D = d_0 p^* + d_1 z^* \quad (35b)$$

$$E = e_0 p^* + e_1 z^* \quad (35c)$$

with z^* and p^* defined as:

$$z^* = \sum z_{\text{reac}}^2 - \sum z_{\text{prod}}^2 \quad (36a)$$

$$p^* = \sum p_{\text{reac}}^2 - \sum p_{\text{prod}}^2 \quad (36b)$$

with *reac* and *prod* indexes referred to reactants and products. The proposed model is based on three main assumptions:

[H1]. It is possible to express the dependence on ionic strength of formation constants by a simple equation, independently of the type of reactants and products, and dependent on the type of reaction only.

Table 2

Parameters of EDH equation (Eqs. (34)–(36)), and their dependence on temperature, valid at $I \leq 1.0$ mol dm⁻³ and $T \leq 318.15$ K^{a)}.

$A = 0.5115 + 9.123 \times 10^{-4} (T - 298.15) + 4.93 \times 10^{-6} (T - 298.15)^2$
$B = 1.5000 + 8.900 \times 10^{-4} (T - 298.15) + 4.195 \times 10^{-6} (T - 298.15)^2$
$c_0 = 0.10 - 3.7 \times 10^{-3} (T - 298.15)$
$c_1 = 0.2095 - 5.4 \times 10^{-4} (T - 298.15)$
$d_1 = -0.0935 + 9.5 \times 10^{-4} (T - 298.15)$

a) In those conditions, $d_0 = e_0 = e_1 = 0$, from ref. [29].

[H2]. All the deviations from the predicted behavior are ascribed to weak complex formation between components and/or species under study and the background ions (e.g., the ionic medium). This implies that pure water is considered as reference state, and some ions as non-interacting with the reactants and/or products involved in the studied equilibrium.

[H3]. Perchlorate does not interact with cationic species, tetraethylammonium cations (and higher tetraalkylammonium analogues) do not with O-donor ligands, and Na⁺ and K⁺ do not with N-donor ligands.

Several evidences collected during more than half a century of investigation on equilibria in solution mainly performed at the Italian Universities of Messina, Torino and Catania (from first papers specifically dedicated to this topic [30,31], to more recent, like, e.g., refs. [6,9, 10,32,33]) demonstrated that the model's assumptions were correct, so that, in the conditions of [H3] (i.e., in non-interacting media) the dependence of the formation constants on ionic strength fulfills [H1] and parameters c_0 , c_1 , d_0 , d_1 , e_0 and e_1 become constant (thus, eq. (34) only depends on p^* and z^*). Their values, together with A and B and their temperature gradients, valid at $I \leq 1.0$ mol dm⁻³ and $T \leq 318.15$ K, are reported in Table 2 [29].

For the above reasons, instead of considering the input of C , D and E for single species, PyES also allows the input of c_0 , c_1 , d_0 , d_1 , e_0 and e_1 in common for all species of the model, in addition to A and B . This is particularly useful when models are built considering stability constants determined in non-interacting media, together with those relative to the formation of weak complexes, allowing calculations in many different ionic media, including multicomponent systems (see, e.g., efs. [11,34], and [35,36] for synthetic seawater, synthetic saliva and urine, respectively).

2.5. Uncertainty propagation

When dealing with real data, it is more and more often required to give the obtained results together with their uncertainty. This functionality was present in ES4 for the soluble species (as reported in the previous work by De Robertis et al. [8]) and it has been extended to precipitates in PyES.

For each component and species present in the system, it is possible to obtain the uncertainty on their concentrations, using the laws of error propagation. This only if the uncertainties relative to the input data, which are those associated with one among or both the analytical concentration of components and the stability constants, are known:

$$\sigma_{[C]}^2 = \sum_i^{N_S} \left(\frac{\partial [C]}{\partial \beta_i} \right)^2 \sigma_{\beta_i}^2 + \sum_r^{N_C} \left(\frac{\partial [C]}{\partial [C_r]_T} \right)^2 \sigma_{[C_r]_T}^2 + \sum_k^{N_P} \left(\frac{\partial [C]}{\partial K_{S_k}} \right)^2 \sigma_{K_{S_k}}^2 \quad (37a)$$

$$\sigma_{[P]}^2 = \sum_i^{N_S} \left(\frac{\partial [P]}{\partial \beta_i} \right)^2 \sigma_{\beta_i}^2 + \sum_r^{N_C} \left(\frac{\partial [P]}{\partial [C_r]_T} \right)^2 \sigma_{[C_r]_T}^2 + \sum_k^{N_P} \left(\frac{\partial [P]}{\partial K_{S_k}} \right)^2 \sigma_{K_{S_k}}^2 \quad (37b)$$

$$\sigma_{[S]}^2 = \sum_i^{N_S} \left(\frac{\partial [S]}{\partial \beta_i} \right)^2 \sigma_{\beta_i}^2 + \sum_r^{N_C} \left(\frac{\partial [S]}{\partial [C_r]_T} \right)^2 \sigma_{[C_r]_T}^2 + \sum_k^{N_P} \left(\frac{\partial [S]}{\partial K_{S_k}} \right)^2 \sigma_{K_{S_k}}^2 \quad (37c)$$

The derivatives present in these equations can be obtained deriving equations relative to the MBE for the system, with respect to the stability constants of the species, and total concentrations of components:

$$\frac{\partial [C_j]_T}{\partial \beta_i} = \frac{\partial [C_j]}{\partial \beta_i} + \sum_r^{N_C} \frac{\partial [C_r]_T}{\partial \beta_i} \sum_l^{N_S} \left(p_{l,r} p_{l,j} \frac{[S_l]}{[C_r]} \right) + p_{i,j} \frac{[S_i]}{b_{i,j}} + \sum_k^{N_P} b p_{k,j} \frac{\partial [P_k]}{\partial \beta_i} \quad (38a)$$

$$\frac{\partial [C_j]_T}{\partial [C_r]_T} = \frac{\partial [C_j]}{\partial [C_r]_T} + \sum_k^{N_C} \frac{\partial x_k}{\partial [C_r]_T} \sum_l^{N_S} \left(p_{l,r} p_{l,j} \frac{[S_l]}{[C_k]} \right) + \sum_l^{N_P} b p_{l,j} \frac{\partial [P_l]}{\partial [C_r]_T} \quad (38b)$$

$$\frac{\partial [C_j]_T}{\partial K s_r} = \frac{\partial [C_j]}{\partial K s_r} + \sum_k^{N_C} \frac{\partial [C_k]_T}{\partial K s_r} \sum_l^{N_S} \left(p_{l,r} p_{l,j} \frac{[S_l]}{[C_k]} \right) + \sum_l^{N_P} b p_{l,j} \frac{\partial [P_l]}{\partial K s_r} \quad (38c)$$

Because $\frac{\partial [C_j]_T}{\partial \beta_i} = 0$, $\frac{\partial [C_j]_T}{\partial [C_r]_T} = \delta_{j,r}$ and $\frac{\partial [C_j]_T}{\partial K s_r} = 0$, it is possible to obtain the derivatives of the free concentration of components with respect to the stability constants and total concentrations by solving a series of systems of equations, obtained by re-arrangement of the previous ones. However, using these equations alone provides indeterminate systems (N_C equations in $N_C + N_P$ unknowns). To balance them, similarly to what was done to achieve the same result in the computation of the Jacobian matrix for the Newton-Raphson technique, a new set of equations is obtained for each one, deriving the saturation index equations of the solid species present with respect to β_i , $[C_r]_T$ and $K s_r$:

$$\frac{\partial S I_j}{\partial \beta_i} = \frac{1}{K s_j} \left(\sum_k^{N_C} b p_{j,k} \frac{S I_j}{[C_k]} \frac{\partial [C_k]}{\partial \beta_i} \right) \quad (39a)$$

$$\frac{\partial S I_j}{\partial [C_r]_T} = \frac{1}{K s_j} \left(\sum_k^{N_C} b p_{j,k} \frac{S I_j}{[C_k]_T} \frac{\partial [C_k]}{\partial [C_r]_T} \right) \quad (39b)$$

$$\frac{\partial S I_j}{\partial K s_r} = \frac{1}{K s_j} \left(-S I_j \delta_{j,r} + \sum_k^{N_C} b p_{j,k} \frac{S I_j}{[C_k]} \frac{\partial [C_k]}{\partial K s_r} \right) \quad (39c)$$

In the end, the result is $i = N_S - N_C$ systems of $AX = B$ type, with:

$$b_j = -p_{i,j} \frac{[S_i]}{\beta_i} \quad j = 1, \dots, N_C \quad (40a)$$

$$b_j = 0 \quad j = N_C + 1, \dots, N_C + N_P$$

$$x_j = \frac{\partial [C_j]}{\partial \beta_i} \quad j = 1, \dots, N_C \quad (40b)$$

$$x_j = \frac{\partial [P_j]}{\partial \beta_i} \quad j = N_C + 1, \dots, N_C + N_P$$

$r = N_C$ systems $AY = D$ with:

$$d_j = \delta_{j,r} \quad j = 1, \dots, N_C \quad (41a)$$

$$d_j = 0 \quad j = N_C + 1, \dots, N_C + N_P$$

$$y_j = \frac{\partial [C_j]}{\partial [C_r]_T} \quad j = 1, \dots, N_C \quad (41b)$$

$$y_j = \frac{\partial [P_j]}{\partial [C_r]_T} \quad j = N_C + 1, \dots, N_C + N_P$$

and $l = N_P$ systems $AZ = E$:

$$e_j = 0 \quad j = 1, \dots, N_C \quad (42a)$$

$$e_j = \delta_{j,l} \frac{S I_j}{K s_l} \quad j = N_C + 1, \dots, N_C + N_P$$

$$z_j = \frac{\partial [C_j]}{\partial K s_l} \quad j = 1, \dots, N_C \quad (42b)$$

$$z_j = \frac{\partial [P_j]}{\partial K s_r} \quad j = N_C + 1, \dots, N_C + N_P$$

having all in common the same matrix of coefficients A , defined as:

$$a_{i,j} = \sum_l^{N_S} p_{l,i} p_{l,j} \frac{[S_l]}{[C_j]} + \delta_{i,j} \quad i, j = 1, \dots, N_C \quad (43a)$$

$$a_{i,j} = b p_{i,j} \quad i = 1, \dots, N_C; j = N_C + 1, \dots, N_C + N_P \quad (43b)$$

$$a_{i,j} = b p_{i,j} \frac{S I_i}{[C_j]} \quad i = N_C + 1, \dots, N_C + N_P; j = 1, \dots, N_C \quad (43c)$$

$$a_{i,j} = 0 \quad i, j = N_C + 1, \dots, N_C + N_P \quad (43d)$$

Using the obtained derivatives for the different components, it is possible to compute the derivatives for the species with respect to β_i and $[C_r]_T$:

$$\frac{\partial [S_i]}{\partial \beta_i} = \delta_{i,i} \frac{[S_i]}{\beta_i} + \sum_j^{N_C} p_{i,j} \frac{[S_i]}{[C_j]} \frac{\partial [C_j]}{\partial \beta_i} \quad (44a)$$

$$\frac{\partial [S_i]}{\partial [C_r]_T} = \sum_j^{N_C} p_{i,j} \frac{[S_i]}{[C_j]} \frac{\partial [C_j]}{\partial [C_r]_T} \quad (44b)$$

$$\frac{\partial [S_i]}{\partial K s_r} = \sum_j^{N_C} p_{i,j} \frac{[S_i]}{[C_j]} \frac{\partial [C_j]}{\partial K s_r} \quad (44c)$$

2.6. Libraries used

From a technical point of view, the software has been developed in the Python programming language, due to its large diffusion in the scientific community, its ease of writing and its cross-platform nature. Even if the produced code results in applications with slower execution times compared to the compiled code counterparts (such as C or BASIC), the use of highly optimized Python libraries, such as Pandas [37] and NumPy [38], allowed to obtain considerable speed-ups that, together with the ease the agile development allowed by Python itself, made it an obvious choice.

The graphical user interface has been implemented using the Qt framework through the use of its Python binding in the form of the PySide6 library [39]. In addition to the above-mentioned Pandas and NumPy libraries, used to manage tabular data and speed up the algorithms, PyQtGraph [40] was used to give the application the ability to chart the results obtained.

Finally, to package the application to allow its installation and execution on machines that do not necessarily have Python installed we used PyInstaller [41].

3. Results and discussion

3.1. Software interface

The software interface is composed by three tabs. Fig. 2a shows the first ("Species"), i.e. that presented to the user at launch. There, it is possible to define the chemical system that is going to be simulated in terms of components and species present, using the two corresponding tables (see Fig. 2a). In addition, using the fields in "Options" (on the left), it is possible to decide whether to take into account the contribution of the ionic strength and if errors arising from the input data uncertainties have to be considered. Selecting one of these options will result in the activation, in the table dedicated to the definition of the various species, of the corresponding fields where the values used in the EDH equation and/or the uncertainties (expressed as standard deviation of the given value) can be defined. These values can both be defined as a

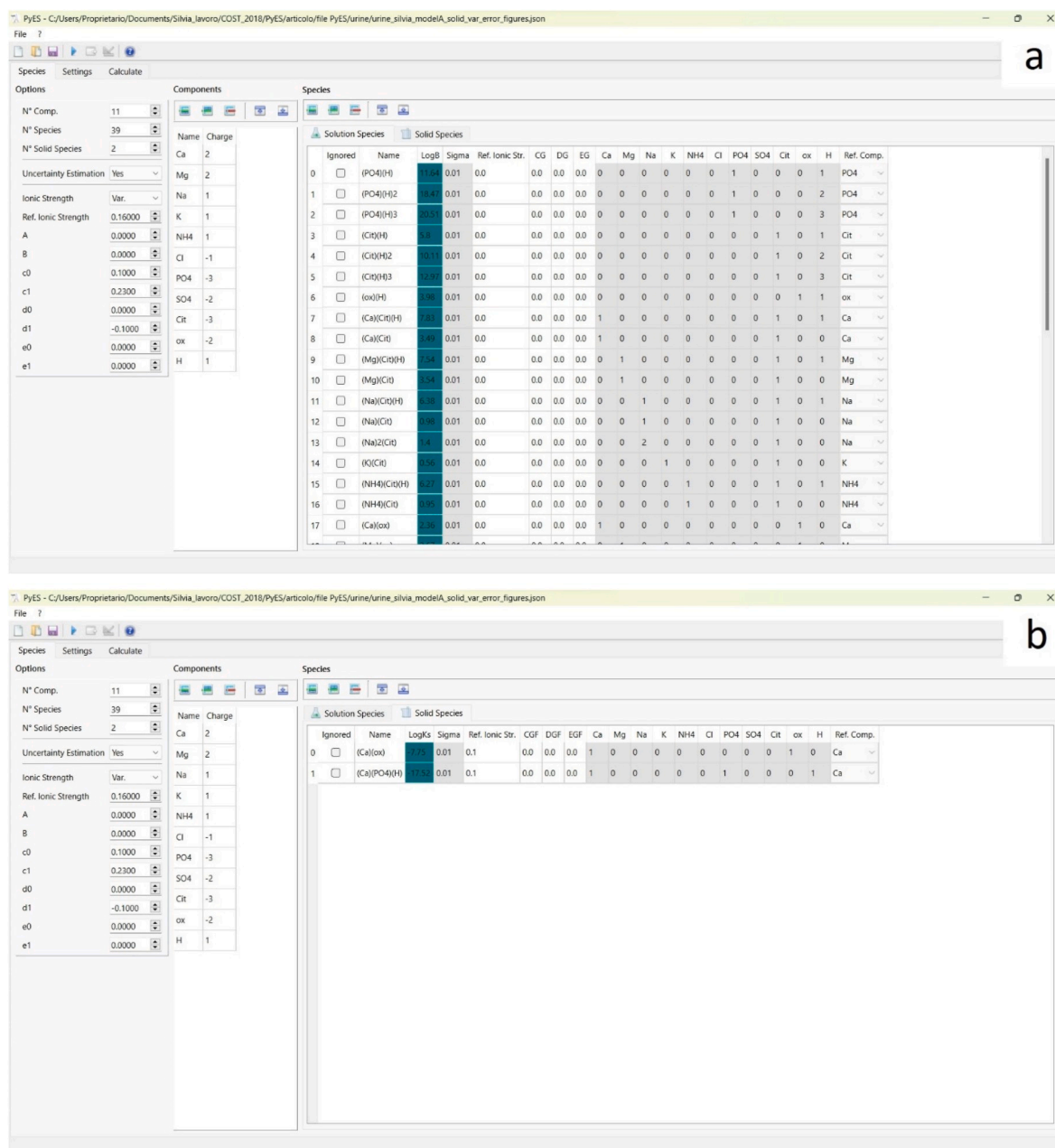


Fig. 2. “Species” tab of PyES’s interface: a) table to define the species in solution; b) table to define the solid species.

project or on a *per species* basis.

An analogous table to that used to define the species in solution is present in a sub-tab to define any solid species that could be present, with the only difference given by the definition of a solubility product instead of a formation constant (see Fig. 2b).

In the second tab (Fig. 3), it is possible to select the desired working mode, i.e., titration simulation (Fig. 3a) or species distribution (Fig. 3b), and input the corresponding required values. In detail, if we intend to simulate a titration, the analytical concentration of each component in both the titration vessel and the titrant is required. In the case of a species distribution, only the analytical concentration for each component is requested. For a titration, it is further required the input of: i) the initial volume, ii) the amount of titrant added at each point, and iii) the number of points to simulate. For a species distribution, the user has to define: i) which component has to be considered independent, ii) in which range of its concentrations the distribution has to be computed, and iii) the number of points to simulate in the range. Additionally, in

both cases, if uncertainties have to be calculated, the uncertainty relative to each component concentration has to be provided.

If it has been chosen in the first tab to consider the effect of the ionic strength, it is necessary to define the concentration of the background ions (if any), which is the sum of the concentrations of all the ions that do not take part in any of the considered equilibria, but for which their contribution to the ionic strength cannot be neglected.

The third and last tab (“Calculate”, Fig. 4) is where the calculation routine can be launched and where the results, and eventually the errors that could arise, will be presented to the user.

Below the result prompt, it is possible to access the “Export Results” menu in which the user can choose the desired format for exporting the results (Fig. 5a), and the chart (in “Plot Results” option, Fig. 5b).

3.2. Example datasets and benchmarks

Various tests were performed to verify the coherence between the

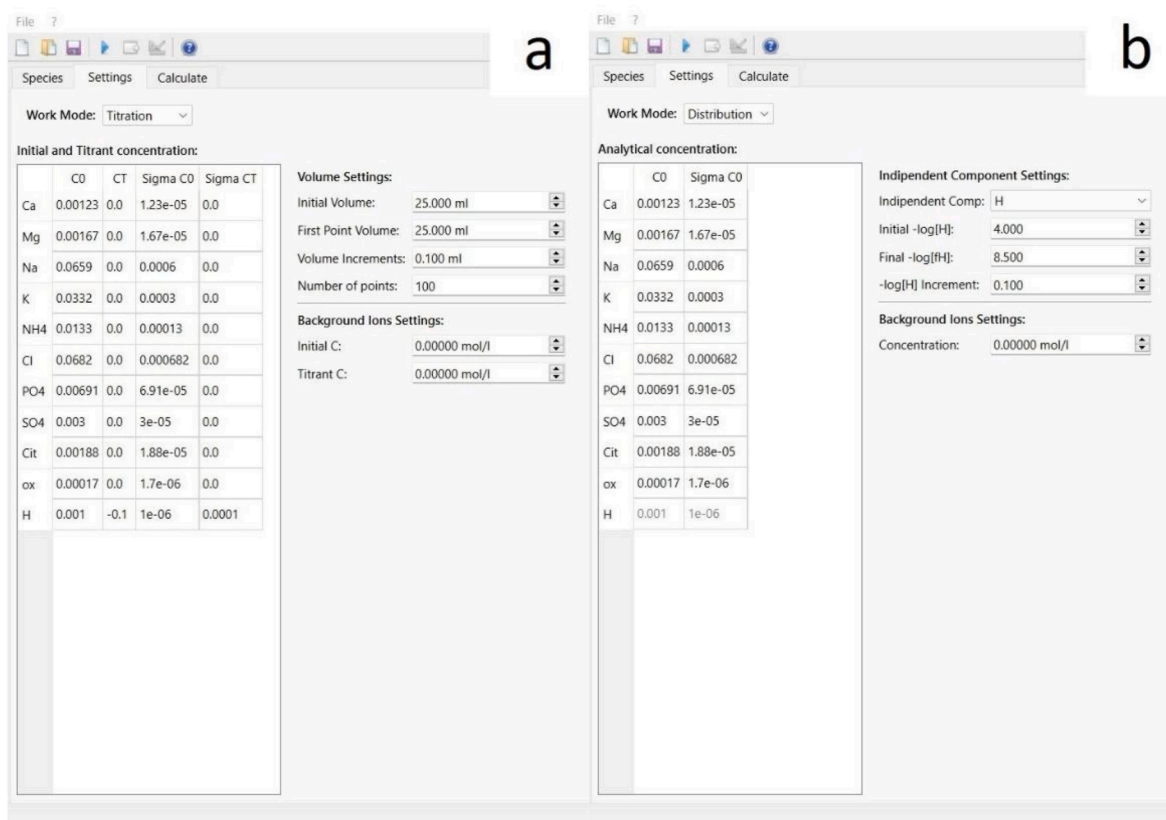
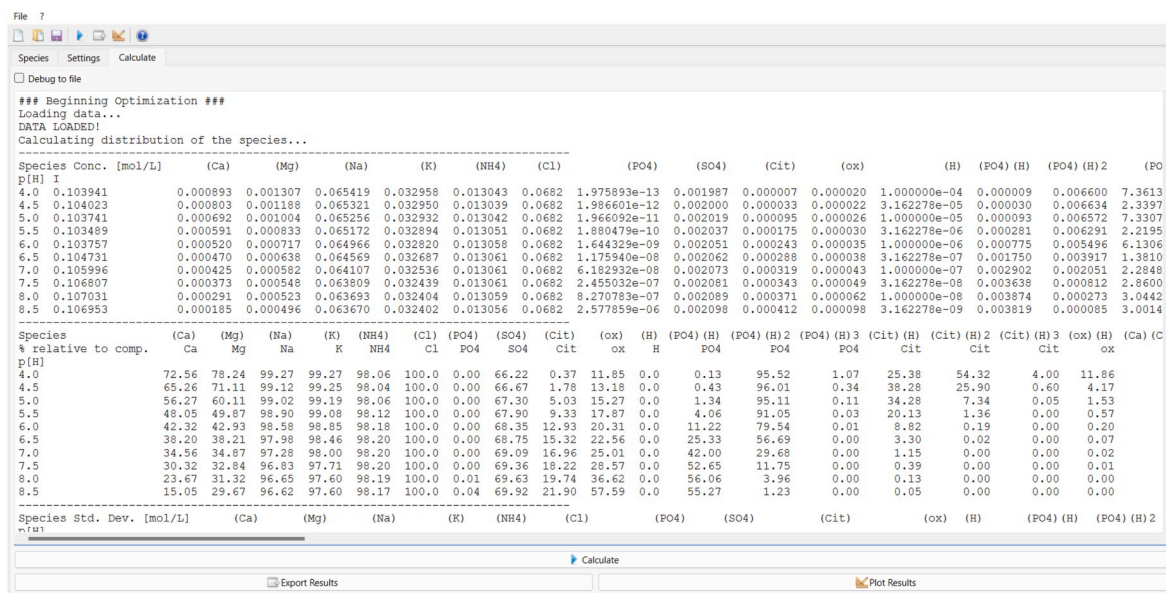


Fig. 3. "Settings" tab of PyES's interface: a) titration simulation; b) species distribution.

results obtained by PyES and those obtained by the well tested ES4 [8], the parent software, and HySS [7], the latter probably being the most widely used and currently available software for these purposes. The chemical systems used as test, whose results are presented in this work are: 1) a monoprotic acid, 2) the phosphoric acid [42], 3) a hypothetical hexaprotic acid, 4) the Cu-glycine system from IUPAC [43], and a chemical model simulating real urine samples [36]. Only the most remarkable comparisons are reported in the main text, but all the outcomes are comprehensively presented as Supplementary Material.

The performances of PyES on the calculation of the species distribution diagrams were tested on different systems, but the comparison between results obtained by PyES and those recently published by Berto et al. [36] for chemical models able to predict the formation of kidney stones from urine analysis proved to be particularly valuable. The relevance of this comparison lies in two main facts: 1) in the aforementioned work, the results were experimentally obtained, and the species distribution were computed by the well validated HySS software [7]; 2) the complexity of the speciation model is particularly complex. In



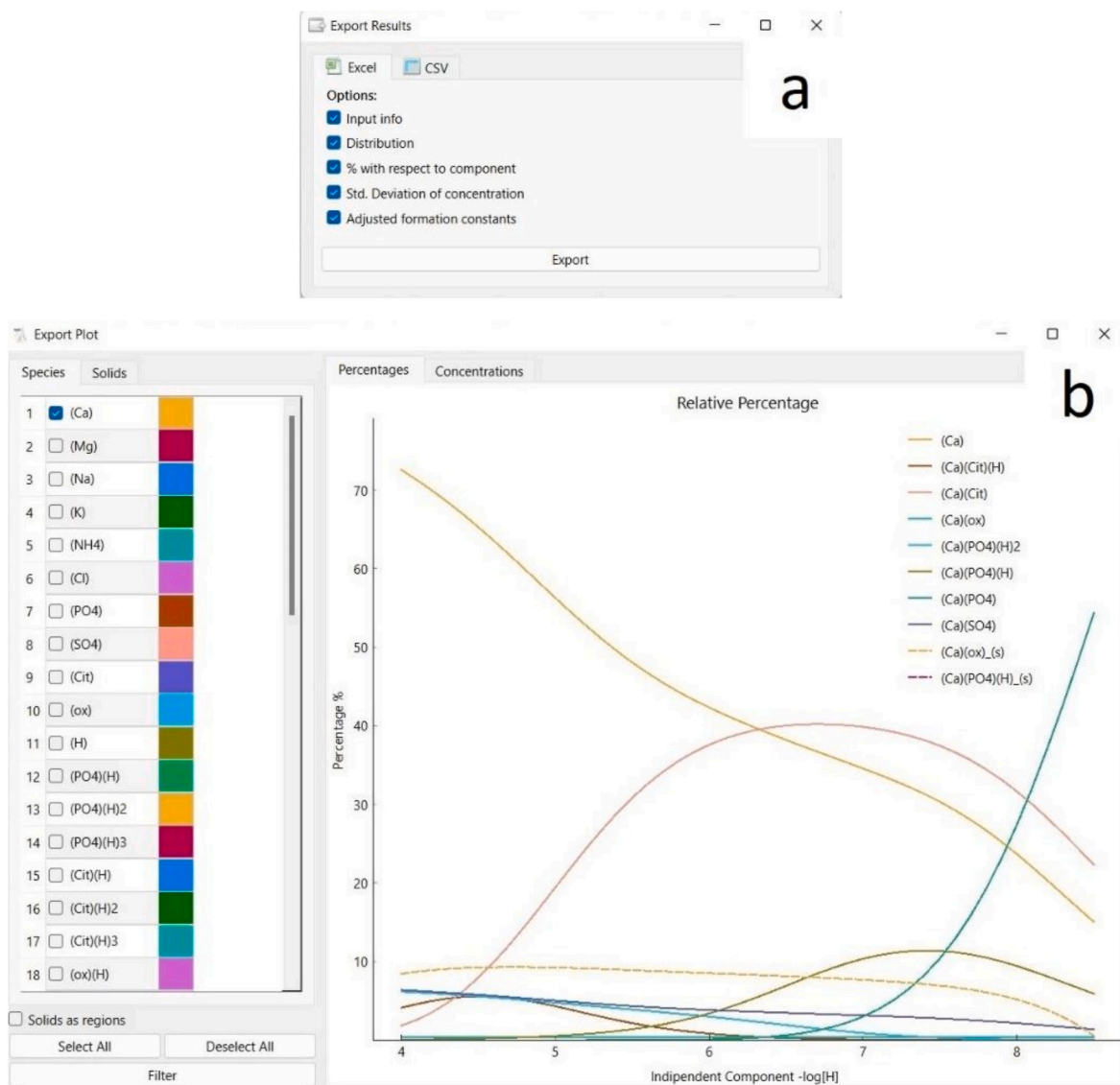


Fig. 5. a) "Export Results" menu of PyES's interface; b) chart obtained by "Plot Results" option.

fact, concerning the latter, the considered chemical model consists of 50 species (of which 11 are components) and 2 solid species. The model takes into account the equilibrium of various anionic systems (phosphate, citrate, oxalate, sulfate) and their interaction with calcium, magnesium, sodium, potassium and ammonium. The comparison between the results obtained by the two software for one of the samples referenced in the original article is presented in Fig. 6.

The data processing was carried out working at fixed ionic strength and neglecting the errors on concentration and formation constants values, consistently with what is allowed by HySS, which has the limitation that it cannot manage these input data. The obtained results appear to be independent of the used software.

The performances of PyES on the titration simulation were tested on three chemical systems containing acidic components with a different number of protogenic groups: a generic monoprotic acid ($c = 1 \times 10^{-2} \text{ mol dm}^{-3}$); a solution of phosphoric acid ($c = 1 \times 10^{-3} \text{ mol dm}^{-3}$) and a hypothetical hexaprotic acid ($c = 1 \times 10^{-3} \text{ mol dm}^{-3}$). Furthermore, the titration simulation function was tested on a solution of Cu^{2+} ($c_{\text{Cu}} = 1 \times 10^{-3} \text{ mol dm}^{-3}$) and glycine ($c_{\text{gly}} = 2 \times 10^{-3} \text{ mol dm}^{-3}$). (All details relative to the chemical models are reported in Supplementary Material, Table 1S and 3S). The results obtained by PyES were compared with those obtained by both ES4 and HySS (Fig. 7) and showed a very good

agreement between the three different computations for all the tested chemical systems. The titration curves obtained overlap perfectly.

The percentage differences between the concentration values estimated by PyES and by the other two software are generally lower than 0.5% for both titrations and species distribution simulations (see details of the Supplementary Material, Table 2S and 4S).

4. Conclusions

The PyES software here presented, besides being an updated version of the parent software ES4, compatible with actual operative systems and easy to use, is a product with interesting new functionalities. It calculates the concentration of species in solution at equilibrium, with virtually no limit to the number of species and/or components. It can manage the variation of the ionic strength of the system, and differently from HySS, can calculate the concentrations of the different species using the stability constants corrected for the change of the ionic strength of the system. Moreover, by improving ES4, it can manage the presence of solid species performing the error propagation to estimate the uncertainties of the species concentrations starting from the uncertainties of stability constants, solubility products and components concentration. Comparison with other tested software suites proved

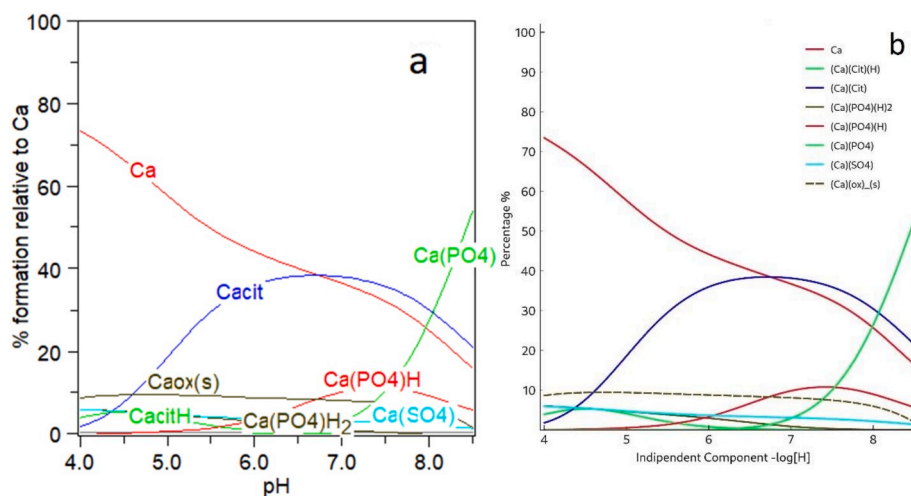


Fig. 6. Species distribution diagrams directly obtained by a) HySS and b) PyES for the same chemical system, simulating a real urine (chemical model and components concentration values are reported in the ref. Berto et al. [36]).

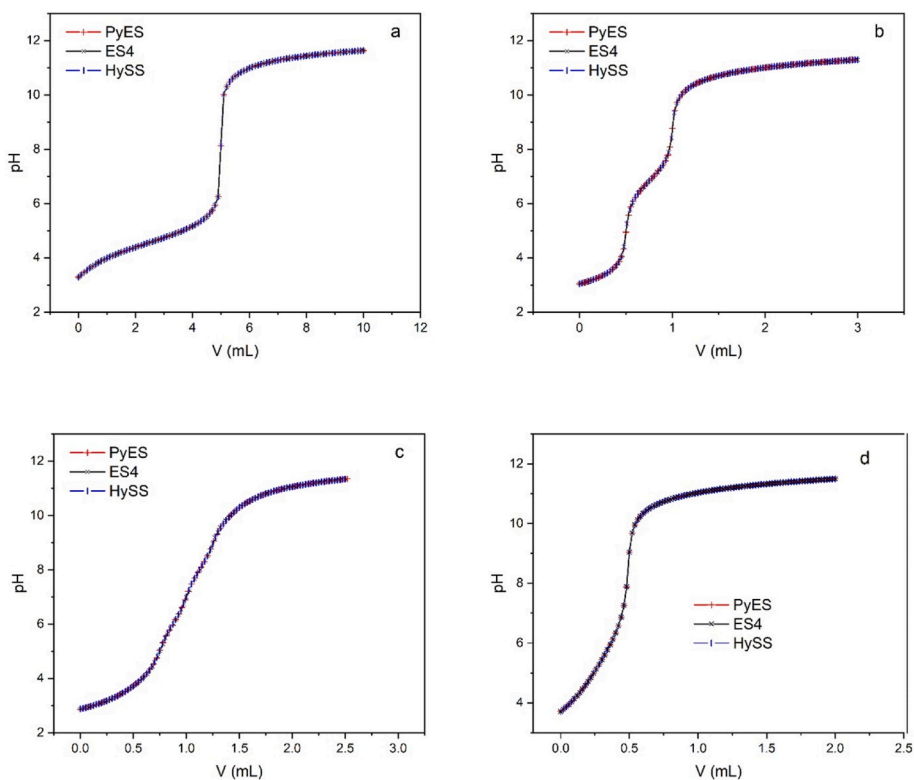


Fig. 7. Simulated titration curves computed by PyES (red cross), ES4 (black X) and HySS (blue vertical dash): a) for 25 cm³ of monoprotic acid 1×10^{-2} mol dm⁻³, titrant KOH 0.05 mol dm⁻³; b) for 25 cm³ of phosphoric acid 1×10^{-3} mol dm⁻³, titrant KOH 0.05 mol dm⁻³; c) for 25 cm³ of hexaprotic acid 1×10^{-3} mol dm⁻³, titrant KOH 0.1 mol dm⁻³; d) for 25 cm³ of Cu²⁺ 1×10^{-3} mol dm⁻³, glycine 2×10^{-3} mol dm⁻³, titrant KOH 0.1 mol dm⁻³. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

successful with results inside an acceptable margin of error. In other words, PyES represents a relevant advance for any researcher dealing with chemical speciation, because it is a modern open-source software, written according to solid mathematics and allowing any user to perform chemical equilibria calculations, taking into account both soluble and insoluble species, with the advantages of considering the variation of ionic strength and computing uncertainties.

PyES and the source code are freely available for download at <https://github.com/Kastakin/PyES>.

The availability of the source code makes it continuously improvable by the scientific community. Thanks to the modularity of the written routines, future new functionalities could be easily implemented expanding the scope of the software itself. Our hope is that, in the near future, PyES will be able to be considered one of the main platforms to

perform the computation for thermodynamic chemical equilibria in solution.

5. Validation

5.1. Prof. Maurizio Remelli¹

1. Dipartimento di Scienze Chimiche, Farmaceutiche ed Agrarie, Università degli Studi di Ferrara, Via Luigi Borsari 46-44121 Ferrara, Italy

The authors present here the version 1.0 of the program PyES, an open source program available for the most common operating systems. The program aims to allow the calculation of species distribution in a

complex system, starting from the speciation model and the stoichiometric concentrations of all the components. PyES also allows to simulate titrations where each of the components can be taken as the titrant. The results are presented both in tabular form, which can be exported to a common spreadsheet, and in chart form, the appearance of which can be customized in many ways. The program also performs the calculation of the ionic strength, point to point, if this parameter is selected as variable, i.e. with reference to real systems, in the absence of a concentrated background electrolyte. The formation of precipitates can also be taken into account.

The interface is intuitive and user-friendly; the use of the program does not require special knowledge of computer science, but only basic knowledge in analytical chemistry related to titration processes and solution equilibria.

The program has been tested on standard systems, such as the nickel/glycine system, that has been the subject of an inter-laboratory study in the past [A. Braibanti, G. Ostacoli et al. (1987) *Pure Appl Chem* 59(12): 1721–1728]; the results have been compared with those provided by the Hyss program [L. Alderighi, P. Gans et al. (1999) *Coord Chem Rev* 184: 311–318] widely used by the scientific community for similar purposes, but available only in executable form.

5.2. Prof. Carmelo Sgarlata² and Ms. Giuseppina D.G. Santonoceta (graduate student)²

2. Laboratory of Calorimetry and Analytical Supramolecular Chemistry (CAsAC) Department of Chemical Sciences, University of Catania Viale A. Doria 6, 95125 Catania, Italy

PyES is a novel application for the determination of the species distribution and the simulation of titration curves from chemical equilibria data, which has its proper place in the framework/family of software commonly used for solution thermodynamics calculations. The program is an advantageous and easy-to-use tool for researchers who work on chemical speciation or need to determine the amount of species formed in a multicomponent solution. Furthermore, the conditions for a titration can be suitably simulated thus making the software convenient for experimental design purposes. The model containing the species in solution (and/or in the solid phase) can be easily defined through a modern and intuitive dialog window.

Unlike similar applications, PyES is able to perform calculations at variable ionic strength and, remarkably, it can estimate the uncertainties on the concentrations of the species from the errors of the input quantities. This is a key point as the uncertainty on the calculated species is a crucial parameter to evaluate the significance of the outcomes.

PyES enables the user to visualize and handle the results through a

clear-cut output file/table which contains both the concentration and the percentage of formation of each species, the associated uncertainties as well as the ionic strength change during a titration simulation. Multiple tests and trials demonstrated that all calculations are correctly performed. All results are displayed as simple plots that may be used for presentation/publication purposes and can be exported as spreadsheet data for further analysis.

The free-to-use model and the availability of the source code will ensure a continuous development of the software through the direct contribution of the scientific community, in line with the open science approach. PyES displays those novelty, robustness and integration features that are currently required by researchers who deal with chemical equilibria and speciation on a daily basis.

CRediT authorship contribution statement

Lorenzo Castellino: Conceptualization; Investigation; Methodology; Software; Writing - original draft; Writing - review & editing. **Eugenio Alladio:** Methodology; Software; Supervision; Writing - review & editing. **Stefano Bertinetti:** Formal analysis; Methodology. **Gabriele Lando:** Validation; Software. **Concetta De Stefano:** Formal analysis; Supervision. **Salvador Blasco:** Formal analysis; Validation. **Enrique García-España:** Formal analysis; Supervision. **Sofia Gama:** Validation; Project administration; Writing - review & editing. **Silvia Berto:** Conceptualization; Data curation; Investigation; Software; Supervision; Validation; Visualization; Writing - original draft; Writing - review & editing. **Demetrio Milea:** Conceptualization; Funding acquisition; Project administration; Resources; Supervision; Validation; Writing - review & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

We have shared link to the software, data code, and example files.

Acknowledgments

This publication is based upon work from COST Action CA18202 NECTAR — Network for Equilibria and Chemical Thermodynamics Advanced Research, supported by COST (European Cooperation in Science and Technology, www.cost.eu, www.cost-nectar.eu).

Symbols

C_i	Component i
S_i	Soluble species i
P_i	Precipitable species i
$[C_i]_T$	Analytical (total) concentration of Component i
N_C	Number of components
N_S	Number of soluble species
N_P	Number of precipitable species
$p_{i,j}$	Stoichiometric coefficient of component j in soluble species i
$bp_{i,j}$	Stoichiometric coefficient of component j in precipitable species i
β_i	Stability constant of soluble species i
K_{Si}	Solubility product of precipitable species i
SI_i	Saturation index of precipitable species i
v_0	Initial volume in titration vessel
v_n	Titration added at titration at point n
σ_x^2	Variance of parameter x

Appendix A. Supplementary data

Supplementary data to this article can be found online at <https://doi.org/10.1016/j.chemolab.2023.104860>.

References

- [1] P. Gans, A. Sabatini, A. Vacca, Investigation of equilibria in solution. Determination of equilibrium constants with HYPERQUAD suite of programs, *Talanta* 43 (1996) 1739–1753.
- [2] Hyperquad suite webpage. <http://www.hyperquad.co.uk/>. Access date: Jan 2023.
- [3] P. Gans, A. Sabatini, A. Vacca, SUPERQUAD: an improved general program for computation of formation constants from potentiometric data, *J. Chem. Soc. Dalton Trans.* (1985) 1195–1200.
- [4] L. Zekany, I. Nagypal, J.P. Fackler Jr., PSEQUAD: a comprehensive program for the evaluation of potentiometric and/or spectrophotometric equilibrium data using analytical derivatives, in: D.J. Leggett (Ed.), *Computational Methods for the Determination of Formation Constants*, Plenum Press, New York, 1985, pp. 291–292.
- [5] C. De Stefano, P. Mineo, C. Rigano, S. Sammartano, Ionic strength dependence of formation constants. XVII. The calculation of equilibrium concentration and formation constants, *Ann. Chim. (Rome)* 83 (1993) 243–262.
- [6] F. Crea, C. De Stefano, A. Irto, D. Milea, A. Pettignano, S. Sammartano, Modeling the acid-base properties of molybdate(VI) in different ionic media, ionic strengths and temperatures, by EDH, SIT and Pitzer equations, *J. Mol. Liq.* 229 (2017) 15–26.
- [7] L. Alderighi, P. Gans, A. Ienco, D. Peters, A. Sabatini, A. Vacca, Hyperquad simulation and speciation (HySS): a utility program for the investigation of equilibria involving soluble and partially soluble species, *Coord. Chem. Rev.* 184 (1999) 311–318.
- [8] A. De Robertis, C. De Stefano, S. Sammartano, C. Rigano, The calculation of equilibrium concentrations in large multimetal/multiligand systems, *Anal. Chim. Acta* 191 (1986) 385–398.
- [9] F. Crea, C. De Stefano, D. Milea, S. Sammartano, Phytate-molybdate(VI) interactions in NaCl(aq) at different ionic strengths: unusual behaviour of the protonated species, *New J. Chem.* 42 (2018) 7671–7679.
- [10] C. Bretti, P. Cardiano, A. Irto, G. Lando, D. Milea, S. Sammartano, Interaction of N-acetyl-L-cysteine with Na⁺, Ca²⁺, Mg²⁺ and Zn²⁺. Thermodynamic aspects, chemical speciation and sequestering ability in natural fluids, *J. Mol. Liq.* 319 (2020), 114164.
- [11] A. De Robertis, C. De Stefano, S. Sammartano, A. Gianguzza, Equilibrium studies in natural fluids. A chemical speciation model for the major constituents of seawater, *Chem. Speciat. Bioavailab.* 6 (1994) 65–84.
- [12] F. Morel, J. Morgan, Numerical method for computing equilibria in aqueous chemical systems, *Environ. Sci. Technol.* 6 (1972) 58–67.
- [13] C.M. Bethke, *Geochemical and Biogeochemical Reaction Modeling*, third ed., Cambridge University Press, Cambridge, UK, 2022.
- [14] J. Carrayrou, R. Mosé, P. Behra, New efficient algorithm for solving thermodynamic chemistry, *AIChE J.* 48 (2002) 894–904.
- [15] M. Marinoni, J. Carrayrou, Y. Lucas, P. Ackerer, Thermodynamic equilibrium solutions through a modified Newton Raphson method, *AIChE J.* 63 (2017) 1246–1262.
- [16] E. Hogfeldt, *Stability Constants of Metal-Ion Complexes. Par. A: Inorganic Ligands*. IUPAC Chemical Data Series, Pergamon Press, Oxford, 1982.
- [17] A.E. Martell, R.M. Smith, R.J. Motekaitis, NIST Standard Reference Database 46, 2004 vers.8, Gaithersburg.
- [18] P.M. May, K. Murray, Database of chemical reactions designed to achieve thermodynamic consistency automatically, *J. Chem. Eng. Data* 46 (2001) 1035–1040.
- [19] P.M. May, D. Rowland, E. Königsberger, G. Hefter, JESS, a Joint Expert Speciation System - IV: a large database of aqueous solution physicochemical properties with an automatic means of achieving thermodynamic consistency, *Talanta* 81 (2010) 142–148.
- [20] D. Pettit, K. Powell, IUPAC Stability Constants Database, Academic Software, Otley, UK, 2004.
- [21] L.G. Sillén, A.E. Martell, *Stability Constants of Metal Ion Complexes. Special Publ. 17*, The Chemical Society, Wiley, London, 1964.
- [22] L.G. Sillén, A.E. Martell, *Stability Constants of Metal Ion Complexes, Supplement Special Publ. 25*, The Chemical Society, Wiley, London, 1964.
- [23] I. Grenthe, I. Puigdomenech, *Modelling in Aquatic Chemistry*, OECD, Paris, 1997.
- [24] C.W. Davies, *Ion Association*, Butterworths, London, 1962.
- [25] L.A. Bromley, Thermodynamic properties of strong electrolytes in aqueous solutions, *AIChE J.* 19 (1973).
- [26] K.S. Pitzer, Thermodynamics of electrolytes. I. Theoretical basis and general equations, *J. Phys. Chem.* 77 (1973) 268–277.
- [27] K.S. Pitzer, *Activity Coefficients in Electrolyte Solutions*, second ed., CRC Press, Inc., Boca Raton, FL, 1991.
- [28] E.A. Guggenheim, The specific thermodynamic properties of aqueous solutions of strong electrolytes, *Philos. Mag. A* 19 (seventh series) (1935) 588–643.
- [29] P.G. Daniele, A. De Robertis, C. De Stefano, S. Sammartano, Ionic strength dependence of formation constants. Xiii. A critical examination of preceding results, in: S. Alegret, J.J. Arias, D. Barcelò, J. Casal, G. Router (Eds.), *Miscellany of Scientific Papers in Honour of Enric Casassas, Professor of Analytical Chemistry, Universitat Autònoma de Barcelona, Bellaterra, Spain, 1991*, pp. 121–126.
- [30] P.G. Daniele, C. Rigano, S. Sammartano, Ionic strength dependence of formation constants. Part II: potentiometric study of the copper(II)-Malonate-Glycinate system in the range 0.01 < I < 1.0, *Transition Met. Chem.* 7 (1982) 109–112.
- [31] P.G. Daniele, C. Rigano, S. Sammartano, Ionic strength dependence of formation constants-I protonation constants of organic and inorganic acids, *Talanta* 30 (1983) 81.
- [32] R.M. Cigala, M. Cordaro, F. Crea, C. De Stefano, V. Fracassetti, M. Marchesi, D. Milea, S. Sammartano, Acid-base properties and alkali and alkaline earth metal complex formation in aqueous solution of diethylenetriamine-N,N,N',N''-pentakis(methylenephosphonic acid) obtained by an efficient synthetic procedure, *Ind. Eng. Chem. Res.* 53 (2014) 9544–9553.
- [33] R.M. Cigala, C. De Stefano, A. Irto, D. Milea, S. Sammartano, Thermodynamic data for the modeling of Lanthanoid(III) sequestration by reduced glutathione in aqueous solution, *J. Chem. Eng. Data* 60 (2015) 192–201.
- [34] F. Crea, C. De Stefano, D. Milea, A. Pettignano, S. Sammartano, SALMO and S3M: a saliva model and a single saliva salt model for equilibrium studies, *Bioinorgan. Chem. Appl.* (2015) 12, 2015.
- [35] P.G. Daniele, C. De Stefano, M. Marangella, C. Rigano, S. Sammartano, URSUS: a computer program for urine speciation, *Biochim. Clin.* 13 (1989) 507–510.
- [36] S. Berto, M. Marangella, C. De Stefano, D. Milea, P.G. Daniele, Critical reappraisal of methods for measuring urine saturation with calcium salts, *Molecules* (2021) 26.
- [37] The pandas development team, *Pandas-dev/pandas: Pandas* (2023), <https://doi.org/10.5281/zenodo.3509134> (v1.5.3).
- [38] C.R. Harris, K.J. Millman, S.J.v.d. Walt, R. Gommers, P. Virtanen, D. Cournapeau, E. Wieser, J. Taylor, S. Berg, N.J. Smith, R. Kern, M. Picus, S. Hoyer, M.H. v. Kerkwijk, M. Brett, A. Haldane, J.F.d. Río, M. Wiebe, P. Peterson, P. Gérard-Marchant, K. Sheppard, T. Reddy, W. Weckesser, H. Abbasi, C. Gohlke, T. E. Oliphant, *Array programming with NumPy*, *Nature* 585 (2020) 357–362.
- [39] Python and Qt simplified. Create a Python GUI in minutes. <https://build-system.fm/an.io/>. Access date: Jan 2023.
- [40] PyQtGraph - Scientific Graphics, GUI Library for Python. <https://www.pyqtgraph.org/>. Access date: Jan 2023.
- [41] PyInstaller - bundle Python applications. <https://www.pyinstaller.org/>. Access date: Jan 2023.
- [42] P.G. Daniele, A. De Robertis, C. De Stefano, A. Gianguzza, S. Sammartano, Salts effects on the protonation of ortho-phosphate between 10 and 50 °C, in aqueous solution. A complex formation model, *J. Solut. Chem.* 20 (1991) 495–515.
- [43] T. Kiss, I. Sovago, A. Gergely, Critical survey of stability-constants of complexes of Glycine, *Pure Appl. Chem.* 63 (1991) 597–638.
- [44] OPIUM webpage. <https://web.natur.cuni.cz/~kyvala/opium.html> (Jan 2023).
- [45] Medusa webpage. <https://www.kth.se/che/medusa/> (Jan 2023).
- [46] PHREEQC webpage. <https://www.usgs.gov/software/phreeqc-version-3> (Jan 2023).
- [47] MINEQL+ webpage. <https://www.mineql.com/index.html> (Jan 2023).