A generalized Poisson and Poisson-Boltzmann solver for electrostatic environments

G. Fisicaro' , L. Genovese, O. Andreussi, N. Marzari, and S. Goedecker

---

A generalized Poisson and Poisson-Boltzmann solver for electrostatic environments

G. Fisicaro,[1,a] L. Genovese,[2] O. Andreussi,[3,4] N. Marzari,[4] and S. Goedecker[1]

[1]*Department of Physics, University of Basel, Klingelbergstrasse 82, 4056 Basel, Switzerland*
[2]*University of Grenoble Alpes, CEA, INAC-SP2M, L_Sim, F-38000 Grenoble, France*
[3]*Institute of Computational Science, Università della Svizzera Italiana, Via Giuseppe Buffi 13, CH-6904 Lugano, Switzerland*
[4]*Theory and Simulations of Materials (THEOS) and National Centre for Computational Design and Discovery of Novel Materials (MARVEL), École Polytechnique Fédérale de Lausanne, Station 12, CH-1015 Lausanne, Switzerland*

The computational study of chemical reactions in complex, wet environments is critical for applications in many fields. It is often essential to study chemical reactions in the presence of applied electrochemical potentials, taking into account the non-trivial electrostatic screening coming from the solvent and the electrolytes. As a consequence, the electrostatic potential has to be found by solving the generalized Poisson and the Poisson-Boltzmann equations for neutral and ionic solutions, respectively. In the present work, solvers for both problems have been developed. A preconditioned conjugate gradient method has been implemented for the solution of the generalized Poisson equation and the linear regime of the Poisson-Boltzmann, allowing to solve iteratively the minimization problem with some ten iterations of the ordinary Poisson equation solver. In addition, a self-consistent procedure enables us to solve the non-linear Poisson-Boltzmann problem. Both solvers exhibit very high accuracy and parallel efficiency and allow for the treatment of periodic, free, and slab boundary conditions. The solver has been integrated into the BigDFT and Quantum-ESPRESSO electronic-structure packages and will be released as an independent program, suitable for integration in other codes. © *2016 AIP Publishing LLC.* [http://dx.doi.org/10.1063/1.4939125]

I. INTRODUCTION

Many important chemical processes take place in solution both in the context of basic and industrial research. The computational study of such chemical reactions in wet environments is therefore of cross-disciplinary interest to physics, chemistry, materials science, chemical engineering, and biology.[1] Computational studies can complement these investigations by giving insight into new processes and materials as well as reducing development times and production costs. Solar-energy harvesting in a dye-sensitized cell or electro-catalytic water splitting are two simple examples of relevance for applications in the energy and environment context.

Molecular properties in the presence of a solution are often very different compared to pure *in vacuum*[2] conditions making vacuum-like *ab initio* calculations an inappropriate approach for such problems. An inclusion of the solute-solvent interaction in *ab initio* simulations is thus mandatory. On the atomistic scale, an explicit inclusion of all solvent molecules in the simulation should be in principle the natural way to account for solvent effects. Due to the very large number of water and possibly other molecules required, this approach enormously increases the computational cost and limits at the same time the size of the system contained in the explicit dielectric medium.[3] The study of the solute-solvent interaction at length scales larger than the molecular sizes would become virtually impossible. Investigations like structure predictions[4] or reaction path determination[5] would become unaffordable in such a purely atomistic approach. Moreover, fully atomistic simulations of solvation effects would need to deal with the extensive sampling, required to characterize liquid configurations, and with the well-known limitations that current state-of-the-art *ab initio* methods present in describing liquid water, in particular regarding its structural and dielectric properties.

An implicit inclusion of the solute-solvent interactions can solve these issues. Starting from the earliest work of Onsager,[6] the quantum chemistry community investigated implicit solvation models[7–9] extensively. In these models, the solvent is introduced as a continuous homogeneous and isotropic medium fully described by a dielectric function. Among them, the polarizable continuum model (PCM) developed by Tomasi and co-workers[7,8] is one of the most popular models. In this approach, a dielectric cavity surrounding the atomistic system is introduced where the permittivity takes on the value of one in regions occupied by atoms and some different value characteristics for the dielectric solvent medium considered outside.

Density functional theory (DFT) is a widely used method to investigate material properties at the atomistic scale. In such *ab initio* calculations, the electronic-structure problem is solved by minimizing the total energy of the system which is a functional of the electronic density

a)giuseppe.fisicaro@unibas.ch

$$E[\rho] = T[\rho] + \int v(\mathbf{r})\rho(\mathbf{r})d\mathbf{r} + \frac{1}{2}\int \rho(\mathbf{r})\phi[\rho]d\mathbf{r} + E_{xc}[\rho],$$
(1)

where the four terms on the right side of Eq. (1) are, respectively, the standard kinetic energy, the interaction energy with an external potential, the electrostatic and the exchange-correlation energies. For gas-phase molecular simulations, the potential $\phi(\mathbf{r})$ generated by a given charge density $\rho(\mathbf{r})$ is given by the solution of the standard Poisson (SPe) equation

$$\nabla^2 \phi(\mathbf{r}) = -4\pi\rho(\mathbf{r}).$$
(2)

An implicit inclusion of the solvent at such DFT level can be obtained by introducing a continuum dielectric cavity by means of a dielectric distribution $\epsilon(\mathbf{r})$. Then, the potential is given by the solution of the generalized Poisson equation (GPe)

$$\nabla \cdot \epsilon(\mathbf{r})\nabla\phi(\mathbf{r}) = -4\pi\rho(\mathbf{r}).$$
(3)

If the system is surrounded by an ionic solution, an extra-term has to be added on the right side of Eq. (3). It accounts for the ionic distribution in the liquid and depends on the local electrostatic potential $\phi(\mathbf{r})$. The resulting non-linear differential equation would therefore become

$$\nabla \cdot \epsilon(\mathbf{r})\nabla\phi(\mathbf{r}) = -4\pi\left(\rho(\mathbf{r}) + \rho^{ions}[\phi](\mathbf{r})\right),$$
(4)

where $\rho^{ions}(\mathbf{r})$ is the local concentration of ions in the dielectric solvent, written as a sum of concentration contributions c_i of ions of type $i \in \{1, 2, \ldots, m\}$ and valence Z_i, which in turn are ϕ-dependent functionals,

$$\rho^{ions}[\phi](\mathbf{r}) = \gamma[\epsilon](\mathbf{r})eN_A\sum_{i=1}^{m} Z_i c_i[\phi](\mathbf{r}).$$
(5)

Here, e is the elementary charge, N_A the Avogadro's number, and $\gamma[\epsilon](\mathbf{r})$ a proper function which guarantees that mobile ion concentrations tend to zero inside the dielectric cavity mapped by $\epsilon(\mathbf{r})$. The most common expression for the $c_i[\phi]$ functional gives rise to the well-known Poisson-Boltzmann equation (PBe).

Whereas several approaches and solvers exist for the SPe,[10–14] efficient and accurate solvers are still missing for the GPe and PBe cases. Some of these solvers handle *sparse* matrices obtained by low-order finite difference discretization of Eq. (3), many of them keeping constant the permittivity in both inner and external regions of the dielectric cavity and then solving a standard Poisson problem. Furthermore, simpler and linearized forms of the Poisson-Boltzmann equation are usually considered, neglecting steric effects and overestimating ionic concentrations close to highly charged surfaces and for multivalent ions.

Fast and accurate GPe and PBe solvers which accurately work for a continuously varying dielectric function could therefore play a key role in the extension of vacuum-based atomistic packages and allowing for quantum simulations in the presence of water, dissolved species, electrolytes, and non-aqueous solvents.

In the present paper, we present a minimization technique which solves the generalized Poisson problem and the linear regime of the Poisson-Boltzmann in some ten iterations of a SPe solver. In combination with a self-consistent procedure, it enables us to solve the non-linear Poisson-Boltzmann problem in a formulation which includes ionic steric effects. The implemented algorithms take advantage of the chosen preconditioner for the minimization procedure. The accuracy of both GPe and PBe solvers has been tested for cases where an analytic solution is available. Two different methods have been implemented to describe the solvent surrounding the atomistic system. We prove the effectiveness of our method in practical DFT calculations of electrostatic solvation energies of various test systems.

II. GENERALIZED POISSON EQUATION

As discussed above, the dielectric medium is described by means of a position-dependent dielectric distribution $\epsilon(\mathbf{r})$. In order to solve numerically Eq. (3), the potential $\phi(\mathbf{r})$, the charge density $\rho(\mathbf{r})$, and the dielectric function $\epsilon(\mathbf{r})$ are generally discretized on a finite grid. In principle also, the generalized Poisson operator

$$\mathcal{A} = \nabla \cdot \epsilon(\mathbf{r})\nabla$$
(6)

should be discretized on the same mesh. It will be shown that depending on the adopted strategy to solve numerically Eq. (3), the discretization of the differential operator $\mathcal{A}$ can be avoided in exchange of an iterative procedure based on a SPe solver.

An alternative would be to solve the GPe iteratively as suggested in Ref. 15. In this approach, the polarization field introduced by the spatially varying dielectric function is added as a source term to the charge density of the ordinary Poisson equation, and the Poisson equation is solved repeatedly until self-consistency between the potential and the polarization charge density induced by it is reached. Our approach completes and simplifies this treatment, reducing considerably the number of SPe iterations, thereby presenting a robust and powerful iterative solver.

Considering that reliable convergence can be an issue in mixing schemes, an alternative approach based on a minimization procedure is desireable. Such an approach can be based on an action integral whose Euler Lagrange equation is the GPe [Eq. (3)],

$$\mathcal{I} = \int \left[\frac{1}{2}\nabla\phi(\mathbf{r})\epsilon(\mathbf{r})\nabla\phi(\mathbf{r}) - 4\pi\rho(\mathbf{r})\phi(\mathbf{r})\right]d\mathbf{r}.$$
(7)

Any numerical minimization scheme can then be applied to solve the electrostatic problem.

In Sec. II B, our strategy to solve Eq. (3), based on a preconditioned conjugate gradient (PCG) algorithm, will be presented. The chosen preconditioner exactly represents the operator in the limit of a slowly varying dielectric constant and is based on the standard Poisson solver of BigDFT.

The possibility of solving the generalized Poisson equation under various boundary conditions (BCs) will be very important. In our approach, the boundary conditions enter in a straightforward way by means of the preconditioner, i.e., through the solution of the SPe.

ALGORITHM 1. Self-consistent (SC) iterative procedure.

1: set ρ_0^{iter}
2: **for** $k = 0, 1, \ldots$ **do**
3:      $\rho_k^{tot} = \rho/\epsilon + \rho_k^{iter}$
4:      solve $\nabla^2 \phi_k = -4\pi \rho_k^{tot}$
5:      $\rho_{k+1}^{iter} = \frac{1}{4\pi} \nabla \ln \epsilon \cdot \nabla \phi_k$
6:      $\rho_{k+1}^{iter} = \eta \rho_{k+1}^{iter} + (1-\eta) \rho_k^{iter}$
7:      $r_{k+1} = \rho_{k+1}^{iter} - \rho_k^{iter}$
8: **end for**

ALGORITHM 2. Preconditioned conjugate gradient (PCG).

1: $r_0 = -4\pi\rho - \mathcal{A}\phi_0, \ p_{-1} = 0$
2: **for** $k = 0, 1, \ldots$ **do**
3:      $v_k = \mathcal{P}^{-1} r_k$
4:      $p_k = v_k + \beta_k p_{k-1}$ (where $\beta_k = \frac{(v_k, r_k)}{(v_{k-1}, r_{k-1})}, k \neq 0$)
5:      $\alpha_k = \frac{(v_k, r_k)}{(p_k, Ap_k)}$
6:      $\phi_{k+1} = \phi_k + \alpha_k p_k$
7:      $r_{k+1} = r_k - \alpha_k \mathcal{A} p_k$
8: **end for**

A. Self-consistent iterative procedure

A strategy to solve the GPe for a given charge density $\rho(\mathbf{r})$ is by means of a self-consistent (SC) iterative procedure.[15] Applying simple algebraic manipulations, Eq. (3) can be rewritten as

$$\nabla^2 \phi(\mathbf{r}) = -4\pi \left[\frac{\rho(\mathbf{r})}{\epsilon(\mathbf{r})} + \rho^{iter}(\mathbf{r}) \right] = -4\pi \left[\rho(\mathbf{r}) + \rho^{pol}(\mathbf{r}) \right], \quad (8)$$

where $\rho^{pol}(\mathbf{r})$ is the polarization charge density. In this approach, an extra-term $\rho^{iter}(\mathbf{r})$,

$$\rho^{iter}(\mathbf{r}) = \frac{1}{4\pi} \nabla \ln \epsilon(\mathbf{r}) \cdot \nabla \phi(\mathbf{r}) \quad (9)$$

induced by the spatially varying dielectric function $\epsilon(\mathbf{r})$ is added as a source to the charge density of the ordinary Poisson equation. Hence, the GPe can be solved by a self-consistent loop on the potential $\phi(\mathbf{r})$, obtained by a SPe solver onto the second member of Eq. (8). The residual r_k, quantifying the convergence, is the difference between the extra terms of Eq. (9) between subsequent iterations. Algorithm 1 describes the procedure. This approach requires a finite difference filter to be applied at step 5. In order to stabilize the iterative method, a linear mixing of the extra-term $\rho^{iter}(\mathbf{r})$ at steps kth and $(k + 1)$th has been introduced tuned by means of the mixing parameter η.

The polarization charge induced in the dielectric medium can be easily related to the extra-term of Eq. (9),

$$\rho^{pol}(\mathbf{r}) = \rho^{iter}(\mathbf{r}) + \frac{1 - \epsilon(\mathbf{r})}{\epsilon(\mathbf{r})} \rho(\mathbf{r}). \quad (10)$$

This charge represents the response of the surrounding implicit dielectric. It lies in the transition region between the inner and outside parts of the cavity and stabilizes the solute density enveloped by the solvent.

B. Preconditioned conjugate gradient

Although a reasonably small number of iterations can be obtained in a self-consistent scheme, minimization techniques can produce more efficient methods to handle Eq. (3) if sophisticated schemes are utilized. In addition, the formulation as a minimization problem would allow to better control the convergence behavior.

Solving this equation with a preconditioned steepest descent (PSD) method is essentially identical to the self-consistency approach of Sec. II A, once a standard Poisson solver is taken as preconditioner. In particular, a good preconditioner for a PSD minimization, whose inverse applied

to a residual vector r_k provides the preconditioned residual v_k, is as follows:

$$\mathcal{P}^{SD} v_k(\mathbf{r}) = \epsilon(\mathbf{r}) \nabla^2 v_k(\mathbf{r}) = -4\pi r_k(\mathbf{r}). \quad (11)$$

Being the Laplacian of $v_k(\mathbf{r})$ related to the residual vector r_k by means of Eq. (11), the generalized Poisson operator becomes

$$\mathcal{A} v_k(\mathbf{r}) = \nabla \cdot \epsilon(\mathbf{r}) \nabla v_k(\mathbf{r}) = \nabla \epsilon(\mathbf{r}) \cdot \nabla v_k(\mathbf{r}) - 4\pi r_k(\mathbf{r}). \quad (12)$$

Such a PSD approach can be described by Algorithm 2 with $\beta_k = 0$. Fixing $\alpha_k = \eta$, it corresponds to the previously described self-consistent approach.

In a PSD scheme, the number of iterations l needed for convergence is proportional to the condition number κ (i.e., the ratio between the largest and the smallest eigenvalue of the product operator $\mathcal{P}^{-1}\mathcal{A}$). Minimization methods with a faster convergence rate than the preconditioned steepest descent algorithm can significantly improve the convergence speed.

We use a preconditioned conjugate gradient scheme, where $l \propto \sqrt{\kappa}$. In such minimization procedure, a good preconditioner can lower κ and, therefore, the overall number of iterations. Algorithm 2 describes the implemented PCG procedure to compute the electrostatic potential $\phi(\mathbf{r})$ starting from a given charge density $\rho(\mathbf{r})$. The minimization procedure starts from an initial gradient r_0 computed on an input guess ϕ_0. $\mathcal{P}$ is the preconditioner whose inverse has to be applied to the residual vector r_k returning the preconditioned residual v_k, and, finally, ϕ_k is the solution of Eq. (3). The convergence criterion is imposed on the Euclidean norm of the residual vector r_k.

Both performances and accuracy in a PCG scheme critically depend on the preconditioner chosen. We implemented a preconditioner based on the solution of the standard Poisson equation (namely, on a standard Poisson solver). Once a residual vector r_k is given at the step 3 of Algorithm 2, we define a preconditioned residual from the following equation:

$$\mathcal{P}^{CG} v_k(\mathbf{r}) = \sqrt{\epsilon(\mathbf{r})} \nabla^2 [v_k(\mathbf{r}) \sqrt{\epsilon(\mathbf{r})}] = -4\pi r_k(\mathbf{r}). \quad (13)$$

This equation has to be solved with respect to $v_k(\mathbf{r})$ once $r_k(\mathbf{r})$ is given.

In addition to speeding up the PCG procedure, the preconditioner defined by Eq. (13) retains a further feature which guarantees accuracy and fast performance for the whole electrostatic solver. In step 7 of Algorithm 2, we have to apply the generalized Poisson operator $\mathcal{A}$ to the preconditioned residue p_k, which means, thanks to step 4, applying it to v_k.

Using a change of variable $v'_k(\mathbf{r}) = \sqrt{\epsilon(\mathbf{r})}v_k(\mathbf{r})$, the GPe becomes

$$\nabla \cdot \epsilon(\mathbf{r})\nabla v_k(\mathbf{r}) = \sqrt{\epsilon(\mathbf{r})}\nabla^2 v'_k(\mathbf{r}) - v'_k(\mathbf{r})\nabla^2\sqrt{\epsilon(\mathbf{r})}. \quad (14)$$

Now simple algebraic manipulations and Eq. (13) allow to rewrite the generalized Poisson operator $\mathcal{A}$ as

$$\mathcal{A}v_k(\mathbf{r}) = \nabla \cdot \epsilon(\mathbf{r})\nabla v_k(\mathbf{r}) \quad (15)$$

$$= -v_k(\mathbf{r})q(\mathbf{r}) - 4\pi r_k(\mathbf{r}), \quad (16)$$

where $q(\mathbf{r}) = \sqrt{\epsilon(\mathbf{r})}\nabla^2\sqrt{\epsilon(\mathbf{r})}$ is calculated once at the beginning of the PCG procedure and kept fixed for the whole minimization loop. Therefore, thanks to the chosen preconditioner, the action of the operator $\mathcal{A}$ can be simplified to a simple multiplication between the potential $v_k(\mathbf{r})$ and the vector $q(\mathbf{r})$ related to the spatially varying dielectric function $\epsilon(\mathbf{r})$. This feature, which provides the exact operator output, makes our PCG procedure robust and fast, avoiding any finite difference differentiation. Furthermore, reducing the PCG algorithm to simple vector operations makes its parallelization straightforward, delegating it to the chosen SPe solver. A similar discussion holds for the boundary conditions, which enter in a natural way by means of the preconditioner, i.e., through the solution of the ordinary Poisson equation, both in the SC and PCG algorithms.

C. Numerical results

Both the self-consistent iterative procedure (Algorithm 1) and the preconditioned conjugate gradient minimization scheme (Algorithm 2) have been implemented and tested. As SPe solver (step 4 of Algorithm 1 and step 3 of Algorithm 2), we used the Interpolating Scaling Function (ISF) Poisson solver, allowing to obtain highly accurate electrostatic potentials for free, wire, surface, and periodic boundary conditions at the cost of $O(N\log(N))$ operations, where N is the number of discretization points (see Ref. 14).

To test both solvers, analytic three dimensional functions have been used. An orthorhombic grid of uniform mesh spacing h_{grid} and (n_x, n_y, n_z) points in each directions has been used. Fig. 1 shows plots of these benchmark fields along a particular direction passing through the box center and parallel to the y axis. All functions depend on the radial distance r from the center of the simulation domain.

A normalized Gaussian function has been chosen for the electrostatic potential $\phi(\mathbf{r})$ (red dashed line in Fig. 1) and the charge density $\rho(\mathbf{r})$ has been derived from the chosen potential and dielectric functions, applying the generalized Poisson differential operator of Eq. (6) (red dotted line). In order to reproduce the dielectric environment typically tackled in electrostatic problems where a solute system is embedded in a solvent (i.e., a cavity where the majority of the atomic charge density is confined), the error function $1 + (\epsilon_0 - 1)h(d_0, \Delta; r)$ has been chosen to represent the spatially varying dielectric constant $\epsilon(\mathbf{r})$ (solid black line in Fig. 1), where

$$h(d_0, \Delta; r) = \frac{1}{2}\left[1 + \mathrm{erf}\left(\frac{r - d_0}{\Delta}\right)\right]. \quad (17)$$

Here, Δ is a parameter which controls the transition region ($\approx 4\Delta$ wide) between the inner and external parts of the cavity
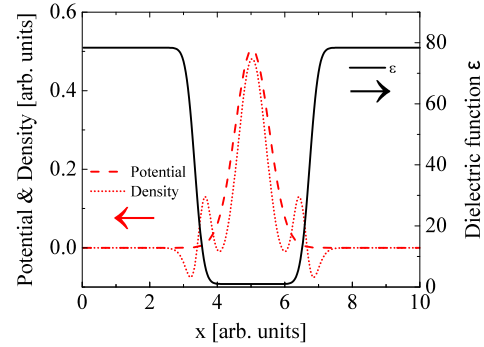

FIG. 1. Analytical three dimensional functions used as benchmark fields for both SC and PCG solvers along a particular direction passing through the box center and parallel to the y axis. Red dashed line: potential $\phi(\mathbf{r})$; red dotted line: charge density $\rho(\mathbf{r})$; black solid line: dielectric function $\epsilon(\mathbf{r})$.

of radius d_0, and ϵ_0 is the dielectric constant of the surrounding medium. These benchmark functions have been used for both free, surface and periodic boundary conditions.

To implement the nabla differential operator needed for Algorithm 1, central, forward and backward finite difference filters of order 16 have been used, which match the accuracy of the underlying SPe solver. We remark that the use of these finite difference filters is not needed for Algorithm 2 as soon as the vector $q(\mathbf{r})$ of Eq. (16) is pre-calculated.

Fig. 2 shows solver performances. Top graphs report the residual norm as function of the iteration number. The residual norm is defined has the Euclidean norm of the residual vector r_k. Graphs in lower panel present the output accuracy as a function of the iteration number. The accuracy in the whole paper is defined as the maximum value of the difference between the final numerical solution and the analytical potential. In this test case, a cubic box of length 10 a.u. has been chosen with $n_x = n_y = n_z = 300$. The Gaussian variance for the potential $\phi(\mathbf{r})$ was $\sigma = 0.5$, and the parameters of $\epsilon(\mathbf{r})$ were $d_0 = 1.7$, $\Delta = 0.3$, and $\epsilon_0 = 78.36$ (all in a.u.). The mixing parameter in step 6 of Algorithm 1 has been fixed to be $\eta = 0.6$, resulting in a robust convergence for all cases. Lower values slow down the convergence for the chosen test functions.

The PCG solver (black squares) exhibits a faster convergence with respect to the SC one (blue circles), reaching an accuracy of $\sim 10^{-10}$ with some ten iterations. Furthermore, its behavior does not change with the boundary conditions as is the case with the SC algorithm. It is worth remembering that each PCG iteration involves only a single solution of the ordinary Poisson equation and as well as of fully parallelizable vector operations. If an accuracy of $\sim 10^{-4}$ to 10^{-5} is enough, then some five iterations solve the electrostatic problem. These features make the developed PCG algorithm together with the chosen preconditioner of Eq. (13) very efficient for atomistic calculations where the generalized Poisson equation needs to be solved repeatedly. Performances of the implemented PCG procedure are also higher than multigrid approaches to solve the GPe, where a number of iterations between 17 and 25 are needed to reach an accuracy of $\sim 10^{-8}$.[16]

For the sake of completeness, the preconditioned steepest descent scheme (Algorithm 2 with $\beta_k = 0$) has been tested
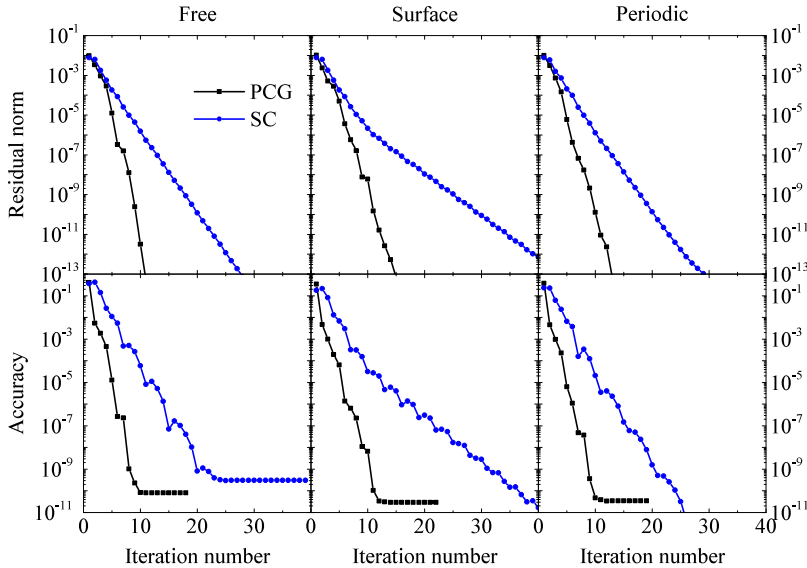
FIG. 2. Euclidean norm of the residual vector r_k (top graphs) and accuracy of the numerical solution (lower graphs) for SC (blue circles) and PCG (black squares) solvers with free, surface and period boundary conditions.

with the preconditioner described by Eq. (11). Residual norm convergence as well as the accuracy of the solution (not reported) behaves like the self-consistent approach (Algorithm 1, blue circles of Fig. 2). An integration of DIIS in the PSD loop effectively lowered the iteration numbers, but did not show better performances with respect to the PCG approach.

III. POISSON-BOLTZMANN EQUATION

The generalized Poisson equation so far discussed holds for a solute plunged in a neutral solution where no mobile charges are present. In order to extend the library to ionic solutions, effects of mobile ions have to be taken into account which, being free to move inside the dielectric medium, modify the spatial distribution of charge and potential close to the interface giving rise to the well known double layer.

In general, mobile charges can be included in the electrostatic problem by means of a continuum mean-field approach, assuming point-like ions in thermodynamic equilibrium. Once the equilibrium is reached, ionic concentrations explicitly depend on the local electrostatic potential $\phi(\mathbf{r})$. Following this idea, the potential $\phi(\mathbf{r})$ generated by a charge density $\rho(\mathbf{r})$ placed in contact with an ionic solution can be extracted solving the generalized Poisson equation [Eq. (4)]. Several models have been proposed in the literature for the ionic bulk concentrations [Eq. (5)]. Gouy[17] and Chapman[18] proposed a Boltzmann distribution

$$c_i[\phi](\mathbf{r}) = c_i^\infty \exp\left(-\frac{Z_i e \phi(\mathbf{r})}{kT}\right), \tag{18}$$

where k is the Boltzmann constant and T is the absolute temperature of the solvent. Combining Eqs. (4), (5), and (18), the well-known Poisson-Boltzmann equation can be recovered. It arises from the equilibrium between thermal and electric forces, which depend, respectively, on the ionic concentration and electrostatic potential. At equilibrium, the total average force must be zero and Eq. (18) holds. In the regions where the electrostatic energy is smaller than

kT, i.e., $Z_i e \phi(\mathbf{r})/kT \ll 1$, the exponential of Eq. (18) can be approximated by a linear function of $\phi(\mathbf{r})$, switching the non-linear problem of Eq. (4) into a linear one.

The Poisson-Boltzmann equation correctly predicts ionic profiles close to the solid-liquid interface with ionic solutions. However, it strongly overestimates ionic concentrations close to highly charged surfaces or multivalent ions. In order to overcome these drawbacks, several models have been proposed.[19] Finite ion size effects can be included in the model by introducing an additional internal force. Using a Bikerman-type expression to model steric effects[20] and imposing that at equilibrium the total average force (thermal, electric, and steric) must go to zero, a Langmuir-type equation for the ionic concentrations can be found

$$c_i[\phi](\mathbf{r}) = \frac{c_i^\infty \exp\left(-\dfrac{Z_i e \phi(\mathbf{r})}{kT}\right)}{1 + \displaystyle\sum_{j=1}^{m} \dfrac{c_j^\infty}{c_j^{\max}}\left[\exp\left(-\dfrac{Z_j e \phi(\mathbf{r})}{kT}\right) + 1\right]}. \tag{19}$$

In Eq. (19), $c_i^{\max}$ are the maximum local concentrations that an ionic species of effective radius R_i can attain. This last is given by

$$c_i^{\max} = \frac{p}{\frac{4}{3}\pi R_i^3 N_A}, \tag{20}$$

where p is the packing coefficient. The combination of Eqs. (4), (5), and (19) gives rise to the so-called modified Poisson-Boltzmann equation (MPBe).[1,21–23] It accounts for finite ion size effects representing an extension of the PBe and preventing ion concentrations to exceed $c_i^{\max}$. Note that if $c_i^{\max} \to \infty \ \forall i \in \{1, 2, \ldots, m\}$, Eq. (19) is reduced to Eq. (18) and the standard Poisson-Boltzmann equation holds.

Both PBe and MPBe equations consider ions in solution as pointlike. A further extension can account for finite size effects in an explicit way, describing ions by means of insulating spheres plunged in the dielectric solvent.[24] This correction allows for a local modification of the solvent permittivity as well as the inclusion of two new forces: one

related to a non-uniform dielectric which tend to move ions into high-permittivity regions; another due to the interaction of the ion dipole and the non-uniform local electric field (dielectrophoretic force).

The Poisson-Boltzmann equation is more difficult to solve than the generalized Poisson equation due to its non-linear nature. If $Z_i e\phi(\mathbf{r})/kT \ll 1$, Eq. (18) can be approximated by a linear function of $\phi(\mathbf{r})$, and the Poisson-Boltzmann problem of Eq. (4) becomes linear. In Sec. II B, a particular preconditioner, i.e., Eq. (13), has been proposed for the solution of the GPe. Furthermore, the generalized Poisson operator, as reformulated in Eq. (16), already contains a linear term with respect to the potential $\phi(\mathbf{r})$. Therefore, Algorithm 2 is expected to solve the linear regime of the PBe, where the $\mathcal{A}$ operator is now given by Eq. (16) with an additional linear term,

$$\mathcal{A}v_k(\mathbf{r}) = \nabla \cdot \epsilon(\mathbf{r})\nabla v_k(\mathbf{r}) + 4\pi \rho^{ions}[\phi](\mathbf{r})$$
$$= -v_k(\mathbf{r})\left(q(\mathbf{r}) + \frac{4\pi e^2 N_A}{kT}\sum_{i=1}^{m}Z_i^2 c_i^{\infty}\right) - 4\pi r_k(\mathbf{r}), \quad (21)$$

where $q(\mathbf{r})$ has been defined in Sec. II B.

In the general non-linear case, at variance to the generalized Poisson equation discussed in Sec. II, there is no general consensus on what flavor of the Poisson-Boltzmann equation is most appropriate for various problems. Therefore, a (M)PBe solver should allow to solve various formulations of this equation. Since it is unlikely that for all variants an action integral can be established (allowing for a minimization scheme), a self-consistent approach should be the most feasible algorithm.

Such a procedure has been implemented and it is detailed in Algorithm 3. The ionic charge density $\rho^{ions}[\phi]$ is included as source term to the charge density of the generalized Poisson equation which, in turn, is solved repeatedly until self-consistency between the potential and the ionic charge induced by it is reached. Starting with an initial input guess ρ_0^{ions} for the ionic charge density, on each self-consistent iteration, a generalized Poisson solver (Algorithm 1 or 2) is applied to numerically compute the electrostatic potential $\phi(\mathbf{r})$. Then, the ionic charge is computed using the new potential at step 5 [using Eq. (5)] and mixed with the old density by means of a linear scheme tuned by the parameter η (step 6).

In order to speed up performances of the (M)PBe solver, an improved version is reported in Algorithm 4. On each iteration k, the electrostatic problem at step 4 is solved only for the previous residual vector r'_{k-1} once ρ_k^{ions}

ALGORITHM 3. Self-consistent iterative procedure for the Poisson-Boltzmann equation.

1: set ρ_0^{ions}
2: **for** $k = 0, 1, \ldots$ **do**
3:     $\rho_k^{tot} = \rho + \rho_k^{ions}$
4:     solve $\nabla \cdot \epsilon \nabla \phi_k = -4\pi \rho_k^{tot}$ (Algorithm 1 or 2)
5:     compute $\rho_{k+1}^{ions} = \rho^{ions}[\phi_k]$
6:     $\rho_{k+1}^{ions} = \eta \rho_{k+1}^{ions} + (1-\eta)\rho_k^{ions}$
7:     $r_{k+1} = \rho_{k+1}^{ions} - \rho_k^{ions}$
8: **end for**

ALGORITHM 4. Improved self-consistent iterative procedure for the Poisson-Boltzmann equation.

1: set $\phi_1 = 0$, $\rho_0^{ions} = 0$, ρ_1^{ions}, $r'_0 = \rho$
2: **for** $k = 1, \ldots$ **do**
3:     $\rho_k^{tot} = r'_{k-1} + \rho_k^{ions} - \rho_{k-1}^{ions}$
4:     solve $\nabla \cdot \epsilon \nabla \phi'_k = -4\pi \rho_k^{tot}$ (Algorithm 2 with a residual vector r'_k)
5:     $\phi_k = \phi_k + \phi'_k$
6:     compute $\rho_{k+1}^{ions} = \rho^{ions}[\phi_k]$
7:     $\rho_{k+1}^{ions} = \eta \rho_{k+1}^{ions} + (1-\eta)\rho_k^{ions}$
8:     $r_{k+1} = \rho_{k+1}^{ions} - \rho_k^{ions}$
9: **end for**

---

has been updated at step 3. Then, the overall solution is given at step 5 as sum over all potential corrections ϕ'_k. This procedure substantially corresponds to the general self-consistent approach of Algorithm 3, now using on each step information of the previous as input guess. It is worth noting that the improved Algorithm 4 can be coupled only with the faster PCG solver (Algorithm 2) for the GPe.

A. Numerical performances

To show performances and accuracy of the Poisson-Boltzmann solver, both Algorithms 3 and 4 have been applied to analytical test cases. Following the strategy reported in Sec. II C, all the involved fields and functions, i.e., the potential, the charge density, and the operator have been discretized on the orthorhombic three dimensional grid and the same parameters have been used to set up all analytical functions. The electrostatic problem lies in a cavity where the great majority of the charge density is confined, described by means of a dielectric error function $\epsilon(\mathbf{r})$ (solid black line of Fig. 1). A normalized Gaussian function has been chosen for the electrostatic potential $\phi(\mathbf{r})$ (red dashed line of Fig. 1), whereas the charge density $\rho(\mathbf{r})$ has been derived from the fixed potential and dielectric functions, applying the Poisson-Boltzmann differential operator. To guarantee that mobile ion concentrations tend to zero inside the dielectric cavity mapped by $\epsilon(\mathbf{r})$, the following functional dependence has been chosen for the $\gamma[\epsilon](\mathbf{r})$ prefactor of Eq. (5),

$$\gamma[\epsilon](\mathbf{r}) = \frac{\epsilon(\mathbf{r})-1}{\epsilon_0 - 1}. \quad (22)$$

A monovalent ($Z_1 = -Z_2 = 1$) binary aqueous electrolyte solution has been considered, with a close packing coefficient $p = 0.74$, effective ionic radius $R_i = 3$ Å, and bulk ion concentrations $c_i^{\infty} = 100$ mol/m³ kept fixed for all ions. As discussed in Sec. III, the PCG algorithm has been applied to solve the linear Poisson-Boltzmann equation with the same GPe preconditioner of Eq. (13). In Fig. 3, the euclidean norm of the residual vector r_k (black squares) and the accuracy of the numerical solution (blue circles) have been reported. Similar performances have been found with respect to the generalized Poisson solver, reaching an accuracy of $\sim 10^{-10}$ with some ten iterations and proving that the PCG procedure is a well suited and fast method also for the linear regime of the PBe.
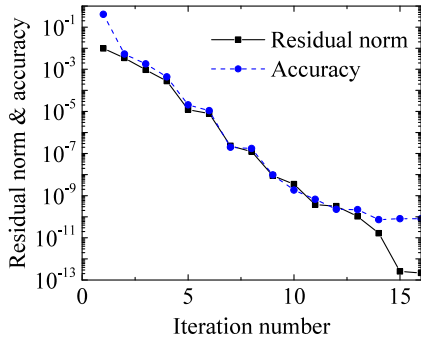
FIG. 3. Euclidean norm of the residual vector r_k (black squares) and accuracy of the numerical solution (blue circles) for the linear Poisson-Boltzmann equation with free boundary conditions (Algorithm 2 has been applied with $\mathcal{A}$ the linear Poisson-Boltzmann operator given by Eq. (21)).

In the general non-linear case, Algorithms 3 and 4 have been applied. An input guess $\rho_0^{ions} = 0$ has been chosen. We found that only a relatively small number of self-consistency iterations is needed in this approach to solve both the Poisson-Boltzmann and the modified Poisson-Boltzmann problems. Fig. 4 shows the Euclidean norm of the residual vector r_k (black squares) and the accuracy of the numerical solution (blue circles) for the modified Poisson-Boltzmann solver with free boundary conditions. The solver exhibits a fast convergence, reaching an accuracy of ~10^{-10} with some five iterations. The inset shows $\rho^{ions}[\phi]$ as given by Eqs. (5) and (19), revealing an ionic density saturation in the solvent for electrostatic potentials whose absolute value is higher than ~0.01 a.u. Numbers between round brackets represent the number of iterations needed to solve the GPe at step 4 of Algorithm 4 using the PCG scheme. The convergence criterion for the GPe solver has been fixed equal to the one of Algorithm 4, except for the first two iterations where very accurate potentials do not change the overall performances of the PBe solver. Furthermore, its behavior does not change with the boundary conditions which are managed by means of the generalized Poisson solver. The latter also fixes the final accuracy of the self-consistent procedure. It is worth noting that performances and parallel efficiency as well as boundary conditions of both GPe and (M)PBe solvers are, eventually, delegated to the underlying SPe solver.
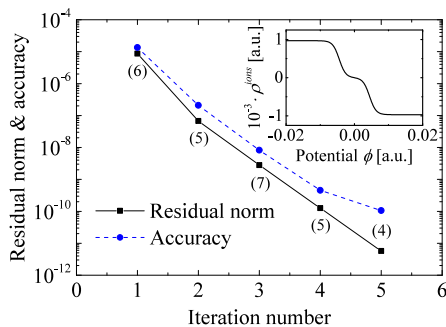

FIG. 4. Euclidean norm of the residual vector r_k (black squares) and accuracy of the numerical solution (blue circles) for the modified Poisson-Boltzmann solver with free boundary conditions. Numbers between round brackets represent the number of iterations needed to solve the GPe at step 4 of Algorithm 4. The inset shows $\rho^{ions}[\phi]$ given by Eqs. (5) and (19).

IV. ELECTRONIC-STRUCTURE COMPUTATIONS

Effects of complex wet environments surrounding an atomistic system can be approximately included into density functional calculations by simply introducing a dielectric cavity surrounding the atomistic system and taking in Eq. (1) the electrostatic potential $\phi[\rho](\mathbf{r})$ as solution of the generalized Poisson or Poisson-Boltzmann equation. The PCG solver for the GPe (Algorithm 2) has been implemented in the electronic-structure package BigDFT,[25–28] extending the capability of the code beyond vacuum-simulations.

Two distinct approaches have been implemented and tested to build up the dielectric cavity enveloping the atomic system. In both approaches, the cavity, mapped by the dielectric function $\epsilon(\mathbf{r})$, is fully differentiable and continuous in the whole simulation domain. In the first approach, the function $\epsilon(\mathbf{r})$ is defined starting from spherical-symmetric atom-centered cavities. Each sphere depends only on the radial distance with respect to a fixed atomic position. The whole rigid cavity is kept fixed during the whole SCF cycle in a DFT simulation.

On the other hand, it could be argued that regions occupied by atoms and their associated electronic charge density are strictly related. In other words, the actual value of the charge density might determine how much "space" is occupied by the solute. Starting from this idea, a cavity can be built up directly from the electronic density, as it has been shown in various publications.[15,29,30] In this second approach, the dielectric function is not explicitly space-dependent, but can be implicitly mapped by means of the electronic charge density.

A. Rigid cavity

The most widespread continuum solvation model, which tries to include the effects of a surrounding dielectric medium in an implicit way, is the polarizable continuum model developed by Tomasi and co-workers.[7,8,31] In the PCM formulation, the cavity surrounding the solute is sharp and discontinuous, and a polarization charge density is exactly localized at the vacuum-dielectric interface. In this way, the dielectric environment is represented by an effective surface polarization charge, reducing the three-dimensional dielectric problem into a two-dimensional one. Furthermore, the cavity can be considered rigid since it depends only on the atomic coordinates which does not vary during a SCF cycle in DFT simulations.

For a first test of the electrostatic solver integration (Algorithms 1 and 2) in an electronic-structure code, a PCM-like cavity has been considered. The dielectric function $\epsilon(\mathbf{r})$ is thus given by the product of spherical-symmetric atom-centered error functions. Differently from the early PCM model, we here define a cavity which is fully differentiable and continuous. In particular, for a system of N atoms of coordinates $\mathbf{R}_i$ (for $i = 1, \ldots, N$), the dielectric function $\epsilon(\mathbf{r})$ can be expressed as

$$\epsilon(\mathbf{r}, \{\mathbf{R}_i\}) = (\epsilon_0 - 1) \left\{ \prod_i h(d_i, \Delta; d(\mathbf{r}, \mathbf{R}_i)) \right\} + 1, \quad (23)$$

where ϵ_0 is the dielectric constant of the surrounding medium and the function h is defined by Eq. (17). In Eq. (23) $d(\mathbf{r}, \mathbf{R}_i) = \|\mathbf{r} - \mathbf{R}_i\|$, d_i are the cavity radii which depend on the particular atom species, and Δ a parameter (kept fix for all atoms) which controls the transition region ($\approx 4\Delta$ wide) from 0 to 1 where the polarization charge is located. Starting from Eq. (23), all vectors which explicitly depend on $\epsilon(\mathbf{r})$ can be analytically computed. The cavity is uniquely defined once $\mathbf{R}_i$, d_i, and Δ are fixed. All environment-dependent fields are calculated once at the start of the solver and kept fixed throughout the procedure.

It has to be noticed that this definition of the cavity relies on the explicit dependence of $\epsilon(\mathbf{r}, \{\mathbf{R}_i\})$ from the atomic coordinates $\mathbf{R}_i$ (and, consequently, of the system Hamiltonian). This dependence gives rise to additional terms when atomic forces are computed. The analytical rigid cavity above described should overcome this problem allowing a direct analytic calculation of such additional contributions. Furthermore, the values of d_i and of Δ have to be tuned by the user, usually by choosing a solvent-dependent scaling factor with respect to empirical Van der Waals radii.[32] A solution that would remove part of this arbitrariness should therefore avoid an explicit use of atomic coordinates in the cavity mapping. This will be the subject of Sec. IV B.

B. Charge-dependent cavity

For this definition of the cavity, the dielectric function does not explicitly depend in the atomic positions, but implicitly via the charge density ρ^{elec},

$$\epsilon(\mathbf{r}) = \epsilon[\rho^{elec}](\mathbf{r}). \quad (24)$$

This approach allows the cavity to self-consistently change during the SCF loop, strictly following the modification of the electronic charge density. A cavity surrounding an atomic system can be generated by means of two threshold parameter, i.e., ρ_{max} and ρ_{min}, fixing $\epsilon(\mathbf{r}) = 1$ in regions when $\rho^{elec}(\mathbf{r}) > \rho_{max}$ and $\epsilon(\mathbf{r}) = \epsilon_0$ when $\rho^{elec}(\mathbf{r}) < \rho_{min}$. Like d_i in the rigid case, ρ_{max} fix the width of the cavity, whilst the extension of the transition region, previously defined by Δ, is now tuned by ρ_{min}.

Several features make the charge-dependent cavity more advantageous with respect to the rigid one. First, once the electron charge density is given, only two parameters uniquely define both the cavity and the transition region for the whole atomic system. Furthermore, since the dielectric function does not explicitly depend on the atomic coordinates, the evaluation of ionic forces can be done without modifications with respect to a simulation in gas-phase.

Among several ways to define the functional dependence on the electronic charge density, the self-consistent continuum solvation (sccs) model developed in Ref. 15 has been implemented. It allows to fit experimental solvation energies on a set of 240 neutral solutes with a mean absolute error of 1.2 kcal/mol,

$$\epsilon(\rho^{elec}) = \begin{cases} 1 & \rho^{elec} > \rho_{max} \\ \epsilon_0^{w(\rho^{elec})} & \rho_{min} < \rho^{elec} < \rho_{max} \, , \\ \epsilon_0 & \rho^{elec} < \rho_{min} \end{cases} \quad (25)$$

where $w(x)$ is a continuous smooth function describing the transition region between vacuum (where atoms are placed) and the full dielectric medium,

$$w(x) = \frac{1}{2\pi} \left[z(x) - \sin(z(x)) \right], \quad (26)$$

$$z(x) = 2\pi \frac{\ln\left(\frac{\rho_{max}}{x}\right)}{\ln\left(\frac{\rho_{max}}{\rho_{min}}\right)}. \quad (27)$$

A good description of this region by means of Eq. (25) is mandatory for the procedure convergence as well as for the mapping of the polarization charge there confined. Since $\epsilon(\mathbf{r})$ explicitly depends on $\rho^{elec}(\mathbf{r})$, its variation needs to be included in the Kohn-Sham (KS) potential. Starting from Eq. (3) and integrating by parts, the electrostatic energy can be rewritten as

$$E_{es}[\rho] = \frac{1}{2} \int \rho \phi[\rho] d\mathbf{r} = \frac{1}{8\pi} \int \epsilon[\rho](\nabla \phi[\rho])^2 d\mathbf{r}. \quad (28)$$

Its functional derivative with respect to ρ gives the electrostatic potential $\phi(\mathbf{r})$ plus an additional term $v_\epsilon(\mathbf{r})$,

$$v_\epsilon(\mathbf{r}) = -\frac{1}{8\pi} \frac{d\epsilon(\rho^{elec}(\mathbf{r}))}{d\rho^{elec}} |\nabla \phi(\mathbf{r})|^2, \quad (29)$$

which has to be added to the KS potential.

C. Solvation free energies

In order to test the integration and performances of the generalized Poisson solver into *ab initio* codes, the whole procedure previously described, i.e., Algorithms 1 and 2 of Secs. II A and II B, rigid and charge-dependent cavities of Secs. IV A and IV B as well as the additional term of Eq. (29) have been integrated in the main BigDFT package.[28] This extension allows to handle complex wet environments in electronic-structure calculations, including implicitly the effects of a solvent surrounding an atomic system.

The electrostatic solvation energy is defined as difference between the total energy of a given atomic system in the presence of the dielectric environment and the energy of the same system in vacuum

$$\Delta G^{el} = G^{el} - G^0. \quad (30)$$

A full comparison with experimental solvation energies needs the inclusion of non-electrostatic contributions. In this case, the main terms in the solute Hamiltonian, as introduced by PCM,[8] are

$$\Delta G^{sol} = \Delta G^{el} + G^{cav} + G^{rep} + G^{dis} + G^{tm} + P\Delta V, \quad (31)$$

where ΔG^{el} is the electrostatic contribution, G^{cav} the cavitation energy, i.e., the energy necessary to build up the solute cavity inside the solvent medium. G^{rep} is a repulsion term representing the continuum counterpart of the short-range interactions induced by the Pauli exclusion principle, whilst G^{dis} reflects van der Waals interactions. The thermal term G^{tm} accounts for the vibrational and rotational changes and, finally, $P\Delta V$ includes volume change in the solute Hamiltonian.

The inclusion of all non-electrostatic contributions goes beyond the aim of the present paper, where testing of the generalized Poisson solver and its integration into first

TABLE I. Electrostatic solvation energies ΔG^{el} (in kcal/mol).

	PCM[a]	Rigid$_{BigDFT}$	sccs$_{QE}^{P}$ [b]	sccs$_{BigDFT}^{F}$
NH$_3$	−6.65	−6.28	−5.39	−5.35
H$_2$O	−8.98	−8.32	−8.21	−8.23
CH$_4$	−0.61	−1.20	−0.68	−0.63
CH$_3$OH	−6.78	−6.57	−5.89	−5.83
CH$_3$NH$_2$	−4.51	−5.71	−4.53	−4.45
CH$_3$CONH$_2$	−12.53	−12.97	−11.87	−11.87

[a]Polarizable continuum model results obtained with GAUSSIAN 09[36] and Pauling's set of atomic radii.[34]
[b]Self-consistent continuum solvation model results from periodic BC calculations in Ref. 15.

principle atomistic calculations is the main goal. Therefore, only electrostatic solvation energies have been computed for a set of small neutral organic molecules.

Water has been chosen as solvent for all cases, with a dielectric constant of $\epsilon_0 = 78.36$ (experimental value at low frequency and ambient conditions). Final energies for all molecules have been extracted after a full geometry optimization both in vacuum and in aqueous solution. In all cases, the surrounding dielectric medium lowers the total energy of the system with respect to vacuum, because the polarization of the dielectric stabilizes the solutes.

Table I reports electrostatic solvation energy ΔG^{el} obtained both with rigid and charge-dependent cavities under free boundary conditions. As previously stated, a critical point of PCM approaches is the choice of shape and size of the cavity, which should mimic the solute incorporating the whole atomic charge density. From its first formulation,[31] PCM atomic radii R_i were fixed proportional to the van der Waals radii, $R_i = f \cdot R_i^{vdW}$. In our rigid model, a proportional factor of $f = 1.2$ has been fixed and the Pauling's set of atomic radii has been considered[33–35] (except for the hydrogen atoms bound to heteroatoms which have a radius value of 1 Å). Having a further degree of freedom with respect to sharp PCM cavities, a $\Delta = 0.5$ a.u. has been tuned and kept constant for all atoms.

As reference, in Table I, PCM calculations together with sccs calculations from Ref. 15 have been reported. The first were performed with the GAUSSIAN 09 code,[36] but using the version of PCM which was the default in GAUSSIAN 03, as specified by the keyword g03defaults. Only electrostatic solvation effects were included in the calculation and, to simplify the comparison, the simple van der Waals surface was adopted with Pauling's set of atomic radii[34] (explicit hydrogens), but without the additional smoothing used to describe the solvent-excluded surface. The Perdew-Berke-Ernzerhof (PBE) functional[37] and the extended triple-zeta 6-311+g(d,p) basis set were used for both geometry optimizations and energy calculations, in vacuum and in solution, consistently with the setup used for the parameterization of the electrostatic solvation energy in sccs.[15]

In all calculations, soft norm-conserving pseudopotentials including non-linear core correction[38,39] along with PBE functional were used to describe the core electrons and exchange-correlation, respectively.
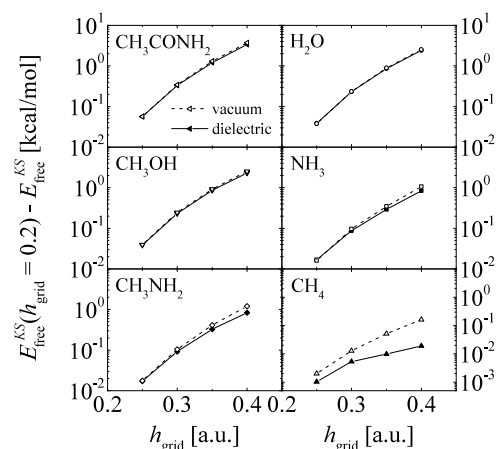

FIG. 5. Accuracy of the Kohn-Sham total energy with respect to the spatial grid h_{grid}. Calculations for all molecules have been done both in vacuum (empty symbols, dashed lines) and with a surrounding dielectric environment (filled symbols, solid lines) in free boundary conditions.

For the charge-dependent cavity, values of $\rho_{max} = 5 \cdot 10^{-3}$ a.u. and $\rho_{min} = 1 \cdot 10^{-4}$ a.u. have been fixed which produce a mean absolute error of 1.20 kcal/mol for the solvation energies of a database of 240 molecules.[15] A perfect agreement has been reached which confirms the performance and reliability of the integrated generalized Poisson solver.

In order to test and validate the whole electrochemical library in BigDFT, the effects of grid resolution and boundary conditions on the electrostatic solvation energies for the same set of neutral molecules have been investigated. The global accuracy is strictly related to the size of the simulation box and the spatial grid resolution h_{grid}. However, a decrease of the latter can affect the whole cost of the calculation both in time and memory usage. Consequently, it is worth to investigate the effects of the dielectric medium's inclusion in a DFT run with respect to the vacuum case.

In that respect, Kohn-Sham total energies have been extracted both in vacuum and in the presence of the solvent (water) as function of the spatial grid h_{grid}. Its accuracy is reported in Fig. 5 as difference with the reference value at $h_{grid} = 0.20$ bohr. Results show that the accuracy is not affected by the surrounding dielectric environment (filled symbols, solid lines) with respect to the vacuum case (empty symbols, dashed lines). Following the same guidelines, the accuracy of the electrostatic solvation energy [Eq. (30)] with respect to the spatial grid h_{grid} has been investigated and reported in Fig. 6. Results indicate that an $h_{grid} = 0.30$ bohr in BigDFT with free boundary condition provides electrostatic solvation energies with an accuracy lower than $\sim 10^{-2}$ kcal/mol.

Since isolated systems embedded in a wet environment are the subject of interest, spurious long range electrostatic interactions with periodic images due to artificially imposed periodicities along certain direction can be problematic. In the developed procedure, the boundary conditions enter through the preconditioner, i.e., the SPe solver. Since in the BigDFT Poisson solver, all the common boundary conditions such as free, wire, and surface are exactly implemented, such spurious interactions do not exist at any stage of our approach.
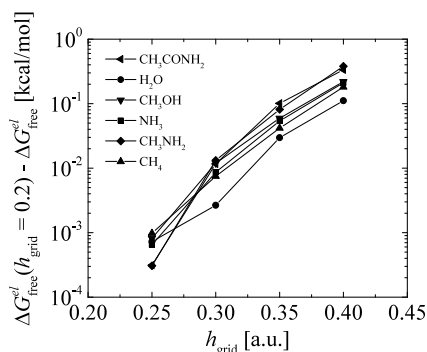
FIG. 6. Accuracy of the electrostatic solvation energy with respect to the spatial grid h_{grid} with free boundary conditions.

TABLE II. Molecule dipole norm in vacuum and water solvent (in Debye).

	Dipole^{vacuum}	Dipole^{water}
CH_3CONH_2	3.88	5.76
H_2O	1.81	2.41
CH_3OH	1.57	2.14
NH_3	1.49	1.98
CH_3NH_2	1.27	1.78
CH_4	0.00	0.00

Fig. 7 reports the difference between the electrostatic solvation energies computed with periodic and free boundary conditions as function of the periodic cell length and keeping fixed the spatial grid $h_{grid} = 0.20$ bohr. Molecules have been ordered according to the strength of their electrostatic dipole, from largest to smallest. Dipole values for each molecule are reported in Table II both in vacuum and in the presence of a water solvent. It can be noticed that the presence of the polar solvent increases the electrostatic dipoles of the molecules.[32] As it might be expected, the free BC calculation represents the asymptotic result for periodic BC run of increasing box sizes. Interactions with image-molecules are less relevant for molecules with small dipole moment like CH_4, but are not negligible for molecules with larger dipoles such as CH_3CONH_2. In such cases unrealistic large periodic boxes are required for periodic BC to reproduce the free boundary condition results with high accuracy.

Numerous processes of practical interest involve surfaces in contact with neutral or ionic solvents, leading to an induced polarization charge on the dielectric medium or an electric double layer. The BigDFT package allows to use exact surface boundary conditions avoiding spurious interaction in the direction orthogonal to the surface. To show a further application of the solvation library in BigDFT, a TiO_2 surface in contact with pure water has been simulated. The full DFT simulation in presence of the solvent has been initialized starting from its relaxed state in vacuum. Fig. 8 shows the

TiO_2 wet surface as well as isosurfaces of the polarization charge density ρ^{pol} as given by Eq. (10). In this test case, an electrostatic solvation energy of -46.60 kcal/mol has been recovered for the TiO_2 slab in contact with the dielectric medium.

D. Performances for a full SCF run

In this section, we report the performances of an entire electronic-structure calculation using the extended BigDFT package containing our new Poisson solver applied to non-vacuum environments. As a test system, we took the protein PDB ID: 1y49 (122 atoms), which was also chosen as a benchmark system in another recent publication on an optimized version of the COSMO[40] Poisson solver for dielectric environments.[41] The approach in this publication is quite different from ours. A small Gaussian type basis set was used, whereas we use a systematic basis set of wavelets. Also
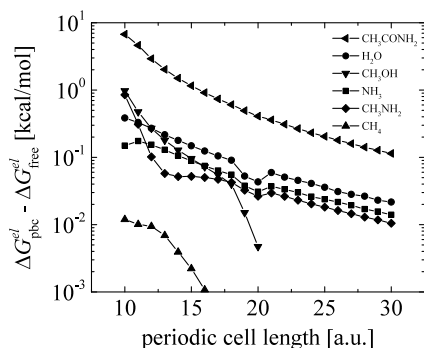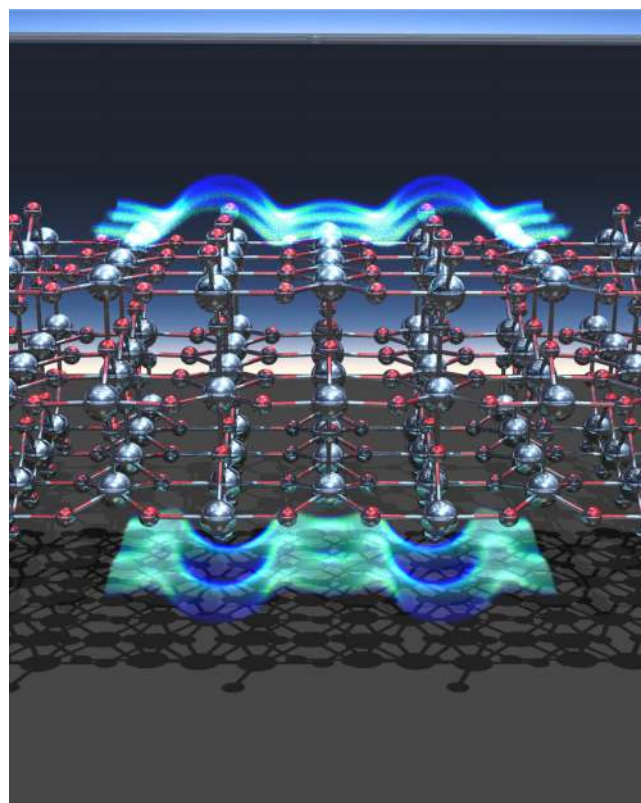

FIG. 7. Difference between the electrostatic solvation energy computed with periodic and free boundary conditions as a function of the periodic cell length. Molecules have been ordered as a function of their electrostatic dipole norm, from largest to smallest (see Table II).

FIG. 8. TiO_2 surface in contact with water: Isosurfaces of the polarization charge ρ^{pol} in the implicit dielectric medium.

the hardware targeted was different. We target traditional massively parallel supercomputers, whereas they targeted powerful single nodes that were enhanced with graphics processing unit (GPU) hardware. In spite of all the differences we will see that in both approaches, the overhead for a simulation in a dielectric medium compared to vacuum is rather small and that the total numerical effort is also quite similar. In our tests, the rigid PCM-like cavity has been used together with the parameter's setting of Sec. IV C and a spatial grid $h_{grid} = 0.40$ bohr. The direct minimization method of BigDFT[25] for the calculation of the converged wavefunction was used.

Both the developed preconditioned conjugate gradient and the self-consistent algorithm have been applied for the electrostatic problem (generalized Poisson equation) with and without the use of an input guess from previous routine calls. As accurate potentials are not needed during the first stages of the SCF wavefunction optimization loop, a dynamic exit threshold from Algorithms 1 and 2 based on the norm of the residual vector r_k has been implemented. It is allowed to vary between τ_{min} and τ_{gnrm}, where the last is the norm of the KS energy gradient from the previous wavefunction SCF iteration. A $\tau_{min} = 10^{-4}$ guarantees a correct SCF convergence while lowering the overall electrostatic calculation time.

The whole electrochemical library, namely, the generalized Poisson and the Poisson-Boltzmann solvers, has been fully embedded in the parallelization scheme of BigDFT, which exploits two levels of parallelization, i.e., MPI and OpenMP. For our test system, which has 179 orbitals, 45 MPI tasks with 4 OpenMP threads were used. The simulations were performed on a Cray XC30 system whose nodes are equipped with 8-core 64-bit Intel SandyBridge CPU (Intel® Xeon® E5-2670) and NVIDIA® Tesla® K20X. Fig. 9 reports a detailed schematic of timings for the total SCF convergence of BigDFT runs both in gas phase and with the inclusion of the

TABLE III. BigDFT timings and performances for 1y49 protein.

Algorithm	Environment	SCF loop (s)	GP solver (s)	Total GPe iterations	No. of SCF iterations
Vacuum solver	Vacuum	42	0.43	1	16
SC	Water	218	97	342	18
SC-IG	Water	91	23	60	18
PCG	Water	71	19	686	16
PCG-IG	Water	54	5.50	178	18

implicit solvent. Each column refers to a particular algorithm (SC and PCG) to solve the generalized Poisson equation (IG means the use of input guess). Colors indicate the load of the different timing categories. SCF vacuum data are also reported for comparison.

Fig. 9 shows the relative cost to include implicit solvent-solute interactions with respect to its gas phase using our generalized Poisson solver. These additional efforts can be visualized by the yellow and dark blue blocks which are related, respectively, to the electrostatic solver and communications. The ratio of the wavefunction optimization runtime in solvent and gas phase strongly varies with respect to the algorithm chosen. Using an input guess from the previous calculation strongly decreases the computational runtime for both approaches (PCG and SC). However, the best performances come out for the PCG algorithm with input guess (PCG-IG) where a ratio of 1.28 is recovered. So thanks to the PCG algorithm, which requires only vacuum Poisson solver calls and fully parallelizable operations (without the use of finite-difference filters like in the SC algorithm), a small overhead for the solvent calculation can be obtained. Similar performance ratios are also found for runs of the same system on a single node of the above mentioned machine, with 8 OpenMP threads, where a vacuum run takes 741 s and a PCG-IG run needed 857 s, i.e., only about 15% slower.

Table III reports run data for all single-point energy evaluations of Fig. 9. It shows the total time spent in the SCF wavefunction optimization loop, the time spent in the electrostatic solver, the sum of all generalized Poisson iterations during the whole SCF loop, and the number of SCF iterations needed to converge both for vacuum and the implicit solvent case.

V. CONCLUSIONS

In the present work, a library to handle complex wet-environments in electronic-structure calculations has been presented. It allows to include on the atomistic scale effects of an aqueous environment in an implicit way. The solver is able to handle both the generalized Poisson and several variants of the Poisson-Boltzmann equation. The core of the generalized Poisson solver is a preconditioned conjugate gradient algorithm which allows to numerically solve the minimization problem with some ten iterations. The same algorithm works for the linear case of the Poisson-Boltzmann
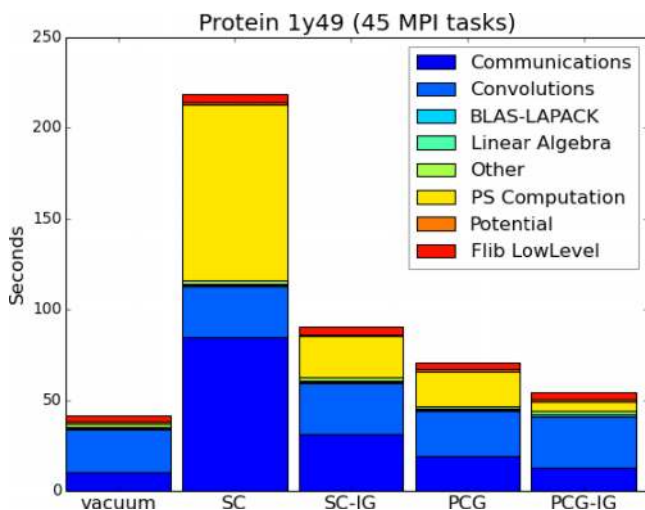

FIG. 9. Relative timings for full SCF convergence in a BigDFT run for a solvated 1y49 protein (122 atoms). Each column refers to a particular algorithm (SC and PCG) to handle the generalized Poisson equation (IG means the use of input guess). Colors indicate the load of the different run categories. SCF vacuum data are also reported for comparison. Absolute timings are reported in Table III.

equation, whilst for the general case, a self-consistent procedure has been implemented. The chosen preconditioner is based on the ISF Poisson solver for the standard Poisson equation, which can handle all common boundary conditions exactly. The code requires a small amount of memory and is very fast and in addition also highly parallelized. We have shown that coupled with BigDFT, our method can correctly reproduce electrostatic solvation energies of a set of small neutral organic molecules. Effects of grid resolution and boundary conditions on the electrostatic solvation energies have been also reported. The whole library will be released as an independent program suitable for integration in other electronic structure codes. A GPU-accelerated version of this software package is also in preparation, following the guidelines indicated in Ref. 42.

ACKNOWLEDGMENTS

[1] I. Dabo, Y. Li, N. Bonnet, and N. Marzari, "*Ab initio* electrochemical properties of electrode surfaces," in *Fuel Cell Science: Theory, Fundamentals and Bio-Catalysis*, edited by A. Wieckowski and J. Nørskov (John Wiley & Sons, Inc., 2010), pp. 415–431.

[2] C. Reichardt, *Solvents and Solvent Effects in Organic Chemistry*, 3rd ed. (WILEY-VCH, Weinheim, 2003).

[3] J. A. White, E. Schwegler, G. Galli, and F. Gygi, J. Chem. Phys. **113**, 4668 (2000).

[4] S. Goedecker, J. Chem. Phys. **120**, 9911 (2004).

[5] B. Schaefer, S. Mohr, M. Amsler, and S. Goedecker, J. Chem. Phys. **140**, 214102 (2014).

[6] L. Onsager, J. Am. Chem. Soc. **58**, 1486 (1936).

[7] J. Tomasi and M. Persico, Chem. Rev. **94**, 2027 (1994).

[8] J. Tomasi, B. Mennucci, and R. Cammi, Chem. Rev. **105**, 2999 (2005).

[9] C. J. Cramer and D. G. Truhlar, Chem. Rev. **99**, 2161 (1999).

[10] L. Greengard and V. Rokhlin, J. Comput. Phys. **73**, 325 (1987).

[11] M. C. Strain, G. E. Scuseria, and M. J. Frisch, Science **271**, 51 (1996).

[12] L. Genovese, T. Deutsch, A. Neelov, S. Goedecker, and G. Beylkin, J. Chem. Phys. **125**, 074105 (2006).

[13] L. Genovese, T. Deutsch, and S. Goedecker, J. Chem. Phys. **127**, 054704 (2007).

[14] A. Cerioni, L. Genovese, A. Mirone, and V. A. Sole, J. Chem. Phys. **137**, 134108 (2012).

[15] O. Andreussi, I. Dabo, and N. Marzari, J. Chem. Phys. **136**, 064102 (2012).

[16] J.-L. Fattebert and F. Gygi, Int. J. Quantum Chem. **93**, 139 (2003).

[17] M. Gouy, J. Phys. Theor. Appl. **9**, 457 (1910).

[18] D. L. Chapman, Philos. Mag. **25**, 475 (1913).

[19] O. Stern, Z. Electrochem. **30**, 508 (1924), http://chemport.cas.org/cgi-bin/sdcgi?APP=ftslink&action=reflink&origin=ACS&version=1.0&coi=1%3ACAS%3A528%3ADyaB2MXls1Wh&md5=0bb6a0352af08f94aa161e9c561d8784.

[20] J. Bikerman, Philos. Mag. **33**, 384 (1942).

[21] M. Otani and O. Sugino, Phys. Rev. B **73**, 115407 (2006).

[22] R. Jinnouchi and A. B. Anderson, Phys. Rev. B **77**, 245417 (2008).

[23] I. Dabo, "Towards first-principles electrochemistry," Ph.D. thesis, MIT, Cambridge, MA (2008).

[24] J. J. López-García, J. Horno, and C. Grosse, Langmuir **27**, 13970 (2011).

[25] L. Genovese, A. Neelov, S. Goedecker, T. Deutsch, S. A. Ghasemi, A. Willand, D. Caliste, O. Zilberberg, M. Rayson, A. Bergman, and R. Schneider, J. Chem. Phys. **129**, 014109 (2008).

[26] L. Genovese, B. Videau, M. Ospici, T. Deutsch, S. Goedecker, and J.-F. Mhaut, C. R. Mec. **339**, 149 (2011).

[27] S. Mohr, L. E. Ratcliff, P. Boulanger, L. Genovese, D. Caliste, T. Deutsch, and S. Goedecker, J. Chem. Phys. **140**, 204110 (2014).

[28] BigDFT code website, www.bigdft.org.

[29] J.-L. Fattebert and F. Gygi, J. Comput. Chem. **23**, 662 (2002).

[30] D. A. Scherlis, J.-L. Fattebert, F. Gygi, M. Cococcioni, and N. Marzari, J. Chem. Phys. **124**, 074103 (2006).

[31] S. Miertuš, E. Scrocco, and J. Tomasi, Chem. Phys. **55**, 117 (1981).

[32] M. Orozco and F. J. Luque, Chem. Rev. **100**, 4187 (2000).

[33] T. Mineva, N. Russo, and E. Sicilia, J. Comput. Chem. **19**, 290 (1998).

[34] L. Pauling, *The Nature of the Chemical Bond*, 3rd ed. (Cornell University Press, Ithaca, NY, 1960).

[35] CRC Handbook of Chemistry and Physics, 96th edition, electronic release, 2015-2016, www.hbcpnetbase.com.

[36] M. J. Frisch, G. W. Trucks, H. B. Schlegel *et al.*, GAUSSIAN 09, REVISION A.1, GAUSSIAN, INC., WALLINGFORD, CT, 2009 .

[37] J. P. Perdew, K. Burke, and M. Ernzerhof, Phys. Rev. Lett. **77**, 3865 (1996).

[38] S. Goedecker, M. Teter, and J. Hutter, Phys. Rev. B **54**, 1703 (1996).

[39] A. Willand, Y. O. Kvashnin, L. Genovese, l. Vázquez-Mayagoitia, A. K. Deb, A. Sadeghi, T. Deutsch, and S. Goedecker, J. Chem. Phys. **138**, 104109 (2013).

[40] A. Klamt and G. Schuurmann, J. Chem. Soc., Perkin Trans. **2**, 799 (1993).

[41] F. Liu, N. Luehr, H. J. Kulik, and T. J. Martínez, J. Chem. Theory Comput. **11**, 3131 (2015).

[42] N. Dugan, L. Genovese, and S. Goedecker, Comput. Phys. Commun. **184**, 1815 (2013).