

Quantum, classical, and hybrid QM/MM calculations in solution: General implementation of the ddCOSMO linear scaling strategy

Cite as: J. Chem. Phys. **141**, 184108 (2014); <https://doi.org/10.1063/1.4901304>

Submitted: 17 September 2014 . Accepted: 29 October 2014 . Published Online: 12 November 2014

Filippo Lipparini, Giovanni Scalmani, Louis Lagardère , Benjamin Stamm, Eric Cancès, Yvon Maday, Jean-Philip Piquemal, Michael J. Frisch, and Benedetta Mennucci 



View Online



Export Citation



CrossMark

ARTICLES YOU MAY BE INTERESTED IN

Domain decomposition for implicit solvation models

The Journal of Chemical Physics **139**, 054111 (2013); <https://doi.org/10.1063/1.4816767>

A new discretization for the polarizable continuum model within the domain decomposition paradigm

The Journal of Chemical Physics **144**, 054101 (2016); <https://doi.org/10.1063/1.4940136>

A consistent and accurate ab initio parametrization of density functional dispersion correction (DFT-D) for the 94 elements H-Pu

The Journal of Chemical Physics **132**, 154104 (2010); <https://doi.org/10.1063/1.3382344>

The Journal
of Chemical Physics

2018 EDITORS' CHOICE

READ NOW!



Quantum, classical, and hybrid QM/MM calculations in solution: General implementation of the ddCOSMO linear scaling strategy

Filippo Lipparini,^{1,2,3,a)} Giovanni Scalmani,⁴ Louis Lagardère,³ Benjamin Stamm,^{1,5} Eric Cancès,⁶ Yvon Maday,^{1,7} Jean-Philip Piquemal,^{2,5} Michael J. Frisch,⁴ and Benedetta Mennucci⁸

¹*Sorbonne Universités, UPMC Univ. Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005 Paris, France*

²*Sorbonne Universités, UPMC Univ. Paris 06, UMR 7616, Laboratoire de Chimie Théorique, F-75005 Paris, France*

³*Sorbonne Universités, UPMC Univ. Paris 06, Institut du Calcul et de la Simulation, F-75005 Paris, France*

⁴*Gaussian, Inc., 340 Quinnipiac St. Bldg. 40, Wallingford, Connecticut 06492, USA*

⁵*CNRS, UMR 7598 and 7616, F-75005 Paris, France*

⁶*Université Paris-Est, CERMICS, Ecole des Ponts and INRIA, 6 & 8 avenue Blaise Pascal, 77455 Marne-la-Vallée Cedex 2, France*

⁷*Institut Universitaire de France, Paris, France and Division of Applied Maths, Brown University, Providence, Rhode Island 02912, USA*

⁸*Dipartimento di Chimica e Chimica Industriale, Università di Pisa, Via Risorgimento 35, 56126 Pisa, Italy*

(Received 17 September 2014; accepted 29 October 2014; published online 12 November 2014)

We present the general theory and implementation of the Conductor-like Screening Model according to the recently developed ddCOSMO paradigm. The various quantities needed to apply ddCOSMO at different levels of theory, including quantum mechanical descriptions, are discussed in detail, with a particular focus on how to compute the integrals needed to evaluate the ddCOSMO solvation energy and its derivatives. The overall computational cost of a ddCOSMO computation is then analyzed and decomposed in the various steps: the different relative weights of such contributions are then discussed for both ddCOSMO and the fastest available alternative discretization to the COSMO equations. Finally, the scaling of the cost of the various steps with respect to the size of the solute is analyzed and discussed, showing how ddCOSMO opens significantly new possibilities when cheap or hybrid molecular mechanics/quantum mechanics methods are used to describe the solute.

© 2014 AIP Publishing LLC. [<http://dx.doi.org/10.1063/1.4901304>]

I. INTRODUCTION

Continuum solvation models^{1–4} are nowadays among the most popular tools in computational chemistry to include the effects of the chemical environment in the description of a molecular property or process. The strength of these models is their simplicity of use and cost-effectiveness, which has granted them a prominent role in a vast field of application, ranging from chemistry to biophysics to materials science.^{1,5,6} Continuum solvation models have been used together with both classical molecular mechanics (MM), quantum mechanics (QM), and hybrid (QM/MM) levels of theory to describe the solute and several different formulations and implementations exist, including fully polarizable QM/MM/Continuum ones.^{7–13} All these methods are based on the electrostatic problem of a density of charge accommodated in a properly shaped cavity Ω surrounded by a uniform, dielectric continuum.¹⁴ Different strategies exist to treat such a problem; among them the most popular is based on the concept of *Apparent Surface Charge* (ASC), a density of charge supported on the cavity's boundary $\Gamma = \partial\Omega$ that represents the polarization of the dielectric. The ASC is the (unique) solution to an integral equation which is equivalent to the elec-

trostatic problem:^{14–16} the main advantage of such a formulation is to reduce the problem to a surface problem, which can be efficiently solved through standard numerical methods such as the Boundary Element Method (BEM). In the last decade, several important advances have been introduced in the numerical treatment of continuum solvation models. In particular, smooth discretizations,^{17–20} i.e., discretizations where the solvation energy is a smooth function of the nuclear coordinates, and fast algorithms²¹ (in particular, the Fast Multipole Method, FMM²²) have extended the range of applicability of continuum solvation models both in terms of the size of the treatable systems and of the properties that can be modeled. Nevertheless, the increase in computational power and the recent advances in hybrid QM/MM models brought a real change of paradigm in computational chemistry, altering in a significant way the limiting size of the systems that can be studied. For systems as large as a protein or a DNA fragment, the original assumption that continuum solvation models provide a cheap way to introduce solvent effects is no longer valid, as the computational cost associated with the solution of the polarization equations, and in particular of the linear system obtained with some discretization technique, can become unaffordable.^{23,24} Indeed, ASC/BEM implementations produce dense—and possibly very ill-conditioned—linear systems:¹⁴ even by using iterative techniques to solve

^{a)}Electronic mail: flippari@uni-mainz.de

them, it is not possible to avoid the cost of large matrix/vector multiplications, which scale quadratically in CPU time with respect to the size of the system; furthermore, while fast summation techniques (in particular, the FMM) can be used to compute the required matrix/vector products with a cost linear in the system's size, the particular problem remains challenging, as such matrix/vector products correspond to compute the potential/field of a set of charges (eventually, smeared as a Gaussian distribution in the most recent implementations) which have a very high density, which is the worst-case scenario for the FMM.

Recently, we have introduced a new discretization technique^{24–26} for the Conductor-like Screening Model (COSMO) by Klamt and Schüürmann^{4,27} (sometimes also referred to as Conductor-like Polarizable Continuum Model,²⁸ or C-PCM, within the PCM family of continuum solvation models). Such a new discretization, which we have called ddCOSMO, demonstrates outstanding computational performances, outspeeding by two to three orders of magnitude even the fastest implementations based on the FMM;²⁴ furthermore, the numerical accuracy is systematically improvable, only a very limited number of parameters control the discretization²⁵ and the potential energy surfaces obtained with ddCOSMO are smooth even when an extreme event, such as a bond dissociation with the formation of two separate cavities, occur.^{25,26} The ddCOSMO algorithm has been implemented in conjunction with both classical force fields (including polarizable ones) and semiempirical methods.²⁴ In this paper, we will discuss the coupling of ddCOSMO with different levels of theory, reviewing the details of classical solutes and semiempirical methods and focusing for the first time on quantum mechanical descriptions where the electronic density is expanded in a set of atomic orbitals. In particular, we will detail the computation of the integrals required to compute the ddCOSMO solvation energy, forces and Fock-matrix contributions and discuss in detail the implementation within the context of Self-Consistent Field (SCF) methods, including Hartree-Fock (HF) and Density Functional Theory (DFT). The extension to hybrid QM/MM models will be briefly analyzed as a straightforward result and directions for post-HF methods coupling will be shortly mentioned.

This paper is organized as follows. In Sec. II, the ddCOSMO discretization is rapidly introduced. In Sec. III, the coupling between ddCOSMO and both a classical and a quantum mechanical description of the solute will be discussed. The quantities needed to compute analytical derivatives with respect to the nuclei positions will be discussed in Sec. IV. Some implementation details and numerical tests will be then presented in Sec. V, followed by some conclusions and perspectives in Sec. VI.

II. A DOMAIN DECOMPOSITION STRATEGY FOR CONTINUUM SOLVATION

In this section, we will briefly outline the main concepts and equations appearing in the ddCOSMO algorithm. We will assume, as customary for continuum solvation, that no charge lies outside the cavity; such an issue has already been discussed in Refs. 29 and 30. The Conductor-like Screening

model treats the solvent as a uniform, infinite conductor which surrounds a molecule-shaped cavity Ω where the solute is accommodated. Such a cavity is usually made by a union of spheres, one per atom:

$$\Omega = \bigcup_{j=1}^M \Omega_j(\mathbf{R}_j, r_j).$$

Here, \mathbf{R}_j is the position of the j th atom and r_j is the radius of the sphere associated to such an atom, which is a parameter of the model; in our implementation, we use the UFF radii³¹ scaled by a factor 1.1. The interaction energy between the solute and the solvent is assumed to be the electrostatic interaction energy E_s between the density of charge ρ of the solute and the polarization potential W of the conductor, usually referred to as the *reaction field*:

$$E_s = \frac{1}{2} f(\varepsilon) \int_{\Omega} \rho(\mathbf{r}) W(\mathbf{r}) d\mathbf{r}, \quad (1)$$

where $f(\varepsilon)$ is an empirical scaling introduced to account for the non-conductor nature of the solvent and ε is its dielectric constant. Notice that, in order to accurately reproduce the solvation energy, other non-electrostatic interactions need to be taken into account. This can be done by adding to the electrostatic solvation energy empirical contributions accounting for the missing interactions, as described, for instance, in Ref. 32. The reaction potential is the unique solution to the following boundary value problem:

$$\begin{cases} -\Delta W(\mathbf{r}) = 0 & \text{for } \mathbf{r} \in \Omega, \\ W(\mathbf{s}) = -\Phi(\mathbf{s}) & \text{for } \mathbf{s} \in \Gamma, \end{cases} \quad (2)$$

where Φ is the electrostatic potential produced by the solute *in vacuo* and $\Gamma = \partial\Omega$ is the boundary of the cavity. Equation (2) is usually solved by recasting it as an integral equation, which is in turn discretized using the BEM;¹⁴ ddCOSMO is a completely different strategy based on Schwarz's Domain Decomposition Method, which tackles the problem in Eq. (2) by decomposing it in a collection of coupled, local problems, where each local problem is defined on a single sphere:²⁵

$$\begin{cases} -\Delta W_j(\mathbf{r}) = 0 & \text{for } \mathbf{r} \in \Omega_j, \\ W_j(\mathbf{s}) = h_j(\mathbf{s}) & \text{for } \mathbf{s} \in \Gamma_j. \end{cases} \quad (3)$$

Here, $\Gamma_j = \partial\Omega_j$ and the function $h_j(\mathbf{s})$ corresponds to minus the solute's potential if \mathbf{s} lies on the external surface Γ , while it is computed as the average reaction potential inside the intersecting spheres if the point \mathbf{s} is buried inside the cavity:

$$h_j(\mathbf{s}) = \begin{cases} -\Phi(\mathbf{s}), & \mathbf{s} \in \Gamma_j^e \\ \sum_{k \in \mathcal{N}_j(\mathbf{s})} \frac{1}{|\mathcal{N}_j(\mathbf{s})|} W_k(\mathbf{s}) & \mathbf{s} \in \Gamma_j^i. \end{cases} \quad (4)$$

In Eq. (4), we have introduced the following notation. Γ_j^e and Γ_j^i denote the portion of the surface Γ_j exposed to the solvent or buried inside the cavity, respectively (see Figure 1 for a pictorial representation); $\mathcal{N}_j(\mathbf{s})$ is the list of the spheres that intersect Ω_j and contain the point $\mathbf{s} \in \Gamma_j$ and $|\mathcal{N}_j(\mathbf{s})|$ is the number of such spheres.

Notice that the right-hand side of Eq. (3) for sphere Ω_j depends on the potential of all the neighboring spheres: an

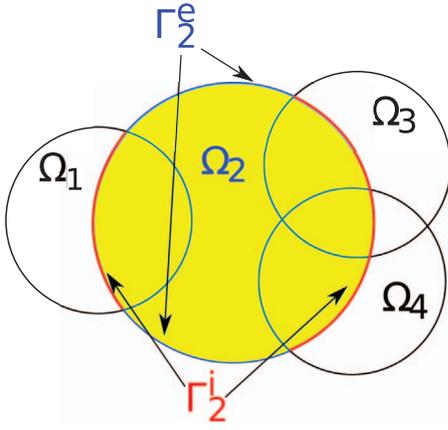


FIG. 1. Example of a van der Waals cavity. The external and internal portions of the surface of sphere 2 are marked in blue and red, respectively.

iterative procedure can be set up in order to solve the coupled problem. Let $\chi_j(\mathbf{r})$ be the characteristic function of sphere Ω_j , i.e.,

$$\chi_j(\mathbf{r}) = \begin{cases} 1 & \mathbf{r} \in \Omega_j \\ 0 & \text{otherwise} \end{cases}$$

For each sphere Ω_j , the function $W_j(\mathbf{r})$ is harmonic for each $\mathbf{r} \in \Omega_j$: it is possible to represent it via a local ASC function defined on the boundary Γ_j :

$$\forall \mathbf{r} \in \Omega_j \quad W_j(\mathbf{r}) = \int_{\Gamma_j} \frac{\sigma_j(\mathbf{s})}{|\mathbf{r} - \mathbf{s}|} d\mathbf{s} = (\tilde{\mathcal{S}}_j \sigma_j)(\mathbf{r}). \quad (5)$$

In particular, at the boundary:

$$\forall \mathbf{s} \in \Gamma_j \quad W_j(\mathbf{s}) = \int_{\Gamma_j} \frac{\sigma_j(\mathbf{s}')}{|\mathbf{s} - \mathbf{s}'|} d\mathbf{s}' = (\mathcal{S}_j \sigma_j)(\mathbf{s}). \quad (6)$$

We have introduced the operators \mathcal{S}_j and $\tilde{\mathcal{S}}_j$ which, given a surface density of charge on Γ_j , produce the potential on Γ_j or in Ω_j , respectively.

We can finally recast the ddCOSMO equation in a more compact way by introducing the normalized weights

$$\forall \mathbf{s} \in \Gamma_j \quad \omega_{jk}(\mathbf{s}) = \frac{\chi_k(\mathbf{s})}{|\mathcal{N}_j(\mathbf{s})|}, \quad U_j(\mathbf{s}) = 1 - \sum_{k \in \mathcal{N}_j(\mathbf{s})} \omega_{jk}(\mathbf{s}).$$

The ddCOSMO equation now reads, $\forall j = 1, \dots, M$:

$$\forall \mathbf{s} \in \Gamma_j \quad (\mathcal{S} \sigma_j)(\mathbf{s}) = -U_j(\mathbf{s})\Phi(\mathbf{s}) + \sum_{k \in \mathcal{N}_j(\mathbf{s})} \omega_{jk}(\mathbf{s})(\tilde{\mathcal{S}}_k \sigma_k)(\mathbf{s}), \quad (7)$$

from which it is apparent that the ddCOSMO method replaces the standard, non-local COSMO integral equation, where all the points on the cavity Γ interact with each other, with a linear system of local integral equations, where only neighboring spheres interact directly. The ddCOSMO equations are discretized by expanding each local ASC in a truncated basis of real spherical harmonics according to the ansatz

$$\sigma_j(\mathbf{R}_j + r_j \mathbf{y}) = \sum_{l=0}^N \sum_{m=-l}^l [X_j]_l^m Y_l^m(\mathbf{y}), \quad (8)$$

where \mathbf{y} is a unit vector and we write a generic point $\mathbf{s} \in \Gamma_j$ as $\mathbf{R}_j + r_j \mathbf{y}$. The ansatz in Eq. (8) allows for an easy computation of the matrix elements of the integral operators in Eq. (7); the details of the computation can be found in Refs. 25 and 26, here we only report the final expressions. The \mathcal{S}_j operator is diagonal in the spherical harmonics basis:

$$[L_{jj}]_{ll'}^{mm'} = \langle lm | \mathcal{S}_j | l' m' \rangle = \frac{4\pi r_j}{2l+1} \delta_{ll'} \delta_{mm'}. \quad (9)$$

To compute the matrix elements of the other operator, i.e., the product of the ω_{jk} weight with the $\tilde{\mathcal{S}}_k$ integral operator, a numerical integration is required. The Lebedev quadrature is particularly suited to the purpose: let $\{\mathbf{y}_n, w_n\}_{n=1}^{N_g}$ be a set of Lebedev points and weights; let

$$\begin{aligned} \mathbf{v}_n^{jk} &= \mathbf{R}_j + r_j \mathbf{y}_n - \mathbf{R}_k, & v_n^{jk} &= |\mathbf{v}_n^{jk}|, \\ t_n^{jk} &= \frac{v_n^{jk}}{r_k}, & \mathbf{s}_n^{jk} &= \frac{\mathbf{v}_n^{jk}}{v_n^{jk}} \end{aligned} \quad (10)$$

and

$$W_{jk}^n = \omega_{jk}(\mathbf{R}_j + r_j \mathbf{y}_n). \quad (11)$$

Notice that it is possible, as discussed in detail in Ref. 26, to regularize the function ω by substituting the characteristic functions $\chi_j(\mathbf{s})$ with a smooth function. The final expression for the discretized $\omega_{jk} \tilde{\mathcal{S}}_k, j \neq k$, is

$$\begin{aligned} [L_{jk}]_{ll'}^{mm'} &= -\langle lm | \omega_{jk} \tilde{\mathcal{S}}_k | l' m' \rangle \\ &= -\sum_{n=1}^{N_g} w_n Y_l^m(\mathbf{y}_n) W_{jk}^n \frac{4\pi r_k}{2l'+1} (t_n^{jk})^{l'} Y_{l'}^{m'}(\mathbf{s}_n^{jk}). \end{aligned} \quad (12)$$

Finally, let $U_n^j = U(\mathbf{R}_j + r_j \mathbf{y}_n)$, $\Phi_n^j = \Phi(\mathbf{R}_j + r_j \mathbf{y}_n)$ and

$$[g_j]_l^m = -\sum_{n=1}^{N_g} w_n Y_l^m(\mathbf{y}_n) U_n^j \Phi_n^j. \quad (13)$$

We can now recast the discretized ddCOSMO equations as a linear system:

$$\begin{pmatrix} L_{11} & \dots & L_{1M} \\ \vdots & \ddots & \vdots \\ L_{M1} & \dots & L_{MM} \end{pmatrix} \begin{pmatrix} X_1 \\ \vdots \\ X_M \end{pmatrix} = \begin{pmatrix} g_1 \\ \vdots \\ g_M \end{pmatrix}, \quad (14)$$

which, for brevity, we will also indicate as

$$LX = g. \quad (15)$$

We will not discuss here the details of the implementation of the ddCOSMO iterative procedure, which has been extensively analyzed in Ref. 26: it is sufficient to say that the L matrix is block-sparse, as the L_{jk} blocks are zero if (and only if) the two spheres Ω_j and Ω_k do not intersect; furthermore, as the diagonal blocks are diagonal, iterative procedures which require the inversion of the diagonal blocks, such as Jacobi iterations, are trivially implemented. From a numerical point of view, such a Jacobi iterative scheme is known as the ‘‘additive Schwarz’’ algorithm and is proven to converge;³³ alternatively,

Gauss-Seidel (“multiplicative Schwarz” algorithm) or conjugate gradient or DIIS algorithms can also be used for faster convergence.

III. QUANTUM AND CLASSICAL SOLUTES: THE COUPLING WITH DDCOSMO

In this section, we will describe how to couple ddCOSMO with different models of the solute. The key quantity here is the solute’s charge density, which can be either a continuous function, a collection of point charges or higher order multipoles or the combination of both. In particular, for a classical solute (i.e., a solute described via a force-field), where the electrostatics is modeled by endowing each atom with a charge or a truncated multipolar distribution, the charge density is

$$\rho(\mathbf{r}) = \sum_{j=1}^M (q_j + \mu_j \cdot \nabla + \dots) \delta(\mathbf{r} - \mathbf{R}_j), \quad (16)$$

where q_j, μ_j, \dots are the charge, dipole, ... associated to the j th of M atoms.

For quantum mechanical models, the charge density is the sum of a classical term due to the nuclei and an electronic term. Assuming that a basis of (Gaussian) atomic orbitals $\{\chi_\mu(\mathbf{r})\}_{\mu=1}^{N_b}$ is employed, the electronic density can be expanded in such a basis introducing the so-called one-particle density matrix $P_{\mu\nu}$: the total density is therefore

$$\rho(\mathbf{r}) = \sum_{j=1}^M Z_j \delta(\mathbf{r} - \mathbf{R}_j) - \sum_{\mu\nu} P_{\mu\nu} \chi_\mu(\mathbf{r}) \chi_\nu(\mathbf{r}). \quad (17)$$

The electrostatic potential of the solute at a point $\mathbf{s} \in \Gamma = \partial\Omega$ is defined as

$$\Phi(\mathbf{s}) = \int_{\mathbb{R}^3} \frac{\rho(\mathbf{r})}{|\mathbf{r} - \mathbf{s}|} d\mathbf{r}. \quad (18)$$

The potential is a standard quantity both for classical solutes, where the integral is replaced by a sum, and quantum solutes, where the potential becomes the sum of the nuclei’s potential and the electronic potential:

$$\Phi(\mathbf{s}) = \sum_{j=1}^M \frac{Z_j}{|\mathbf{R}_j - \mathbf{s}|} - \sum_{\mu\nu} P_{\mu\nu} \int_{\mathbb{R}^3} \frac{\chi_\mu(\mathbf{r}) \chi_\nu(\mathbf{r})}{|\mathbf{r} - \mathbf{s}|} d\mathbf{r}. \quad (19)$$

The one-electron integrals in Eq. (19) are standard quantities for quantum chemical computations. We will refer to the integral

$$V_{\mu\nu}(\mathbf{s}) = \int_{\mathbb{R}^3} \frac{\chi_\mu(\mathbf{r}) \chi_\nu(\mathbf{r})}{|\mathbf{r} - \mathbf{s}|} d\mathbf{r}$$

as the “uncontracted potential,” since the electronic potential can be obtained by contracting such integrals with the electronic density matrix. Here, we point out that the definition in Eq. (19) is general and does not depend on the specific method used for the solute: the information on the level of theory is in fact fully contained in the density matrix, which can be the SCF density matrix for HF or Kohn-Sham DFT descriptions, or a correlated density matrix for more refined post-HF levels

of theory, such as Many-Body Perturbation or Coupled Cluster theory.^{34–36}

The computation of the solvation energy is more subtle in the case of ddCOSMO as compared to the usual COSMO implementations. The solvation energy is in fact defined through the integral in Eq. (1), which we repeat here for convenience:

$$E_s = \frac{1}{2} f(\varepsilon) \int_{\Omega} \rho(\mathbf{r}) W(\mathbf{r}) d\mathbf{r}. \quad (20)$$

The ddCOSMO scheme allows one to compute easily the reaction potential in each sphere Ω_j using Eq. (5), which we also repeat here:

$$\forall j, \forall \mathbf{r} \in \Omega_j \quad W_j(\mathbf{r}) = \int_{\Gamma_j} \frac{\sigma_j(\mathbf{s})}{|\mathbf{r} - \mathbf{s}|} d\mathbf{s}; \quad (21)$$

introducing the ddCOSMO ansatz, Eq. (8), and by using the spherical harmonics addition theorem, one gets easily to

$$\begin{aligned} W_j(\mathbf{r}) &= \sum_{lm} [X_j]_{lm}^m \int_{\Gamma_j} \frac{Y_l^m\left(\frac{\mathbf{s}}{r_j}\right)}{|\mathbf{s} - \mathbf{r}|} d\mathbf{s} \\ &= \sum_{lm} \frac{4\pi}{2l+1} \frac{r^l}{r_j^{l+1}} [X_j]_{lm}^m Y_l^m\left(\frac{\mathbf{r}}{|\mathbf{r}|}\right), \end{aligned} \quad (22)$$

where we introduced the compact notation

$$\sum_{lm} = \sum_{l=0}^N \sum_{m=-l}^l$$

and, for convenience, we translate the reference framework in the center of Ω_j whenever we write an integral on Γ_j , with the change of variables $\mathbf{r} - \mathbf{R}_j \rightarrow \mathbf{r}$. For classical solutes, the peculiar form of the charge density allows one to compute the solvation energy in an inexpensive way, as a sum of single-sphere contributions, which is ideal in the ddCOSMO framework. Through some elementary algebra:

$$E_s = \frac{1}{2} f(\varepsilon) \sum_{j=1}^M \frac{4\pi}{2l+1} \frac{1}{r_j^{l+1}} \sum_{l=0}^{L_s} \sum_{m=-l}^l [X_j]_{lm}^m [\Theta_j]_{lm}^m, \quad (23)$$

where L_s is the maximum angular momentum in the solute’s multipolar representation and $[\Theta_j]_{lm}^m$ is the l, m (spherical) multipole associated to the j th atom. It is convenient here to introduce the vector Ψ , which is defined as follows:

$$\begin{aligned} [\Psi_j]_{lm}^m &= \frac{4\pi}{2l+1} \frac{1}{r_j^{l+1}} [\Theta_j]_{lm}^m, \quad l \leq L_s \leq N; \\ [\Psi_j]_{lm}^m &= 0 \quad \text{otherwise.} \end{aligned} \quad (24)$$

Such a vector is the multipolar representation of the potential produced by the fraction of the solute’s density contained in Ω_j , and allows the energy to be written as

$$E_s = \frac{1}{2} f(\varepsilon) \sum_{j=1}^M \sum_{lm} [\Psi_j]_{lm}^m [X_j]_{lm}^m. \quad (25)$$

For QM solutes, the evaluation of the ddCOSMO energy is more complex. As one needs to evaluate the reaction potential using Eq. (21), it is necessary to decompose the integral in Eq. (20) as a sum of single-sphere contributions; however,

a naive decomposition of this integral would produce double (or multiple) counting in the regions where various spheres intersect. In order to overcome this difficulty, a multicentric quadrature scheme can be used where the points belonging to an intersection region are suitably weighed; furthermore, it is possible to choose these weights in such a way that they vary smoothly with the distance from the integration centers—i.e., the position of the atoms—in order to have a smoothly varying energy and being able to compute analytical derivatives. The latter topic will be discussed in Sec. IV. The numerical quadrature procedure used in the framework of density functional theory, such as the one proposed by Becke,³⁷ suits our purposes as long as the integration domain is extended to the whole space, i.e., as long as one replaces the integral in Eq. (20) with

$$E_s = \frac{1}{2} f(\varepsilon) \int_{\mathbb{R}^3} \rho(\mathbf{r}) W(\mathbf{r}) d\mathbf{r}. \quad (26)$$

This allows one to exploit in a black-box way the efficient procedures used to compute the exchange-correlation energy in DFT. We will now give some further detail on the numerical procedure. We refer the reader to Refs. 37 and 38 for a complete review of the quadrature scheme that we use; here it is sufficient to say that Becke's quadrature introduces, for each atom j , a set of points and weights $\{\mathbf{x}_n^j, \tau_n^j\}_{n=1}^{N_b^j}$ such that the integral of the electronic density in the whole space can be written as

$$\int_{\mathbb{R}^3} \rho(\mathbf{r}) d\mathbf{r} = -N_e = \sum_{j=1}^M \sum_{n=1}^{N_b^j} \tau_n^j \rho(\mathbf{x}_n^j). \quad (27)$$

Notice that Becke's scheme is designed to correctly integrate the density to (minus) the total number of electrons N_e : the double counting issue is therefore handled correctly.

Using the Becke's scheme, the ddCOSMO energy is computed as

$$E_s = \frac{1}{2} f(\varepsilon) \sum_{j=1}^M \sum_{n=1}^{N_b^j} \tau_n^j \rho(\mathbf{x}_n^j) W(\mathbf{x}_n^j). \quad (28)$$

As already mentioned, the Becke scheme is multicentric and the integral in Eq. (28) is assembled as a sum of single-atom, and therefore single-sphere, contributions: for each atom j we use the local ASC coefficients X_j in order to evaluate the reaction potential at $W(\mathbf{x}_n^j)$. However, as the integration has been extended in the whole space, we need to evaluate, for each atom h , the reaction potential at points that can be either inside the sphere Ω_j or outside. If a point is inside, Eq. (22) can be used; for the points \mathbf{x}_n^j exterior to Ω_j , we replace $W(\mathbf{x}_n^j)$ by the value of the harmonic extension outside Ω_j of the function W_j defined by Eq. (3) (note that this harmonic extension is easily computed by an external multipolar expansion). This is of course an approximation, but we have checked numerically that the error generated by this approximation is small for the points \mathbf{x}_n^j located inside Ω . For the points \mathbf{x}_n^j located outside Ω , it can be shown mathematically that this choice leads to a correction term partially balancing the escaped charge error

(a similar analysis was carried out in Ref. 29 for the standard ASC method).

More in detail, let $\mathbf{x}_n^j = x_n^j \mathbf{s}_n^j$, where $x_n^j = |\mathbf{x}_n^j|$ and $\mathbf{s}_n^j = \frac{\mathbf{x}_n^j}{x_n^j}$. The reaction potential at a Becke's point is

$$\begin{aligned} W(\mathbf{x}_n^j) &= \sum_{lm} [X_j]_l^m \int_{\Gamma_j} \frac{Y_l^m\left(\frac{\mathbf{s}}{|\mathbf{s}|}\right)}{|\mathbf{s} - \mathbf{x}_n^j|} d\mathbf{s} \\ &= \sum_{lm} \frac{4\pi}{2l+1} \frac{x_{<}^l}{x_{>}^{l+1}} [X_j]_l^m Y_l^m(\mathbf{s}_n^j), \end{aligned} \quad (29)$$

where $x_{<} = x_n^j$ and $x_{>} = r_j$ if the Becke's point is inside Ω_j and vice versa if the point is outside. By putting everything together, we get, for the ddCOSMO energy:

$$E_s = \frac{1}{2} f(\varepsilon) \sum_{j=1}^M \sum_{lm} \frac{4\pi}{2l+1} [X_j]_l^m \sum_{n=1}^{N_b^j} \tau_n^j \rho(\mathbf{x}_n^j) \frac{x_{<}^l}{x_{>}^{l+1}} Y_l^m(\mathbf{s}_n^j), \quad (30)$$

which can be recast in a more compact fashion by generalizing the definition of the Ψ vector to

$$[\Psi_j]_l^m = \frac{4\pi}{2l+1} \sum_{n=1}^{N_b^j} \tau_n^j \rho(\mathbf{x}_n^j) \frac{x_{<}^l}{x_{>}^{l+1}} Y_l^m(\mathbf{s}_n^j) \quad (31)$$

as, again

$$E_s = \frac{1}{2} f(\varepsilon) \sum_{j=1}^M \sum_{lm} [\Psi_j]_l^m [X_j]_l^m = \frac{1}{2} f(\varepsilon) \langle \Psi, X \rangle. \quad (32)$$

For polarizable solutes, i.e., described either with a QM method or with a polarizable force field, one also has to modify the equations that determine the solute's density in order to account for mutual polarization. We will take here as an example the coupling of ddCOSMO with the Hartree-Fock method (the generalization to DFT is straightforward) and discuss briefly semiempirical methods as a special case. The coupling of polarizable continuum solvation models to SCF-based methods has been extensively discussed in the literature^{1,39} and can be derived by minimizing, under the usual constraints, the following energy functional with respect to the density matrix \mathbf{P} :

$$\mathcal{G}(\mathbf{P}) = \text{tr} \mathbf{h} \mathbf{P} + \frac{1}{2} \text{tr} \mathbf{G}(\mathbf{P}) \mathbf{P} + E_s(\mathbf{P}), \quad (33)$$

where \mathbf{h} and \mathbf{G} are the customary one and two electron integrals matrices and the dependence of the solvation energy E_s on the density matrix is due to the dependence of the electrostatic potential, and therefore of the reaction potential, on the electronic density.³⁹ The minimization of the functional in Eq. (33) leads to the SCF equations

$$\tilde{\mathbf{F}} \mathbf{C} = \mathbf{S} \mathbf{C} \mathbf{E}, \quad \mathbf{P} = \mathbf{C} \mathbf{C}^\dagger, \quad (34)$$

where \mathbf{S} is the overlap matrix, \mathbf{C} is the matrix of the molecular orbital coefficients, \mathbf{E} is the diagonal matrix of the orbital energies and the Fock Matrix $\tilde{\mathbf{F}}$, which accounts for the solvent, is defined as the gradient of the functional in Eq. (33) with respect to the density matrix:

$$\tilde{\mathbf{F}} = \frac{\partial \mathcal{G}(\mathbf{P})}{\partial \mathbf{P}} = \mathbf{h} + \mathbf{G}(\mathbf{P}) + \frac{\partial E_s(\mathbf{P})}{\partial \mathbf{P}} = \mathbf{F} + \mathbf{F}^s. \quad (35)$$

In order to compute the ddCOSMO Fock matrix contribution \mathbf{F}^s , the derivatives of Eq. (32) with respect to the density matrix are to be evaluated:³⁹

$$F_{\mu\nu}^s = \frac{\partial E_s}{\partial P_{\mu\nu}} = \frac{1}{2} f(\varepsilon) \left[\left\langle \frac{\partial \Psi}{\partial P_{\mu\nu}}, X \right\rangle + \left\langle \Psi, \frac{\partial X}{\partial P_{\mu\nu}} \right\rangle \right]. \quad (36)$$

The first term requires a numerical integration, as it contains the derivatives of Ψ , and can be computed, by using Becke's scheme, as follows:

$$\left\langle \frac{\partial \Psi}{\partial P_{\mu\nu}}, X \right\rangle = \sum_{j=1}^M \sum_{lm} \frac{4\pi}{2l+1} [X_j]_l^m \times \sum_{n=1}^{N_b^j} \tau_n^j \chi_{\mu}(\mathbf{r}_n^j) \chi_{\nu}(\mathbf{r}_n^j) \frac{x_{<}^l}{x_{>}^{l+1}} Y_l^m(\mathbf{s}_n^j). \quad (37)$$

Equation (37) can conveniently be implemented by precomputing the l, m sum introducing a vector

$$\eta_n^j = \sum_{lm} \frac{4\pi}{2l+1} \frac{x_{<}^l}{x_{>}^{l+1}} Y_l^m(\mathbf{s}_n^j) [X_j]_l^m,$$

so that

$$\left\langle \frac{\partial \Psi}{\partial P_{\mu\nu}}, X \right\rangle = \sum_{j=1}^M \sum_{n=1}^{N_g} \tau_n^j \eta_n^j \chi_{\mu}(\mathbf{r}_n^j) \chi_{\nu}(\mathbf{r}_n^j).$$

The second term can be computed analytically, as it involves the derivatives of the solute's potential. Let $[S_j]_l^m$ be the solution to the adjoint ddCOSMO system $L^*S = \Psi$:

$$\left\langle \Psi, \frac{\partial X}{\partial P_{\mu\nu}} \right\rangle = \left\langle S, \frac{\partial g}{\partial P_{\mu\nu}} \right\rangle. \quad (38)$$

Here,

$$\frac{\partial [g_j]_l^m}{\partial P_{\mu\nu}} = \sum_{n=1}^{N_g} w_n Y_l^m(\mathbf{y}_n) U_n^j V_{n,\mu\nu}^j,$$

where we recall that $\{\mathbf{y}_n, w_n\}_{n=1}^{N_g}$ are the points and weights of a Lebedev integration grid, the U_n^j are switching factors and $V_{n,\mu\nu}^j$ is the uncontracted potential at the n th grid point on sphere j . Once the adjoint ddCOSMO equations have been solved,²⁶ it is of course convenient to precompute the l, m sum and therefore to assemble directly

$$\sum_{j=1}^M \sum_{n=1}^{N_g} \left(w_n U_n^j \sum_{lm} [S_j]_l^m Y_l^m(\mathbf{y}_n) \right) V_{n,\mu\nu}^j = \sum_{j=1}^M \sum_{n=1}^{N_g} \xi_n^j V_{n,\mu\nu}^j. \quad (39)$$

Semiempirical methods can be seen as a special case of SCF-based methods where the integrals in the SCF equations are parametrized and the electronic density is approximated with a multipolar distribution which is obtained via a parametrized transformation of the density matrix:⁴⁰

$$[\Theta_j]_l^m = \sum_{\mu\nu} \Lambda_{jlm,\mu\nu} P_{\mu\nu}, \quad (40)$$

where the linear transformation Λ contains the semiempirical parameters. The equations for classical solutes can then be

used to compute ddCOSMO-related quantities: in particular, the potential can be assembled as a simple sum of multipolar terms:

$$\Phi_n^j = \sum_{k=1}^M \sum_{l=0}^{L_s} \sum_{m=-l}^l \frac{4\pi}{2l+1} \frac{[\Theta_k]_l^m}{(v_n^{jk})^{l+1}} Y_l^m(\mathbf{s}_n^{jk}), \quad (41)$$

where the quantities v_n^{jk} and \mathbf{s}_n^{jk} have been defined in Sec. II and L_s here is the maximum angular momentum of the semiempirical multipolar expansion, which is related to the choice of the method. Furthermore, Eq. (24) can be used for the Ψ vector. The ddCOSMO contribution to the Fock operator for semiempirical methods can be obtained exploiting the following chain rule:

$$\mathbf{F}^s = \frac{\partial E^s}{\partial \mathbf{P}} = \frac{\partial E^s}{\partial \Theta} \frac{\partial \Theta}{\partial \mathbf{P}} = \mathbf{\Lambda}^\dagger \mathbf{\Xi},$$

where we have introduced the multipolar Fock matrix $\mathbf{\Xi}$. Such a matrix is assembled as the derivative of the ddCOSMO energy with respect to the multipoles Θ :

$$[\Xi_i]_l^m = \frac{\partial E^s}{\partial [\Theta_i]_l^m}. \quad (42)$$

Again, there are two different contributions, the first arising from the derivatives of Ψ , the other from the derivatives of X . Through some algebra, one gets

$$[\Xi_i]_l^m = \frac{\partial E^s}{\partial [\Theta_i]_l^m} = \frac{1}{2} f(\varepsilon) \left[\frac{4\pi}{2l+1} [X_i]_l^m \delta_{l \leq L_s} + \sum_{j=1}^M \sum_{l'm'} [S_j]_{l'}^{m'} \frac{\partial [g_j]_{l'}^{m'}}{\partial [\Theta_i]_l^m} \right], \quad (43)$$

where the derivative of the right-hand side g are

$$\begin{aligned} \frac{\partial [g_j]_{l'}^{m'}}{\partial [\Theta_i]_l^m} &= \sum_{n=1}^{N_g} w_n Y_{l'}^{m'}(\mathbf{y}_n) U_n^j \frac{\partial \Phi_n^j}{\partial [\Theta_i]_l^m} \\ &= \sum_{n=1}^{N_g} w_n Y_{l'}^{m'}(\mathbf{y}_n) U_n^j \frac{4\pi}{2l+1} \frac{1}{(v_n^{jk})^{l+1}} Y_l^m(\mathbf{s}^{ji}). \end{aligned} \quad (44)$$

We recall that we are assuming that the $[\Theta_j]_l^m$ are spherical multipoles. We conclude this section with a remark about polarizable force fields. In polarizable force fields, the multipoles, or some of the multipoles, describing the solute are determined by solving some polarization equations, which are usually derived by minimizing an energy functional with respect to the polarizable multipoles (usually, fluctuating charges or induced dipoles).^{41,42} The coupling of polarizable force fields with ddCOSMO can be achieved in the same exact way it is achieved for semiempirical methods, where the multipoles $[\Theta_j]_l^m$ are to be replaced with the polarizable multipoles and where the SCF equations have to be replaced with the polarization equations. The computation of the potential, of Ψ and of the ddCOSMO contribution to the polarization equations remains unchanged. Notice finally that, for what concerns the ddCOSMO implementation, a mixed QM/MM description does not introduce significant complications, as

one will just have to use the appropriate equations to compute the potential and Ψ vector and their derivatives for the QM and MM atoms, respectively: all other quantities only depend on the geometry of the system and are not affected by the specific level of theory.

IV. ANALYTICAL DERIVATIVES

In this section, we will derive the ddCOSMO contribution to the forces as analytical derivatives of the energy. We will focus on QM solutes; the derivation for classical solutes is a sub-case which has already been dealt with in Ref. 26. We will denote the derivation with respect to a generic nuclear coordinate with the superscript x . In order to compute the analytical derivatives of the energy, three contributions have to be assembled:

$$E_s^x = \frac{1}{2} f(\varepsilon) [\langle \Psi^x, X \rangle + \langle \Psi, (L^{-1})^x g \rangle + \langle \Psi, L^{-1} g^x \rangle]. \quad (45)$$

The second contribution involves only geometrical quantities and has been thoroughly discussed in Ref. 26. The third contribution can be assembled as follows:

$$\begin{aligned} & \langle \Psi, L^{-1} g^x \rangle \\ &= \sum_{j=1}^M \sum_{n=1}^{N_g} w_n \sum_{lm} [S_j]_l^m Y_l^m(\mathbf{y}_n) [(U_n^j)^x \Phi_n^j + U_n^j (\Phi_n^j)^x]. \end{aligned} \quad (46)$$

Again, there are two separate contributions: the first involves the derivative of the potential weights and is a purely geometrical term: we refer the reader to Ref. 26 for a detailed derivation. The second contribution of the third term involves the derivatives of the molecular potential. By introducing the vector ξ_n^j as in Eq. (39), this term can be recast as

$$\sum_{j=1}^M \sum_{n=1}^{N_g} \xi_n^j (\Phi_n^j)^x, \quad (47)$$

where

$$\begin{aligned} (\Phi_n^j)^x &= \sum_{\mu\nu} P_{\mu\nu} \left(\int_{\Omega} \frac{[\chi_{\mu}(\mathbf{r})\chi_{\nu}(\mathbf{r})]^x}{|\mathbf{R}_j + r_j \mathbf{y}_n - \mathbf{r}|} d\mathbf{r} \right. \\ & \quad \left. - \int_{\Omega} \frac{\chi_{\mu}(r)\chi_{\nu}(r)}{|\mathbf{R}_j + r_j \mathbf{y}_n - \mathbf{r}|^3} (\mathbf{R}_j + r_j \mathbf{y}_n - \mathbf{r})^x d\mathbf{r} \right). \end{aligned} \quad (48)$$

In Eq. (48), the first term gives a nonzero contribution only when at least one of the basis functions is centered on the atom which the x coordinate belongs to; also, the second term vanishes if such an atom is not the j th atom.

The first term of Eq. (45) only appears in quantum mechanical computations, where it involves the derivatives of Eq. (31). In particular, both the derivatives of the basis functions and the derivatives of the Becke's quadrature weights, which depend on the geometry, have to be computed: as already mentioned in Sec. III, the integration weights in

Becke's scheme are smooth functions of the distance between atoms,^{37,38} which makes their differentiation possible. Through some algebra, one obtains

$$\begin{aligned} \langle \Psi^x, X \rangle &= \sum_{\mu\nu} P_{\mu\nu} \sum_{j=1}^M \sum_{lm} \frac{4\pi}{2l+1} [X_j]_l^m \\ & \times \sum_{n=1}^{N_b^j} [\tau_n^j \chi_{\mu}(\mathbf{x}_n^j) \chi_{\nu}(\mathbf{x}_n^j)]^x \frac{x_{<}^l}{x_{>}^{l+1}} Y_l^m(\mathbf{s}_n^j). \end{aligned} \quad (49)$$

As for the evaluation of the Fock matrix contributions, it is convenient to precompute the l, m sum by assembling an intermediate quantity

$$\eta_n^j = \sum_{lm} \frac{4\pi}{2l+1} \frac{x_{<}^l}{x_{>}^{l+1}} Y_l^m(\mathbf{s}_n^j) [X_j]_l^m$$

and the density derivatives

$$\rho^x(\mathbf{x}_n^j) = \sum_{\mu\nu} P_{\mu\nu} (\chi_{\mu}^x(\mathbf{x}_n^j) \chi_{\nu}(\mathbf{x}_n^j) + \chi_{\mu}(\mathbf{x}_n^j) \chi_{\nu}^x(\mathbf{x}_n^j))$$

so that we can directly assemble

$$\langle \Psi^x, X \rangle = \sum_{j=1}^M \sum_{n=1}^{N_b^j} [\tau_n^j \rho^x(\mathbf{x}_n^j) + (\tau_n^j)^x \rho(\mathbf{x}_n^j)] \eta_n^j. \quad (50)$$

V. IMPLEMENTATION AND NUMERICAL RESULTS

In this section we will detail some crucial aspects of the numerical implementation and we will demonstrate the linear scaling properties of the ddCOSMO algorithm. The numerical procedure used to solve the ddCOSMO equations and to compute the contributions to the forces arising from the derivatives of the ddCOSMO matrix and of the weights have been extensively discussed in Ref. 26; here we will focus on the computation of the right-hand sides Φ and Ψ , on the evaluation of the Fock matrix contributions and on the relative impact of the various steps of a ddCOSMO computation on the overall computational cost.

A. Implementation

A ddCOSMO computation is made of various points that we recapitulate here.

1. Preliminary computations

Various geometrical quantities and intermediates need to be computed only once, given a molecular geometry. The resulting information remains available for further computations, i.e., for subsequent SCF cycles or for the computation of response properties. In order to produce a code able to run even on dated or limited machines, only the quantities that require a very small amount of memory are always precomputed; other ones are only assembled and stored if enough memory is available. The preliminary steps are summarized as follows:

1. Build various geometrical quantities, including a neighbor list for the spheres, the Lebedev grid and weights

and the U_n^i weights. At this step, it is also convenient to assemble, for each grid point, a set of spherical harmonics $\{Y_l^m(\mathbf{y}_n)\}$ to be used for numerical integrations as in Eq. (12). All these quantities require an amount of memory which is either constant or linear in the number of atoms.²⁶

2. If enough memory is available, it is also convenient to precompute the W_n^{jk} and t_n^{jk} as in Eqs. (10) and (11). The amount of memory required is, for each vector, $N_g \times M \times \langle N_{ng} \rangle$, i.e., a quantity proportional to the number of atoms where the prefactor depends on the number of grid points and on the average number of neighbors per sphere $\langle N_{ng} \rangle$. The latter quantity, for standard molecular systems, is usually between 20 and 25: the memory requirements for this step are rather low.
3. If even more memory is available, it is also possible to precompute all the spherical harmonics at the \mathbf{s}_n^{jk} points. Sparse storage can be used in order to limit the memory requirements, by only storing the spherical harmonics that are needed in the order they are used. This can be accomplished as follows: for each sphere Ω_j , loop over the neighbors Ω_k , $k \in \mathcal{N}_j$ and over the grid points. If a grid point on Γ_j is inside Ω_k , compute \mathbf{s}_n^{jk} and assemble the $(N+1)^2$ spherical harmonics and store them; then increment a counter ic . At the end of the two loops, $ic+1$ will be the starting position for the spherical harmonics on the next sphere. Let N_{int} be the final value of ic : the quantity of memory needed to store all the spherical functions will be $N_{int} \times (N+1)^2$, which is linear in the number of atoms. Notice however that the constant of proportionality can be rather large, as for a molecular system the number of internal points per sphere is usually roughly 200-300 using a grid with $N_g = 110$, and N ranges, depending on the precision required, from 6 to 10: for each atom, up to 30 000 integer words of memory can be required.
4. If the adjoint ddCOSMO system has to be solved, the previous two points have to be repeated with reverse indexes, i.e., assembling W_n^{kj} , t_n^{kj} , and $Y_l^m(\mathbf{s}_n^{kj})$ and therefore doubling the required quantity of memory.

We remark again that if not enough memory is available, the points 2-4 can be skipped. We have a very efficient procedure²⁶ to assemble a set of spherical harmonics at a point \mathbf{s} which only requires the evaluation of one square root and no transcendent function evaluation in order to produce $\{Y_l^m(\mathbf{s})\}$, $0 \leq l \leq N$, $-l \leq m \leq l$: the evaluation of the spherical harmonics is not the bottleneck of the computation.

2. Main computations

In order to include ddCOSMO in an energy computation, up to three main steps need to be carried out in order to compute up to six fundamental quantities. More in detail:

1. Assemble the right-hand sides. Compute the solute's potential Φ , according to either Eq. (19) or Eq. (41) depending on the level of theory; notice that this step can

be performed using the FMM (possibly in its continuous generalization) to achieve, for large enough systems, a linear scaling in the computational cost. Then, compute the Ψ vector according to either Eq. (24) or Eq. (31). The classical (or semiempirical) case is trivial; the quantum mechanical one requires the numerical evaluation of an integral. The cost associated with such an operation is, for large enough systems, still linear in the size of the system, but with a large constant prefactor which depends on the number of grid points per atom and on the maximum angular momentum of the spherical harmonics to be evaluated.

2. Solve the equations. The ddCOSMO local ASC X is obtained by solving the ddCOSMO equations (15) according to the iterative procedure described in Ref. 26; for quantum mechanical computations only, in order to compute the ddCOSMO Fock matrix contributions, it is also required to solve the adjoint ddCOSMO equations in order to compute the S coefficients. Here, we point out that the efficiency of this step depends on whether the precomputation described in the previous paragraph are feasible; the subtlety here is that the major advantage of precomputing the spherical harmonics and geometrical quantities is not just the obvious time saving due to not having to recompute them at each iteration, but also a more efficient vectorization of the resulting code by the compiler. The latter aspect is particularly important because the evaluation of a multipolar expansion, which is needed for the solution of the ddCOSMO equations, is indeed the most demanding part of the ddCOSMO algorithm.
3. For quantum mechanical solutes only, the ddCOSMO Fock matrix contributions have to be computed according to either Eqs. (37) and (38) or Eq. (43). Notice that there are two different ddCOSMO contributions to the Fock matrix: for the analytical one (Eq. (38)), which we will label in the following F1, the same considerations than for Φ can be made, including the use of the FMM to accelerate the computation, while for the numerical one (Eq. (37)), which we will label in the following F2, what has been stated for the computation of Ψ applies.

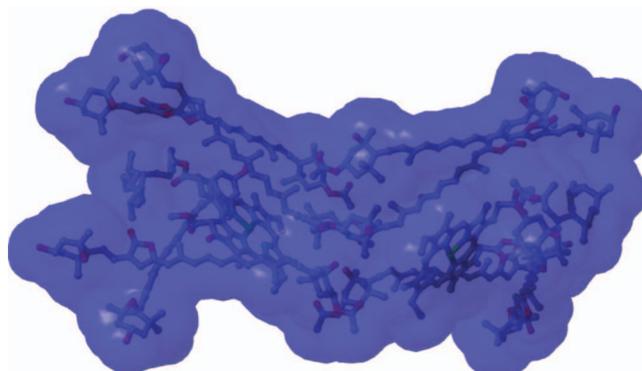


FIG. 2. Light-harvesting complex of the PCP protein within its van der Waals cavity.

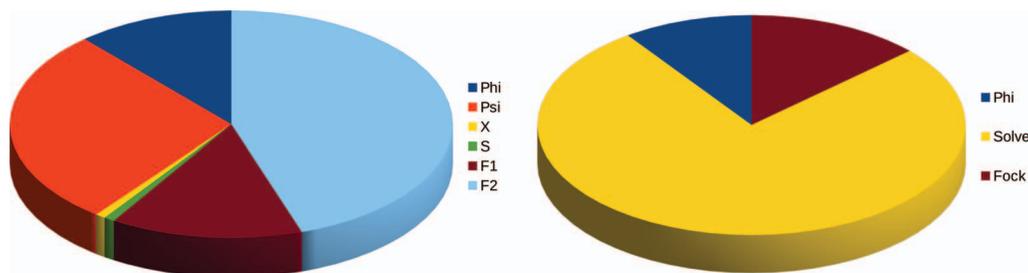


FIG. 3. Relative computational times for the various tasks involved in a ddCOSMO (left, total elapsed time: 7.07 s) and in a traditional (right, total elapsed time 7.12 s) computation for a pure QM computation. Phi (blue): solute's potential; Psi (orange): Ψ vector as in equation (31); X/Solve (yellow): solution to the direct ddCOSMO equations or to the CSC equations, respectively; S (green): solution to the adjoint ddCOSMO equations; F1/Fock (burgundy): analytical contribution to the Fock operator as in Eq. (39) or CSC Fock matrix contribution, respectively; F2 (light blue): numerical contribution to the Fock operator as in Eq. (37).

We conclude this section by remarking that in a traditional COSMO computation, three main steps involving only three fundamental quantities are involved: computing the solute's potential, solving the COSMO equations (the COSMO matrix is symmetric) and assembling the COSMO Fock-matrix contribution.

B. Numerical results

We will now present some examples in order to show the impact of the various contributions and the overall computational properties of the ddCOSMO method in various prototypical computations. As an example, we compute the SCF energy of the 611 peridinine of the light-harvesting complex of the peridinin chlorophyll protein⁴³ (PCP), either alone or together with the other chromophores included at the MM level. A pictorial representation of the chromophores cluster is reported in Figure 2. The ddCOSMO algorithm has been implemented in a locally modified development version of the Gaussian⁴⁴ suite of programs. All the calculations have been performed on a dual Xeon E2650 (2.0 GHz, 8 cores per processor) workstation equipped with 64 GB of 1.6 GHz DDR3 RAM; all the data reported refer to the first SCF cycle. There was always enough memory to precompute all the quantities described at the beginning of this section. We will compare the ddCOSMO implementation with the most recent and efficient traditional one, based on the continuous surface charge formalism¹⁸ (CSC), using the FMM to accelerate the

computation,²¹ for three different levels of theory which are detailed here as follows:

- Pure QM (DFT): we use the B3LYP⁴⁵ functional with Pople's 6-31G* basis,⁴⁶ the 611-peridinine (135 atoms) is used as a test system;
- Pure semiempirical: we use the PM6 hamiltonian;⁴⁷ the 611-peridinine is used as a test system;
- QM/MM: DFT B3LYP/6-31G* for the QM part, Universal Force Field³¹ (UFF) with QEq⁴⁸ charges for the MM one. The QM part is the 611 peridinine, the MM part consists of the other chromophores of the PCP, for a total of 1038 atoms.

In Figure 3, we report the relative computational effort spent in performing the various tasks associated to both a ddCOSMO and a CSC computation. The same data are reported for a semiempirical computation in Figure 4 and for a QM/MM one in Figure 5.

The absolute timings are reported for each step of each computation in Table I.

The first thing that appears from the graphs is that the solution to the COSMO equations, which is by far the bottleneck of a traditional computation, does not represent anymore the slow step in a ddCOSMO one. While two systems of linear equations ("X" and "S" labels) have to be solved in order to compute the ddCOSMO energy and Fock matrix contributions, the procedure is so efficient that, compared to the traditional one ("solve" label), it becomes a small part of the overall computational time. We remark here that, for

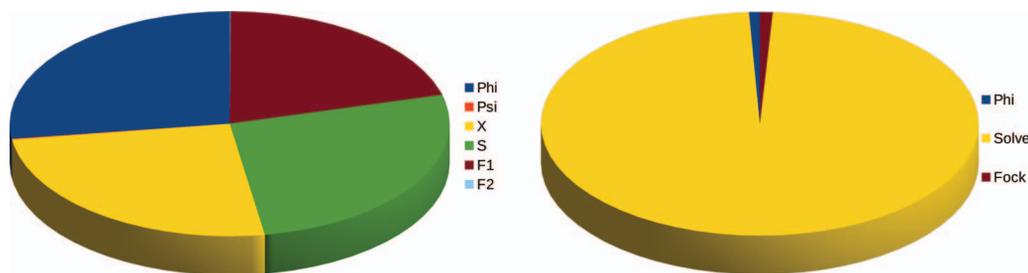


FIG. 4. Relative computational times for the various tasks involved in a ddCOSMO (left, total elapsed time 0.20 s) and in a traditional (right, total elapsed time 5.04 s) computation for a pure Semiempirical computation. Phi (blue): solute's potential; Psi (orange): Ψ vector as in equation (24); X/Solve (yellow): solution to the direct ddCOSMO equations or to the CSC equations, respectively; S (green): solution to the adjoint ddCOSMO equations; F1/Fock (burgundy): analytical contribution to the Fock operator as in Eq. (39) or CSC Fock matrix contribution, respectively; F2 (light blue): second contribution to the Fock operator as in Eq. (43).

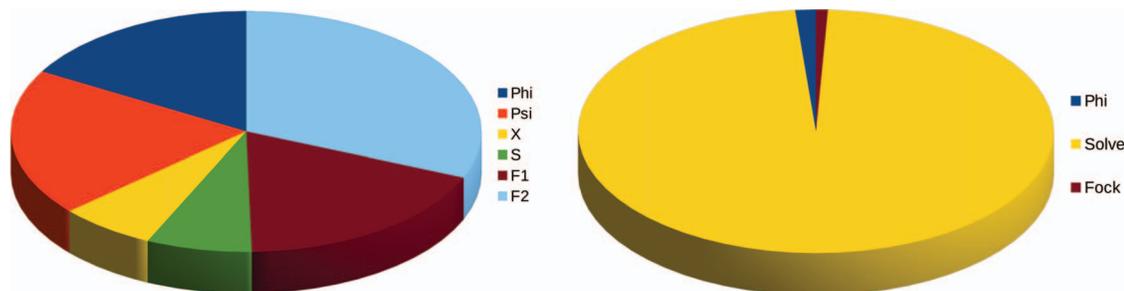


FIG. 5. Relative computational times for the various tasks involved in a ddCOSMO (left, total elapsed time 10.03 s) and in a traditional (right, total elapsed time 173.20 s) computation for a hybrid QM/MM computation. Phi (blue): solute’s potential; Psi (orange): Ψ vector as in Eq. (31) or Eq. (24); X/Solve (yellow): solution to the direct ddCOSMO equations or to the CSC equations, respectively; S (green): solution to the adjoint ddCOSMO equations; F1/Fock (burgundy): analytical contribution to the Fock operator as in Eq. (39) or CSC Fock matrix contribution, respectively; F2 (light blue): numerical contribution to the Fock operator as in Eq. (37) or (43).

the CSC computations, we are using the FMM; normally, either matrix inversion or $\mathcal{O}(N^2)$ iterative techniques are used to solve the COSMO equations: nevertheless, ddCOSMO dramatically changes the relative importance of the various step in terms of CPU time. This is particularly noteworthy for semiempirical and hybrid QM/MM computations, where, in traditional computations, the solution to the COSMO equations is responsible of almost the whole computational effort.

However, ddCOSMO introduce two numerical integration steps (“Psi” and “F2” labels) that are not present in the traditional computations: such steps have a strong impact on the overall computational time: while some code optimization effort could alter slightly the picture, such steps are the bottleneck of a ddCOSMO computation. For a pure QM description, the impact of the two numerical integrations is such that it balances the gain introduced by the ddCOSMO algorithm, making the traditional and ddCOSMO procedures roughly equivalent in terms of elapsed time, as can be inferred from Table I. However, when either a semiempirical method is used and no numerical integration is required, ddCOSMO becomes immediately advantageous even on medium-sized systems such as the ones analyzed.

The breakthrough performances of ddCOSMO are however even more apparent in the QM/MM computation: not only the computational effort needed to solve the ddCOSMO equations is relatively small—13% of the total, to be compared with 98% for CSC—but also the overhead due to the

presence of the MM part is very limited. In a QM/MM computation, in fact, the numerical integration has only to be carried out for the QM atoms; furthermore, assembling the MM part of the Ψ vector is extremely cheap. As a result, the time needed to assemble the Ψ vector and the numerical contribution to the Fock matrix for a QM/MM computation is the same than for an equivalent pure QM computation (i.e., a computation where the system is the QM portion of the QM/MM computation, as it is here), as it can be seen from Table I: in this situation, the efficiency of the ddCOSMO algorithm makes a huge difference on the overall computational time, which is reduced from almost 3 min to 10 s. Notice that the time to compute the solute’s potential and the analytical Fock matrix contributions is roughly the same between ddCOSMO and CSC, with small difference imputable to the operating system activity.

It is worth here to stress that ddCOSMO is especially suited for large systems which are accessible through hybrid methods, as in any case a pure QM computation on a system composed by thousands or tens of thousands of atoms would be so expensive that the fraction of computer time associated with the solvation method would be negligible independent of the implementation: on the contrary, if a hybrid scheme is used, the solution to the COSMO equations via the traditional CSC implementation can easily become the real computational bottleneck even by using the FMM, while a very small difference is observed in the overall computer time between a pure QM and a QM/MM computation if ddCOSMO is used.

To summarize, ddCOSMO introduces a non-negligible overhead in pure QM computations due to the necessity of computing numerically two integrals; on the other hand, this is compensated by a much faster solution to the solvation equations. In particular, as in a QM/MM simulation the ddCOSMO overhead is only associated to the QM portion, the new ddCOSMO algorithms vastly outperforms any traditional BEM implementation, including the ones based on the FMM.

TABLE I. Absolute timings (in seconds) for the various ddCOSMO and CSC tasks for a pure QM, pure semiempirical (SE), and hybrid QM/MM computation. Keys as in Figure 3.

Task	SE		QM		QM/MM	
	dd	CSC	dd	CSC	dd	CSC
Phi	0.0530	0.0703	0.8134	0.6969	1.6740	1.3666
Psi	0.0002		1.9835		1.9852	
X/Solve	0.0494	4.9328	0.0522	5.4797	0.6652	170.2033
S	0.0519		0.0557		0.7274	
F1/Fock	0.0406	0.0405	0.9942	0.9400	1.8419	1.6333
F2	0.0001		3.1696		3.1350	
Total	0.1951	5.0436	7.0687	7.1165	10.0288	173.2032

C. Scaling properties

We conclude this section by analyzing the scaling of the various contributions with respect to the size of the system. For the QM computations, we used the B3LYP

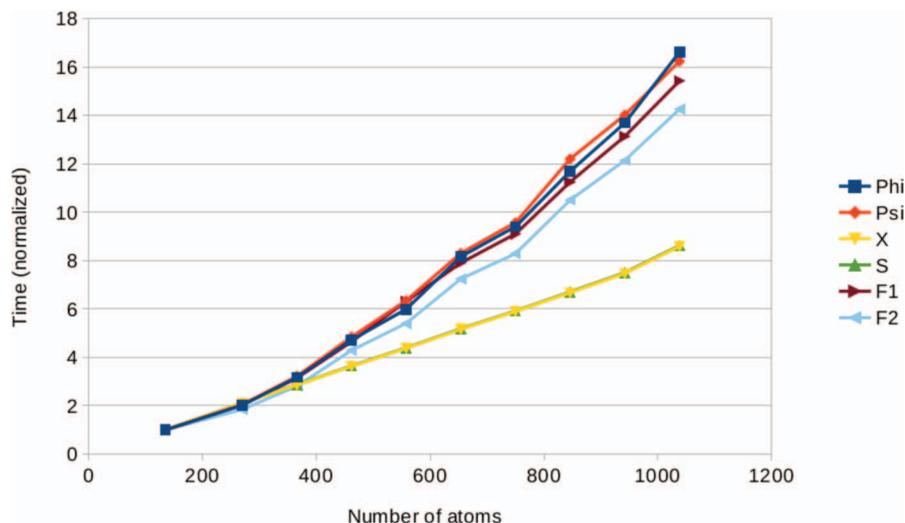


FIG. 6. Scaling of the various ddCOSMO contributions as a function of the system size. Keys as in Fig. 3.

functional in conjunction with the 6-31G* basis set, which means more than 8000 basis functions for the largest system. For the semiempirical computations, we used the PM6 Hamiltonian. In Figure 6, the timings (normalized with respect to the time needed for the smallest system for graphical reasons) of the various ddCOSMO steps are reported as a function of the system size for the DFT computations; the same analysis is repeated for the PM6 computations in Figure 7. Notice that, for the semiempirical method, the “Psi” and “F2” timings are not reported, as they are negligible. We notice from both Figures 6 and 7 that the various steps of a ddCOSMO computation show, for the shape and size of the systems analyzed, slightly different scaling properties. Among the various quantities, two are computed by using the FMM (Phi and F1), two by numerical quadrature (Psi and F2) and two with the ddCOSMO algorithm itself. The FMM quantities exhibit a scaling which deviates slightly from linearity. Unfortunately, the global scaling of the whole potential computation is difficult to analyze, as in preliminary operations on the electronic density and basis functions some weakly quadratic steps are

involved; moreover, increasing the size of the system can require a further level of boxification, which makes the computation more expensive. Finally, the linear scaling regime is achieved for large enough systems, which might not be the case for the smaller ones reported in Figures 6 and 7. Nevertheless, the computation of the potential and of the analytical Fock Matrix contributions greatly benefits from the use of the FMM and the overall scaling is much better than quadratic. For semiempirical computations this analysis is simplified as, being the solute’s density approximated by nuclear point multipoles, assembling the potential is a pure FMM computation and no quadratic step is involved. It is possible to see in Fig. 7 that the linear scaling regime is quickly achieved starting from roughly 400 atoms both for the potential and for the analytical Fock matrix contribution.

The numerical quadrature shows as well a small deviation from the ideal linear scaling with respect to the size of the system; its behavior is slightly closer to the ideal one than for the FMM and overall very good. We expect that this result can be further improved by using localized orbitals and for larger

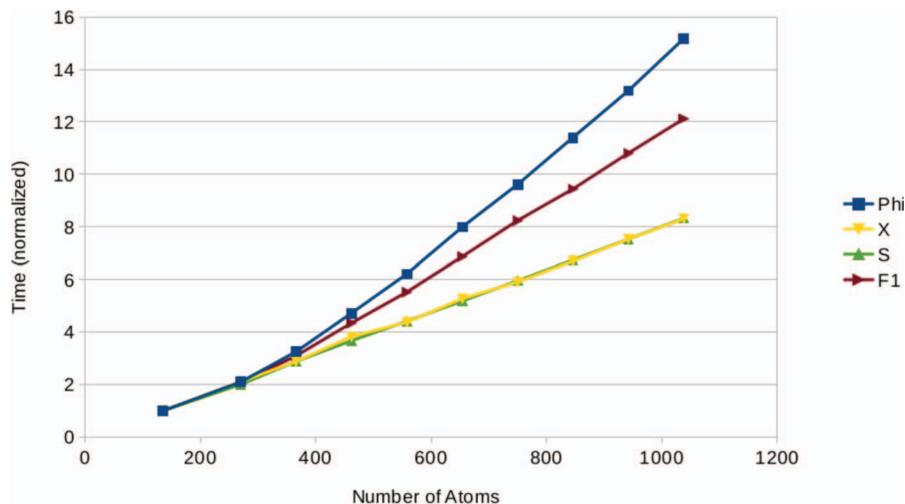


FIG. 7. Scaling of the various ddCOSMO contributions as a function of the system size. Keys as in Fig. 4.

systems. It is here worth pointing out that a full QM computation on a 1000 atoms system is *per se* a quite challenging computation on standard computers: the overhead represented by the solvent is therefore not so important as for large, cheap QM/MM or semiempirical computations.

A perfectly linear behavior is finally observed for the two ddCOSMO computations: the scaling properties of ddCOSMO, as it has been shown in Ref. 26, do not depend on the molecular shape (i.e., ddCOSMO enjoys the same scaling properties for both linear and globular molecules), but rather on the connectivity; furthermore, as little as one hundred atoms are enough for achieving the linear scaling regime. We want here to stress again how the solution to the polarization equations is, for standard discretizations, by far the bottleneck of the overall computation: improving the scaling and the overall cost of such a task is therefore the most important goal in order to extend the applicability of polarizable continuum solvation models.

VI. CONCLUSION AND PERSPECTIVES

In this paper we have presented the comprehensive implementation of ddCOSMO, providing every element needed to implement ddCOSMO in conjunction with different levels of theory for the solute. In particular, we have recapitulated the classical and semiempirical cases and we have focused on solutes described via a continuous charge density, such as the electronic density. The equations that we have provided for the solute-solvent coupling are general and can be used not only with SCF-based methods such as Hartree-Fock or Density Functional Theory, but, in principle, even with the most sophisticated post-Hartree-Fock methods, provided that the equations are modified to account for the solvent. Such modifications, however, do not depend on the specific discretization of the COSMO equations and have been discussed elsewhere for various levels of theory (see, for instance, Refs. 34–36). We have also discussed the implementation of analytical gradients; as the implementation of electric response properties and electronic transition energies can be obtained with the same integrals and procedures presented here, such properties have not been explicitly discussed.

From the analysis of the computational costs of the various tasks associated with a ddCOSMO computation, we have showed that ddCOSMO is competitive with the fastest available implementations of COSMO, i.e., the one that rely on the FMM to achieve linear scaling in CPU time, for pure QM solutes; however, ddCOSMO exhibits its real strength in hybrid, QM/MM computations, where a very small computational overhead is introduced by the MM part thanks to the very efficient domain decomposition discretization: when compared to a traditional implementation, ddCOSMO becomes rapidly very advantageous even on medium sized systems. We point out that we have based our comparisons on a real life example and not on one tailored purposely to show the good properties of ddCOSMO, which would have been a larger-sized system or, for the hybrid computation, a model with a much larger MM region: we believe that ddCOSMO represents the ideal paradigm to perform hybrid computations, as it allows

an atomistic description of a large system without worrying about the increase in simulation time.

ddCOSMO is a relatively new algorithm, having been introduced about one year ago, and while its implementation starts to be mature and various properties can already be computed with different levels of theory, further extensions are still under active investigation. In particular, as briefly mentioned in this paper, a first implementation of ddCOSMO coupled to a dipole-based polarizable force field has already shown performances good enough to perform polarizable molecular dynamics simulations within a polarizable continuum: these results will be object of a future publication. The coupling of ddCOSMO with a polarizable force field can also be exploited for a fully polarizable, three-layered QM/MM/Continuum model: this aspect is also under active investigation in our labs with a particular focus on excited states properties and phenomena. More specifically about ddCOSMO, various improvements are being considered. The extension of the domain decomposition formalism to the dielectric model is under close scrutiny and a pilot code is being developed. This is particularly important as the COSMO model accounts for the dielectric nature of solvents only in an approximate way, by introducing a scaling factor: while this approximation is in general very good for very polar solvents, it can introduce an error when non-polar solvents are considered. New definitions of the molecular cavity, with particular emphasis on the solvent excluded surface, have also been considered: while there are several mathematical difficulties both in the parametrization of the surface and in finding a suitable way to introduce the domain decomposition paradigm, some progress has been done and the topic is under analysis. With the perspective of describing very large systems, some elementary coarse-graining of the van der Waals cavity has also been explored. Finally, a parametrization of the non-electrostatic contributions to the solvation energy specific for ddCOSMO is under development.

ACKNOWLEDGMENTS

This work was supported in part by the ANR Manif and the French state funds managed by CALSIMLAB and the ANR within the Investissements d’Avenir program under reference ANR-11-IDEX-0004-02. F.L. and B.M. gratefully acknowledge Gaussian, Inc. for funding.

¹J. Tomasi, B. Mennucci, and R. Cammi, *Chem. Rev.* **105**, 2999 (2005).

²C. Cramer and D. Truhlar, *Chem. Rev.* **99**, 2161 (1999).

³B. Mennucci, *WIREs Comput. Mol. Sci.* **2**, 386 (2012).

⁴A. Klamt, *WIREs Comput. Mol. Sci.* **1**, 699 (2011).

⁵*Continuum Solvation Models in Chemical Physics*, edited by B. Mennucci and R. Cammi (Wiley, New York, 2007).

⁶B. Mennucci, *J. Phys. Chem. Lett.* **1**, 1666 (2010), and references therein.

⁷A. H. Steindal, K. Ruud, L. Frediani, K. Aidas, and J. Kongsted, *J. Phys. Chem. B* **115**, 3027 (2011).

⁸F. Lipparini, C. Cappelli, and V. Barone, *J. Chem. Theory Comput.* **8**, 4153 (2012).

⁹C. Curutchet, A. Muñoz-Losa, S. Monti, J. Kongsted, G. D. Scholes, and B. Mennucci, *J. Chem. Theory Comput.* **5**, 1838 (2009).

¹⁰S. Caprasecca, C. Curutchet, and B. Mennucci, *J. Chem. Theory Comput.* **8**, 4462 (2012).

¹¹E. Boulanger and W. Thiel, *J. Chem. Theory Comput.* **8**, 4527 (2012).

¹²F. Lipparini, C. Cappelli, G. Scalmani, N. De Mitri, and V. Barone, *J. Chem. Theory Comput.* **8**, 4270 (2012).

- ¹³F. Lipparini, C. Cappelli, and V. Barone, *J. Chem. Phys.* **138**, 234108 (2013).
- ¹⁴E. Cancès, in *Continuum Solvation Models in Chemical Physics*, edited by B. Mennucci, and R. Cammi (Wiley, New York, 2007), Chap. 1.2, pp. 29–48.
- ¹⁵E. Cancès, B. Mennucci, and J. Tomasi, *J. Chem. Phys.* **107**, 3032 (1997).
- ¹⁶E. Cancès and B. Mennucci, *J. Math. Chem.* **23**, 309 (1998).
- ¹⁷D. York and M. Karplus, *J. Phys. Chem. A* **103**, 11060 (1999).
- ¹⁸G. Scalmani and M. J. Frisch, *J. Chem. Phys.* **132**, 114110 (2010).
- ¹⁹A. W. Lange and J. M. Herbert, *J. Phys. Chem. Lett.* **1**, 556 (2010).
- ²⁰A. W. Lange and J. M. Herbert, *J. Chem. Phys.* **133**, 244111 (2010).
- ²¹G. Scalmani, V. Barone, K. Kudin, C. Pomelli, G. Scuseria, and M. Frisch, *Theor. Chem. Acc.* **111**, 90 (2004).
- ²²L. Greengard and V. Rokhlin, *J. Comput. Phys.* **73**, 325 (1987).
- ²³F. Lipparini, G. Scalmani, B. Mennucci, E. Cancès, M. Caricato, and M. J. Frisch, *J. Chem. Phys.* **133**, 014106 (2010).
- ²⁴F. Lipparini, L. Lagardère, G. Scalmani, B. Stamm, E. Cancès, Y. Maday, J.-P. Piquemal, M. J. Frisch, and B. Mennucci, *J. Phys. Chem. Lett.* **5**, 953 (2014).
- ²⁵E. Cancès, Y. Maday, and B. Stamm, *J. Chem. Phys.* **139**, 054111 (2013).
- ²⁶F. Lipparini, B. Stamm, E. Cancès, Y. Maday, and B. Mennucci, *J. Chem. Theory Comput.* **9**, 3637 (2013).
- ²⁷A. Klamt and G. Schuurmann, *J. Chem. Soc., Perkin Trans. 2* **1993**, 799.
- ²⁸M. Cossi, N. Rega, G. Scalmani, and V. Barone, *J. Comput. Chem.* **24**, 669 (2003).
- ²⁹D. Chipman, *J. Chem. Phys.* **110**, 8012 (1999).
- ³⁰E. Cancès and B. Mennucci, *J. Chem. Phys.* **115**, 6130 (2001).
- ³¹A. Rappe, C. Casewit, K. Colwell, W. Goddard, and W. Skiff, *J. Am. Chem. Soc.* **114**, 10024 (1992).
- ³²A. V. Marenich, C. J. Cramer, and D. G. Truhlar, *J. Phys. Chem. B* **113**, 6378 (2009).
- ³³P.-L. Lions, in *Proceedings of the First International Symposium on Domain Decomposition Methods for Partial Differential Equations*, Paris, France (SIAM, 1988), pp. 1–42.
- ³⁴F. Lipparini, G. Scalmani, and B. Mennucci, *Phys. Chem. Chem. Phys.* **11**, 11617 (2009).
- ³⁵R. Cammi, *J. Chem. Phys.* **131**, 164104 (2009).
- ³⁶M. Caricato, B. Mennucci, G. Scalmani, G. W. Trucks, and M. J. Frisch, *J. Chem. Phys.* **132**, 084102 (2010).
- ³⁷A. D. Becke, *J. Chem. Phys.* **88**, 2547 (1988).
- ³⁸R. E. Stratmann, G. E. Scuseria, and M. J. Frisch, *Chem. Phys. Lett.* **257**, 213 (1996).
- ³⁹F. Lipparini, G. Scalmani, B. Mennucci, and M. J. Frisch, *J. Chem. Theory Comput.* **7**, 610 (2011).
- ⁴⁰J. Khandogin, B. A. Gregersen, W. Thiel, and D. M. York, *J. Phys. Chem. B* **109**, 9799 (2005).
- ⁴¹F. Lipparini and V. Barone, *J. Chem. Theory Comput.* **7**, 3711 (2011).
- ⁴²F. Lipparini, L. Lagardère, B. Stamm, E. Cancès, M. Schnieders, P. Ren, Y. Maday, and J.-P. Piquemal, *J. Chem. Theory Comput.* **10**, 1638 (2014).
- ⁴³E. Hofmann, P. M. Wrench, F. P. Sharples, R. G. Hiller, W. Welte, and K. Diederichs, *Science* **272**, 1788 (1996).
- ⁴⁴M. J. Frisch, G. W. Trucks, H. B. Schlegel *et al.*, Gaussian, Revision h.36, Gaussian, Inc., Wallingford, CT, 2010.
- ⁴⁵A. Becke, *J. Chem. Phys.* **98**, 5648 (1993).
- ⁴⁶M. M. Francl, W. J. Pietro, W. J. Hehre, J. S. Binkley, M. S. Gordon, D. J. DeFrees, and J. A. Pople, *J. Chem. Phys.* **77**, 3654 (1982).
- ⁴⁷J. J. P. Stewart, *J. Mol. Model.* **13**, 1173 (2007).
- ⁴⁸A. Rappe and W. Goddard, *J. Phys. Chem.* **95**, 3358 (1991).