Linearly scaling computation of ddPCM solvation energy and forces using the fast multipole method.

A. Mikhalev,[1] M. Nottoli,[2] and B. Stamm[1, a]

[1] *Department of Mathematics, RWTH Aachen University, Schinkelstr. 2,*
52062 Aachen, Germany

[2] *Dipartimento di Chimica e Chimica Industriale, Università di Pisa,*
Via G. Moruzzi 13, 56124 Pisa, Italy

This paper proposes the first linear scaling implementation for the domain decomposition approach of the polarizable continuum model (ddPCM) for the computation of the solvation energy and forces. The ddPCM-equation consists of a (non-local) integral equation on the van der Waals (vdW) or solvent accessible surface (SAS) of the solute's cavity resulting in a dense solution matrix and, in turn, one matrix-vector multiplication has a quadratic arithmetic complexity with respect to the number of atoms of the solute molecule. The use of spherical harmonics as basis functions makes it natural to employ the fast multipole method (FMM) in order to provide an asymptotically linear scaling method. In the present paper, we employ the FMM in a non-uniform manner with a clusterization based on a recursive inertial bisection. We present some numerical tests illustrating the accuracy and scaling of our implementation.

[a] Electronic mail: best@acom.rwth-aachen.de

I. INTRODUCTION

The numerical computation of the energy, forces and properties based on derivatives of the energy of a given molecule is key in computational chemistry. However, most of the studied phenomena and reactions happen in the condensed phase implying that it is important to not only account for the studied molecule itself but also for the surrounding environment as the latter has a significant impact on the former. Within this setting, multiscale models, where the molecule of interest is described accurately and where the remaining part consisting of the environment is presented by a less accurate but cheaper description, are very effective in capturing all the important effects in a numerically efficient way for certain specific classes of problems.

Polarizable continuum solvation models (PCSMs)[7,12,27,38,39], as one class of multiscale models, have become a standard tool in computational chemistry. The polarizable continuum model (PCM)[28] that is considered in this article is one of such PCSMs. Of course, PCM is not the only polarizable continuum model. Indeed, the (linearized or non-linear) Poisson-Boltzmann model[5,10] is a generalization thereof formulated decades earlier taking the ionic strength of the solvent into account and the COSMO (model)[17,18] is a simplification of PCM for highly polar solvents. Implementations of PCSMs are routinely available in quantum chemistry codes and are widely used for diverse applications. However, polarizable continuum models, and thus also PCMs, can only be employed if such a homogeneous bulk environment can accurately model all effects of the solvent and the applicability is thus restricted to suitable cases.

The polarizable continuum solvation model (PCM) describes the electrostatic contribution to the solvation energy[38] by replacing the environment with a homogeneous, polarizable and continuum medium which fills all the space except the region occupied by the solute molecule. The environment is then characterized by its macroscopic bulk properties such as the dielectric permittivity. Given their characteristics, PCMs are particularly suited for the description of solvated molecules, where the environment indeed can be properly described by an average property such as the bulk dielectric permittivity. The PCMs present some numerical advantages over explicit descriptions of the environment: firstly PCMs depend on few parameters and can be easily employed, secondly they naturally take into account a statistical sampling over all the degrees of freedom of the environment, which on the contrary has to be taken into account when using explicit descriptions.

The classical electrostatic problem of PCMs can be formulated as a partial differential equation (PDE) on the entire domain $\mathbb{R}^3$ or by an equivalent boundary integral equation (BIE) on the boundary of the solute molecule which is also known as the integral equation formulation of PCM (IEF-PCM)[3]. The IEF-PCM has to be approximated and solved to determine the apparent surface density on the surface of the cavity due to the solvent polarization caused by the solute's density. Several numerical methods as direct discretizations of the IEF-PCM have been proposed[6,20,21,34,35].

This paper builds on the existing (meshless) domain decomposition paradigm for PCM[36], the derivation of analytical gradients[9] and its coupling to quantum mechanical models for the solute[31]. The domain decomposition paradigm first applies a domain decomposition of the computational domain and then discretizes each subdomain, combined with coupling conditions between the subdomains. While the previous contributions were focused on generalizing the method to a wider range of possible applications, a performant implementation was not yet considered. The discretization of the PCM boundary integral equations, such as the one used in ddPCM, has the following drawback: the corresponding matrices of resulting linear systems are dense leading to a natural quadratically scaling with respect to the number of atoms of the solute as reported in[9,31], in contrast to the ddCOSMO-method[4,25,37] which leads to a sparse solution matrix. In this paper we present, for the first time, a fast and efficient linearly scaling implementation of the ddPCM based on the fast multipole method (FMM) for the computation of solvation energies and analytical gradients.

The ddPCM is a part of the domain-decomposition paradigm that started with the linearly scaling ddCOSMO method[4,25] for the computation of the energy and forces and that has been successfully coupled to different models of the solute[23,24,37]. Besides the extension to the ddPCM that is discussed in this article, the domain decomposition framework has also been developed for the linearized Poisson-Boltzmann model[33] and a PCM[32] with variable dielectric constant that accounts for the Solvent Excluded Surface (SES) as separation between the solute and the solvent.

This paper is organized as follows. Without going into the derivation of the ddPCM-method, Section II sheds light on the quadratically scaling (with respect to the number of atoms) parts of the ddPCM-method and formalizes different operations that need to be computed by the FMM. Section III explains how our implementation achieves linear scaling in the computation of the ddPCM-energy and analytical forces using an adaptive binary treecode-based FMM. Section IV provides numerical tests that benchmark our implementation, while Section V is left for the conclusions.

II. THE DOMAIN DECOMPOSITION METHOD FOR THE PCM AND THE QUADRATICALLY SCALING BOTTLENECK

The domain decomposition approach for the polarized continuum solvent model (the ddPCM[31]) is an extension of the domain decomposition approach for the conductor-like screening solvent model (the ddCOSMO[24]): whereas the ddPCM allows a description of the solvent using an arbitrary, and possibly low, value of the bulk dielectric permittivity, the ddCOSMO restricts the cases to high conducting dielectrics as it is based on an infinite dielectric constant, as for a conducting material, as an approximation. This generalization comes with a price: while the ddCOSMO scales linearly with the number of atoms of the solute molecule for proteins for both energy and forces calculations, the ddPCM contains a quadratically scaling asymptotic bottleneck in terms of the complexity of the numerical method. This is due to the fact that the ddPCM-equations are non-local boundary integral equations on the solute's surface, which results in dense matrices after discretization.

Since the ddPCM numerical method for the computation of the energy and forces has already been reported in the literature[9,31,36], we only provide a brief introduction of the essential parts of the algorithm required for the subsequent discussion. In the following, we will highlight and discuss the additional steps compared to the ddCOSMO that are required for the ddPCM and which are of quadratic complexity, but we first need some basic notions to be defined.

We assume that the solute molecule consists of M atoms, each endowed of a partial charge q_j and located at a position $\mathbf{x}_j$. We also assume that the solute is placed in a cavity composed of interlocked balls which is characterized by the dielectric permittivity of the vacuum, whereas the rest of the environment is characterized by a dielectric permittivity of ε. Each ball $\Omega_j = B_{r_j}(\mathbf{x}_j)$ has a radius r_j and is centered on a corresponding atom j. The total cavity of the molecule is defined as $\Omega = \bigcup_{j=1}^{M} \Omega_j$, and depending on the values used as radii, it can be a van der Waals (vdW) cavity or a solvent accessible (SAS) cavity.

A. Solvation Energy

A high level description of how to compute the solvation energy by the ddPCM is presented in Algorithm 1, and we refer to 31 for a precise definition of the different quantities. Steps 3 and 4 are marked as the ddPCM steps, which are required to be performed by the ddPCM approach in addition to the linearly scaling ddCOSMO method. These two steps scale quadratic with respect to the number of atoms but, in the case of the ddCOSMO where $\varepsilon = \infty$ and thus $R_\varepsilon = R_\infty$, they vanish due to invertibility of R_∞ implying $\Phi_\varepsilon = \Phi$.

Algorithm 1 ddPCM computation of a solvation energy. The steps marked with "[ddPCM]" are specific of ddPCM, whereas those marked with "[ddCOSMO]" are common to both ddPCM and ddCOSMO.

```
1: [ddCOSMO] Compute the molecular potential at external grid points
2: [ddCOSMO] Perform the Lebedev quadrature to assemble $\Phi$
3: [ddPCM] Compute $\Phi_\infty = R_\infty \Phi$                {quadratic complexity}
4: [ddPCM] Solve $R_\varepsilon \Phi_\varepsilon = \Phi_\infty$                {quadratic complexity}
5: [ddCOSMO] Solve $LX = -\Phi_\varepsilon$
6: [ddCOSMO] Compute $\Psi$
7: [ddCOSMO] Obtain the energy $\frac{1}{2}\Psi^\top X$
```

While step 3 requires only a single matrix-vector multiplication, step 4 requires several of them to solve the linear system iteratively. Assuming that the amount of iterations does not grow with the number of atoms M (which is observed in practice), the quadratic scaling arises then from the matrix-vector products involving the matrices R_ε and R_∞.

B. Forces

Like in the previous subsection, Algorithm 2 presents a sketch of how to compute the forces as analytical gradients of the energy from the initial paper[31].

Algorithm 2 ddPCM computation of analytical gradients. The steps marked with "[ddPCM]" are specific of ddPCM, whereas those marked with "[ddCOSMO]" are common to both ddPCM and ddCOSMO.

```
 1: Execute algorithm 1 to acquire $\Psi, \Phi$ and $\Phi_\varepsilon$
 2: [ddCOSMO] Solve $L^\top S = \Psi$
 3: [ddPCM] Solve $R_\varepsilon^\top Y = S$                      {quadratic complexity}
 4: [ddPCM] Assemble $Q = S - \frac{4\pi}{\varepsilon-1} Y$          {linear complexity}
 5: [ddCOSMO] Expand $S$ at Lebedev grid points
 6: [ddPCM] Assemble $g = \Phi_\varepsilon - \Phi$                  {linear complexity}
 7: for $i = 1, \ldots, M$ do
 8:     [ddCOSMO] Contract $S^\top (\nabla_i L) X$
 9:     [ddPCM] Contract $Y^\top (\nabla_i R_\varepsilon) g$         {quadratic complexity}
10:     [ddCOSMO] Contract $-Q^\top \nabla_i (\Phi^A + \Phi^B)$
11:     [ddCOSMO] Contract $(\nabla_i \Psi)^\top X$
12: end for
```

Since the linear complexity of the forces calculations corresponding to ddCOSMO has already been reported in 25 and the derivation of the forces of ddPCM in detail in 9,31, we focus here again only on the quadratically scaling parts of the method:

- Step 3: requires the solution of the dense adjoint linear system involving the transposed/adjoint matrix $R_\varepsilon^\top$,

- Step 9: contains M contractions with different sparse 3-dimensional tensors $\nabla_i R_\varepsilon$ each one with $O(M)$ non-zero entries.

Note that the other ddPCM-related steps 4 and 6 are not mentioned, as they consist only of a subtraction of vectors of a length proportional to M.

C. The quadratically scaling bottlenecks

The quadratically scaling parts identified in Sections II A and II B are related to matrix-vector multiplications involving R_∞ and R_ε and the contraction of two vectors with $\nabla_i R_\varepsilon$ for all i. To further discuss the quadratic complexity of the mentioned parts, we need additional details which are first presented in the following.

The surface of the cavity is denoted by $\Gamma = \partial\Omega$ and for each atom j, the set $\Gamma_j^{\mathrm{e}} = \Gamma_j \cap \Gamma$ denotes the external part of Γ_j while $\Gamma_j^{\mathrm{i}} = \Gamma_j \cap \Omega$ consists of the internal part of Γ_j.

Further, the set $\{(s_n, \omega_n)\}_{n=1}^{N_g}$ of N_g grid points $s_n \in \mathbb{S}^2$ and weights $\omega_n > 0$ defines the Lebedev quadrature rule[22], which is used in ddCOSMO and ddPCM to numerically approximate integrals on the spheres.

On each Γ_i, the function $U_i(x)$ is a regularized characteristic function of Γ_i^{e} defined on Γ_i that depends on the positions and radii of all atoms. We define the external grid points as those Lebedev quadrature points where the characteristic function U_i is non-zero, i.e., $U_i(\mathbf{x}_i + r_i s_n) > 0$, and the switching area grid points are those Lebedev points where the gradient of the regularized characteristic function is non-zero, i.e., $\nabla_{\mathbf{x}_j} U_i(\mathbf{x}_i + r_i s_n) \neq 0$ for some j.

The discretization is based on real-valued L^2-orthonormal spherical harmonics as basis functions and the discretization parameter $\ell_{\max}$ denotes their maximal degree (to which we also refer to as the degree of the modelling harmonics). The spherical harmonics $Y_\ell^m(s)$ with $\ell = 0, \dots, \ell_{\max}$ and $m = -\ell, \dots, \ell$ is defined for a point $s \in \mathbb{S}^2$ on the unit sphere with spherical coordinates (θ, φ) as

$$Y_\ell^m(s) = \begin{cases} (-1)^m \sqrt{2} \sqrt{\frac{2\ell+1}{4\pi}} \sqrt{\frac{(n-m)!}{(n+m)!}} P_\ell^m(\cos\theta)\cos(m\varphi), & m > 0, \\ \sqrt{\frac{2\ell+1}{4\pi}} P_\ell^0(\cos\theta), & m = 0, \\ (-1)^m \sqrt{2} \sqrt{\frac{2\ell+1}{4\pi}} \sqrt{\frac{(n-|m|)!}{(n+|m|)!}} P_\ell^{|m|}(\cos\theta)\sin(|m|\varphi), & m < 0. \end{cases} \tag{1}$$

We further introduce by $\mathcal{P}_{\ell m}(x, y; r)$ the (single layer) potential generated by the spherical harmonic Y_ℓ^m on the (source) sphere of radius r centered at the (source) point y and evaluated at the (target) point x outside the ball $B_r(y)$:

$$\forall x \in \mathbb{R}^3 \setminus \overline{B_r(y)} : \quad \mathcal{P}_{\ell m}(x, y; r) = \frac{4\pi}{2\ell + 1} \left(\frac{r}{|x - y|} \right)^{\ell+1} Y_\ell^m \left(\frac{x - y}{|x - y|} \right). \tag{2}$$

It is then convenient to define $Y_{\ell m}^j(x) := Y_\ell^m \left(\frac{x - \mathbf{x}_j}{|x - \mathbf{x}_j|} \right)$, $r_j^\ell(x) := \left(\frac{|x - \mathbf{x}_j|}{r_j} \right)^\ell$ to shorten notation of the potential $\mathcal{P}_{\ell m}$:

$$\forall x \in \mathbb{R}^3 \setminus \overline{\Omega_j} : \quad \mathcal{P}_{\ell m}^j(x) := \mathcal{P}_{\ell m}(x, \mathbf{x}_j; r_j) = \frac{4\pi}{2\ell + 1} r_j^{-(\ell+1)}(x)\, Y_{\ell m}^j(x). \tag{3}$$

Additionally, we introduce the following notation

$$\sum_{\ell m}^{\ell_{\max}} = \sum_{\ell=0}^{\ell_{\max}} \sum_{m=-\ell}^{\ell} \tag{4}$$

to further shorten all the upcoming equations. We are now ready to summarize the quadratic operations in this formalism.

1. *Operation 1: The primal linear system*

Let us note, that both R_∞ and R_ε are $M \times M$ block matrices with $(\ell_{\max}+1)^2 \times (\ell_{\max}+1)^2$ blocks and a difference $R_\varepsilon - R_\infty = \frac{4\pi}{\varepsilon-1} I$ is just a scaled identity matrix. Hence, a matrix-vector product can be split into a linearly scaling diagonal and quadratically scaling off-diagonal parts. Let us therefore denote the matrix R as the matrix identical to R_∞ and R_ε for the off-diagonal blocks and with zero diagonal blocks:

$$R := \begin{bmatrix} 0 & R_{12} & \dots & R_{1M} \\ R_{21} & 0 & \dots & R_{2M} \\ \vdots & \vdots & \ddots & \vdots \\ R_{M1} & R_{M2} & \dots & 0 \end{bmatrix}. \tag{5}$$

An element on the (ℓ, m)-multi-indexed row and the (ℓ', m')-multi-indexed column of the off-diagonal block R_{ij} of the i-th block row and the j-th block column of R is defined as follows:

$$[R_{ij}]_{\ell\ell'}^{mm'} = -\sum_{n=1}^{N_g} \omega_n Y_\ell^m(s_n) U_i(\mathbf{x}_i + r_i s_n) \ell' [\mathcal{P}_{ij}^n]_{\ell'}^{m'} = -\sum_{n=1}^{N_g} [w_i^n]_\ell^m \ell' [\mathcal{P}_{ij}^n]_{\ell'}^{m'} \tag{6}$$

with

$$[w_i^n]_\ell^m := \omega_n Y_\ell^m(s_n) U_i(\mathbf{x}_i + r_i s_n) \tag{7}$$

and

$$[\mathcal{P}_{ij}^n]_{\ell'}^{m'} := \mathcal{P}_{\ell' m'}^j(\mathbf{x}_i + r_i s_n). \tag{8}$$

Hence, a matrix-vector product of the matrix R by some vector v becomes:

$$[(Rv)_i]_\ell^m = \sum_{\substack{j=1 \\ j\neq i}}^{M} \sum_{\ell' m'}^{\ell_{\max}} [R_{ij}]_{\ell\ell'}^{mm'} [v_j]_{\ell'}^{m'} = -\sum_{n=1}^{N_g} [w_i^n]_\ell^m \mathcal{P}^i(x_i + r_i s_n; \boldsymbol{v}_\ell), \tag{9}$$

where

$$\forall x \in \Gamma_i^{\mathrm{e}}: \qquad \mathcal{P}^i(x; \boldsymbol{v}_\ell) := \sum_{\substack{j=1\\ j\neq i}}^{M} \sum_{\ell' m'}^{\ell_{\max}} [(\boldsymbol{v}_\ell)_j]_{\ell'}^{m'} \, \mathcal{P}^j_{\ell' m'}(x), \tag{10}$$

and

$$[(\boldsymbol{v}_\ell)_j]_{\ell'}^{m'} := \ell' \, [v_j]_{\ell'}^{m'}. \tag{11}$$

The quantity $\mathcal{P}^i(x; \boldsymbol{v}_\ell)$ can be interpreted as the single layer potential at the given point x on the exterior part $\Gamma_i^{\mathrm{e}} = \Gamma_i \cap \partial\Omega$ of the i-th sphere generated by the charge distributions with moments $\ell'[v_j]_{\ell'}^{m'}$ on all other spheres $j \neq i$. Thus, the matrix-vector product can be divided into 3 subsequent steps:

1. Scale the input coefficients of spherical harmonics to obtain $\boldsymbol{v}_\ell$ as is described in (11). This step is linear by M in arithmetic complexity.

2. Compute the potential following equation (10) for every external grid point, i.e., for every i and n such that $U_i(x_i + r_i s_n) > 0$. This step requires $O(M^2)$ operations.

3. On each sphere i, sum (i.e., numerically integrate) the obtained potentials along with the local quantity $[w_j^n]_\ell^m$ according to (9). Each (numerical) integral is local resulting in $O(M)$ operations.

2. Operation 2: The adjoint linear system

In correspondence to the equation (6) the adjoint off-diagonal matrix-vector product reads:

$$[(R^\top z)_j]_{\ell'}^{m'} = \sum_{\substack{i=1\\ i\neq j}}^{M} \sum_{\ell m}^{\ell_{\max}} [R_{ij}]_{\ell\ell'}^{mm'} [z_i]_\ell^m = -\ell' \sum_{\substack{i=1\\ i\neq j}}^{M} \sum_{n=1}^{N_g} [\mathcal{P}_{ij}^n]_{\ell'}^{m'} (\boldsymbol{w}_z)_i^n, \tag{12}$$

where

$$(\boldsymbol{w}_z)_i^n := \sum_{\ell m}^{\ell_{\max}} [w_i^n]_\ell^m [z_i]_\ell^m. \tag{13}$$

By introducing further notation for the adjoint potential

$$\forall j, \ell', m' : \quad \overline{\mathcal{P}}^j(\ell', m'; \boldsymbol{w}_z) = \sum_{\substack{i=1\\ i\neq j}}^{M} \sum_{n=1}^{N_g} [\mathcal{P}_{ij}^n]_{\ell'}^{m'} (\boldsymbol{w}_z)_i^n \tag{14}$$

we can simplify the adjoint off-diagonal matrix-vector product as follows:

$$[(R^\top z)_j]_{\ell'}^{m'} = -\ell' \, \overline{\mathcal{P}}^j(\ell', m'; \boldsymbol{w}_z). \tag{15}$$

Thus, the adjoint matrix-vector product can be divided again into 3 following steps:

1. For every i, assemble the vector $(\boldsymbol{w}_z)_i$ by the equation (13). The complexity of this step for all i together is linear in M.

2. Compute the adjoint potentials following (14) for every j, ℓ', m'. Due to the sum over all spheres, each adjoint potential needs $O(M)$ operations. Hence, the accumulated number of operations over all atoms is $O(M^2)$.

3. Scale the result by the corresponding degree ℓ' of the harmonics in accordance to (15). This step scales linearly by the number of atoms M.

3. Operation 3: Contraction of differential matrices

Further, the contraction of $\nabla_k R_\varepsilon$ with any two vectors z and v is given by

$$\begin{aligned} z^\top (\nabla_k R_\varepsilon) v &= \sum_{i=1}^{M} \sum_{\substack{j=1\\ j\neq i}}^{M} \sum_{\ell m}^{\ell_{\max}} \sum_{\ell' m'}^{\ell_{\max}} [z_i]_\ell^m \, [(\nabla_k R_\varepsilon)_{ij}]_{\ell\ell'}^{mm'} \, [v_j]_{\ell'}^{m'} \\ &= \sum_{i=1}^{M} \sum_{\substack{j=1\\ j\neq i}}^{M} \sum_{\ell' m'}^{\ell_{\max}} \sum_{n=1}^{N_g} \nabla_k \left(\underbrace{\left(\sum_{\ell m}^{\ell_{\max}} [z_i]_\ell^m [w_i^n]_\ell^m \right)}_{(\boldsymbol{w}_z)_i^n} [\mathcal{P}_{ij}^n]_{\ell'}^{m'} \right) [(\boldsymbol{v}_\ell)_j]_{\ell'}^{m'}. \end{aligned}$$

Using the product rule for differentiation, we then obtain three contributions

$$z^\top(\nabla_k R_\varepsilon)v = \sum_{i\in N_k}\sum_{n=1}^{N_g} \nabla_k(\boldsymbol{w}_z)_i^n \underbrace{\left(\sum_{\substack{j=1\\ j\neq i}}^{M}\sum_{\ell' m'}^{\ell_{\max}} [\mathcal{P}_{ij}^n]_{\ell'}^{m'}[(\boldsymbol{v}_\ell)_j]_{\ell'}^{m'}\right)}_{\mathcal{P}^i(\mathbf{x}_i+r_i s_n;\boldsymbol{v}_\ell)} \tag{16}$$

$$+\sum_{n=1}^{N_g}(\boldsymbol{w}_z)_k^n \underbrace{\left(\sum_{\substack{j=1\\ j\neq k}}^{M}\sum_{\ell' m'}^{\ell_{\max}} \nabla_k[\mathcal{P}_{kj}^n]_{\ell'}^{m'}[(\boldsymbol{v}_\ell)_j]_{\ell'}^{m'}\right)}_{\nabla_k\mathcal{P}^k(\mathbf{x}_k+r_k s_n;\boldsymbol{v}_\ell)} \tag{17}$$

$$+\sum_{\ell' m'}^{\ell_{\max}}[(\boldsymbol{v}_\ell)_k]_{\ell'}^{m'} \underbrace{\left(\sum_{\substack{i=1\\ i\neq k}}^{M}\sum_{n=1}^{N_g} \nabla_k[\mathcal{P}_{ik}^n]_{\ell'}^{m'}(\boldsymbol{w}_z)_i^n\right)}_{\nabla_k\overline{\mathcal{P}}^k(\ell',m';\boldsymbol{w}_z)} \tag{18}$$

where N_k denotes the set of neighbouring/intersecting spheres of the k-th sphere (excluding itself). Such a set of contractions is computed for every $k = 1,\dots,M$. As one can notice, the quadratic complexity of obtaining all contractions together comes from the potentials $\mathcal{P}^i$, the gradients of the potentials $\nabla_k\mathcal{P}^k$ as well as the gradients of the adjoint potentials $\nabla_k\overline{\mathcal{P}}^k$. All other operations are local on each atom, and we thus have:

1. Compute the potentials $\mathcal{P}^i(\mathbf{x}_i+r_is_n;\boldsymbol{v}_\ell)$ and their gradients $\nabla_i\mathcal{P}^i(\mathbf{x}_i+r_is_n;\boldsymbol{v}_\ell)$ at every external grid point as well as the gradients of the adjoint potentials $\nabla_j\overline{\mathcal{P}}^j(\ell',m';\boldsymbol{w}_z)$ for all modeling harmonics, i.e., for every j,ℓ',m'. The complexity of this step is $O(M^2)$.

2. Compute, for every k, the vectors $(\boldsymbol{v}_\ell)_k$ and $(\boldsymbol{w}_z)_k$ following equations (11) and (13) respectively resulting in $O(M)$ operations for all possible k.

3. For every atom k, loop through the intersecting spheres $i \in N_k$ and contract the acquired potentials at external grid points in the switching area of the i-th sphere with the gradient of the local entity $\nabla_k(\boldsymbol{w}_z)_i^n$ in order to obtain the right hand side of (16). By the nature of matter, the total number of intersecting pairs of spheres is $O(M)$. It makes the computational complexity of the entire step linear in M.

4. For every atom k, contract the acquired gradients of the potentials $\nabla_k\mathcal{P}^k$ with the

vector $(\boldsymbol{w}_z)_k$ and the gradients of adjoint potentials $\nabla_k \overline{\mathcal{P}}^k$ with the vector $(\boldsymbol{v}_\ell)_k$ in order to obtain (17) and (18). For all k together, this step requires $O(M)$ operations.

4. Summary

The quadratically scaling bottlenecks in Operations 1–3 can be formalized as the computation of the following quantities:

1. The computation of the potentials

$$\mathcal{P}^i(\mathbf{x}_i + r_i s_n; \boldsymbol{v}_\ell) \tag{19}$$

   and their gradients

$$\nabla_i \mathcal{P}^i(\mathbf{x}_i + r_i s_n; \boldsymbol{v}_\ell) \tag{20}$$

   at all external grid points.

2. The computation of the adjoint potentials

$$\overline{\mathcal{P}}^i(\ell, m; \boldsymbol{w}_z) \tag{21}$$

   and their gradients

$$\nabla_i \overline{\mathcal{P}}^i(\ell, m; \boldsymbol{w}_z) \tag{22}$$

   for every modeling harmonic function $Y_{\ell m}^i$.

In order to achieve the linear scaling of these operations with respect to the number of atoms M, we will approximate the computations of the above quantities by fast summation techniques. We will use the fast multipole method (FMM) which will be discussed in the next section.

III. IMPLEMENTATION OF THE LINEARLY SCALING DDPCM

A. Overview of the FMM

As the Fast Multipole Method (FMM)[11] is a well-known and well-reported algorithm, we present only its basic overview without any in-depth discussion. First, the original FMM computes the Coulomb potentials for a given set of N point-charges. In three dimensions, it is equivalent to finding the following sums:

$$\forall i = 1, \ldots, N : \qquad p_i = \sum_{\substack{j=1 \\ j \neq i}}^{N} \frac{q_j}{|\mathbf{x}_i - \mathbf{x}_j|}, \tag{23}$$

where $\mathbf{x}_j$ and q_j are a location and a charge of the j-th particle respectively and p_i is the Coulomb potential at a location of the i-th particle generated by all other point-charges. A straightforward evaluation of such potentials requires $O(N^2)$ floating point operations.

In the 3-dimensional case, the FMM uses spherical harmonic expansions to separate the variables of the Coulomb potential $f(x, y) = \frac{1}{|x-y|}$ of a set of source point-charges in a ball $B_{r_y}(y_0)$ of a radius r_y with a center y_0 and a set of distant target locations in a ball $B_{r_x}(x_0)$ of a radius r_x with a center x_0. Such expansions around the source $B_{r_y}(y_0)$ and inside the target $B_{r_x}(x_0)$ areas are called the *multipole* and the *local* expansions, respectively. For a pair of a source and a target balls that are geometrically well-separated, i.e., far from each other, all the corresponding potentials from the sources at the target locations are called the *far-field* potentials/interactions. The *far-field* interactions can be approximated by truncating the *multipole* and the *local* expansions at controllable loss of accuracy. The number of terms of the expansion does not depend neither on a number of the source point-charges nor on a number of the target locations but on the separation distance of the two balls. On the other hand, if the source and the target balls are close enough to each other, or even intersect, then such a separation of the variables is impossible. In this case, all the potentials are called the *near-field* potentials/interactions and must be evaluated directly. The arithmetic complexity of the *far-field* interactions depends on the sum of the number of sources and the number of targets, and the number of operations required to calculate the *near-field* interactions is the product of the number of sources and targets. The FMM partitions all the interaction-potentials into either *far-field* or *near-field* interactions. The sets of sources and targets are decided by a partitioning of the space and, furthermore, a hierarchical partitioning (a cluster tree) is employed to improve the efficiency: when the targets and sources are farther apart, a coarser clustering is used which reduces the number of potentials to be evaluated, while retaining the same accuracy. In this way, an approximated result is acquired in $O(N)$ operations. Generally speaking, the fewer potentials are classified as *near-field* interactions, the lower the arithmetic complexity is. The following Algorithm 3 summarizes schematically all the major points of the FMM.

Algorithm 3 The FMM

```
Require: $N$ particles with locations $\{\mathbf{x}_i\}_{i=1}^N$ and charges $\{q_i\}_{i=1}^N$.
Ensure: Potentials $\{p_i\}_{i=1}^N$ at all locations $\{\mathbf{x}_i\}_{i=1}^N$.
Build a hierarchical cluster tree.
Particles-to-multipole (P2M) step: compute the multipole expansion coefficients for each leaf node of the cluster tree.
Multipole-to-multipole (M2M) step: obtain the multipole expansions coefficients hierarchically in a bottom-to-top pass through the cluster tree.
Multipole-to-local (M2L) step: transfer the multipole expansions coefficients to the local expansions coefficients for the *far-field* interactions.
Local-to-local (L2L) step: pass the local expansions coefficients to the node leaves of the cluster tree in a top-to-bottom manner.
Local-to-particles (L2P) step: translate the local expansion of each leaf node into the potential of the corresponding particles.
Particle-to-particle (P2P) step: evaluate potentials from all the *near-field* interactions.
```

B. Adaptive tree-based FMM for ddPCM

The main distinguishing features of our implementation with respect to the classical 3-dimensional FMM are summarized in TABLE I. In the following subsections we emphasize details on how we construct a cluster tree, partition it into the far-field and the near-field interactions and implement the M2M, M2L and L2L operations. We would also like to stress out, that both the ddPCM and the FMM rely on spherical harmonics, and thus on a truncation parameter. However, these harmonics serve different purposes, and therefore we distinguish them as $\ell_{\max}$ for the maximal degree of the discretization to represent the unknowns within ddPCM, and $p_{\max}$ for the maximal degree used in the FMM.

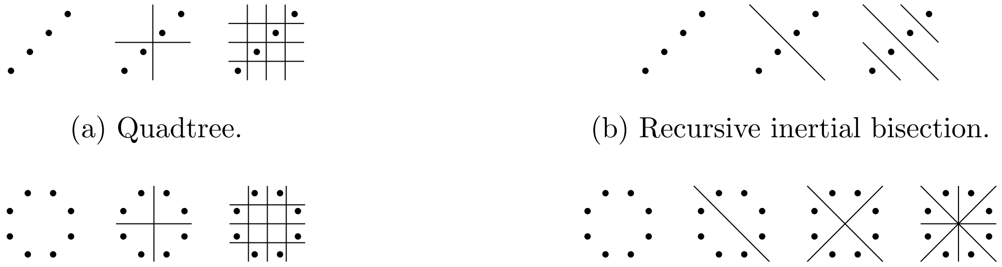
	Proposed approach	Original FMM
sources	distribution on spheres	point-charges
targets	points on input spheres	input points
cluster tree	binary, adaptive	octree, uniform
radii of harmonics	variable	unit

TABLE I: Main differences between the proposed approach and the original FMM.

1. *Recursive inertial bisection*

Since the spheres building the cavity of the solute's molecule are arranged according to a non-uniform distribution, we are motived to use an adaptive cluster tree, in a recursively bisected form, as an alternative to the standard octree. A uniform clusterization has an obvious advantage — namely that M2M, M2L and L2L steps are identical among all nodes of a single level of a cluster tree. For a non-uniform tree, all M2M, M2L and L2L translations have different transformation matrices which need to be computed at each node separately. On the other side, a non-uniform tree better adapts to the non-uniform distribution of the spheres (and its integration points) and thus theoretically leads to a smaller amount of operations. Therefore, it is a fine weighing of advantages and disadvantages that justifies our choice.

The inertial bisection is a naive approach of dividing a set of point-masses by a hyperplane into two subgeometries by minimizing the moment of inertia. This approach cuts perpendicularly to the principal axis through the center of mass and thus interprets a non-uniform distribution of points better than any coordinate-based subdivision method. Applying the bisection further to the subgeometries in a recursive manner defines the *recursive inertial bisection*. A simple comparison of the *recursive inertial bisection* with a quadtree, which is standard for the FMM on a plane, is schematically presented in the FIG. 1. As is illustrated, the quadtree for a set of equally weighted points aligned on the diagonal or a circle has many empty nodes/boxes while the *recursive inertial bisection* reflects the one-dimensional structure. To construct the cluster tree, we assume all the atom centers $\{\mathbf{x}_i\}_{i=1}^M$ have a unit mass and apply the *recursive inertial bisection* to this set until each leaf node contains only a single point. Our approach is formalized in Algorithm 5 of Appendix A.



(a) Quadtree.

(b) Recursive inertial bisection.

(c) Quadtree.

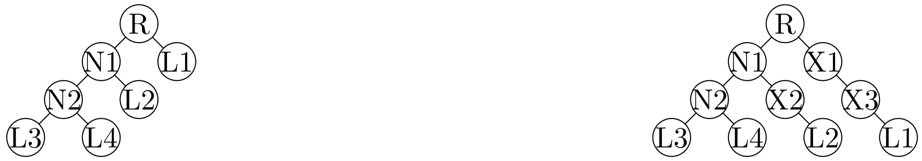
(d) Recursive inertial bisection.

FIG. 1: Comparison of a quadtree and a recursive inertial bisection for a line and a circle of equally-weighted particles.

Once the cluster tree is constructed, we have to define the domains of convergence of the multipole and the local expansions of each node. Like in the FMM, we use bounding spheres for this purpose: every leaf node contains only a single input atom by the construction and thus the bounding sphere of every leaf node is simply the corresponding input sphere itself, while the bounding sphere of a non-leaf node is the sphere of a minimal radius that contains both bounding spheres of its two children nodes. This construction method is formalized in Algorithm 6 of Appendix A.

When all the bounding spheres are constructed, the partitioning into the far-field and the near-field interactions is done recursively. This process requires one last item — an *admissibility condition.* For our numerical tests, we used the following condition: two bounding spheres are admissibly far, i.e., geometrically well-separated, if the distance between them is not smaller than the maximum of their radii. The partitioning is done with the help of a list of test pairs, which is initialized by the pair $\{(\mathrm{R}, \mathrm{R})\}$ with R denoting the root node of the tree. The result is given as two lists: a list of *admissibly far pairs* of nodes and a list of *admissibly near pairs* of nodes. While the test list is non-empty, each pair is checked for the admissibility condition. If it is satisfied, then the pair is moved to the list of all *admissibly far pairs.* If it is not satisfied and both nodes are non-leaf nodes, then the test list is expanded by all possible combinations of pairs of children nodes. Otherwise, the pair is moved to the list of all *admissible near pairs.* These steps are formalized in Algorithm 7 of Appendix A.

The recursive inertial bisection leads to a possibly unbalanced tree, where some leaf nodes do not belong to the bottom level of the tree. Such an example is presented in the FIG. 2a,



(a) Unbalanced tree

(b) Balanced tree

FIG. 2: Balancing a cluster tree by adding mock nodes X1, X2 and X3. For the balanced tree, all the leaf nodes are on the bottom level, which implies a smaller amount of the near-field interactions and thus lowers the total complexity of the FMM matrix-vector product.

where N1 and N2 denote non-leaf nodes and L1, L2, L3 and L4 denote leaf nodes. In this case, the disadvantage is the increased amount of the near-field interactions. However, it is possible to count some of them as the far-field interactions simply by adding mock nodes with the help of Algorithm 8 presented in Appendix A. An example of the balanced tree is presented in the FIG. 2b, where mock nodes X1, X2 and X3 are added to the previously unbalanced tree.

The entire procedure of building the cluster tree with partitioning into the far-field and the near-field interactions is presented in Algorithm 4.

Algorithm 4 Construction and partitioning of a cluster tree.

```
Require: Set of $M$ spheres with centers $\{\mathbf{x}_j\}_{j=1}^M$ and radii $\{r_j\}_{j=1}^M$
Ensure: Cluster tree $\mathcal{T}$ with lists of near $\mathcal{N}$ and far $\mathcal{F}$ admissible pairs.
Construct the binary cluster tree $\mathcal{T}$ by the Algorithm 5
Compute bounding spheres of all nodes of $\mathcal{T}$ by the Algorithm 6
Add mock nodes by the Algorithm 8
Generate lists of far $\mathcal{F}$ and near $\mathcal{N}$ admissible pairs by the Algorithm 7
```

2. Scaled Solid Harmonics and representation of the potential

The original FMM works with solid harmonics with a provided convergence radius that defines an area of validity of the expansion. The potential of a multipole expansion caused by real normalized spherical harmonics of maximal degree $\ell_{\max}$, with coefficients $[\widehat{\mathsf{M}}_j]_\ell^m$, a center $\mathbf{x}_j$ and a convergence radius r_j outside a ball $B_{r_j}(\mathbf{x}_j)$ is given by:

$$\forall x \notin B_{r_j}(\mathbf{x}_j): \quad \mathcal{P}(x) = \sum_{\ell m}^{\ell_{\max}} \frac{4\pi}{2\ell+1} \frac{[\widehat{\mathsf{M}}_j]_\ell^m}{|x - \mathbf{x}_j|^{\ell+1}} Y_{\ell m}^j(x). \tag{24}$$

The potential of a corresponding local expansion caused by harmonics of degree up to $p_{\max}$, with coefficients $[\widehat{\mathsf{L}}_j]_\ell^m$ inside the ball $B_{r_j}(\mathbf{x}_j)$ is defined as follows:

$$\forall x \in B_{r_j}(\mathbf{x}_j): \quad \mathcal{L}(x) = \sum_{\ell m}^{p_{\max}} \frac{4\pi}{2\ell+1} [\widehat{\mathsf{L}}_j]_\ell^m |x - \mathbf{x}_j|^\ell Y_{\ell m}^j(x). \tag{25}$$

Note that we used the maximal degree $\ell_{\max}$ for the multipole expansion as it is connected with the modeling harmonics (discretization) and the maximal degree $p_{\max}$ for the local expansion as it is related to the potential that is approximated by the FMM. These equations, combined with the finite machine precision, cause a series of overflow and underflow in the evaluation of the expressions $|x - \mathbf{x}_j|^{-\ell-1}$ and $|x - \mathbf{x}_j|^\ell$, which actually limits the maximum allowed degree of spherical harmonics. To overcome the problem, we rewrite equation (24) with a relative distance to the ball $B_{r_j}(\mathbf{x}_j)$ as follows:

$$\forall x \notin B_{r_j}(\mathbf{x}_j): \quad \mathcal{P}(x) = \sum_{\ell m}^{\ell_{\max}} [\mathsf{M}_j]_\ell^m \, \mathcal{P}_{\ell m}^j(x), \qquad [\mathsf{M}_j]_\ell^m = \frac{[\widehat{\mathsf{M}}_j]_\ell^m}{r_j^{\ell+1}}, \tag{26}$$

where potential $\mathcal{P}_{\ell m}^j(x)$ was previously defined in the equation (3). We apply the same technique to the local expansion of the equation (25):

$$\forall x \in B_{r_j}(\mathbf{x}_j): \quad \mathcal{L}(x) = \sum_{\ell m}^{p_{\max}} [\mathsf{L}_j]_\ell^m \, \mathcal{L}_{\ell m}^j(x), \qquad [\mathsf{L}_j]_\ell^m = [\widehat{\mathsf{L}}_j]_\ell^m r_j^\ell, \tag{27}$$

where

$$\forall x \in \Omega_j: \quad \mathcal{L}_{\ell m}^j(x) := \mathcal{L}_{\ell m}(x, \mathbf{x}_j; r_j) = \frac{4\pi}{2\ell+1} r_j^\ell(x) \, Y_{\ell m}^j(x), \tag{28}$$

and

$$\forall x \in B_r(y): \quad \mathcal{L}_{\ell m}(x, y; r) := \frac{4\pi}{2\ell+1} \left(\frac{|x-y|}{r} \right)^\ell Y_\ell^m \left(\frac{x-y}{|x-y|} \right). \tag{29}$$

Hence, the representation of the unit (equations (24) and (25)) and the scaled (equations (26) and (27)) solid harmonics of the same degree differ only by a scalar factor for both, for the multipole and the local expansions. This implies that we can use the original FMM by applying proper scaling of the solid harmonics before and after each translation of the expansions.

3. FMM-operators

Let us now formally define the M2M translation. First of all, this is a linear operator with bounded maximal degree of harmonics, so it can be represented simply by a matrix. Therefore, we denote the M2M translation from the source spherical harmonics of a radius R_S located at the point x to the target spherical harmonics of a radius R_T at the origin as $\mathrm{M2M}(x, R_S, R_T)$. Such an explicit representation is used in the next section, where we compute its gradient. To be more specific, a translation of the spherical harmonic Y_ℓ^m of a unit weight from the source sphere into the weight of the spherical harmonic $Y_{\ell'}^{m'}$ on the target sphere is denoted as $[\mathrm{M2M}(x, R_S, R_T)]_{\ell'\ell}^{m'm}$. Generally, the value $[\mathrm{M2M}(x, R_S, R_T)]_{\ell'\ell}^{m'm}$ is non-zero for all possible indices ℓ, m, ℓ' and m', which leads to $O(p_{\max}^4)$ operations for each M2M translation.

As is well known, there are different techniques to reduce the complexity of the FMM translations. In this paper, we focus on a rotation-based approach that first rotates the system of coordinates to align the centers of the source and target spherical harmonics along the OZ axis, then applies a corresponding translation along the the OZ direction and finally finishes by applying the backward rotation. Such an approach is formalized as a product of three matrices of identical size $(p_{\max} + 1)^2 \times (p_{\max} + 1)^2$:

$$\mathrm{M2M}(x, R_S, R_T) = Q(x, p_{\max})^\top \cdot \mathrm{M2M}(\|x\| e_z, R_S, R_T) \cdot Q(x, p_{\max}), \tag{30}$$

where $Q(x, p_{\max})$ denotes the rotation matrix of coefficients of harmonics of a degree up to $p_{\max}$, corresponding to the rotation of the vector x to the vector $\|x\| e_z = [0, 0, \|x\|]^\top$. A rotated spherical harmonic Y_ℓ^m is a linear combination of harmonics $Y_{\ell'}^{m'}$ with the same degree $\ell' = \ell$ and different orders $m' = -\ell, \dots, \ell$. Hence, the matrix $Q(x, p_{\max})$ is a block diagonal matrix with $O(p_{\max}^3)$ non-zero elements. The M2M translation is easy to adapt to the case of the M2L translation:

$$\mathrm{M2L}(x, R_S, R_T) = Q(x, p_{max})^\top \cdot \mathrm{M2L}(\|x\| e_z, R_S, R_T) \cdot Q(x, p_{\max}), \tag{31}$$

and to the case of the L2L translation:

$$\mathrm{L2L}(x, R_S, R_T) = Q(x, p_{\max})^\top \cdot \mathrm{L2L}(\|x\| e_z, R_S, R_T) \cdot Q(x, p_{\max}). \tag{32}$$

Our implementation of $Q(x, p_{\max})$ consists even of 2 elementary rotations: first, we rotate around the OZ axis and then rotate around the OY axis to align the centers along the OZ axis. The first rotation requires only $O(p_{\max}^2)$ operations as each pair of harmonics Y_ℓ^m and Y_ℓ^{-m} is transformed into itself. The second rotation, however, consists of $O(p_{\max}^3)$ floating point operations: for any given degree ℓ, the set of harmonics Y_ℓ^m with a non-negative order $m \geq 0$ and a set of harmonics Y_ℓ^m with a negative order $m < 0$ are transformed into themselves. Transformations of the second rotation are computed in a recursive manner, as it is proposed in 14 and 15. The backward rotation is the adjoint of the forward rotation and is therefore performed as the OY rotation followed by the OZ rotation, both in backward directions.

Following the theorems for the translation operations described in 1, we derive the OZ translation of the M2M step over a distance ρ as a matrix-vector product. Considering a scaled spherical harmonics of radius R_S as the source, and a scaled spherical harmonics of radius R_T as the target, the matrix $\mathrm{M2M}(\rho e_z, R_S, R_T)$ reads:

$$[\mathrm{M2M}(\rho e_z, R_S, R_T)]_{\ell'\ell}^{m'm} = \begin{cases} 0, & m \neq m' \text{ or } \ell > \ell', \\ \frac{R_S^{\ell+1}\rho^{\ell'-\ell}}{R_T^{\ell'+1}} \frac{A_{\ell'-\ell}^0 A_\ell^m}{A_{\ell'}^m} \sqrt{\frac{2\ell'+1}{2\ell+1}}, & \text{otherwise}, \end{cases} \tag{33}$$

where the multiindices (ℓ, m) and (ℓ', m') are the source and the target harmonic indices respectively and A_ℓ^m is defined as

$$A_\ell^m = \frac{1}{\sqrt{(\ell-m)!(\ell+m)!}}. \tag{34}$$

In the same manner, we derive the matrix of the OZ translation of the M2L step as

$$[\mathrm{M2L}(\rho e_z, R_S, R_T)]_{\ell'\ell}^{m'm} = \begin{cases} 0, & m \neq m', \\ (-1)^{\ell+m} \frac{R_S^{\ell+1} R_T^{\ell'}}{\rho^{\ell'+\ell+1}} \frac{A_\ell^m A_{\ell'}^m}{A_{\ell'+\ell}^0} \sqrt{\frac{2\ell'+1}{2\ell+1}}, & \text{otherwise}. \end{cases} \tag{35}$$

And, finally, the L2L step operation along the OZ axis is given by

$$[\mathrm{L2L}(\rho e_z, R_S, R_T)]_{\ell'\ell}^{m'm} = \begin{cases} 0, & m \neq m' \text{ or } \ell < \ell', \\ (-1)^{\ell+\ell'} \frac{R_T^{\ell'}\rho^{\ell-\ell'}}{R_S^{\ell}} \frac{A_{\ell-\ell'}^0 A_{\ell'}^m}{A_\ell^m} \sqrt{\frac{2\ell'+1}{2\ell+1}}, & \text{otherwise}. \end{cases} \tag{36}$$

All of the above provided OZ translation matrices are sparse with $O(p_{\max}^3)$ non-zero elements.

Since the OZ translation of the M2M step is non-zero only for $m = m'$, a single element of a general M2M matrix can be written as follows:

$$[\mathrm{M2M}(x, R_S, R_T)]_{\ell'\ell}^{m'm} = \sum_{k=-\ell}^{\ell} [Q(x,\ell')^\top]_{\ell'\ell'}^{m'k} [Q(x,\ell)]_{\ell\ell}^{km} \tag{37}$$

$$\times [\mathrm{M2M}(\|x\| e_z, R_S, R_T)]_{\ell'\ell}^{kk}. \tag{38}$$

Since it is clear how to generate the corresponding formulae for the M2L and the L2L operators, we omit such representations.

4. Gradients of the multipole and the local expansions

Gradients of the multipole expansions are essential if forces are to be computed which require the comptuation of (20) and (22). It is obvious that the gradient of the multipole expansion with respect to the source of the expansion is the negative gradient of the same expansion with respect to the target location due to the symmetric nature of the Coulomb potential. One of the possible ways to compute such gradients is just to differentiate the potentials based on the explicit formula. However, we propose to use a different approach which is explained in detail in Appendix B. Indeed, the derivative of a general potential of the form $\sum_{\ell m}^{\ell_{\max}} [\mathsf{M}]_\ell^m \mathcal{P}_{\ell m}(t, s; r)$ with respect to s can be expressed as

$$\nabla_s \sum_{\ell m}^{\ell_{\max}} [\mathsf{M}]_\ell^m \mathcal{P}_{\ell m}(t, s; r) = \frac{1}{r} \sum_{\ell m}^{\ell_{\max}+1} [\mathsf{d}\mathcal{P}\,\mathsf{M}]_\ell^m \mathcal{P}_{\ell m}(t, s; r), \tag{39}$$

where $\mathsf{d}\mathcal{P}\,\mathsf{M} \in \mathbb{R}^{3\times(\ell_{\max}+2)^2}$ represents the concatenation of three matrix-vector products with the same vector $\mathsf{M} \in \mathbb{R}^{(\ell_{\max}+1)^2}$ and whose definition is given in Appendix B.

In the same manner, the derivative of a general local expansion of the form $\sum_{\ell m}^{p_{\max}} [\mathsf{L}]_\ell^m \mathcal{L}_{\ell m}(t, s; r)$ with respect to s can be expressed as

$$\nabla_s \sum_{\ell m}^{p_{\max}} [\mathsf{L}]_\ell^m \mathcal{L}_{\ell m}(t, s; r) = \frac{1}{r} \sum_{\ell m}^{p_{\max}-1} [\mathsf{d}\mathcal{L}\,\mathsf{L}]_\ell^m \mathcal{L}_{\ell m}(t, s; r). \tag{40}$$

Again, $\mathsf{d}\mathcal{L}\,\mathsf{L} \in \mathbb{R}^{3\times(p_{\max})^2}$ represents the the concatenation of three matrix-vector products with the same vector $\mathsf{L} \in \mathbb{R}^{(p_{\max}+1)^2}$ and whose details are given in Appendix B.

C. Eliminating the quadratically scaling bottlenecks

The FMM allows to compute not only potentials at any given target point, but the entire potential field around them, which is represented by the sum of the near-field term and the far-field term. For the sake of simple notation of the far-field term, let us denote the successive application of tree-wise M2M, M2L and L2L operators that approximate the far-field term as a single combined operator:

$$\mathrm{L}_\mathrm{M} = \mathrm{L2L} \circ \mathrm{M2L} \circ \mathrm{M2M}, \tag{41}$$

which transforms an input vector of coefficients of spherical harmonics of all input atoms of a degree up to $\ell_{\max}$ into an output vector of weights of spherical harmonics of all input atoms of a degree up to $p_{\max}$:

$$\mathrm{L_M}: \; M \times (\ell_{\max}+1)^2 \longrightarrow M \times (p_{\max}+1)^2. \tag{42}$$

The input coefficients correspond to the multipole expansions while the output coefficients are related to the local expansions. The adjoint operator $\mathrm{L}_\mathrm{M}^\top$ applies the adjoint M2M, adjoint M2L and adjoint L2L in reversed order:

$$\mathrm{L}_\mathrm{M}^\top = \mathrm{M2M}^\top \circ \mathrm{M2L}^\top \circ \mathrm{L2L}^\top, \tag{43}$$

with the input and the output of the following sizes:

$$\mathrm{L}_\mathrm{M}^\top : \; M \times (p_{\max}+1)^2 \longrightarrow M \times (\ell_{\max}+1)^2. \tag{44}$$

Due to the linear nature of the M2M, M2L and L2L operators that consist of a chain of matrix-vector products, both the primal $\mathrm{L_M}$ and the adjoint $\mathrm{L}_\mathrm{M}^\top$ are linear operators.

Following the notation of Algorithm 7, presented in Appendix A, we introduce $\mathcal{N}_i$ as the list of all input spheres that form the near admissible pair with the i-th input sphere excluding the i-th sphere itself. The value $|\mathcal{N}_i|$ represents the number of elements of the list $\mathcal{N}_i$.

1. The potentials

The potential $\mathcal{P}^i(x; \boldsymbol{v}_\ell)$ in the vicinity of the i-th ball at every required external grid point $\mathbf{x}_i + r_i s_n$ is approximated as the sum of the near-field and the far-field potentials:

$$\forall i, n: \qquad \mathcal{P}^i(\mathbf{x}_i + r_i s_n; \boldsymbol{v}_\ell) = \sum_{\substack{j=1 \\ j\neq i}}^{M} \sum_{\ell m}^{\ell_{\max}} [(\boldsymbol{v}_\ell)_j]_\ell^m \mathcal{P}_{\ell m}^j(\mathbf{x}_i + r_i s_n) \tag{45}$$

$$\approx \sum_{j \in \mathcal{N}_i} \sum_{\ell m}^{\ell_{\max}} [(\boldsymbol{v}_\ell)_j]_\ell^m \mathcal{P}_{\ell m}^j(\mathbf{x}_i + r_i s_n) \tag{46}$$

$$+ \sum_{\ell m}^{p_{\max}} [(\mathrm{L_M}(\boldsymbol{v}_\ell))_i]_\ell^m \mathcal{L}_{\ell m}^i(\mathbf{x}_i + r_i s_n). \tag{47}$$

It is worth noting that the term

$$\mathcal{L}_{\ell m}^i(\mathbf{x}_i + r_i s_n) = \frac{4\pi}{2\ell+1} r_i^\ell(\mathbf{x}_i + r_i s_n) Y_{\ell m}^i(\mathbf{x}_i + r_i s_n) = \frac{4\pi}{2\ell+1} Y_\ell^m(s_n)$$

in the far-field potential does not depend on the index i. In turn, we denote it as follows:

$$[\mathcal{L}^n]_\ell^m = \mathcal{L}^i_{\ell m}(\mathbf{x}_i + r_i s_n), \tag{48}$$

and rewrite the approximation of the potential as:

$$\mathcal{P}^i(\mathbf{x}_i + r_i s_n; \boldsymbol{v}_\ell) \approx \sum_{j\in\mathcal{N}_i} \sum_{\ell m}^{\ell_{\max}} [(\boldsymbol{v}_\ell)_j]_\ell^m [\mathcal{P}^n_{ij}]_\ell^m \tag{49}$$

$$+ \sum_{\ell m}^{p_{\max}} [(\mathrm{L_M}\boldsymbol{v}_\ell)_i]_\ell^m [\mathcal{L}^n]_\ell^m. \tag{50}$$

Finally, to calculate potentials at all the external grid points we need to:

A1. Compute $\mathrm{L_M}\boldsymbol{v}_\ell$ by the FMM in $O(Mp^3_{\max})$ operations,

A2. Account all the near-field potentials for every atom (49) in $O(|\mathcal{N}_i|N_g\ell^2_{\max})$ operations per i-th sphere,

A3. Add up approximation of the far-field potentials (50) in $O(N_g p^2_{\max})$ operations for each atom.

The total arithmetic complexity of the potentials scales linearly with the number of atoms M of the molecule.

2. The gradients of the potentials

Let us remind, that we also require the vector of the gradients of all the potentials $\mathcal{P}^i(x, \boldsymbol{v}_\ell)$ at appropriate external grid points:

$$\forall i, n: \qquad \nabla_{\mathbf{x}_i}\mathcal{P}^i(\mathbf{x}_i + r_i s_n; \boldsymbol{v}_\ell) = \sum_{\substack{j=1\\ j\neq i}}^{M} \sum_{\ell m}^{\ell_{\max}} [(\boldsymbol{v}_\ell)_j]_\ell^m \nabla\mathcal{P}^j_{\ell m}(\mathbf{x}_i + r_i s_n) \tag{51}$$

$$\approx \sum_{j\in\mathcal{N}_i} \sum_{\ell m}^{\ell_{\max}} [(\boldsymbol{v}_\ell)_j]_\ell^m \nabla\mathcal{P}^j_{\ell m}(\mathbf{x}_i + r_i s_n) \tag{52}$$

$$+ \sum_{\ell m}^{p_{\max}} [(\mathrm{L_M}\boldsymbol{v}_\ell)_i]_\ell^m \nabla\mathcal{L}^i_{\ell m}(\mathbf{x}_i + r_i s_n). \tag{53}$$

A way of computing gradients

$$\nabla\mathcal{P}^j_{\ell m}(t) = \nabla_t\mathcal{P}_{\ell m}(t, \mathbf{x}_j; r_j) = -\nabla_{\mathbf{x}_j}\mathcal{P}_{\ell m}(t, \mathbf{x}_j; r_j) \tag{54}$$

and

$$\nabla\mathcal{L}^i_{\ell m}(t) = \nabla_t \mathcal{L}_{\ell m}(t, \mathbf{x}_i; r_i) = -\nabla_{\mathbf{x}_i}\mathcal{L}_{\ell m}(t, \mathbf{x}_i; r_i) \tag{55}$$

for all possible pairs (ℓ, m) was already discussed in the previous Section III B 4. Hence we utilize the operators $\mathsf{d}\mathcal{P}$ and $\mathsf{d}\mathcal{L}$ to obtain the gradients in a fast way:

$$\forall i, n: \qquad \nabla_{\mathbf{x}_i}\mathcal{P}^i(\mathbf{x}_i + r_i s_n; \boldsymbol{v}_\ell) \approx -\sum_{j\in\mathcal{N}_i}\frac{1}{r_j}\sum_{\ell m}^{\ell_{\max}+1}[\mathsf{d}\mathcal{P}\left((\boldsymbol{v}_\ell)_j\right)]^m_\ell [\mathcal{P}^n_{ij}]^m_\ell \tag{56}$$

$$-\frac{1}{r_i}\sum_{\ell m}^{p_{\max}-1}[\mathsf{d}\mathcal{L}\left((\mathrm{L_M}\boldsymbol{v}_\ell)_i\right)]^m_\ell\,[\mathcal{L}^n_i]^m_\ell\,. \tag{57}$$

Note that the matrix-vector products $\mathsf{d}\mathcal{P}\left((\boldsymbol{v}_\ell)_j\right)$ and $\mathsf{d}\mathcal{L}\left((\mathrm{L_M}\boldsymbol{v}_\ell)_i\right)$ are defined in Appendix B by (79) and (87) respectively. Assuming that the vector $\mathrm{L_M}\boldsymbol{v}_\ell$ is already computed in $O(Mp^3_{max})$ operations, the following steps are needed to obtain all necessary gradients for each atom:

B1. Compute $\mathsf{d}\mathcal{P}((\boldsymbol{v}_\ell)_j)$ in $O(\ell^3_{\max})$ operations,

B2. Account for the near-field interactions (56) in $O(|\mathcal{N}_i|N_g\ell^2_{\max})$ operations,

B3. Acquire $\mathsf{d}\mathcal{L}(\mathrm{L_M}\boldsymbol{v}_\ell)_i$ in $O(p^3_{\max})$ operations,

B4. Approximate the gradient of the far-field interactions by the equation (57) in $O(N_g p^2_{\max})$ operations.

Due to locality of the above steps, total complexity of computing the gradients of the potentials for all spheres together is again linear in the number of atoms M.

3. *The adjoint potentials*

The adjoint potential $\overline{\mathcal{P}}^j$ takes values of potentials at external grid points and translates them into spherical harmonics:

$$\overline{\mathcal{P}}^j(\ell', m'; \boldsymbol{w}_z) = \sum_{\substack{i=1\\ i\neq j}}^{M}\sum_{n=1}^{N_g}\mathcal{P}^j_{\ell' m'}(\mathbf{x}_i + r_i s_n)(\boldsymbol{w}_z)^n_i, \tag{58}$$

where the vector $(\boldsymbol{w}_z)^n_i$ was previously defined in equation (13). Due to linearity of the primal potential $\mathcal{P}^i(\mathbf{x}_i + r_i s_n; \boldsymbol{v}_\ell)$, its adjoint can be computed by applying its suboperations adjointly in the reversed order. While it is easy to acquire adjoints of the near-field potentials, the far-field potentials require certain additional operations. The computation of the primal far-field potentials requires two steps: (A1.) the evaluation of $\mathrm{L}_\mathrm{M}\boldsymbol{v}_\ell$ with the linear operator L_M and (A3.) the matrix-matrix product by the matrix $\boldsymbol{\mathcal{L}}$. Thus, the adjoint far-field potentials can be accounted for also in two steps: the adjoint matrix-matrix product by the matrix $\boldsymbol{\mathcal{L}}$ followed by applying the adjoint operator $\mathrm{L}_\mathrm{M}^\top$. The first operation (adjoint of A3.) can be shortly represented by introducing the following auxiliary matrix $\widehat{\boldsymbol{w}_z}$:

$$\forall i : \qquad [(\widehat{\boldsymbol{w}_z})_i]_\ell^m = \sum_{n=1}^{N_g} (\boldsymbol{w}_z)_i^n [\mathcal{L}^n]_\ell^m, \tag{59}$$

such that the entire adjoint potential becomes

$$\overline{\mathcal{P}}^j(\ell', m'; \boldsymbol{w}_z) \approx \sum_{i \in \mathcal{N}_j} \sum_{n=1}^{N_g} [\mathcal{P}_{ij}^n]_{\ell'}^{m'} (\boldsymbol{w}_z)_i^n + \left[\left(\mathrm{L}_\mathrm{M}^\top \widehat{\boldsymbol{w}_z}\right)_j\right]_{\ell'}^{m'}. \tag{60}$$

Hence we summarize the cost of computing the adjoint potentials:

C1. Precompute the adjoint matrix-matrix product (59) in $O(MN_g p_{\max}^2)$ operations,

C2. Calculate the near-field term of the adjoint potentials (60) in $O(|\mathcal{N}_j| N_g \ell_{\max}^2)$ operations per atom,

C3. Execute the adjoint far-field operator $\mathrm{L}_\mathrm{M}^\top$ in $O(Mp_{\max}^3)$ operations to take into account the far-field term of the adjoint potentials (60).

Steps C1. and C3. are shared among all the atoms and, hence, are processed only once, while the step number C2. is local to each sphere. Therefore the computation of the adjoint potentials $\overline{\mathcal{P}}^j(\ell', m'; \boldsymbol{w}_z)$ for all possible values of j, ℓ' and m' scales linearly with the size M of the molecule.

4. The gradients of the adjoint potentials

The gradient over the center of the j-th atom of the adjoint potential is as follows:

$$\nabla_{\mathbf{x}_j} \overline{\mathcal{P}}^j(\ell', m'; \boldsymbol{w}_z) = \sum_{\substack{i=1 \\ i \neq j}}^{M} \sum_{n=1}^{N_g} \nabla_{\mathbf{x}_j} \mathcal{P}_{\ell' m'}(\mathbf{x}_i + r_i s_n, \mathbf{x}_j; r_j) (\boldsymbol{w}_z)_i^n. \tag{61}$$

By contracting it with the vector $(\boldsymbol{v}_\ell)_j$ and taking into account equation (40), we get:

$$\sum_{\ell' m'}^{\ell_{\max}} [(\boldsymbol{v}_\ell)_j]_{\ell'}^{m'} \nabla_{\mathbf{x}_j} \overline{\mathcal{P}}^j(\ell', m'; \boldsymbol{w}_z) = \frac{1}{r_j} \sum_{\ell' m'}^{\ell_{\max}+1} [\mathsf{d}\mathcal{P}\left((\boldsymbol{v}_\ell)_j\right)]_{\ell'}^{m'} \overline{\mathcal{P}}^j(\ell', m'; \boldsymbol{w}_z). \tag{62}$$

This simply implies that to compute the entire term of the analytical forces, presented in the equation (18), we just need to contract a vector of the adjoint potentials, which is already precomputed, with the gradient of the corresponding multipole coefficients. Therefore, the arithmetical complexity of obtaining the gradients of the adjoint potentials is also linear in the size of the molecule.

IV. NUMERICAL EXPERIMENTS

In order to demonstrate that our implementation of the FMM-accelerated ddPCM is accurate and linearly scaling, we present benchmark results in this section. More precisely, we report the relative errors of the solvation energies and the forces, the scaling of a single matrix-vector product of the ddPCM and the ddCOSMO linear systems as well as the scaling of the entire algorithm for computing the energies and the forces with respect to the number of atoms.

The ddCOSMO and the ddPCM linear systems are solved iteratively using a Jacobi/DIIS approach both for the FMM-accelerated and the baseline implementations with a relative stopping criterion based on the tolerance $\tau = 10^{-8}$. All molecular structures, details and parameters used for the numerical tests are presented in Appendix C for the purpose of reproducibility. For each computation, we present the relative errors of the energy and the forces given by $\frac{|\widehat{E}-E|}{E}$ and $\frac{\|\widehat{F}-F\|}{\|F\|}$, where E and F denote the reference energy and forces while $\widehat{E}$ and $\widehat{F}$ denote the approximated values and $\|\cdot\|$ stands for the Frobenius norm. The meaning of the reference values are explicitly noted for each plot.

All numerical tests are based on a scaled van der Waals (vdW) cavity of the solute molecules, in which each vdW radius taken from ref.[2] is scaled by 1.1. We also set the dielectric permittivity of the environment to 2.0, a value which corresponds to an apolar solvent. The choice is motivated by the fact that for low values of ε the COSMO approximation is not accurate anymore, and to get the correct solvation energy one has to use PCM.

To begin with, we compare our FMM-based approach with the FMM harmonics of a degree as large as $p_{\max} = 20$ against the non-accelerated (quadratically scaling) ddPCM implementation on small molecular structures in order to verify the correctness of the implementation. The corresponding relative errors of the solvation energy and the forces for the modeling harmonics $\ell_{\max}$ equal to 6, 8, 10 and 12 are presented in FIG. 3 on the left panel. The right panel of the same FIG. 3 represents the speedup of the FMM-accelerated code with the same value $p_{\max} = 20$, which is measured as a ratio of the total computational time in seconds of the FMM-accelerated code against the baseline non-accelerated implementation. We observe that the relative error in energy and forces is at most 10^{-7}. We therefore deduce the correctness of our FMM-accelerated implementation.

Since the initial ddPCM scales quadratically, it becomes exhaustively time-consuming to perform any tests for molecules with more than 1000 atoms. Hence, we use our FMM-accelerated approach with the large value of $p_{\max} = 20$ as a baseline method that provides reference energies and forces for larger molecules as presented in the FIG. 4, and benchmark it against computations using a lower $p_{\max}$. This figure shows that the accuracy of the solvation energies and forces can be systematically controlled by increasing the FMM maximal degree $p_{\max}$. Even more, the relative errors decay exponentially with the growth of the parameter $p_{\max}$. The presented errors of the energies are much lower compared to errors of the forces and the reason can be explained as follows: we only approximate the off-diagonal part of the solution matrix, while the diagonal part is taken as it is. Provided with the value $\varepsilon = 2$ of the bulk dielectric permittivity, our linear systems of interest are heavily diagonally dominant. Therefore, the overall impact of the off-diagonal part for the energies is more or less negligible. For the forces, on the other hand, the dominant diagonal constant vanishes and the accuracy drops to the level of the FMM-approximation.

In terms of the algorithmic complexity, we show first the average time of a single matrix-vector product by the primal and the adjoint PCM and COSMO linear systems in the FIG. 5. As can be seen, the FMM makes every single matrix-vector product by the PCM system linearly scaling, roughly one order of magnitude more expensive with respect to ddCOSMO (when using the very high $p_{\max} = 20$).

As reported in FIG. 6 (left),the total number of iterations required for the ddPCM step is the more or less constant regardless of the molecular system, whereas the number of iterations required for the ddCOSMO step slightly increases when going to larger systems, but despite this it reaches a plateau at around 75–80 iterations. FIG. 6 (right) reports the total run time, which again is linear scaling, and hence, using the FMM accelerated ddPCM it is possible to run calculations on very large systems.

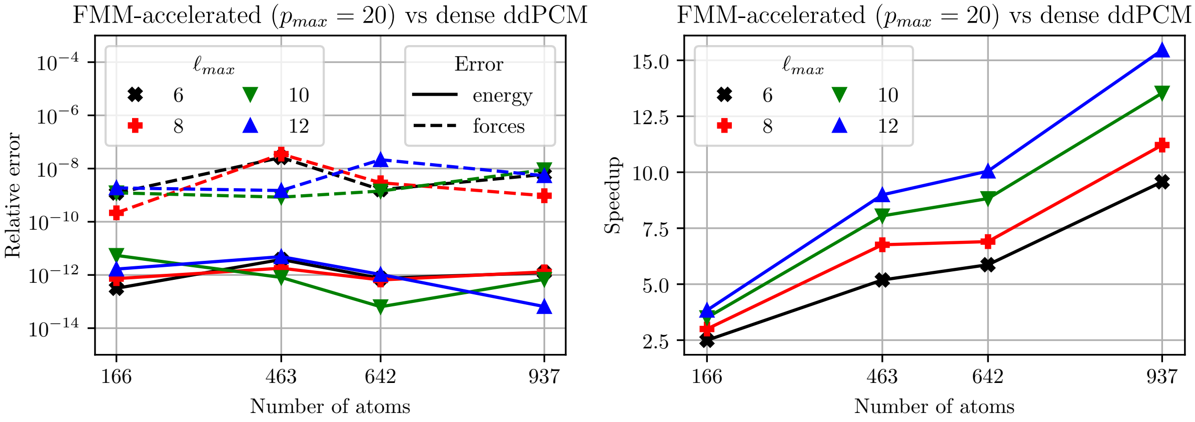


FIG. 3: Relative error of the solvation energy and the forces (left) and the speedup (right) of the FMM-accelerated ($p_{\max} = 20$) ddPCM solver against the dense (non-accelerated) ddPCM solver. Relative stopping criterion is $\tau = 10^{-8}$.

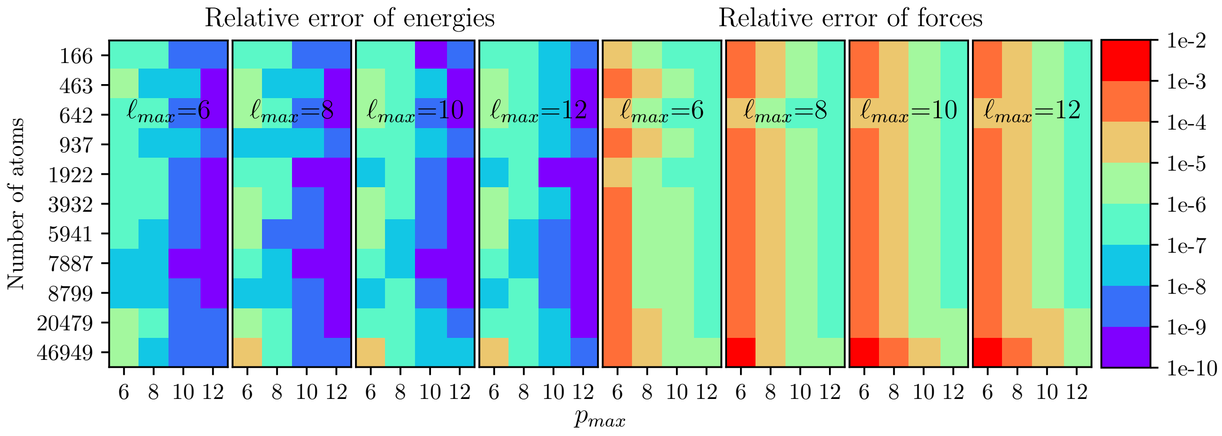


FIG. 4: Relative error of the solvation energy (left) and the forces (right) of the FMM-accelerated ddPCM solver with different $p_{\max}$ against the FMM-accelerated ddPCM solver with $p_{\max} = 20$.

V. CONCLUSION

We presented a FMM-based acceleration of the baseline ddPCM approach yielding linear scaling for the computation of the ddPCM-solvation energy and forces. Numerical experiments show that the accelerated version outperforms the baseline method in terms of execution time both for large and small molecules with a controllable error. Our code is publicly available within the ddX-library that combines implementations of the ddCOSMO,

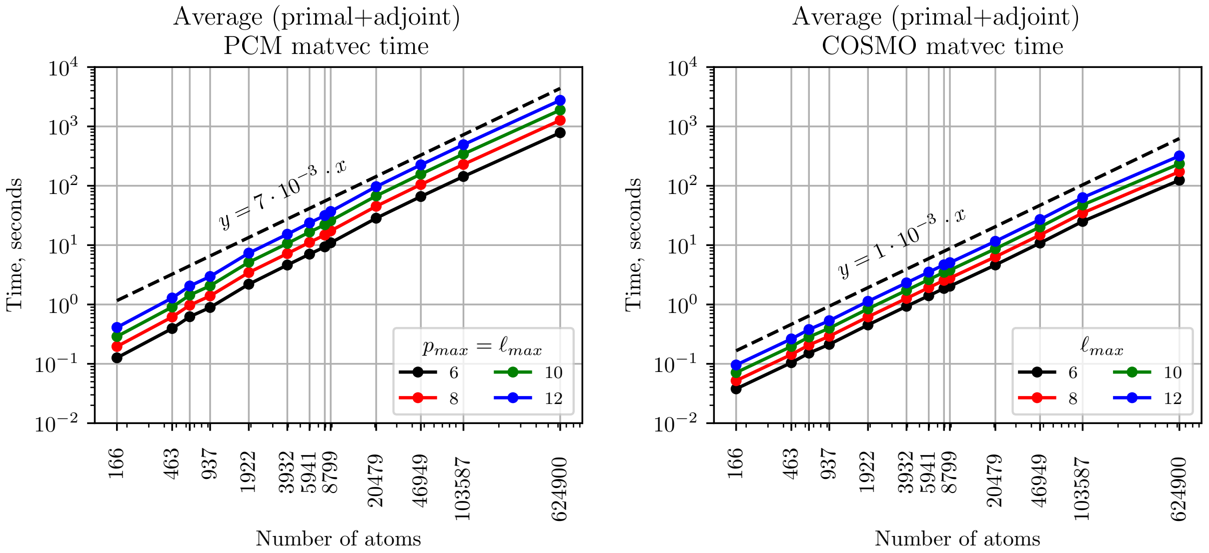


FIG. 5: Average time of a single PCM-related (left) and COSMO-related (right) matrix-vector product.

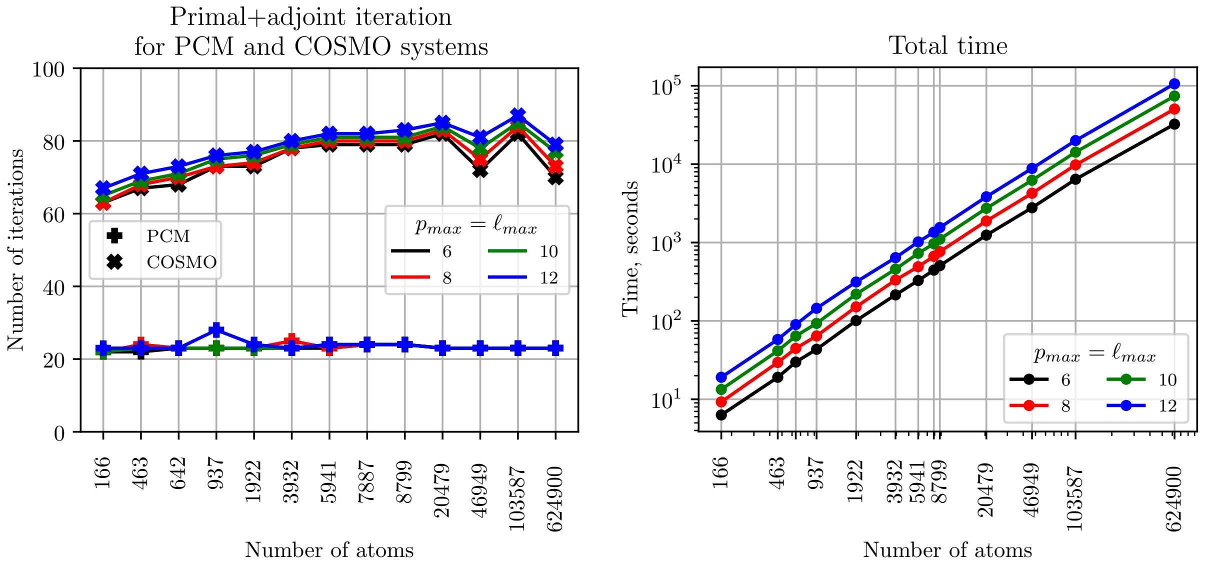


FIG. 6: The number of iterations to solve both the primal and the adjoint PCM and the COSMO linear systems (left) and the total time (right) of the entire execution.

ddPCM and ddLPB. Our FMM-implementation is modular and can be easily adapted to new kernels such as the Yukawa-kernel that is required for the linearized Poisson-Boltzmann equation, which is subject to ongoing work to achieve linear scaling for ddLPB. We have several further possible extensions to the current state of research in our minds: to support Solvent Excluded Surfaces (SES) of cavities, to enable parallelism for shared-memory systems to reduce wall execution time for large molecules and to provide a priori or a posteriori strategy of choosing the maximal degree of the FMM harmonics, which is currently a user-chosen internal parameter of the new ddPCM. A different direction of improvement is testing alternative, coarse-grained, cavity definitions, which use a lower number of spheres with respect to the one-to-one correspondence used in this and in the preceding works. Using coarse grained cavities, where one sphere contains several atoms, on one side lowers the computational cost and on the other side allows for more physically accurate shapes.

DATA AVAILABILITY STATEMENT

The data that support the findings of this study are openly available in Zenodo at http://doi.org/10.5281/zenodo.6656429, which includes the implementation of ddPCM in the ddX-library, the data and scripts to run the results and the jupyter notebook to create the plots. Further, we provide some details, including on the hardware, in Appendix C.

ACKNOWLEDGEMENTS

AM and BS are acknowledging support by the German Research Foundation (DFG) under project 440641818. MN acknowledges financial support from Gaussian, Inc.

REFERENCES

[1] Rick Beatson and Leslie Greengard. A short course on fast multipole methods. *Wavelets, Multilevel Methods Elliptic PDEs*, 1:1–37, 1997. 20

[2] A. Bondi. van der waals volumes and radii. *J. Phys. Chem.*, 68(3):441–451, 1964. 26

[3] E. Cancès, B. Mennucci, and J. Tomasi. A new integral equation formalism for the polarizable continuum model: Theoretical background and applications to isotropic and anisotropic dielectrics. *J. Chem. Phys.*, 107(8):3032–3041, 1997. 3

[4] Eric Cancès, Yvon Maday, and Benjamin Stamm. Domain decomposition for implicit solvation models. *J. Chem. Phys.*, 139(5):054111, 2013. 3

[5] David Leonard Chapman. A contribution to the theory of electrocapillarity. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 25(148):475–481, 1913. 2

[6] Maurizio Cossi, Giovanni Scalmani, Nadia Rega, and Vincenzo Barone. New developments in the polarizable continuum model for quantum mechanical and classical calculations on molecules in solution. *J. Chem. Phys.*, 117(1):43–54, 2002. 3

[7] Christopher J. Cramer and Donald G. Truhlar. Implicit solvation models: equilibria, structure, spectra, and dynamics. *Chem. Rev.*, 99(8):2161–2200, 1999. 2

[8] Ingrid Dreveny, Aleksandra S. Andryushkova, Anton Glieder, Karl Gruber, and Christoph Kratky. Substrate binding in the FAD-dependent hydroxynitrile lyase from almond provides insight into the mechanism of cyanohydrin formation and explains the absence of dehydrogenation activity. *Biochemistry*, 48(15):3370–3377, 2009. 43

[9] Paolo Gatto, Filippo Lipparini, and Benjamin Stamm. Computation of forces arising from the polarizable continuum model within the domain-decomposition paradigm. *J. Chem. Phys.*, 147(22):224108, 2017. 3, 4, 6

[10] M. Gouy. Sur la constitution de la charge électrique à la surface d'un électrolyte. *J. De Physique Théorique Appliquée*, 9(1):457–468, 1910. 2

[11] Leslie Greengard and Vladimir Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73(2):325–348, 1987. 12

[12] John M. Herbert. Dielectric continuum methods for quantum chemistry. *WIREs Comput. Mol. Sci.*, 11(4):e1519, 2021. 2

[13] Shinya Honda, Toshihiko Akiba, Yusuke S. Kato, Yoshito Sawada, Masakazu Sekijima, Miyuki Ishimura, Ayako Ooishi, Hideki Watanabe, Takayuki Odahara, and Kazuaki Harata. Crystal structure of a ten-amino acid protein. *J. Am. Chem. Soc.*, 130(46):15327–15331, 2008. 42

[14] Joseph Ivanic and Klaus Ruedenberg. Rotation matrices for real spherical harmonics. direct determination by recursion. *J. Phys. Chem.*, 100(15):6342–6347, 1996. 20

[15] Joseph Ivanic and Klaus Ruedenberg. Rotation matrices for real spherical harmonics. direct determination by recursion. *J. Phys. Chem. A*, 102(45):9099–9100, 1998. 20

[16] Christian Jelsch, Martha M. Teeter, Victor Lamzin, Virginie Pichon-Pesme, Robert H. Blessing, and Claude Lecomte. Accurate protein crystallography at ultra-high resolution: Valence electron distribution in crambin. *Proc. Natl. Acad. Sci.*, 97(7):3171–3176, 2000.

42

[17] A. Klamt and G. Schüürmann. COSMO: a new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient. *J. Chem. Soc., Perkin Trans. 2*, (5):799–805, 1993. 2

[18] Andreas Klamt. The COSMO and COSMO-RS solvation models. *WIREs Comput. Mol. Sci.*, 1(5):699–709, 2011. 2

[19] Guennadi Kozlov, Long Nguyen, Tong Lin, Gregory De Crescenzo, Morag Park, and Kalle Gehring. Structural basis of ubiquitin recognition by the ubiquitin-associated (UBA) domain of the ubiquitin ligase EDD. *J. Biol. Chem.*, 282(49):35787–35795, 2007. 42, 43

[20] Adrian W. Lange and John M. Herbert. Polarizable continuum reaction-field solvation models affording smooth potential energy surfaces. *J. Phys. Chem. Lett.*, 1(2):556–561, 2009. 3

[21] Adrian W. Lange and John M. Herbert. A smooth, nonsingular, and faithful discretization scheme for polarizable continuum models: The switching/gaussian approach. *J. Chem. Phys.*, 133(24):244111, 2010. 3

[22] V. I. Lebedev and D. N. Laikov. A quadrature formula for the sphere of the 131st algebraic order of accuracy. In *Doklady Mathematics*, volume 59, pages 477–481. Pleiades Publishing, Ltd., 1999. 7

[23] Filippo Lipparini, Louis Lagardère, Giovanni Scalmani, Benjamin Stamm, Eric Cancès, Yvon Maday, Jean-Philip Piquemal, Michael J Frisch, and Benedetta Mennucci. Quantum calculations in solution for large to very large molecules: A new linear scaling qm/continuum approach. *J. Phys. Chem. Lett.*, 5(6):953–958, 2014. 3

[24] Filippo Lipparini, Giovanni Scalmani, Louis Lagardère, Benjamin Stamm, Eric Cancès, Yvon Maday, Jean-Philip Piquemal, Michael J Frisch, and Benedetta Mennucci. Quantum, classical, and hybrid qm/mm calculations in solution: General implementation of the ddcosmo linear scaling strategy. *J. Chem. Phys.*, 141(18):184108, 2014. 3, 4

[25] Filippo Lipparini, Benjamin Stamm, Eric Cancès, Yvon Maday, and Benedetta Mennucci. Fast domain decomposition algorithm for continuum solvation models: Energy and first derivatives. *J. Chem. Theory Comput.*, 9(8):3637–3648, 2013. 3, 6

[26] Sheena McGowan, Christine A. Oellig, Woldeamanuel A. Birru, Tom T. Caradoc-Davies, Colin M. Stack, Jonathan Lowther, Tina Skinner-Adams, Artur Mucha, Pawel Kafarski, Jolanta Grembecka, Katharine R. Trenholme, Ashley M. Buckle, Donald L. Gardiner,

John P. Dalton, and James C. Whisstock. Structure of the plasmodium falciparum m17 aminopeptidase and significance for the design of drugs targeting the neutral exopeptidases. *Proc. Natl. Acad. Sci.*, 107(6):2449–2454, 2010. 43

[27] Benedetta Mennucci. Polarizable continuum model. *WIREs Comput. Mol. Sci.*, 2(3):386–404, 2012. 2

[28] S. Miertuš, E. Scrocco, and J. Tomasi. Electrostatic interaction of a solute with a continuum. a direct utilizaion of AB initio molecular potentials for the prevision of solvent effects. *Chem. Phys.*, 55(1):117–129, 1981. 2

[29] Aleksandr Mikhalev, Michele Nottoli, and Benjamin Stamm. ddX: Linearly scaling computation of ddPCM solvation energy and forces using the fast multipole method. Zenodo, 2022. 42

[30] Ki-Hyun Nam, Oh Yeun Kwon, Kanako Sugiyama, Won-Ho Lee, Young Kwan Kim, Hyun Kyu Song, Eunice Eunkyung Kim, Sam-Yong Park, Hyesung Jeon, and Kwang Yeon Hwang. Structural characterization of the photoswitchable fluorescent protein dronpa-c62s. *Biochem. Biophys. Res. Comm.*, 354(4):962–967, 2007. 43

[31] Michele Nottoli, Benjamin Stamm, Giovanni Scalmani, and Filippo Lipparini. Quantum calculations in solution of energies, structures, and properties with a domain decomposition polarizable continuum model. *J. Chem. Theory Comput.*, 15(11):6061–6073, 2019. 3, 4, 5, 6

[32] Chaoyu Quan, Benjamin Stamm, and Yvon Maday. A domain decomposition method for the polarizable continuum model based on the solvent excluded surface. *Math. Models Methods Appl. Sci.*, 28(07):1233–1266, jun 2018. 3

[33] Chaoyu Quan, Benjamin Stamm, and Yvon Maday. A domain decomposition method for the poisson–boltzmann solvation models. *SIAM J. Sci. Comput.*, 41(2):B320–B350, jan 2019. 3

[34] Giovanni Scalmani, Vincenzo Barone, Konstantin N. Kudin, Christian S. Pomelli, Gustavo E. Scuseria, and Michael J. Frisch. Achieving linear-scaling computational cost for the polarizable continuum model of solvation. *Theor. Chem. Acc.*, 111(2-6):90–100, 2004. 3

[35] Giovanni Scalmani and Michael J. Frisch. Continuous surface charge polarizable continuum models of solvation. i. general formalism. *J. Chem. Phys.*, 132(11):114110, 2010. 3

[36] Benjamin Stamm, Eric Cancès, Filippo Lipparini, and Yvon Maday. A new discretization for the polarizable continuum model within the domain decomposition paradigm. *J. Chem. Phys.*, 144(5):054101, 2016. 3, 4

[37] Benjamin Stamm, Louis Lagardère, Giovanni Scalmani, Paolo Gatto, Eric Cancès, Jean-Philip Piquemal, Yvon Maday, Benedetta Mennucci, and Filippo Lipparini. How to make continuum solvation incredibly fast in a few simple steps: A practical guide to the domain decomposition paradigm for the conductor-like screening model. *Int. J. Quantum Chem.*, 119(1):e25669, 2018. 3

[38] Jacopo Tomasi, Benedetta Mennucci, and Roberto Cammi. Quantum mechanical continuum solvation models. *Chem. Rev.*, 105(8):2999–3094, 2005. 2

[39] Jacopo Tomasi and Maurizio Persico. Molecular interactions in solution: An overview of methods based on continuous distributions of the solvent. *Chem. Rev.*, 94(7):2027–2094, 1994. 2

[40] Ling Zhu, Xiangxi Wang, Jingshan Ren, Claudine Porta, Hannah Wenham, Jens-Ola Ekström, Anusha Panjwani, Nick J. Knowles, Abhay Kotecha, C. Alistair Siebert, A. Michael Lindberg, Elizabeth E. Fry, Zihe Rao, Tobias J. Tuthill, and David I. Stuart. Structure of ljungan virus provides insight into genome packaging of this picornavirus. *Nat. Comm.*, 6(1), 2015. 43

APPENDIX A: ALGORITHMS

Algorithm 5 Recursive inertial bisection of the input atoms.

```
Require: Set of centers $\{\mathbf{x}_j\}_{j=1}^{M}$ of the input $M$ atoms.
Ensure: Binary cluster tree $\mathcal{T}$ with a single sphere in each leaf node.
Init the root node by indexes of all spheres $\mathrm{R} = \{1, \ldots, M\}$
Init a temporary list of nodes $\mathcal{L} = \{R\}$
while $\mathcal{L} \neq \varnothing$ do
    $p = \mathrm{pop}(\mathcal{L})$
    if $|p| > 1$ then {do not divide clusters with a single input sphere}
        Compute the center of mass: $\mathbf{c} = \frac{1}{|p|} \sum_{j=1}^{|p|} \mathbf{x}_{p_j}$.
        Form the $3 \times |p|$ matrix $Y$ with shifted coordinates: $Y_{:,j} = \mathbf{x}_{p_j} - \mathbf{c}$.
        Get the first right singular vector $v$ of the matrix $Y$.
        Let $n = \arg(v < 0)$ {set of indices of negative values of $v$}
        Let $\overline{n} = \{1, \ldots, |p|\} \setminus n$ {set of indices of non-negative values of $v$}
        Extend set $\mathcal{L}$ by two subsets: $\{p_{n_i}\}_{i=1}^{|n|}$ and $\{p_{\overline{n}_i}\}_{i=1}^{|\overline{n}|}$
    end if
end while
```

Algorithm 6 Computing bounding spheres of a binary cluster tree

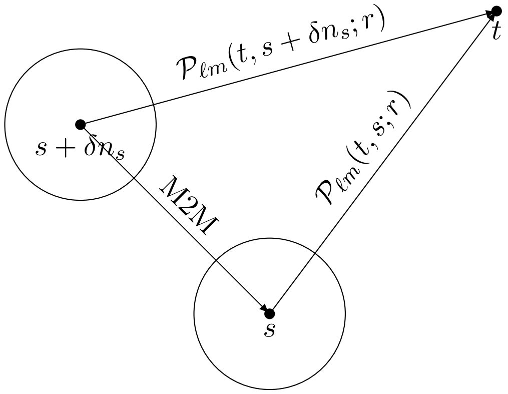
```
Require: Binary cluster tree $\mathcal{T}$.
Ensure: Bounding spheres for all nodes of the tree $\mathcal{T}$.
Let $h = \text{height}(\mathcal{T})$ be a number of levels
for $i = h, \ldots, 1$ do {from the bottom of the tree to the top}
    for all leaf nodes $p$ of the level $i$ of the tree $\mathcal{T}$ do
        Use the input sphere of the node $p$ as its bounding sphere
    end for
    for all non-leaf nodes $p$ of the level $i$ of the tree $\mathcal{T}$ do
        Let $(c_1, r_1)$ and $(c_2, r_2)$ be bounding spheres of children nodes of $p$
        Let $d = \|c_1 - c_2\|$ {Euclidean distance between the spheres}
        if $r_1 \geq r_2 + d$ then {The 2nd sphere is inside the 1st one}
            $(c, r) = (c_1, r_1)$ {Use the 1st sphere}
        else if $r_2 \geq r_1 + d$ then {The 1st sphere is inside the 2nd one}
            $(c, r) = (c_2, r_2)$ {Use the 2nd sphere}
        else
            Let $r = (r_1 + r_2 + d)/2$
            Let $c = c_2 + (d - r_2)\frac{c_1 - c_2}{d}$
        end if
        Use a sphere $(c, r)$ as the bounding sphere of the node $p$
    end for
end for
```

Algorithm 7 Partitioning of a cluster tree into admissible pairs of nodes.

```
Require: Cluster tree $\mathcal{T}$ with the root node R.
Ensure: Lists of far $\mathcal{F}$ and near $\mathcal{N}$ admissible pairs of nodes.
Initialize $\mathcal{N} = \varnothing$ and $\mathcal{F} = \varnothing$.
Initialize the test list $\mathcal{L} = \{(R, R)\}$.
while $\mathcal{L} \neq \varnothing$ do
  $(p_1, p_2) = \mathrm{pop}(\mathcal{L})$ {Get the first pair of nodes to be tested}
  Let $(c_1, r_1)$ and $(c_2, r_2)$ be bounding spheres of $p_1$ and $p_2$ respectively
  Let $d = \|c_1 - c_2\|$ {Distance between centers of spheres}
  if $d - r_1 - r_2 \geq \max(r_1, r_2)$ then {Check the admissibility condition}
    Push the pair $(p_1, p_2)$ into the set $\mathcal{F}$.
  else if $p_1$ and $p_2$ are leaf nodes of the bottom level of the tree $\mathcal{T}$ then
    Push the pair $(p_1, p_2)$ into the set $\mathcal{N}$.
  else {Check all possible pairs of children nodes}
    for all $c_1 \in \mathrm{children}(p_1)$ do
      for all $c_2 \in \mathrm{children}(p_2)$ do
        Push the pair $(c_1, c_2)$ into the set $\mathcal{L}$.
      end for
    end for
  end if
end while
```

Algorithm 8 Balancing a cluster tree by adding mock nodes

```
Require: Unbalanced cluster tree $\mathcal{T}$.
Ensure: Balanced cluster tree $\mathcal{T}$.
Let $h = \mathrm{height}(\mathcal{T})$ be a number of levels
for $i = 1, \ldots, h - 1$ do {top to bottom, except the bottom of the tree}
  for all leaf nodes $p$ of the level $i$ of the tree $\mathcal{T}$ do
    Add child to the node $p$ with the same input atom and bounding sphere.
  end for
end for
```



$\mathcal{P}_{\ell m}(t, s + \delta n_s; r)$
t
$s + \delta n_s$
$\mathcal{P}_{\ell m}(t, s; r)$
M2M
s

FIG. 7: Computing the directional derivative along the direction n_s of the multipole expansion at t due to a multipole located at s with the help of the M2M translation from the sphere at the center $s + \delta n_s$ to the one centered at s.

APPENDIX B: GRADIENT COMPUTATIONS OF THE POTENTIAL AND ADJOINT-POTENTIAL

We will show how we compute gradients of multipole expansions

$$\sum_{\ell m}^{\ell_{\max}} [\mathsf{M}]_{\ell}^{m} \mathcal{P}_{\ell m}(t, s; r)$$

and local expansions

$$\sum_{\ell m}^{p_{\max}} [\mathsf{L}]_{\ell}^{m} \mathcal{L}_{\ell m}(t, s; r)$$

with respect to the position of the source point s. Here, t denotes the target point of the potential induced by the multipole expansion from the source sphere at the center s with radius r.

We first consider the directional derivative of each mode $\mathcal{P}_{\ell m}(t, s, r)$ of the multipole expansion centered at s with respect to a given normalized direction n_s which can be expressed as the following limit:

$$\frac{\partial}{\partial n_s} \mathcal{P}_{\ell m}(t, s; r) := \lim_{\delta \to 0} \frac{\mathcal{P}_{\ell m}(t, s + \delta n_s; r) - \mathcal{P}_{\ell m}(t, s; r)}{\delta}. \tag{63}$$

The source $s + \delta n_s$ of the potential $\mathcal{P}_{\ell m}(t, s + \delta n_s; r)$ can be translated back to the original (fixed) location s using the M2M operation:

$$\mathcal{P}_{\ell m}(t, s + \delta n_s; r) = \sum_{\ell' m'}^{\infty} \mathcal{P}_{\ell' m'}(t, s; r) [\mathrm{M2M}(\delta n_s, r, r)]_{\ell' \ell}^{m' m}, \tag{64}$$

where $[\mathrm{M2M}(\delta n_s, r, r)]^{m'm}_{\ell'\ell}$ is the M2M translation of a solid harmonics of degree (ℓ, m) with scaling radius r from the point δn_s to the solid harmonics of degree (ℓ', m') of the same radius r at the origin.

By substituting equation (64) twice in the equation (63) we get

$$\frac{\partial}{\partial n_s}\mathcal{P}_{\ell m}(t, s; r) = \sum_{\ell' m'}^{\infty} \mathcal{P}_{\ell' m'}(t, s; r) \tag{65}$$

$$\times \lim_{\delta \to 0} \frac{[\mathrm{M2M}(\delta n_s, r, r)]^{m'm}_{\ell'\ell} - [\mathrm{M2M}(0, r, r)]^{m'm}_{\ell'\ell}}{\delta}. \tag{66}$$

This principle is illustrated in FIG. 7.

The M2M operation can now be rewritten in the form of a rotation, OZ translation and a backward rotation as in equation (37):

$$\frac{\partial}{\partial n_s}\mathcal{P}_{\ell m}(t, s; r) = \sum_{\ell' m'}^{\infty} \mathcal{P}_{\ell' m'}(t, s; r) \sum_{k=-\ell}^{\ell} [Q(n_s, \ell')^\top]^{m'k}_{\ell'\ell'} [Q(n_s, \ell)]^{km}_{\ell\ell} \tag{67}$$

$$\times \lim_{\delta \to 0} \frac{[\mathrm{M2M}(\delta e_z, r, r)]^{kk}_{\ell'\ell} - [\mathrm{M2M}(0, r, r)]^{kk}_{\ell'\ell}}{\delta}. \tag{68}$$

The M2M translation over the OZ axis was previously introduced in equation (33). Every element of this translation is a polynomial function of $\rho = \delta$ and we substitute the limits of the differences by the derivative:

$$\lim_{\delta \to 0} \frac{[\mathrm{M2M}(\delta e_z, r, r)]^{kk}_{\ell'\ell} - [\mathrm{M2M}(0, r, r)]^{kk}_{\ell'\ell}}{\delta} = \frac{\mathsf{d}}{\mathsf{d}\delta} \left([\mathrm{M2M}(\delta e_z, r, r)]^{kk}_{\ell'\ell} \right)_{\delta=0}. \tag{69}$$

The values of such derivatives are easy to obtain from equation (33):

$$\frac{\mathsf{d}}{\mathsf{d}\delta} \left([\mathrm{M2M}(\delta e_z, r, r)]^{kk}_{\ell'\ell} \right)_{\delta=0} = 0, \quad \text{if } \ell' \neq \ell + 1, \tag{70}$$

and in the case $\ell' = \ell + 1$ we have

$$\frac{\mathsf{d}}{\mathsf{d}\delta} \left([\mathrm{M2M}(\delta e_z, r, r)]^{kk}_{(\ell+1)\ell} \right)_{\delta=0} = \frac{\sqrt{(\ell + 1 - k)(\ell + 1 + k)(2\ell + 3)}}{r\sqrt{2\ell + 1}}. \tag{71}$$

As is clearly seen, every pair of indices (ℓ, k) yields only a single non-zero derivative of the OZ translation part of the M2M operation. Then, the directional derivative of the single mode of the multipole expansion can be simplified as follows:

$$\frac{\partial}{\partial n_s}\mathcal{P}_{\ell m}(t,s;r) = \frac{1}{r}\sum_{m'=-\ell-1}^{\ell+1}\mathcal{P}_{(\ell+1)m'}(t,s;r) \tag{72}$$

$$\times \sum_{k=-\ell}^{\ell}[Q(n_s,\ell+1)^\top]_{(\ell+1)(\ell+1)}^{m'k}[Q(n_s,\ell)]_{\ell\ell}^{km} \tag{73}$$

$$\times \frac{\sqrt{(\ell+1-k)(\ell+1+k)(2\ell+3)}}{\sqrt{2\ell+1}}. \tag{74}$$

It is important to note that in the case $\ell = \ell_{\max}$, the backward rotation is performed for harmonics of a degree $\ell_{\max}+1$. Let us introduce the following sparse $(\ell_{\max}+2)^2 \times (\ell_{\max}+1)^2$ matrix:

$$[\Lambda_{\mathcal{P}}]_{\ell'\ell}^{k'k} = \begin{cases} 0 & \text{if } (\ell',k') \neq (\ell+1,k), \\ \frac{\sqrt{(\ell+1-k)(\ell+1+k)(2\ell+3)}}{\sqrt{2\ell+1}} & \text{if } (\ell',k') = (\ell+1,k). \end{cases} \tag{75}$$

Then, the directional derivative of the single mode of degree ℓ of the multipole expansion is just a linear combination of all modes of degree $\ell+1$ of the same multipole expansion, i.e.

$$\frac{\partial}{\partial n_s}\mathcal{P}_{\ell m}(t,s;r) = \frac{1}{r}\sum_{m'=-\ell-1}^{\ell+1}\mathcal{P}_{(\ell+1)m'}(t,s;r)[Q(n_s,\ell_{\max}+1)^\top\Lambda_{\mathcal{P}}Q(n_s,\ell_{\max})]_{(\ell+1)\ell}^{m'm}. \tag{76}$$

Therefore, the directional derivative of the entire multipole expansion of degree up to $\ell_{\max}$ is the multipole expansion of maximal degree $\ell_{\max}+1$:

$$\frac{\partial}{\partial n_s}\sum_{\ell m}^{\ell_{\max}}[\mathsf{M}]_\ell^m\mathcal{P}_{\ell m}(t,s;r) = \frac{1}{r}\sum_{\ell m}^{\ell_{\max}+1}[\mathsf{d}\mathcal{P}(n_s)\,\mathsf{M}]_\ell^m\,\mathcal{P}_{\ell m}(t,s;r), \tag{77}$$

where the matrix $\mathsf{d}\mathcal{P}(n_s) \in \mathbb{R}^{(\ell_{\max}+2)^2\times(\ell_{\max}+1)^2}$ is the following product of three matrices:

$$\mathsf{d}\mathcal{P}(n_s) := Q(n_s,\ell_{\max}+1)^\top\Lambda_{\mathcal{P}}Q(n_s,\ell_{\max}), \tag{78}$$

and $[\mathsf{d}\mathcal{P}(n_s)\,\mathsf{M}]_\ell^m$ the (ℓ,m)-th coefficient of the matrix-vector product between $\mathsf{d}\mathcal{P}(n_s)$ and M. It is natural to extend such a notation to a case of the gradient:

$$\mathsf{d}\mathcal{P} = [\mathsf{d}\mathcal{P}(e_x),\, \mathsf{d}\mathcal{P}(e_y),\, \mathsf{d}\mathcal{P}(e_z)]^\top \in \mathbb{R}^{3\times(\ell_{\max}+2)^2\times(\ell_{\max}+1)^2}, \tag{79}$$

which is useful to represent the gradient of the multipole expansion:

$$\nabla_s\sum_{\ell m}^{\ell_{\max}}[\mathsf{M}]_\ell^m\mathcal{P}_{\ell m}(t,s;r) = \frac{1}{r}\sum_{\ell m}^{\ell_{\max}+1}[\mathsf{d}\mathcal{P}\,\mathsf{M}]_\ell^m\mathcal{P}_{\ell m}(t,s;r), \tag{80}$$

where $\mathsf{d}\mathcal{P}\,\mathsf{M} \in \mathbb{R}^{3\times(\ell_{\max}+2)^2}$ with

$$[\mathsf{d}\mathcal{P}\,\mathsf{M}]_\ell^m = \Big[[\mathsf{d}\mathcal{P}(e_x)\mathsf{M}]_\ell^m,\, [\mathsf{d}\mathcal{P}(e_y)\mathsf{M}]_\ell^m,\, [\mathsf{d}\mathcal{P}(e_z)\mathsf{M}]_\ell^m\Big]^\top \in \mathbb{R}^3. \tag{81}$$

On the other hand we also need to obtain gradients of the local expansions with harmonics of a degree up to $p_{\max}$. It is intuitive to repeat all the above multipole-related steps of computing the gradient to the case of the local expansion. The only difference is that the derivative of the L2L translation over the OZ axis is zero in the case where the target degree ℓ' being not equal to the source degree ℓ minus one:

$$\frac{\mathsf{d}}{\mathsf{d}\delta}\left([\mathrm{L2L}(\delta e_z, r, r)]_{\ell'\ell}^{kk}\right)_{\delta=0} = 0, \quad \text{if } \ell' \neq \ell - 1. \tag{82}$$

In the case $\ell' = \ell - 1$, the derivative equals the following expression:

$$\frac{\mathsf{d}}{\mathsf{d}\delta}\left([\mathrm{L2L}(\delta e_z, r, r)]_{(\ell-1)\ell}^{kk}\right)_{\delta=0} = -\frac{\sqrt{(\ell-k)(\ell+k)(2\ell-1)}}{r\sqrt{2\ell+1}}. \tag{83}$$

Effectively, this implies that the directional derivative of the single mode of degree ℓ of the local expansion is a linear combination of all modes of degree $\ell - 1$ of the same local expansion. Therefore, we introduce the following sparse $p_{\max}^2 \times (p_{\max}+1)^2$ matrix:

$$[\Lambda_{\mathcal{L}}]_{\ell'\ell}^{k'k} = \begin{cases} 0 & \text{if } (\ell', k') \neq (\ell-1, k), \\ \frac{-\sqrt{(\ell-k)(\ell+k)(2\ell-1)}}{\sqrt{2\ell+1}} & \text{if } (\ell', k') = (\ell-1, k). \end{cases} \tag{84}$$

Now, we can formally define the operator $\mathsf{d}\mathcal{L}(n_s) : \mathbb{R}^{(p_{\max}+1)^2} \to \mathbb{R}^{p_{\max}^2}$ as follows

$$\mathsf{d}\mathcal{L}(n_s) := Q(n_s, p_{\max}-1)^\top \Lambda_{\mathcal{L}} Q(n_s, p_{\max}). \tag{85}$$

Once again, it is natural to define a gradient-related tensor $\mathsf{d}\mathcal{L} \in \mathbb{R}^{3\times p_{\max}^2 \times (p_{\max}+1)^2}$:

$$\mathsf{d}\mathcal{L} = [\mathsf{d}\mathcal{L}(e_x),\, \mathsf{d}\mathcal{L}(e_y),\, \mathsf{d}\mathcal{L}(e_z)]^\top, \tag{86}$$

with a tensor-vector product $\mathsf{d}\mathcal{L}\,\mathsf{L}$ defined as

$$[\mathsf{d}\mathcal{L}\,\mathsf{L}]_\ell^m = \Big[[\mathsf{d}\mathcal{L}(e_x)\mathsf{L}]_\ell^m,\, [\mathsf{d}\mathcal{L}(e_y)\mathsf{L}]_\ell^m,\, [\mathsf{d}\mathcal{L}(e_z)\mathsf{L}]_\ell^m\Big]^\top. \tag{87}$$

In the end, the gradient of the local expansion is rewritten in a short form:

$$\nabla_s \sum_{\ell m}^{p_{\max}} [\mathsf{L}]_\ell^m \mathcal{L}_{\ell m}(t, s; r) = \frac{1}{r} \sum_{\ell m}^{p_{\max}-1} [\mathsf{d}\mathcal{L}\,\mathsf{L}]_\ell^m \mathcal{L}_{\ell m}(t, s; r), \tag{88}$$

which is analogous to its multipole counterpart (80).

By the construction of the operators $\mathsf{d}\mathcal{P}$ and $\mathsf{d}\mathcal{L}$, their arithmetic complexities are $O(\ell_{\max}^3)$ and $O(p_{\max}^3)$ respectively since both are based on rotations and diagonal scaling of harmonics.

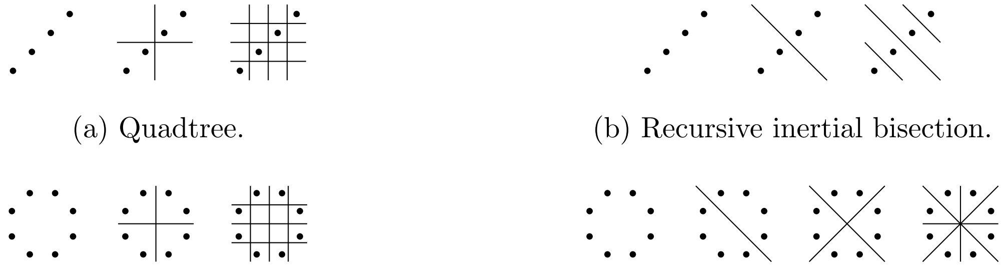
APPENDIX C: REPRODUCIBILITY

For reproducibility, we provide all examples publicly in 29, which includes the implementation of ddPCM in the ddX-library, the data and scripts to run the results and the jupyter notebook to create the plots. Further, we provide some details in the following:

- Our software: ddX library, `https://github.com/ACoM-Computational-Mathematics/ddX` commit 524668e8033c6c9c53809b81e04e4739ac289950
- Other software.
  - Operating system: CentOS Linux 7, kernel 3.10.0-1160.59.1.el7.x86_64,
  - Compilers: GNU Fortran and gcc 8.3.0
  - Third-party libraries: BLAS 3.4.2, LAPACK 3.4.2
  - Miscellaneous executables: CMake 3.22.1,
- Hardware. Only a single cluster node with the following specification was used:
  - Motherboard: Dell Inc. 04DK47 A06,
  - CPU: 2 x AMD EPYC 7282 16-Core Processor @ 2.30GHz,
  - Memory: 512 GB,
- Inputs. For each of the input molecules we prepared a VdW surface by scaling the Van der Waals radii by 1.1. The list of test molecules is the following (number of atoms, name, pdb code):
  1. $M = 166$, chignolin, 2rvd[13]
  2. $M = 463$, melittin monomer, 2mlt
  3. $M = 642$, crambin, 1ejg[16]
  4. $M = 937$, melittin complete, 2mlt
  5. $M = 1922$, ubiquitin A subunit, 2qho[19]
  6. $M = 3932$, ubiquitin AB subunits, 2qho[19]
  7. $M = 5941$, ubiquitin ABC subunits, 2qho[19]

8. $M = 7887$, ubiquitin ABCD subunits, 2qho[19]
9. $M = 8799$, ubiquitin complete, 2qho[19]
10. $M = 20479$, hydroxynitrile lyase, 3gdp[8]
11. $M = 11827$, green fluorescent protein, 2gx2[30]
12. $M = 103587$, hydrolase, 3gdp[26]
13. $M = 624900$, Ljungan virus, 3jb4[40]

- Other ddPCM parameters:
  - Number of the Lebedev grid points: $N_g = 590$,
  - Dielectric permittivity of the solute: $\varepsilon = 2.0$,
  - Relative stop criterion for the Jacobi iterative solver: 10^{-8},
  - Number of DIIS extrapolation points: 25.

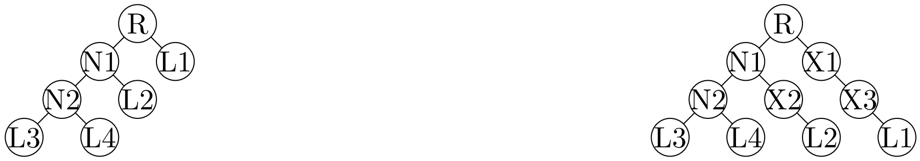

(a) Quadtree.

(b) Recursive inertial bisection.

(c) Quadtree.

(d) Recursive inertial bisection.

FIG. 1: Comparison of a quadtree and a recursive inertial bisection for a line and a circle of equally-weighted particles.



(a) Unbalanced tree

(b) Balanced tree

FIG. 2: Balancing a cluster tree by adding mock nodes X1, X2 and X3. For the balanced tree, all the leaf nodes are on the bottom level, which implies a smaller amount of the near-field interactions and thus lowers the total complexity of the FMM matrix-vector product.

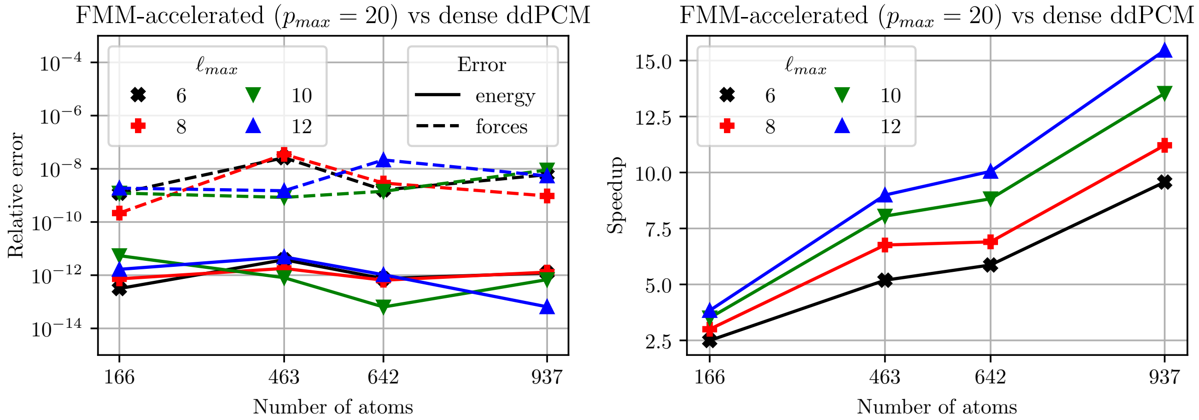


FIG. 3: Relative error of the solvation energy and the forces (left) and the speedup (right) of the FMM-accelerated ($p_{\max} = 20$) ddPCM solver against the dense (non-accelerated) ddPCM solver. Relative stopping criterion is $\tau = 10^{-8}$.

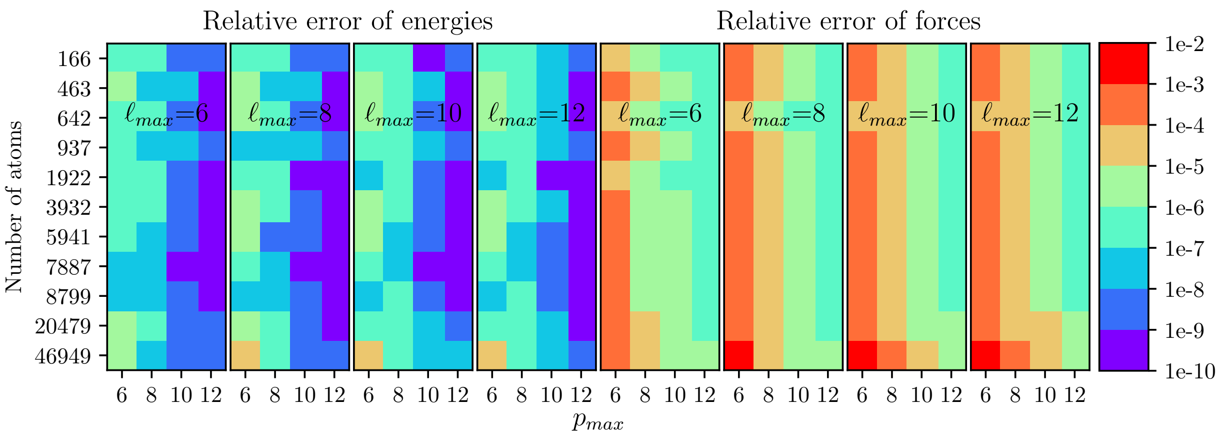


FIG. 4: Relative error of the solvation energy (left) and the forces (right) of the FMM-accelerated ddPCM solver with different $p_{\max}$ against the FMM-accelerated ddPCM solver with $p_{\max} = 20$.

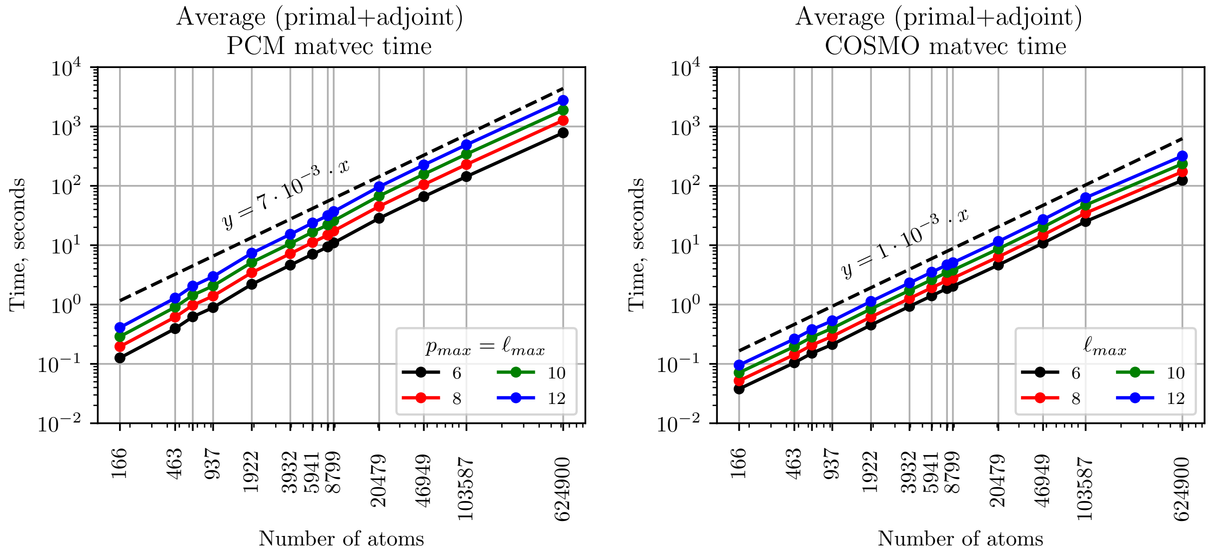


FIG. 5: Average time of a single PCM-related (left) and COSMO-related (right) matrix-vector product.

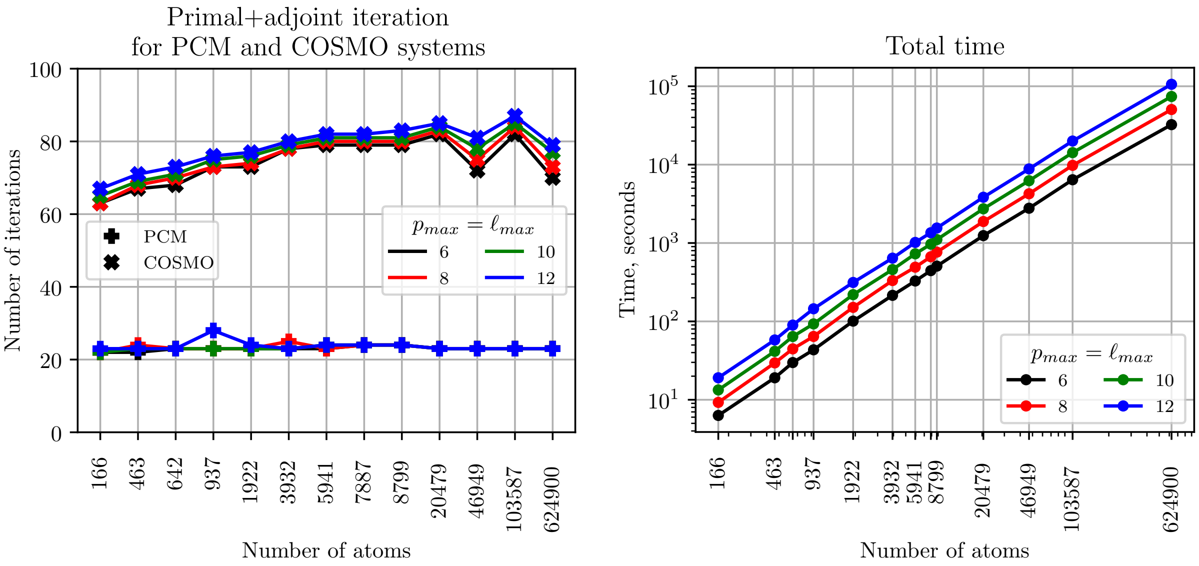


FIG. 6: The number of iterations to solve both the primal and the adjoint PCM and the COSMO linear systems (left) and the total time (right) of the entire execution.

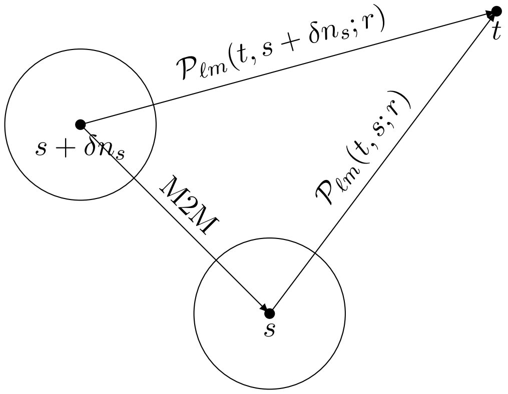

$\mathcal{P}_{\ell m}(t, s + \delta n_s; r)$
t
$s + \delta n_s$
M2M
$\mathcal{P}_{\ell m}(t, s; r)$
s

FIG. 7: Computing the directional derivative along the direction n_s of the multipole expansion at t due to a multipole located at s with the help of the M2M translation from the sphere at the center $s + \delta n_s$ to the one centered at s.