"This document is the Accepted Manuscript version of a Published Work that appeared in final form in Journal of Chemical Theory and Computation, copyright © American Chemical Society after peer review and technical editing by the publisher. To access the final edited and published work see http://pubs.acs.org/doi/abs/10.1021/acs.jctc.5b00949."

Multigrid-Based Methodology for Implicit Solvation Models in Periodic DFT

Miquel Garcia-Ratés,* and Núria López*

Institute of Chemical Research of Catalonia (ICIQ), The Barcelona Institute of Science and Technology, Avinguda dels Països Catalans 16, 43007 Tarragona, Spain

> E-mail: mgarciar@iciq.es; nlopez@iciq.es Phone: +34 977 920229. Fax: +34 977 920231

KEYWORDS: Solvation, Periodic Boundary Conditions, Plane Waves, Multigrid

Abstract

Continuum solvation models have become a widespread approach for the study of environmental effects in Density Functional Theory (DFT) methods. Adding solvation contributions mainly relies on the solution of the Generalized Poisson Equation (GPE) governing the behavior of the electrostatic potential of a system. Although multigrid methods are especially appropriate for the solution of partial differential equations, up to now, their use is not much extended in DFT-based codes because of their high memory requirements. In this Article, we report the implementation of an accelerated multigrid solver-based approach for the treatment of solvation effects in the Vienna ab initio Simulation Package (VASP). The stated implicit solvation model, named VASP-MGCM (VASP-Multigrid Continuum Model), uses an efficient and transferable algorithm for the product of sparse matrices that highly outperforms serial multigrid solvers. The calculated solvation free energies for a set of molecules, including neutral and ionic species, as well as adsorbed molecules on metallic surfaces, agree with experimental data and with simulation results obtained with other continuum models.

^{*}To whom correspondence should be addressed

1 Introduction

Most processes in the field of physics, chemistry and biology occur in the presence of a liquid phase. Solvent effects play a major role in phenomena related to electrochemistry, photochemistry, heterogeneous catalysis, energy conversion/storage, and many chemical reactions take place at the solid/liquid interface.^{1,2} Liquid molecules (i) stabilize or destabilize adsorbed states on metal surfaces thus affecting their reactivity and selectivity,³ (ii) influence surface structure of electrodes in batteries⁴ or (iii) affect the self-assembly of nanoparticles which, in turn, modifies the size and shape of the resulting superstructures.⁵ Having a detailed molecular description of the mechanisms governing these phenomena is essential for a tailored design of new materials with different functionalities for a wide variety of applications.

Computer simulations, and particularly density functional theory (DFT) based methods, allow the understanding of the thermodynamics and kinetics of adsorbed molecules on surfaces.⁶ Such methods, when properly used, can retrieve even quantitative parameters for gas-solid reactions. However, these models present a weak point as modeling the liquid phase and introducing solvent contributions is far from trivial. An accurate treatment of solvation effects in computational studies of systems in contact with liquid phases is of paramount importance to obtain meaningful results to be compared with experimental data.⁷ In this context, the inclusion of the solvent can be done either explicitly or implicitly. The first approach involves the computation of solute-solvent and solvent-solvent interactions with a subsequent increase in the computational cost of the calculations as compared to the gas phase situation. In the case of periodic systems, simulation boxes need to be large enough to avoid undesired interactions between periodic images. The remaining vacuum has to be filled, then, with a huge number of solvent molecules. Moreover, either longtime scale molecular dynamics (MD) or Monte Carlo (MC) simulations need to be performed to average microscopic quantities for the calculation of thermodynamic properties of interest.⁸ However, in the case of quantum systems, the use of first principles-molecular dynamics simulations is restricted to rather small systems (up to 10^3 atoms, and short time scales, 20 ps) for which only one or very few trajectories are sampled. The accessible dynamic properties and correlation lengths are then very small.⁹ This makes this method almost impractical for the modeling of complex reaction networks taking place for solvated species. The second strategy encompasses the so-called continuum models, where the solute is treated quantum mechanically, while the solvent is represented as a structureless continuum medium in which the solvent properties are implicitly averaged. In order to retain the solvent configuration around the solute, which is especially important when solvent molecules form hydrogen bonds with it, the first solvation shells are, in general, explicitly accounted for (see Figure 1).



Figure 1: Schematic view of implicit solvation. The solute, originally in the vacuum, is placed in a cavity (gray area) up to the first solvation layer (orange area) within the solvent, which is modeled as a continuous medium. The solvent molecules in the orange area (not shown for simplicity) are treated explicitly to retain the particular solute-solvent interactions.

The most used continuum solvent approach is the polarizable continuum model (PCM).^{10–12} In this method, the solute is placed inside a cavity built, in general, as a set of interlocking spheres centered on solute atoms.^{13–15} The cavity surface is divided into small tesserae, and the polarization of the solvent is represented by surface charges placed on their center. Although PCM provides an accurate way of calculating physical properties of solvated systems, it has a major drawback: the choice of the solute cavity, which can be defined in different ways.^{16,17} Furthermore, discontinuities in the calculated atomic forces can appear due to the fact that charges are not distributed continuously over the surface cavity. Such discontinuities in the potential energy surface of the solute have been overcome by introducing switching functions for the surface grid points^{18,19} or using a continuous surface charge (CSC) formalism.²⁰ However, even if the use of PCM is appropriate when dealing with isolated systems, it is not the most convenient method when trying to simulate periodic systems, where the main quantities (like the electrostatic potential generated by

the surface charges) should be represented as a sum of plane waves (PWs).

In the present study, we have implemented from scratch a fast multigrid solver for the electrostatic problem into the Vienna ab initio Simulation Package (VASP) for solvent PCM in PW.²¹ VASP is a plane-wave based DFT program, aimed at performing ab initio total-energy and MD calculations for a large number of different systems, that is mainly used in the field of heterogeneous catalysis. The aim of this study is the incorporation of solvation effects into VASP using the Fattebert and Gygi implicit solvation model including nonelectrostatic contributions to the free energy of solvation.^{22,23} Although multigrid solvers usually rely on parallel algorithms, we propose a serial multigrid scheme, called VASP-MGCM (VASP-Multigrid Continuum Model), that solves the electrostatic problem in real space at a comparable cost. This approach is justified by the fact that the largest number of grid points in the simulated systems is about 10⁷, a number that a serial solver can afford. The efficient programming of the algorithm, that involves the derivation of a fast computation method for the product of large sparse matrices, results in a low increase in the computational cost for the calculations as compared to the same simulations in vacuum. Our results for some of the systems are compared to those obtained both from a recent implementation of solvation into VASP made by Mathew et al.,²⁴ and to available experimental data.

2 Revision of the state-of-the-art of including PCM in PW

Since the past decade, a set of continuum models aiming at including solvation effects into DFT periodic plane-wave (PW) based codes has been developed (see scheme in Figure 2). The first relevant model was proposed by Fattebert and Gygi. It defined the dielectric medium as a smooth function of the solute electronic charge density.^{22,23} In this way, both the dielectric permittivity of the medium and the solute charge density depend self-consistently one on the other. Extensions of this model have been done, mainly, to add nonelectrostatic contributions to the solvation free energy and to solve convergence issues in the electronic self-consistent cycle.^{25–27}

Despite the fact that continuum solvation models involve a drastic reduction in the computa-



Figure 2: Implicit solvation in periodic systems. As opposed to PCM, identical copies of the simulation box are distributed throughout each of the three space directions. For simplicity, we just show the periodic images of the solute atoms in one direction (pale-colored atoms), the solute cavity is shown in gray.

tional cost of DFT calculations as compared to the explicit approach, the solution of the electrostatic equations with enough accuracy can also become very demanding. The partial differential equations (PDEs) to be handled have, in general, a large number of unknowns and, in practice, two methodologies are adopted to solve them numerically: Fast Fourier Transform (FFT) based algorithms and multigrid methods. The former is the most popular approach to solve PDEs on 3D grids due to the scalability of its parallel formulation.²⁸ FFT highly optimized parallel libraries like FFTW²⁹ and MKL³⁰ are available and the corresponding subroutines can be called at any stage in a DFT code. Multigrid methods, on the other hand, are an effective group of algorithms for solving PDEs using a set of grids with different sizes.³¹ While basic iterative methods rapidly reduce the high frequency components of the solution error rapidly, they are not good at smoothing out the long wavelength part of the error. The basic idea behind multigrid is to approximate the low frequency components of the error by projecting them to coarse versions of the original fine grid where the numerical problem is to be solved. Therefore, the multigrid scheme, combined with iterative methods, accelerates the convergence of direct and iterative methods, especially for large systems.

When solvation effects need to be included in electronic-structure PW-based calculations, the main step is the solution of the Generalized Poisson Equation (GPE) (see eq 3 in Section 3) on a 3D grid. Due to the scalability of FFT schemes, some strategies to solve the GPE are based on the use of FFTs. In particular, solvation models have been implemented into well-known DFT codes by using, for instance, FFT-based versions of the preconditioned conjugate gradient algorithm,²⁴ or iterative procedures that make extensive use of the FFT.^{26,27} Although the arithmetic cost of multigrid is of $\mathcal{O}(N)$, while FFT-based methods require $\mathcal{O}(N \log N)$ operations (N is the system size), the use of such methods in the context of implicit solvation models is less extended, and usually restricted to small systems.^{25,32} Indeed, the multigrid algorithm depends on a large number of parameters (e.g., the order of accuracy of the transfer operators between grids, the discretization order of the fine-grid differential equation, the total number of grids, or the number of smoothing cycles). Therefore, an inefficient implementation of the algorithm with respect to all these parameters can result in a high computational cost. Nonetheless, multigrid methods present many advantages in the solution of PDEs as they can be adapted to manage any kind of geometries, including irregularly-shaped domains and any type of boundary conditions, and can be applied to nonlinear problems. The use of multigrid techniques is, thus, especially suited for solving the electrostatic problem in quantum systems provided that the algorithm is efficient enough.

3 Theoretical Basis

3.1 Electrostatics

When a solute is immersed in a dielectric medium, the Gauss law, in CGS units, reads as

$$\nabla \cdot \mathbf{E} = 4\pi (\rho(\mathbf{r}) + \rho_{\text{pol}}(\mathbf{r})) \tag{1}$$

with **E** being the total electric field vector, ρ the solute charge density, and ρ_{pol} the polarization charge density. This last term arises from the fact that the dielectric medium is polarized due to the electric field generated by the solute. Since $\rho_{pol}(\mathbf{r}) = -\nabla \cdot \mathbf{P}$ where **P** is the polarization vector, and $\mathbf{P} = \chi \mathbf{E}$ with χ being the susceptibility of the medium, eq 1 can be rewritten as

$$\nabla \cdot \left[(1 + 4\pi \chi(\mathbf{r}))\mathbf{E} \right] = 4\pi \rho(\mathbf{r}) \tag{2}$$

Here, the term $1 + 4\pi\chi(\mathbf{r})$ is also known as dielectric permittivity, that we denote by $\varepsilon(\mathbf{r})$, and gives a measure of how the interaction between two charges is screened with respect to the vacuum when being placed in a dielectric medium. At the same time, the total electric field can be described as the gradient of the total electrostatic potential ϕ ; $\mathbf{E} = -\nabla \phi(\mathbf{r})$. Eq 2 then becomes

$$\nabla \cdot [\boldsymbol{\varepsilon}(\mathbf{r}) \nabla \boldsymbol{\phi}(\mathbf{r})] = -4\pi \boldsymbol{\rho}(\mathbf{r}) \tag{3}$$

which is known as the Poisson equation for an inhomogeneous medium or Generalized Poisson Equation (GPE). If $\varepsilon(\mathbf{r}) = 1$ everywhere, the Poisson equation $\nabla^2 \phi(\mathbf{r}) = -4\pi \rho(\mathbf{r})$ is recovered. Once the value for $\phi(\mathbf{r})$ is known, and introducing the electric displacement vector $\mathbf{D} = \mathbf{E} + 4\pi \mathbf{P}$, the electrostatic energy of the system, E_{el} , is calculated straightforwardly

$$E_{\rm el} = \frac{1}{8\pi} \int \mathbf{E} \cdot \mathbf{D} \, d\mathbf{r} = \frac{1}{8\pi} \int \mathbf{E} \cdot \mathbf{E} \, d\mathbf{r} + \frac{1}{2} \int \mathbf{E} \cdot \left(\frac{\boldsymbol{\varepsilon}(\mathbf{r}) - 1}{4\pi}\right) \mathbf{E} \, d\mathbf{r} = \frac{1}{8\pi} \int \boldsymbol{\varepsilon}(\mathbf{r}) |\nabla \phi(\mathbf{r})|^2 d\mathbf{r} \quad (4)$$

3.2 Density Functional Theory

In the context of DFT calculations, the term in eq 4 is added to the Kohn-Sham (KS) energy functional E_{KS}^{33}

$$E_{\rm KS} = K + E_{\rm xc} + E_{\rm el} + E_{\rm nel} + E_{\rm ext}$$
⁽⁵⁾

where K and E_{xc} account for the electronic kinetic and exchange-correlation energies, E_{nel} for the nonelectrostatic term, and E_{ext} is the energy associated to an external potential. The functional

derivative of $E_{\rm KS}$ with respect to the electronic charge density $\rho_{\rm el}$ results in a kinetic term plus the so called Kohn-Sham effective potential, $V_{\rm eff}$

$$V_{\rm KS} = \frac{\delta E_{\rm KS}}{\delta \rho_{\rm el}} = V_K + \underbrace{V_{xc} + V_{\rm el} + V_{\rm nel} + V_{\rm ext}}_{V_{\rm eff}} \tag{6}$$

The term $V_{\rm el} = \delta E_{\rm el} / \delta \rho$ can be further written as

$$\frac{\delta E_{el}}{\delta \rho_{el}} = \phi(\mathbf{r}) - \frac{1}{8\pi} |\nabla \phi(\mathbf{r})|^2 \frac{\delta \varepsilon(\mathbf{r})}{\delta \rho_{el}(\mathbf{r})}$$
(7)

Due to the presence of the dielectric medium, the force \mathbf{F}_i acting on an atom *i* of the solute located at \mathbf{R}_i is also different to that acting on the solute in the vacuum. According to the Hellman-Feynman theorem, and using the result in eq 4, the \mathbf{F}_i forces are given by

$$\mathbf{F}_{i} = -\frac{\partial E_{\mathrm{el}}}{\partial \mathbf{R}_{i}} = -\int \phi(\mathbf{r}) \frac{\partial \rho}{\partial \mathbf{R}_{i}} d\mathbf{r}$$
(8)

For a complete demonstration of eq 8 see ref [26].

The nonelectrostatic contribution, E_{nel} , to E_{KS} is calculated as a unique term accounting for cavitation, dispersion, and repulsion effects. We have adopted the formulation of Cococcioni et al.³⁴ (extended in refs. [25, 26]) that states that E_{nel} is equal to the quantum surface of the solute up to an effective surface tension. Although E_{nel} is shown to be nonlinear with the solvent-accessible area for protein-protein complexes,³⁵ such definition is extended in continuum-solvent models yielding good results for solvation energies for large sets of molecules.^{24–27} The expressions for E_{nel} and V_{nel} are provided in the SI.

The last point that should be covered to include the effect of the solvent on a DFT code is the functional form of the dielectric function $\varepsilon(\mathbf{r})$. This function must fulfill some conditions like: being flat inside the solute and the solvent, being smooth enough and reaching a value of 1 inside the solute and ε_0 , the permittivity of the bulk, in the solvent. Instead of defining the solutesolvent interface as a cavity made from interlocking spheres centered on solute atoms, as done in the polarizable continuum model (PCM), we have adopted the dielectric function proposed by Fattebert and Gygi^{22,23} which is an explicit function of ρ_{el} ,

$$\boldsymbol{\varepsilon}(\mathbf{r}) = 1 + \frac{\boldsymbol{\varepsilon}_0 - 1}{1 + (\boldsymbol{\rho}_{el}(\mathbf{r})/\boldsymbol{\rho}_0)^{2\beta}}$$
(9)

and depends on three parameters ε_0 , β , and ρ_0 . The parameter ρ_0 corresponds to the critical density in the middle of the solute-solvent interface, while β controls the width of the interface (see Figure 3). To avoid solvent penetration into the region occupied by the solute atoms, an additional charge density can be added to ρ_{el} in eq 9.



Figure 3: Variation of the dielectric permittivity $\varepsilon(\rho)$ with respect to the parameter β (the values of ε_0 and ρ_0 are fixed to 78.5 and $7.8 \cdot 10^{-4}$ a.u., respectively). The intersection between the three curves falls in the middle of the solute-solvent interface. The width of the interface for $\beta = 1$ and $\beta = 2$ is delimited by dashed lines.

3.3 Corrections

At this point, one has all the ingredients to incorporate solvent effects into a plane-wave DFT based code. This is done just by adding the corrections given in eqs 3, 4, 7 and 8, and the nonelectrostatic terms into the corresponding equations in the vacuum. In summary, if we use the subscript 0 for quantities computed in the vacuum, we should make the following main changes into the code:

• The KS local potential should be corrected by the term $V_{\rm corr}$

$$V_{\rm corr}(\mathbf{r}) = \phi(\mathbf{r}) - \frac{1}{8\pi} |\nabla\phi(\mathbf{r})|^2 \frac{\delta\varepsilon(\mathbf{r})}{\delta\rho_{\rm el}(\mathbf{r})} - \phi_0(\mathbf{r}) + V_{\rm nel}$$
(10)

• A correction E_{corr} is made on the KS energy functional

$$E_{\text{corr}} = \frac{1}{8\pi} \int \left[\varepsilon(\mathbf{r}) |\nabla \phi(\mathbf{r})|^2 - |\nabla \phi_0(\mathbf{r})|^2 \right] d\mathbf{r} + E_{\text{nel}}$$
(11)

• Hellman-Feynman forces are corrected by the term \mathbf{F}_{corr}

$$\mathbf{F}_{\text{corr}} = -\int (\phi(\mathbf{r}) - \phi_0(\mathbf{r})) \frac{\partial \rho(\mathbf{r})}{\partial \mathbf{R}} d\mathbf{r}$$
(12)

As mentioned above, the terms that account for nonelectrostatic effects are just shown in the Supporting Information for simplicity.

4 Solver Implementation and Computational Details

Once the implicit solvation model is stated, eq 3 has to be solved numerically to include the effect of solvation into the VASP code. For that purpose, we have programmed a set of Fortran 90 subroutines that have been coupled to VASP. Most of the calculations have been done in real space and in serial mode. However, we benefit from the Fast Fourier Transform (FFT) parallel libraries to transfer quantities between reciprocal and real space. To solve eq 3, we first rewrite it by making the scalar product $(\frac{\partial}{\partial x_1}, \frac{\partial}{\partial x_2}, \frac{\partial}{\partial x_3}) \cdot (\varepsilon \frac{\partial \phi}{\partial x_1}, \varepsilon \frac{\partial \phi}{\partial x_2}, \varepsilon \frac{\partial \phi}{\partial x_3})$, where $x_1 = x, x_2 = y, x_3 = z$, and using the product rule for the derivatives $(\varepsilon \phi')'$

$$\sum_{i=1}^{3} \left[\frac{\partial \varepsilon}{\partial x_i} \frac{\partial \phi}{\partial x_i} + \varepsilon \frac{\partial^2 \phi}{\partial x_i^2} \right] = -4\pi\rho \tag{13}$$

The solute charge density, ρ in the right hand side is the sum of an electronic and an ionic charge density. For the latter, denoted as ρ_{ion} , we have modeled it by gaussian distributions centered on the position of solute atoms. For an ion *i* with effective charge Z_i , $\rho_{ion}(\mathbf{r})$ reads as

$$\rho_{\text{ion}}(\mathbf{r}) = -\frac{Z_i}{(2\pi\sigma^2)^{3/2}} \exp\left(-\frac{(\mathbf{r}-\mathbf{R}_i)^2}{2\sigma^2}\right)$$
(14)

with σ being the width of the gaussian. We have chosen a value of σ around 5 times the grid spacing.

Eq 13 is discretized by finite differences on a cell-centered grid of grid spacing h_1 , h_2 , h_3 along x_1 , x_2 , and x_3 directions. Periodic boundary conditions are applied in each space direction. The derivatives of ε and ϕ are approximated to sixth order using the weights derived in ref [36].

$$\frac{d^m f}{dx_i^m}\Big|_{x_i=x_{i_0}} = \frac{1}{h_i^m} \sum_{n=-3}^3 w_{m,n} f(x_{i_0} + nh_i)$$
(15)

Here, *m* is the order of the derivative, and $w_{m,n}$ are the corresponding weights (see SI for the values). Equation 13 is rewritten using the approximations for the derivatives in eq 15, and a system of *N* equations with *N* unknows is obtained

$$A_h \phi_h = f_h \tag{16}$$

where A_h accounts for the left hand side matrix acting on the electrostatic potential ϕ , while $f = -4\pi\rho$. The subscript *h* stands for the grid spacing vector $\mathbf{h} = (h_1, h_2, h_3)$. Details for the expression of A_h are available in the SI. To solve eq 16 we employ a serial multigrid solver programmed from scratch. Multigrid methods accelerate the numerical convergence rate for the solution of a system of equations with respect to direct or iterative methods. For that purpose, we use

a hierarchy of grids $\Omega_1, \Omega_2, \ldots, \Omega_{N_g}$ where Ω_1 corresponds to the finest grid, $2\mathbf{h}^{\Omega_{l+1}} = \mathbf{h}^{\Omega_l}$ and N_g is the total number of grids. As VASP uses an even number of grid points, we have adopted a cellcentered approach. Regarding transfer operators between grids, we employ trilinear interpolation to map function values from a coarse to a fine grid. Restriction operator R is constructed as a scaled adjoint of interpolation operator I as indicated in ref [31]. The corresponding stencils are provided in the SI. Left hand side matrices A in the coarse grid equation are obtained through the Galerkin coarse grid approximation; $A_i = RA_{i-1}I$, with $i = 2, ..., N_g$. Because of the large size of matrices R, A and I in the finest grids, their product is the most time-consuming operation in a multigrid algorithm. Nevertheless, as all three matrices are sparse, we have implemented an algorithm that speeds up sparse matrix-matrix products considerably. Here, the value and column for the nonzero elements in a row are stored in two matrices. With the aid of a third auxiliary matrix, that controls which elements of both matrices need to be multiplied, the matrix-matrix product is performed in an efficient and fast way (see Appendix A for details). Multigrid V-cycles with Gauss-Seidel smoothing³¹ are performed until a convergence criterium for ϕ is reached. For that purpose, at the end of each multigrid cycle we calculate the percentage error, e_{ϕ} , of the electrostatic potential. If this quantity is lower than a chosen tolerance, the iterative process stops. Gauss-Jordan elimination³⁷ is used to solve the system in the coarsest grid. A general scheme of the process followed to compute the electrostatic potential into VASP is shown in Figure 4.

Once the electrostatic potential is calculated, the additional term defined in eq 10 is added to the Kohn-Sham Hamiltonian in VASP, and the total free energy and force are corrected. To test the convergence of the multigrid solver we have considered a maximum of $N_g = 6$ grids and 8 V-cycles, the number of pre- and post-smoothing Gauss-Seidel iterations being equal to 10. As we choose water as the solvent, the corresponding parameters in eq 9 are those determined in ref [22]; that is $\varepsilon_0 = 78.5$, $\beta = 1.3$, and $\rho_0 = 4 \cdot 10^{-4}$ a.u.

Solvation energies have been calculated through DFT calculations using the VASP code,²¹ version 5.3.5, with the Perdew-Burke-Ernzerhof (PBE) GGA exchange-correlation functional.³⁸ The valence electrons are described by plane waves with a cutoff energy of 450 eV while the core



Figure 4: Calculation of the electrostatic potential, $\phi(\mathbf{G})$, into VASP. The vector \mathbf{G} accounts for a reciprocal space vector. A detailed scheme of the multigrid solver algorithm is provided in the SI.

electrons are represented by projector augmented wave (PAW) pseudopotentials.³⁹ For the Pt(111) surface the slab contains 48 atoms in four layers, while for the (0001) surface of Ru a slab with 36 atoms in four layers was employed. In any case, the vacuum between slabs is 13Å and the Brillouin zone is sampled by a $3x_3x_1$ Γ -centered k-points mesh constructed through the Monkhorst-Pack scheme.⁴⁰ The convergence criteria for electronic and ionic minimization is set to 10^{-6} eV and 0.015 eV/Å, respectively. A Gaussian smearing function with a width of 0.03 eV is used to broaden the energy levels near the Fermi level. For the case of isolated molecules, we have carried out geometry optimizations, both in the gas phase and using the SMD implicit solvent,⁴¹ with the Gaussian 09 package at the PBE/6-31G(d) level.^{38,42,43} The SMD model separates the free energy of solvation in two contributions: (i) a bulk electrostatic energy that involves the solution of the GPE in terms of the integral-equation-formalism polarizable continuum model (IEF-PCM), (ii) a second contribution, proportional to the solvent-accesible are arising from short-range interactions between the solute and the solvent molecules (cavity-dispersion solvent structure term). This last point makes the model especially suited for an accurate calculation of the free energy of solvation for charged and uncharged molecular species.

5 Results

To validate the implementation of our solver we have performed some tests for a set of systems employing water as solvent. Simulated systems range from molecular systems to adsorbed molecules



Figure 5: Side and top views for the bare Pt(111) surface (a), $H_2O/Pt(111)$ (b) $CH_3OH/Pt(111)$ (c), $H_2O/Ru(0001)$ (d) and $CH_3OH/Ru(0001)$ (e) systems. Green spheres stand for Pt atoms, golden for Ru, red for O, white for H and grey for C.

on metal surfaces. For the first set of molecules, we show the results for water, methanol, and hydronium, hydroxide, Cl and Na⁺ ions, while for the second set the results are shown for water and methanol adsorbed on the Pt(111) and Ru(0001) surfaces (see Figure 5).

5.1 Multigrid Solver Convergence

The first point to be tested is the convergence of the multigrid algorithm. In Figure 6, the percentage error, e_{ϕ} , of the calculated electrostatic potential ϕ is shown as a function of the V-cycle for some systems. Here, $e_{\phi} = 100 \cdot \langle \sum_{i=1}^{N} |(f_i - \sum_j A_{ij}\phi_j)/f_i|/N \rangle$, where the brackets stand for an average



Figure 6: Percentage error e_{ϕ} as a function of the multigrid V-cycle for the case of an isolated water (black circles) and a methanol molecule adsorbed on a Pt-(111) surface (red squares).

over electronic cycles. Although real-space meshes involve a large number of gridpoints, in all cases we observe a high convergence rate, with $e_{\phi} \approx 10^{-6}$ after 5 V-cycles. The convergence rate is almost constant up to 7 cycles, where the percentage error reaches a plateau around a value of 10^{-8} . Only the first electronic cycle, not included in the average, requires more multigrid cycles for the electrostatic potential to converge. As the computational effort needed to compute a single cycle is minimal, we have chosen $e_{\phi} = 10^{-8}$ as the stopping criterium for the multigrid algorithm. The CPU time needed for an electronic cycle to converge (without starting from the vacuum wavefunctions) is approximately 1.5 times that for the unsolvated systems for all the tested machine architectures (nodes consisting on Intel Xeon X3360, E5530 and E5645 processors), and about one third of the time spent by the multigrid subroutine is consuming part of the whole implementation as it requires the allocation of big matrices. Nevertheless, if we consider that we

are using a serial 3D multigrid solver, the cost of a VASP calculation for a solvated system is not significantly larger than the one in gas-phase. Indeed, the solvation model was written trying to maximize computational efficiency, particularly for the fine grid calculations, for which serial solvers have a poor performance. A more detailed study of the robustness of our solver is presented in the Supporting Information.

5.2 Solvation Energies

Free energies of solvation have been calculated for the systems described above. The results are compared to experimental data,^{44–46} and to simulation results using Gaussian 09 SMD and the implicit solvation model of Mathew et al.²⁴, named VASPsol. For this model we have considered the solvent default parameters. For the case of the metallic surfaces, the results are just compared to the solvation model. The free energies, ΔG^{s} , are calculated as the difference between the total energies for the solvated system, E^{s} , and those for the same system in the gas-phase, E^{0} , that is,

$$\Delta G^{\rm s} = E^{\rm s} - E^0 \tag{17}$$

Experimental energies correspond to the Gibbs free energy of solvation, ΔG^* , as defined by Ben-Naim⁴⁷ (ideal gas at gas-phase concentration of 1 mol·L⁻¹ \rightarrow ideal solution at a liquid phase concentration of 1 mol·L⁻¹). All these energies are calculated considering the value of Tissandier et al. (-264.0 kcal/mol, -11.45 eV) for the solvation energy of the proton (H⁺).⁴⁸ The results for ΔG^s and ΔG^* are presented in Table 1 and Table 2. Corrections to forces have been monitored during the calculations in order to check the correctness of the results. No anomalous drift is observed in the forces along any of the space directions.

The calculated solvation free energies for the neutral isolated species (water and methanol) agree closely both with the experimental and other simulation data, as seen in the values for the percentage error, δ_{exp}^{MGCM} (less than 15% in both cases). Notice that for any of the methods discussed here the errors are larger or equal the one reported by our implementation.

Table 1: Solvation free energies in eV units for molecular systems. ΔG_{MGCM}^{s} , $\Delta G_{VASPsol}^{s}$ and ΔG_{g09}^{s} stand for the calculated energies with our model, VASPsol and Gaussian 09 package using the SMD model, respectively, while $\Delta G_{exp.}^{*}$ corresponds to experimental data. The superscripts *a*,*b*, and *c* correspond to refs. [44], [45], and [46]. An energy cutoff of $E_c = 450$ eV is set for both VASP-MGCM and VASPsol calculations, while a $3 \times 3 \times 1$ *k*-point mesh is set for the case of adsorbed molecules on metal surfaces.

System	H_2O	CH ₃ OH	H_3O^+	OH^-	Cl^{-}	Na ⁺
ΔG_{MGCM}	-0.305	-0.210	-3.710	-3.150	-2.233	-3.087
$\Delta G_{VASPsol}$	-0.314	-0.199	-3.549	-2.294	-1.955	-2.978
ΔG_{g09}	-0.347	-0.177	-4.203	-4.574	-2.931	-3.132
$\Delta G^*_{exp.}$	-0.270 ^a	-0.220 ^a	-4.779 ^b	-4.553 ^b -4.456 ^c	-3.235 ^b	-3.783 ^c
$\delta^{\rm MGCM}_{exp}(\%)$	12.96	4.54	22.37	30.81 29.31	31.00	18.39

Table 2: Solvation free energies in eV units for adsorbed molecules on metal surfaces. ΔG_{MGCM}^s and $\Delta G_{VASPsol}^s$ stand for the calculated energies with our model and the VASPsol package, respectively. An energy cutoff of $E_c = 450$ eV is set for both VASP-MGCM and VASPsol calculations, while a $3 \times 3 \times 1$ *k*-point mesh is set for the case of adsorbed molecules on metal surfaces.

System	Pt(111)	H ₂ O/Pt(111)	CH ₃ OH/Pt(111)	$\mathbf{H}_{2}\mathbf{O}/\mathbf{Ru}(0001)$	CH ₃ OH/Ru(0001)
ΔG_{MGCM}	-0.030	-0.270	-0.148	-0.294	-0.128
$\Delta G_{VASPsol}$	0.047	-0.237	-0.136	-0.261	-0.119

Solvation of charged species is more challenging as direct bonds between the solvates and the solvaten can occur. Indeed, ions in solution can be regarded as $M(H_2O)_n^{\pm}$ entities, where *n* stands for the number of water molecules in the solvation layers and M for the ion. Within this shell, water molecules are strongly oriented towards the solute and their behavior is largely different from molecules located far away. A popular and efficient strategy to improve the performance of continuum models in the calculation of solvation free energies is to add explicit water molecules to the ionic solute (explicity solvent shell).^{51,52} However, for the sake of simplicity we are not considering this approach in any of the calculations presented here VASP-MGCM, VASPsol, Gaussian09-SMD. As expected, the level of agreement between experiments and calculations is not observed in charged species, where simulation results underestimate the experimental values for all ions in our study (20% < δ_{exp}^{MGCM} < 30%). The difficulties in retrieving solvation energies for neutral and charged species with the same degree of accuracy have been described in the literature.^{27,49,50}

Moreover, the amount of available experimental solvation data for ionic solutes is more limited than for neutral species. This adds uncertainty in the comparison of simulated and experimental data for ionic solutes in solution.⁵¹ Our methodology provides solvation energy data that is within 25-30% of the Gaussian09-SMD values. More importantly, methodologies with periodic boundary conditions have some issues with charged systems, therefore it is much easier to work on models where the total system is neutral. If we compare the total reaction $2H_2O_{solv} \leftarrow H_3O^+ + OH^-$ the differential solvation contribution from Gaussian09-SMD and VASP-MGCM is below 6%. The situation is not that favorable for VASPsol due, in part, to the intrinsic solvation of the hydroxyl anion.

In a second step, we have applied the methodology to a true periodic-boundary model, i.e. adsorption on a metal surface. In this case no experimental values exist and thus the comparison of our VASP-MGCM to VASPsol. The system chosen to illustrate the performance of our continuum model is the adsorption of molecular methanol and water on the Pt(111) and Ru(0001) surfaces. The adsorption of water and methanol is exothermic by -0.27 and -0.15 eV, respectively, and the energy differences between present and previous methodologies are in the range of 0.05 eV. In particular, we have calculated the adsorption energy E_{ads}^s of methanol (*m*) on the metal surface in the presence of water as

$$E_{\rm ads}^s = E_{\rm m+surf}^s - E_{\rm m}^s - E_{\rm surf}^s \tag{18}$$

where E_{m+surf}^{s} , E_{m}^{s} , and E_{surf}^{s} stand for the total energy of the molecule on the metal, the molecule, and the bare surface solvated systems, respectively. For the sake of consistency with the results in ref [53], we have considered van der Waals (vdW) interactions through Grimme's DFT-D2 method⁵⁴ with our Pt and Ru C_6 parameters,⁵⁵ while the vdW parameters for the H, C, and O elements are those tabulated by Grimme for the PBE functional.⁵⁴

The calculated E_{ads}^s with the VASP-MGCM is equal to -0.53 eV. This value is not far from the adsorption energy (-0.67 eV) for the unsolvated methanol+water covered Pt(111) system calculated in ref [53]. However, it should be remarked that in the water covered Pt(111) system there is no solvent apart from a first explicit water layer. Then, neither the environment nor the orientation for

the adsorbed methanol molecule are the same than in our calculations, and a small shift is expected between both adsorption energies.

The present methodology also highlights the difficulties when addressing reactivity on the liquid-metal interface. Up to now, most of the solvent effects were introduced in an indirect way⁵⁸ however our simulations show that solvation might influence crucial elements in the modeling. For instance, the gas-phase adsorption of water and methanol on Pt(111) is exothermic by -0.34 and -0.46 eV respectively,^{56,57} but if the reaction takes place using water as solvent, the adsorption strength is reversed and water is preferred (-0.27 eV) over methanol (-0.15 eV).

As for Ru(0001), the solvent contribution is more favorable for molecularly adsorbed water (by about -0.02 eV) and less for molecularly adsorbed methanol (0.02 eV) when compared to Pt(111). On Ru(0001) water is known to dissociate easily. Therefore, we have investigated the contribution of the solvent to the molecular-dissociated energy balance. For the system without solvent the dissociated state is -0.389 eV more stable than the molecular one. When solvent is included the dissociation is still favorable but by a smaller amount, -0.285 eV. This is in line with the fact that a single molecule on Ru(0001) is always dissociated while for a full water bilayer only about 50% of the molecules are dissociated on the surface.⁵⁹

6 Comparison to previous models

As stated in the previous section, a similar implicit solvation approach to VASP-MGCM, named VASPsol, has been recently published by Mathew et al.²⁴ The main idea behind VASPsol is the same than in VASP-MGCM; the GPE is solved by taking into account a relative permittivity that is a functional of the electronic charge density ρ_{el} . Once the electrostatic potential ϕ for the solvated system is calculated, energies and forces are corrected. Nonelectrostatic effects are also considered through an energy term proportional to the solvent-accessible area.

Although the general structure of both models is almost the same, there exist some differences in the calculation of the different physical quantities. The main difference comes from: (i) the com-

putation of the relative permittivity, (ii) the way the GPE is solved, and (iii) the calculation of the nonelectrostatic terms. Regarding the last point, it does not have a huge effect on the convergence rate of the calculations and we will not discuss it further. With respect to the relative permittivity ε , VASPsol uses a very similar functional form than that proposed by Fattebert and Gygi.⁶⁰ In this context, as done in VASP-MGCM, an additional density charge is added to ρ_{el} for the calculation of ε . As a consequence, the solvent entrance in the core atom region is prevented for both models and changing the definition of ε does not change the final results up to the precision showed (we have tested various functional forms for $\varepsilon(\rho(\mathbf{r}))$). The key difference between both VASP-MGCM and VASPsol is the way the GPE is solved. In VASPsol the GPE is solved through a preconditioned conjugate gradient algorithm. The solver mades use of the FFT subroutines, and the equation is solved in reciprocal space. This method is quite efficient as the energies for solvated systems converge in almost the same time than those for their unsolvated counterparts when starting from vacuum wave functions.²⁴ However, the CPU time needed when calculations start from scratch is on the same order of magnitude than that for the VASP-MGCM.

In general, the problems related to the convergence of a solvent calculation arise from the sharpness of the charge density surface. If the problem is not solved when adding the corresponding countercharge in the total charge density (that becomes smoother) for a given set of parameters of the GPE solver, then a slight change with respect to them can lead to a converged solution. In this sense, the number of tunable physical parameters is almost the same for both the VASP-MGCM and VASPsol models. VASP-MGCM also allows adjusting some of the parameters related to the multigrid solver (i.e. the number of grids) which can help obtaining a converged solution for cases with sharp charge density surfaces. This situation only occurs in few cases but is an example of the certain adaptability of VASP-MGCM to each system when solving the GPE equation.

7 Conclusions

We have implemented a continuum solvation model into the VASP plane-wave package VASP-MGCM. The electrostatic problem, which describes the effect of the environment on the solute, is solved by means of multigrid methods. The computational cost of such approach is minimized by a fast and efficient method for the product of sparse matrices. Few multigrid cycles are required for the solution of the generalized Poisson equation. Hence, only a slight increase (around 30%) in the computation time is observed as compared to calculations in vacuum, making the solvation model applicable even to large systems with dense grids. The calculated solvation energies for a set of isolated and adsorbed molecules on metallic surfaces are consistent with experimental data and simulation results using other implicit solvation models. The present approach can be improved by adding the possibility of dealing with non homogeneous grids. This would involve the creation of denser grids at the solute atoms positions, and coarser grids at the regions with associated smaller charge density. Different functional forms for the dielectric constant and the non electrostatic contributions to the free energy are also to be added to make the model more flexible to the users' purposes. In summary, the implemented VASP-MGCM presents the following advantages: it is (i) robust in equation solving, (ii) adaptable to large cells since the electrostatic problem is solved in the direct space, (iii) transferable to other codes with similar characteristics, (iv) reasonably computationally demanding, (v) adaptable to any kind of periodic boundary conditions, (vi) versatile to incorporate coarse-grain strategies. All the above characteristics allow the use of the present code to the study of liquid-solid heterogeneous reactions that are fundamental to industrial and energy related problems. VASP-MGCM will be available soon on our website as a patch to the VASP code.

A. Appendix

A.1. Sparse Matrices

Operations involving sparse matrices are common in many fields such as graph theory, combinatorial optimization or multigrid methods, among others. In particular, the sparse matrix-vector (SpMV) product and the sparse matrix-matrix (SpMM) product are required operations when solving linear systems of equations regardless of the chosen strategy. Because of the large amount of zero entries contained in such matrices, sparse storage schemes are adopted to reduce the memory usage in numerical methods. The most common strategy is to exploit the particular sparsity structure of the involved matrices (their symmetry, if they are banded, ...) to make their product in the most efficient way. Adopted storage formats for sparse matrices are (i) the compressed row storage (CRS), (ii) the compressed column storage (CCS), (iii) the block compressed row storage (BCRS), or (iv) the Jagged diagonal storage (JDS) format.⁶¹ Sequential algorithms for the SpMM product like that described by Gustavson⁶² or those implemented in the Sparse Matrix Multiplication Package (SMMP)⁶³ make use of the CRS format. It is also the case for several packages aimed at solving large sparse linear systems like ITPACK,⁶⁴ SPARSEKIT⁶⁵ or Sparse BLAS.⁶⁶ Given a sparse matrix B of order N with a total number of NNZ nonzero entries, three one-dimensional matrices are used in the CRS scheme. A first array B^* of length NNZ stores the nonzero values in B. The column indices for such values are stored in an integer matrix JB of length NNZ. Finally, a third array IB containing N + 1 elements is used to list the starting locations of each row in B in such a way that IB(i) denotes the beginning of row *i* and IB(i+1) its end.

In the present study, we adopt a multigrid scheme to solve the electrostatic problem. In this context, the Galerkin approach involves the *RAI* sparse triple-matrix product which represents a large percentage of the total cost for solving our system of linear equations. Available libraries like BLAS⁶⁶ or LAPACK,⁶⁷ which can be called at any stage in VASP subroutines, contain high-performance linear algebra routines that perform the SpMV product, but not the SpMM product. To this end, we have implemented a sequential algorithm to perform the RAI triple sparse-matrix

product in a fast way, even for dense grids.

A.2. Sparse Matrix-Matrix Product Algorithm

To access the data in the sparse *R*, *A*, and *I* matrices we have adopted a modified version of the CRS format. Instead of employing one-dimensional matrices for the values and column indices of their nonzero elements, we define R^* , A^* and I^* matrices of size (N/8, 64), (N, M) and (N, 8), respectively (M = 19 in our case, see Supporting Information for more details). The corresponding column indices are called *IDR*, *ID* and *IDI* for *R*, *A*, and *I*. Once the data structure for all three matrices is created, the algorithm makes the *RAI* product in a per-row based method. For each row *i* in R^* we obtain the corresponding row *i* in the final *RAI* matrix (the column indices being stored in *IDRAI*). These operations are done in a fast way through the use of a third small one-dimensional array, called *IEL*, that is allocated during both the R^*A^* and the $(R^*A^*)I^*$ products. We can know, then, before making the sparse products, which columns in a row of the resulting matrix will contain nonzero elements. An scheme of our algorithm is shown in Figure 7 (for a detailed presentation of the algorithm see the Supporting Information).

- For each row i = 1, N/8:
 - 1. Identify columns in the $R^*(i,:)A^*$ product containing nonzero elements and store the column indices in *IEL*.
 - 2. Make the $R^*(i,:)A^*$ product just for the columns stored in *IEL*.
 - 3. Deallocate *IEL*.
 - 4. Identify columns in the $R^*A^*(i,:)I^*$ product containing nonzero elements and store the column indices in reallocated *IEL*.
 - 5. Make the $R^*A^*(i,:)I^*$ product and store the values in *RAI* matrix and their column indices in *IDRAI*.
- end

Figure 7: Algorithm for the SpMM product. The symbol ':' in $R^*(i,:)$ refers to all the columns in row *i* for a matrix R^* .

Unlike the CRS format, our algorithm does not require a third matrix of row pointers of length

N + 1 per each stored sparse matrix. Indeed, the dynamic array *IEL* that is created during the triple SpMM product, contains a much lower number of elements (~ $10^{-5}N$ for the densest grid) making this operation very efficient. Because of the discretization of A^* , and the chosen stencils for R^* and I^* (see Supporting Information), once the nonzero elements in the first row of *RAI* are known, this number remains constant for the rest of rows. Hence, there is no need of requesting an extra storage as it occurs for other sequential algorithms performing the SpMM product.⁶² At the same time, our algorithm has been programmed in such a way that there exists just an if statement, located in the *IEL* calculation loop. The number of calls to if statements is then comparable or lower than those for the general sequential SpMM product algorithms.^{62,63}

8 Acknowledgments

The authors acknowledge the European Research Council (ERC-2010-StG-258406) and MINECO (CTQ2012-33826) for financial support. We are grateful to Prof. F. J. Luque for critically reading the manuscript and giving us valuable suggestions.

Associated Content

Further details about the discretization of the GPE, the transfer operators between grids and the nonelectrostatic contributions to the Kohn-Sham energy (sections S1-S3), details of the algorithm for the SpMM product (section S4), a general scheme of the multigrid V-cycle (section S5), performance of VASP-MGCM with respect to numerical parameters (section S6), and possible parallelization of our multigrid algorithm (section S7).

References

 Nørskov, J. K.; Bligaard, T.; Rossmeisl, J.; Christensen, C. H. Towards the Computational Design of Solid Catalysts. *Nat. Chem.* 2009, *1*, 37-46.

- Zope, B. N.; Hibbitts, D. D.; Neurock, M.; Davis, R. J. Reactivity of the Gold/Water Interface During Selective Oxidation Catalysis. *Science* 2010, *330*, 74-78.
- Rossmeisl, J.; Logadottir, A.; Nørskov, J. K. Electrolysis of Water on (Oxidized) Metal Surfaces. *Chem. Phys.* 2005, *319*, 178-184.
- Hoermann, N.; Jaeckle, M.; Gossenberger, F.; Roman, T.; Forster-Tonigold, K.; Naderian, M.; Sakong, S.; Gross, A. Some Challenges in the First-Principles Modeling of Structures and Processes in Electrochemical Energy Storage and Transfer. *J. Power Sources* 2015, 275, 531-538.
- Grzelczak, M.; Pérez-Juste, J.; Mulvaney, P.; Liz-Marzán, L. M. Shape Control in Gold Nanoparticle Synthesis. *Chem. Soc. Rev.* 2008, *37*, 1783-1791.
- 6. Martin, R. M. *Electronic Structure: Basic Theory and Practical Methods*; Cambridge University Press: Cambridge, U.K., 2004; pp 11-51.
- Jensen, F. Introduction to Computational Chemistry; John Wiley & Sons Ltd.: Chichester, U.K., 2007; pp 475-486.
- Frenkel, D.; Smit, B. Understanding Molecular Simulation, from Algorithms to Applications, 2nd ed.; Academic: San Diego, CA, U.S.A., 1996; pp 23-105.
- Marx, D.; Hutter, J. Ab Initio Molecular Dynamics: Basic Theory and Advanced Methods; Cambridge University Press: Cambridge, U.K., 2009; pp 1-75.
- 10. Tomasi, J.; Persico, M. Molecular Interactions in Solution: An Overview of Methods Based on Continuous Distributions of the Solvent. *Chem. Rev.* **1994**, *94*, 2027-2094.
- Cancès, E.; Mennucci, B.; Tomasi, J. A New Integral Equation Formalism for the Polarizable Continuum Model: Theoretical Background and Applications to Isotropic and Anisotropic Dielectrics. J. Chem. Phys. 1997, 107, 3032-3041.

- Tomasi, J.; Mennucci, B.; Cammi, R. Quantum Mechanical Continuum Solvation Models. *Chem. Rev.* 2005, *105*, 2999-3094.
- Pascual-Ahuir, J. L.; Silla. E. An Improved Description of Molecular Surfaces. I. Building the Spherical Surface Set. J. Comput. Chem. 1990, 11, 1047-1060.
- Silla, E.; Villar, F.; Nilsson, O. Molecular Volumes and Surfaces of Biomacromolecules via GEPOL: A Fast and Efficient Algorithm. *J. Mol. Graphics* 1990, 8, 168-172.
- Silla, E.; Tuñón, I.; Pascual-Ahuir, J. L. GEPOL: An Improved Description of Molecular Surfaces II. Computing the Molecular Area and Volume. *J. Comput. Chem.* 1991, *12*, 1077-1088.
- Cossi, M.; Barone, V.; Cammi, R.; Tomasi, J. Ab Initio Study of Solvated Molecules: A New Implementation of the Polarizable Continuum Model. *Chem. Phys. Lett.* **1996**, 255, 327-335.
- Mennucci, B.; Cancès, E.; Tomasi, J. Evaluation of Solvent Effects in Isotropic and Anisotropic Dielectrics and in Ionic Solutions with a Unified Integral Equation Method: Theoretical Bases, Computational Implementation, and Numerical Applications. *J. Phys. Chem.* 1997, 101, 10506.
- York, D. M.; Karplus, M. A Smooth Solvation Potential Based on the Conductor-Like Screening Model. J. Phys. Chem. A 1999, 103, 11060-11079.
- 19. Lange, A. W.; Herbert, J. M. Polarizable Continuum Reaction-Field Solvation Models Affording Smooth Potential Energy Surfaces. *J. Phys. Chem. Lett.* **2010**, *1*, 556-561.
- Scalmani, G.; Frisch, M. J. Continuous Surface Charge Polarizable Continuum Models of Solvation. I. General Formalism. J. Chem. Phys. 2010, 132, 114110.
- 21. Kresse, G.; Furtmüller, J. Efficiency of Ab-Initio Total Energy Calculations for Metals and Semiconductors Using a Plane-Wave Basis Set. *Comput. Mater. Sci.* **1996**, *6*, 15-50.

- 22. Fattebert, J.-L.; Gygi, F. Density Functional Theory for Efficient Ab Initio Molecular Dynamics Simulations in Solution. *J. Comput. Chem.* **2002**, *23*, 662-666.
- Fattebert, J.-L.; Gygi, F. First-Principles Molecular Dynamics Simulations in a Continuum Solvent. *Int. J. Quant. Chem.* 2003, *93*, 139-147.
- Mathew, K.; Sundararaman, R.; Letchworth-Weaver, K.; Arias, T. A.; Hennig, R. G. Implicit Solvation Model for Density-Functional Study of Nanocrystal Surfaces and Reaction Pathways. J. Chem. Phys. 2014, 140, 084106.
- Scherlis, D. A.; Fattebert, J.-L.; Gygi, F.; Cococcioni, M.; Marzari, N. A Unified Electrostatic and Cavitation Model for First-Principles Molecular Dynamics in Solution. *J. Chem. Phys.* 2006, 124, 074103.
- Andreussi, O., Dabo, I., Marzari, N. Revised Self-Consistent Continuum Solvation in Electronic-Structure Calculations. J. Chem. Phys. 2012, 136, 064102.
- Dupont, C.; Andreussi, O.; Marzari, N. Self-Consistent Continuum Solvation (SCCS): The Case of Charged Systems. J. Chem. Phys. 2013, 139, 214110.
- Cooley, J. W.; Tukey, J. W. An Algorithm for the Machine Calculation of Complex Fourier Series. *Math. Comput.* 1965, 19, 297-301.
- Frigo, M.; Johnson, S. G. The Design and Implementation of FFTW3. *Proc. IEEE* 2005, 93, 216-231.
- Intel Math Kernel Library. https://software.intel.com/en-us/intel-mkl (accesed Aug 28, 2015).
- Wesseling, P. An Introduction to Multigrid Methods; John Wiley & Sons Ltd.: Chichester, U.K., 1992; pp 1-201.
- 32. Sánchez, V. M.; Sued, M.; Scherlis, D. A. First-Principles Molecular Dynamics Simulations at Solid-Liquid Interfaces with a Continuum Solvent. *J. Chem. Phys.* **2009**, *131*, 174108.

- Kohn, W.; Sham, L. Self-Consistent Equations Including Exchange and Correlation Effects. *Phys. Rev.* 1965, 140, A1133-A1138.
- Cococcioni, M.; Mauri, F.; Ceder, G.; Marzari, N. Electronic-Enthalpy Functional for Finite Systems Under Pressure. *Phys. Rev. Lett.* 2005, *94*, 145501.
- 35. Harris, R. C.; Pettitt, B. M. Examining the Assumptions Underlying Continuum-Solvent Models. *J. Chem. Theory Comput.* **2015**, *11*, 4593-4600.
- Fornberg, B. Generation of Finite Difference Formulas on Arbitrarily Spaced Grids. *Math. Comp.* 1988, 51, 699-706.
- Press, W. H.; Teukolsky, S. A.; Vetterling, W. T.; Flannery, B. P. Numerical Recipes in Fortran 90. The Art of Parallel Scientific Computing, 2nd ed.; Cambridge University Press: New York, U.S.A., 1999.
- Perdew, J. P.; Burke, K.; Ernzerhof, M. Generalized Gradient Approximation Made Simple. *Phys. Rev. Lett.* 1996, 77, 3865-3868.
- 39. Blöchl, P. E. Projector Augmented-Wave Method. Phys. Rev. B 1994, 50, 17953-17979.
- Monkhorst, H. J.; Pack. J. D. Special Points for Brillouin-Zone Integrations. *Phys. Rev. B* 1976, 13, 5188-5192.
- 41. Marenich, A. V.; Cramer, C. J.; Truhlar, D. G. Universal Solvation Model Based on Solute Electron Density and on a Continuum Model of the Solvent Defined by the Bulk Dielectric Constant and Atomic Surface Tensions. J. Phys. Chem. B 2009, 113, 6378-6396.
- Frisch, M. J.; Trucks, G. W.; Schlegel, H. B.; Scuseria, G. E.; Robb, M. A.; Cheeseman, J. R.; Scalmani, G.; Barone, V.; Mennucci, B.; Petersson, G. A.; et al. *Gaussian 09, Revision D.01*; Gaussian, Inc.: Wallingford CT, U.S.A., 2009.

- Ditchfield, R.; Hehre, W. J.; Pople, J. A. Self-Consistent Molecular-Orbital Method IX. An Extended Gaussian-Type Basis for Molecular Orbital Studies of Organic Molecules. *J. Chem. Phys.* 1971, 54, 724-728.
- Tannor, D. J.; Marte, B.; Murphy, R.; Friesner, R. A.; Sitkoff, D.; Nicholls, A.; Ringnalda, M.; Goddard, III, W. A.; Honig, B. Accurate First Principles Calculation of Molecular Charge Distributions and Solvation Energies from Ab Initio Quantum Mechanics and Continuum Dielectric Theory. J. Am. Chem. Soc. 1994, 116, 11875-11882.
- 45. Pliego, J. R., Jr.; Riveros, J. M. Gibbs Energy of Solvation of Organic Ions in Aqueous and Dimethyl Sulfoxide Solutions. *Phys. Chem. Chem. Phys.* **2002**, *4*, 1622-1627.
- Marcus, Y. Thermodynamics of Solvation of Ions. Part 5. Gibbs Free Energy of Hydration at 298.15 K. J. Chem. Soc. Faraday Trans. 1991, 87, 2995-2999.
- 47. Ben-Naim, A. Solvation Thermodynamics; Plenum Press: New York, U.S.A., 1987; pp 1-40.
- Tissandier, M. D.; Cowen, K. A.; Feng, W. Y.; Gundlach, E.; Cohen, M. H.; Earhart, A. D.; Coe, J. V.; Tuttle, T. R. The Proton's Absolute Aqueous Enthalpy and Gibbs Free Energy of Solvation from Cluster-Ion Solvation Data. *J. Phys. Chem. A* 1998, *102*, 7787-7794.
- Dziedzic, J.; Helal, H. H.; Skylaris, C. -K.; Mostofi, A. A.; Payne, M. C. Minimal Parameter Implicit Solvent for Ab Initio Electronic Structure Calculations. *Europhys. Lett.* 2011, 95, 43001.
- Curutchet, C.; Bidon-Chanal, A.; Soteras, I.; Orozco, M.; Luque, F. J. MST Continuum Study of the Hydration Free Energies of Monovalent Ionic Species. J. Phys. Chem. B 2005, 109, 3565-3574.
- Kelly, C. P.; Cramer, C. J.; Truhlar, D. G. SM6: A Density Functional Theory Continuum Solvation Model for Calculating Aqueous Solvation Free Energies of Neutrals, Ions, and Solute-Water Clusters. J. Chem. Theory Comput. 2005, 1, 1133-1152.

- 52. Kelly, C. P.; Cramer, C. J.; Truhlar, D. G. Aqueous Solvation Free Energies of Ions and Ion-Water Clusters Based on an Accurate Value for the Absolute Aqueous Solvation Free Energy of the Proton. *J. Phys. Chem. B* 2006, *110*, 16066-16081.
- Błoński, P.; López, N. On the Adsorption of Formaldehyde and Methanol on a Water-Covered Pt(111): a DFT-D Study. *J. Phys. Chem. C* 2012, *116*, 15484-15492.
- Grimme, S. Semiempirical GGA-Type Density Functional Constructed with a Long-Range Dispersion Correction. J. Comput. Chem. 2006, 27, 1787-1799.
- Almora-Barrios, N.; Carchini, G.; Błoński, P.; López, N. Costless Derivation of Dispersion Coefficients for Metal Surfaces. J. Chem. Theory Comput. 2014, 10, 5002-5009.
- Revilla-López, G.; López, N. A Unified Study of Water Adsorption on Metals: Meaningful Models from Structural Motifs. *Phys. Chem. Chem. Phys.* 2014, *16*, 18933-18940.
- 57. García-Muelas, R.; López, N. Collective Descriptors for the Adsorption of Sugar-Alcohols on Pt and Pd(111). J. Phys. Chem. C 2014, 118, 17531-17537.
- Kandoi, S.; Greeley, J.; Simonetti, D.; Shabaker, J.; Dumesic, J. A.; Mavrikakis, M. Reaction Kinetics of Ethylene Glycol Reforming over Platinum in the Vapor versus Aqueous Phases. *J. Phys. Chem. C* 2011, *115*, 961-971.
- Revilla-López, G.; Błoński, P.; López, N. Free Energy Assessment of Water Structures and Their Dissociation on Ru(0001). J. Phys. Chem. C 2015, 119, 5478-5483.
- 60. Petrosyan, S. A.; Rigos, A. A.; Arias, T. A. Joint Density-Functional Theory: Ab Initio Study of Cr₂O₃ Surface Chemistry in Solution. *J. Phys. Chem. B* **2005**, *109*, 15436-15444.
- 61. Bai, Z.; Demmel, J.; Dongarra, J.; Ruhe, A.; van der Vorst, H. *Templates for the Solution of Algebraic Eigenvalue Problems: a Practical Guide*; SIAM: Philadelphia, U.S.A., 2000.
- 62. Gustavson, F. G. Two Fast Algorithms for Sparse Matrices: Multiplication and Permuted Transposition. *Trans. Math. Soft.* **1978**, *4*, 250-269.

- Bank, R. E.; Douglas, C. C. Sparse MAtrix Multiplication Package (SMMP). Adv. Comput. Math. 1993, 1, 127-137.
- Kincaid, D. R.; Respess, J. R.; Young, D. M.; Grimes, R. G. ITPACK 2C: A Fortran Package for Solving Large Sparse Linear Systems by Adaptative Accelerated Iterative Methods. *ACM Trans. Math, Soft.* 1982, *8*, 302-322.
- SPARSKIT. http://www-users.cs.umn.edu/~saad/software/SPARSKIT (accessed Nov 20, 2015).
- Dongarra, J.; Du Croz, J.; Duff, I.; Hammarling, S. A Set of Level 3 Basic Linear Algebra Subprograms. ACM Trans. Math. Soft. 1990, 16, 1-17.
- Anderson, E.; Bai, Z.; Bischof, C.; Demmel, J.; Dongarra, J.; Du Croz, J.; Greenbaum,
 A.; Hammarling, S.; McKenney, A.; Ostrouchov, S.; et al. *LAPACK Users' Guide, 3rd ed.*;
 SIAM: Philadelphia, U.S.A., 1999.

For Table of Contents Use Only

