JCTC Journal of Chemical Theory and Computation

Article

pubs.acs.org/JCTC

Scalable Evaluation of Polarization Energy and Associated Forces in Polarizable Molecular Dynamics: I. Toward Massively Parallel Direct Space Computations

Filippo Lipparini,*[,†,‡,§] Louis Lagardère,[§] Benjamin Stamm,[†,∥] Eric Cancès,[⊥] Michael Schnieders,[#] Pengyu Ren,[▽] Yvon Maday,*[,†,○,◆] and Jean-Philip Piquemal*[,†,∥]

[†]Sorbonne Universités, UPMC Univ. Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France

[‡]Sorbonne Universités, UPMC Univ. Paris 06, UMR 7616, Laboratoire de Chimie Théorique, F-75005, Paris, France

[§]Sorbonne Universités, UPMC Univ. Paris 06, Institut du Calcul et de la Simulation, F-75005, Paris, France

[∥]CNRS, UMR 7598 and 7616, F-75005, Paris, France

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

[#]Departments of Biomedical Engineering and Biochemistry, The University of Iowa, Iowa City, Iowa 52358, United States

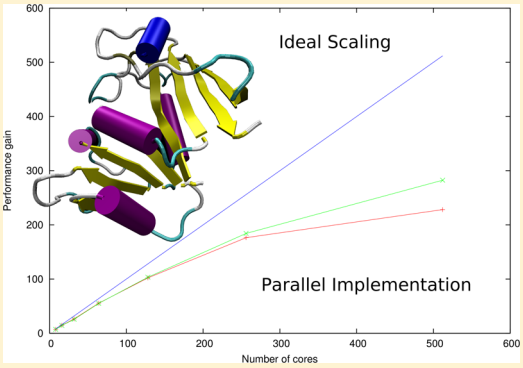
[▽]Department of Biomedical Engineering, The University of Texas at Austin, Austin, Texas 78712, United States

[○]Institut Universitaire de France, Paris, France

[◆]Division of Applied Mathematics, Brown University, Providence, Rhode Island 02912, United States

Supporting Information

ABSTRACT: In this paper, we investigate various numerical strategies to compute the direct space polarization energy and associated forces in the context of the point dipole approximation (including damping) used in polarizable molecular dynamics. We present a careful mathematical analysis of the algorithms that have been implemented in popular production packages and applied to large test systems. We show that the classical Jacobi Over-Relaxation method (JOR) should not be used as its convergence requires a proper value of the relaxation parameter, whereas other strategies should be preferred. On a single node, Preconditioned Conjugate Gradient methods (PCG) and Jacobi algorithm coupled with the Direct Inversion in the Iterative Subspace (JI/DIIS) provide reliable stability/convergence and are roughly twice as fast as JOR. Moreover, both algorithms are suitable for massively parallel implementations. The lower requirements in terms of processes communications make JI/DIIS the method of choice for MPI and hybrid OpenMP/MPI paradigms for real life tests. Furthermore, using a predictor step as a guess along a molecular dynamics simulation provides another inexpensive, yet very effective, form of convergence acceleration. Overall, two to three orders of magnitude in time can be gained compared to the initial JOR single node approach to the final PGC or JI/DIIS parallel one combined with the predictors MD refinements. Such a speedup traces a new route for the high performance implementation of polarizable molecular dynamics and therefore extends the applicability of the technique as it will facilitate future multiscale QM/MM/continuum computations.



1. INTRODUCTION

In recent years, the development of Anisotropic Polarizable Molecular Mechanics[1−4] (APMM) has been of prime importance as the methods have reached the ability of treating small proteins. Among the various strategies employed to deal with polarization, fluctuating charges approaches[5−9] and Drude-type[10,11] models have been shown to scale reasonably well from the computational point of view due to the relative simplicity of their functional form that retains a point charge description of the electrostatics. For methods going beyond the point charge approximation, one has to deal with distributed multipoles and/or with Hermite Gaussian-based distributed densities. For such methods, polarizability is usually included through the point dipole approximation, which has been the most popular strategy as it ensures control on the polarization. Force fields such as SIBFA,[3] EFP,[12] or AMOEBA[13,14] use this technique in conjunction with damping functions to avoid the short-range polarization catastrophe.[15−17] While such polarizable force fields are really robust and ensure the capability to get transferable and quantitative results,[3] as compared to *ab initio* energy calculation, the evaluation of the induced point dipoles

Received: December 19, 2013
Published: February 28, 2014

still represents a computational bottleneck for polarizable force fields, as it requires to solve a linear system whose dimension is very large, being three times the number of polarizable sites (i.e., bearing polarizability). Such an issue has been rather little addressed by the community from a numerical point of view, with the noteworthy exception of a paper by Skeel and Wang[18] that was following the pioneering work of Sagui and Darden dedicated to PME.[19] However, if one is not able to find efficient and massively parallel algorithms to deal with modern computers, the future applicability of polarizable force fields will remain limited despite their advantages, their natural coupling with quantum mechanics (including QM/MM[11,20−25]), and their applicability to complex systems encompassing charged molecules, metals, heterogeneous biological systems, and difficult liquids such as water. Fortunately, the machinery in place has been well studied by applied mathematicians, and mature and robust solutions exist. The purpose of this paper, which is the first of a series, is to review the numerical problem, analyze it thoroughly and propose iterative procedures, which allow for the scalable evaluation of both the polarization energy and of the relative forces to perform numerically stable molecular dynamics (MD) simulations relying on solid applied mathematics. The algorithms presented here are the basis for a new, incoming, massively parallel, implementation of both the SIBFA and AMOEBA force fields within the Tinker[26] and FFX[27,28] packages.

In polarizable MD, two frameworks are possible to deal with the electrostatic and polarization interactions: (i) periodic boundary condition simulations, the electrostatics being normally treated by using the Particle-Mesh Ewald (PME) approach[19,29,30] that requires a reciprocal space evaluation of the induced dipoles or (ii) nonperiodic simulations using continuum solvation models,[31−35] where only direct space computations are performed to evaluate the induced dipoles and the polarization energy. This first paper addresses the second point in tandem with a new fast algorithm to perform simulation in a *polarizable* continuum solvent, described by the Conductor-like Screening Model[36] (COSMO), for which an algorithm fast enough to be employed in MD simulations (ddCOSMO) has been recently proposed by some of us.[37,38] Indeed, with a significantly fast evaluation of the continuum energy and derivatives, the evaluation of the induced dipoles of the MM system can easily become the real bottleneck of the overall computation. In particular, as the scaling of the ddCOSMO algorithm is linear with respect to the size of the system, while the evaluation of the polarization energy is quadratic (however, with a much smaller proportionality factor), the optimization and parallelization of the evaluation of the polarization energies and forces are a required step to develop a multiscale strategy. This remains true also when the APMM/Continuum approach represents the classical part of a QM/MM computation. This paper addresses such a task, providing a proof of concept of effectiveness on modern computers, and is organized as follows. In Section 2, the polarization model is presented in both its traditional and variational formulations, and the forces are derived. In Section 3, the properties of the polarization matrix are discussed, and several iterative procedures are presented and analyzed. In Section 4, several numerical examples are illustrated to show the performances of the various algorithms presented. Finally, in Section 5, some conclusions are drawn and some perspectives given.

2. EVALUATION OF THE POLARIZATION ENERGY AND OF ITS DERIVATIVES: THE NUMERICAL PROBLEM

2.1. Classical Models for Polarization Energies.

Notation. Throughout this paper, we will use the following notation. We will indicate vector quantities with an arrow if the vectors are in $\mathbb{R}^3$ and with bold font if they represent a collection of three-dimensional vectors. For instance, $\boldsymbol{\mu}$ will be a $3M$-dimensional column vector $(\vec{\mu}_1^T,..., \vec{\mu}_M^T)^T$, where each $\vec{\mu}_i = (\mu_i^x,\mu_i^y,\mu_i^z)^T$ is a three-dimensional column vector. We will use Latin indexes (e.g., i, $1 \le i \le M$) as a subscript to refer to a particular atom (with index i) and Greek indexes as superscripts to indicate the Cartesian components of a vector, so that μ_i^α will be the α-th Cartesian component of the dipole associated with the i-th atom.

In this section, we will discuss the polarization equations according to the polarizable point dipoles model and to the smeared Thole's model, which is used to avoid the short-range polarization catastrophe.[16,17] Thole's model has been recently widely used in the context of AMOEBA,[13,14] as well as of other polarizable force fields, including the ones used for QM/MM simulations.[23−25] In a dipole-based polarizable force field, the parameters describing each atom's electrostatic properties are a collection of static multipoles and a polarizability, which is a rank 2 symmetric tensor describing the linear response of such an atom to an external electric field. Depending on the force field, the static multipoles can include either only a point charge or multipoles up to the quadrupole. For the sake of generality, in this communication, we will assume that each atom i is endowed with point charge q_i, static point dipole $\vec{\mu}_{s,i}$, static point quadrupole Θ_i, and polarizability α_i. Notice that the static dipole and quadrupole and, if not isotropic, the polarizability, are vector or tensor quantities, and to define the electrostatics of a molecule, each atomic term has to be rotated in the lab frame by defining a local frame in terms of the positions of some neighboring atoms. This will be discussed in full detail in Appendix B.

The static multipoles create, at each atom i, an electric field $\vec{E}_i$ that is responsible for an induced dipole $\vec{\mu}_i$, the unknown of the polarization problem, in such a way that the (favorable) interaction between the induced dipoles and the inducing field is maximized, while both the work to polarize each site and the repulsion between the induced dipoles are minimized. In other words, the electrostatic equilibrium is reached when the total polarization energy $\mathcal{E}(\mu)$ is minimized, i.e.,

$$\mathcal{E}_{\min} = \min_{\boldsymbol{\mu}\in\mathbb{R}^N} \mathcal{E}(\mu) \tag{1}$$

where as a first classical model, the expression of the energy $\mathcal{E}$ is

$$\mathcal{E} = -\sum_{i=1}^{N} E_i^\alpha \mu_i^\alpha + \frac{1}{2}\sum_{i=1}^{M} [\alpha_i^{-1}]^{\alpha\beta}\mu_i^\alpha\mu_i^\beta + \frac{1}{2}\sum_{i=1}^{M}\sum_{j\neq i} T_{ij}^{\alpha\beta}\mu_i^\alpha\mu_j^\beta \tag{2}$$

Each one of the three terms appearing in eq 2 has a well-defined physical meaning: the first represents the dipoles-field interaction, the second the polarization self-energy and the third the interaction between induced dipoles. Here, the inducing field at each atom i is

$$E_i^\alpha = \sum_{j\neq i} T_{ij}^\alpha q_j + T_{ij}^{\alpha\beta}\mu_{s,j}^\beta + T_{ij}^{\alpha\beta\gamma}\Theta_j^{\beta\gamma} \tag{3}$$

where we use Einstein's summation convention on Greek indexes α, β...,

$$T_{ij}^{\alpha} = \frac{\partial}{\partial r_i^{\alpha}}\frac{1}{r_{ij}} = -\frac{r_{ij}^{\alpha}}{r_{ij}^{3}},\ T_{ij}^{\alpha\beta} = \frac{\partial}{\partial r_i^{\alpha}}T_{ij}^{\beta},\ T_{ij}^{\alpha\beta\gamma} = \frac{\partial}{\partial r_i^{\alpha}}T_{ij}^{\beta\gamma} \tag{4}$$

$\vec{r}_{ij} = \vec{r}_i - \vec{r}_j$ is the vector pointing from atom j to atom i, and $r_{ij} = |\vec{r}_{ij}|$ is its length.

2.2. Damping Scheme for the Energy. The matrices defined in eq 4 can become highly singular when two atoms get too close to each other. In this circumstance, the energy and the magnitude of the induced dipoles would diverge, leading to the so-called *polarization catastrophe*. A possible way to avoid such an unphysical behavior has been first introduced by Thole in his 1981 paper.[16] Thole proposes to look at the induced dipoles not as point dipoles but as exponentially smeared charge distribution, a choice that finds a strong support in the quantum mechanical description of the electronic density. A comprehensive and detailed discussion on the possible damping schemes and on their advantages can be found in ref 17. Here, we will limit ourselves to Thole's original exponential damping scheme, which can be implemented by replacing the Coulomb interaction with a damped Coulomb interaction:

$$\mathcal{T}_{ij}^{\alpha} = \lambda_3(u_{ij})T_{ij}^{\alpha} \tag{5}$$

where

$$\lambda_3(u_{ij}) = 1 - e^{-au_{ij}^3} \tag{6}$$

and $u_{ij} = r_{ij}/(\langle\alpha_i\rangle\langle\alpha_j\rangle)^{1/6}$ is the effective distance as a function of the averaged polarizabilities of sites i and j; $\langle\alpha_i\rangle = 1/3$ tr α_i. Other damping schemes can be obtained by replacing the definition of λ_3 with a different one. Here, a is a nondimensional parameter that determines the strength of the damping. Higher order interaction tensors can be obtained by differentiating eq 5. The complete expressions are reported in Appendix A. The energy $\mathcal{E}_d$ (d as damped) for the Thole's polarizable dipoles is defined as

$$\mathcal{E}_d = -\sum_{i=1}^{N} E_i^{\alpha}\mu_i^{\alpha} + \frac{1}{2}\sum_{i=1}^{M}[\alpha_i^{-1}]^{\alpha\beta}\mu_i^{\alpha}\mu_i^{\beta} + \frac{1}{2}\sum_{i=1}^{M}\sum_{j\neq i}\mathcal{T}_{ij}^{\alpha\beta}\mu_i^{\alpha}\mu_j^{\beta} \tag{7}$$

where

$$E_i^{\alpha} = \sum_{j\neq i}\mathcal{T}_{ij}^{\alpha}q_j + \mathcal{T}_{ij}^{\alpha\beta}\mu_{s,j}^{\beta} + \mathcal{T}_{ij}^{\alpha\beta\gamma}\Theta_j^{\beta\gamma} \tag{8}$$

and the dipoles are obtained by minimizing the functional defined in eq 7 with respect to the dipoles

$$\frac{\partial\mathcal{E}_d}{\partial\mu_i^{\alpha}} = [\alpha_i^{-1}]^{\alpha\beta}\mu_i^{\beta} + \sum_{j\neq i}\mathcal{T}_{ij}^{\alpha\beta}\mu_j^{\beta} - E_i^{\alpha} = 0 \tag{9}$$

We can recast eq 9 in a more compact form by introducing the $3M$ -dimensional matrix $\mathbf{T}$, which we will refer to as the polarization matrix, according to

$$\mathbf{T} = \begin{pmatrix} \alpha_1^{-1} & \mathcal{T}_{12} & \mathcal{T}_{13} & \dots & \mathcal{T}_{1M} \\ \mathcal{T}_{21} & \alpha_2^{-1} & \mathcal{T}_{23} & \dots & \mathcal{T}_{2M} \\ \mathcal{T}_{31} & \mathcal{T}_{32} & \ddots & & \\ \vdots & \vdots & & & \vdots \\ \mathcal{T}_{M1} & \mathcal{T}_{M2} & & \dots & \alpha_M^{-1} \end{pmatrix} \tag{10}$$

where the $\mathcal{T}_{ij}\in\mathbb{R}^{3\times3}$ blocks are defined in Appendix A, and the $3M$ -dimensional vectors

$$\mathbf{E} = \begin{pmatrix} \vec{E}_1 \\ \vec{E}_2 \\ \vdots \\ \vec{E}_M \end{pmatrix},\ \mu = \begin{pmatrix} \vec{\mu}_1 \\ \vec{\mu}_2 \\ \vdots \\ \vec{\mu}_M \end{pmatrix} \tag{11}$$

which allow us to write the polarization linear system as

$$\mathbf{T}\mu = \mathbf{E} \tag{12}$$

and the polarization energy functional as

$$\mathcal{E}_d = \frac{1}{2}\mu^{\dagger}\mathbf{T}\mu - \mathbf{E}^{\dagger}\mu \tag{13}$$

Note that the matrix $\mathbf{T}$ is symmetric.

2.3. Analytical Derivatives of the Polarization Energy. The variational formulation makes the derivation of the forces particularly easy,[39] as it is possible to exploit the following chain rule

$$\frac{\mathrm{d}\mathcal{E}_d}{\mathrm{d}r_k^{\alpha}} = \frac{\partial\mathcal{E}_d}{\partial r_k^{\alpha}} + \frac{\partial\mathcal{E}_d}{\partial\boldsymbol{\mu}}\cdot\frac{\partial\boldsymbol{\mu}}{\partial r_k^{\alpha}} \tag{14}$$

It is crucial to observe that the second term on the right-hand side of eq 14 contains the derivatives of the energy with respect to the induced dipoles, which at the stationary point vanish; therefore, only the explicit derivatives of the energy need to be computed and the forces will be

$$F_k^{\alpha} = -\frac{\partial\mathcal{E}_d}{\partial r_k^{\alpha}}$$

This can be seen as a classical equivalent to the Hellmann–Feynman theorem for variational wave functions. By expanding, we get

$$\begin{aligned} F_k^{\alpha} &= \boldsymbol{\mu}^{\dagger}\frac{\partial\mathbf{E}^{\alpha}}{\partial r_k^{\alpha}} - \frac{1}{2}\boldsymbol{\mu}^{\dagger}\frac{\partial\mathbf{T}}{\partial r_k^{\alpha}}\boldsymbol{\mu} \\ &= \sum_{i=1}^{N}\frac{\partial E_i^{\beta}}{\partial r_k^{\alpha}}\mu_i^{\beta} - \frac{1}{2}\sum_{i=1}^{N}\frac{\partial[\alpha_i^{-1}]^{\beta\gamma}}{\partial r_k^{\alpha}}\mu_i^{\beta}\mu_i^{\gamma} \\ &\quad - \frac{1}{2}\sum_{i=1}^{N}\sum_{j\neq i}\frac{\partial\mathcal{T}_{ij}^{\beta\gamma}}{\partial r_k^{\alpha}}\mu_i^{\beta}\mu_i^{\gamma} \end{aligned} \tag{15}$$

The first term in eq 15 involves the derivatives of the electric field. There are two families of contributions arising from this term: explicit ones, which are to be assembled via the derivatives of the interaction matrices, and implicit ones, which depends on the rotation used to define the static dipoles and quadrupoles in the lab frame starting from their values in a proper local frame. For the second contributions, let R_i be the rotation matrix associated with the local to global frame

transformation, i.e., the matrix that has for columns the coordinates of the basis of the local frame in the global frame basis. A variety of choices for the definition of the local frame, together with the associated rotation matrices and their derivatives, is reported in Appendix B. A documented Fortran routine that computes all the aforementioned quantities is also included in the Supporting Information. Any vector $\vec{v}_i$ or tensor M_i defined in the local system is expressed, in the global frame, as respectively

$$\vec{v}_i = R_i\vec{v}_i^0,\ M_i = R_iM_i^0R_i^{-1} = R_iM_i^0R_i^{\dagger}$$

where we have exploited the fact that as both the local and the lab frames are orthonormal R_i is an orthogonal matrix. As already mentioned, such quantities include static point dipoles and quadrupoles, as well as the atomic polarizability tensors. While the parameters do not depend on the positions of the various atoms, the rotation matrices do, and the forces will therefore include contributions due to the derivatives of R, sometimes improperly referred to as a *torque*. Here, we give the expression of such a contribution for the static dipole and for the static quadrupole to provide an example for both vector and tensor quantities

$$\frac{\partial\mu_{i,s}^{\beta}}{\partial r_k^{\alpha}} = \frac{\partial R_i^{\beta\gamma}}{\partial r_k^{\alpha}}\mu_{i,s}^{\gamma,0},\ \frac{\partial\Theta_i^{\beta\gamma}}{\partial r_k^{\alpha}} = \frac{\partial R_i^{\beta\delta}}{\partial r_k^{\alpha}}\Theta_i^{\delta\epsilon,0}R_i^{\gamma\epsilon} - \Theta_i^{\beta\delta}\frac{\partial R_i^{\delta\epsilon}}{\partial r_k^{\alpha}}R_i^{\gamma\epsilon} \tag{16}$$

The second term in eq 15 is only present if the atomic polarizabilities are anisotropic and need to be rotated in the lab frame. For isotropic (i.e., scalar) polarizabilities, there is no rotation matrix contribution. Finally, the third term involves the derivatives of the $\mathcal{T}_{ij}^{\alpha\beta}$ matrix

$$\frac{\partial\mathcal{T}_{ij}^{\beta\gamma}}{\partial r_k^{\alpha}} = (\delta_{ik} - \delta_{jk})\mathcal{T}_{ij}^{\alpha\beta\gamma} \tag{17}$$

3. NUMERICAL PROPERTIES AND ITERATIVE SOLUTION OF THE POLARIZATION EQUATIONS: FROM CHEMISTRY TO MATHEMATICS

3.1. Some General Considerations. In this section, we will discuss some possible numerical procedures to solve the polarization equation eq 12 in a robust and efficient way. For large systems, iterative procedures are mandatory, as the $O(N^3)$ matrix inversion becomes rapidly infeasible. Notice that as iterative solution schemes rely on matrix−vector multiplications, fast summation techniques[29,40,41] can be involved in the procedure. In this paper, we will not deal with such techniques; nevertheless, all the considerations made in this section also apply when such techniques are used. Some of the aspects of the possible iterative solutions were discussed by Wang and Skeel in a recent paper.[18] Indeed, Wang and Skeel showed in the context of PBC simulations that iterative methods alternative to the traditionally implemented ones could be successfully used to accelerate convergence of the polarization equations. In particular, they suggested the use of a preconditioned conjugate gradient iterative scheme, which we will also analyze in this section. Iterative methods can be grouped in two main families: stationary methods, like Jacobi Iterations (JI), Gauss−Seidel iterations (GS), or the Jacobi Over-Relaxation method (the latter being currently the most used one in the context of the AMOEBA force field), and Krylov subspace methods, such as the Conjugate Gradient (CG), Preconditioned CG (PCG), or the Generalized Minimal RESidual (GMRES) methods. An intermediate scheme can be obtained by coupling the JI with Pulay's direct inversion in the iterative subspace (DIIS) method; such a strategy is especially interesting because it maintains the conceptual simplicity of JI while enjoying superior convergence properties[42] or even ensuring convergence in case JI may not converge.

The JI method is probably the simplest iterative procedure to solve a N-dimensional linear system $\mathbf{Tx} = \mathbf{b}$. Let $\mathbf{D}$ be the diagonal matrix whose elements are the diagonal elements of $\mathbf{T}$, and let $\mathbf{O} = \mathbf{D} - \mathbf{T}$. Jacobi iterations are defined as follows

$$\mathrm{x}^{[n]} = \mathbf{D}^{-1}\mathbf{O}\mathrm{x}^{[n-1]} + \mathbf{D}^{-1}\mathbf{b} \tag{18}$$

It is easily shown that JI converge if the spectral radius $\rho(\mathbf{D}^{-1}\mathbf{O})$, i.e., the modulus of the eigenvalue with the largest modulus, of the iteration matrix $\mathbf{D}^{-1}\mathbf{O}$ is smaller than 1, and the convergence rate is as $\rho(\mathbf{D}^{-1}\mathbf{O})^n$. This means that if the largest eigenvalue of $\mathbf{D}^{-1}\mathbf{O}$ has a modulus larger or equal to 1, the JI will not converge. Furthermore, if one eigenvalue is of a modulus lower but close to 1, convergence will be very slow. Jacobi iterations are granted to converge if $\mathbf{T}$ is strictly diagonal dominant, i.e., if for each row i, $T_{ii} > \sum_j T_{ij}$. We will show through an example that this is often not the case for the polarization problem. A simple way to improve JI (or GS) iterations is to mix at each step the solution with the vector obtained at the previous step by setting

$$\tilde{\mathrm{x}}^{[n]} = \omega\mathrm{x}^{[n]} + (1-\omega)\tilde{\mathrm{x}}^{[n-1]} \tag{19}$$

with

$$\mathrm{x}^{[n]} = \mathbf{D}^{-1}\mathbf{O}\tilde{\mathrm{x}}^{[n-1]} + \mathbf{D}^{-1}\mathbf{b}$$

This method is known as the Jacobi Over-Relaxation (JOR) when based on the Jacobi Iterations, Successive Over-Relaxation (SOR) when based on Gauss−Seidel iterations, and Symmetric Successive Over-Relaxation (SSOR) when applied to the symmetric Gauss-Seidel iterations. It is easily proven that for a positive definite symmetric matrix there always exists some relaxation parameter ω with $0 < \omega < 2/\rho(\mathbf{D}^{-1}\mathbf{O})$, such that the JOR iterations converge; however, the optimal parameter ω_{opt} depends on the spectral properties of the specific matrix and is therefore *not universal*. Again, convergence is assured if $\mathbf{T}$ is strictly diagonal dominant but not for the general case.

Krylov subspace methods are more complex than fixed point methods but enjoy superior robustness as they are always granted to converge in the absence of rounding errors when applied to positive definite matrices. Their convergence rate depends again on the spectral properties of the $\mathbf{T}$ matrix. For symmetric positive definite matrices, the optimal choice is achieved by the conjugate gradient method, which corresponds to a CG minimization of the functional

$$\frac{1}{2}\mathrm{x}^{\dagger}\mathbf{T}\mathrm{x} - \mathrm{x}^{\dagger}\mathbf{b}$$

which for the linear system associated with the induced dipoles corresponds to the energy functional defined in eq 13. It is apparent how having a symmetric and positive definite matrix is mandatory to use the CG algorithm. The functional would not be bounded from below if the matrix had any negative eigenvalue, and its gradient would not be the residual of the linear system if it was not symmetric. Notice, therefore, that if the matrix $\mathbf{T}$ is computed via numerical techniques that are

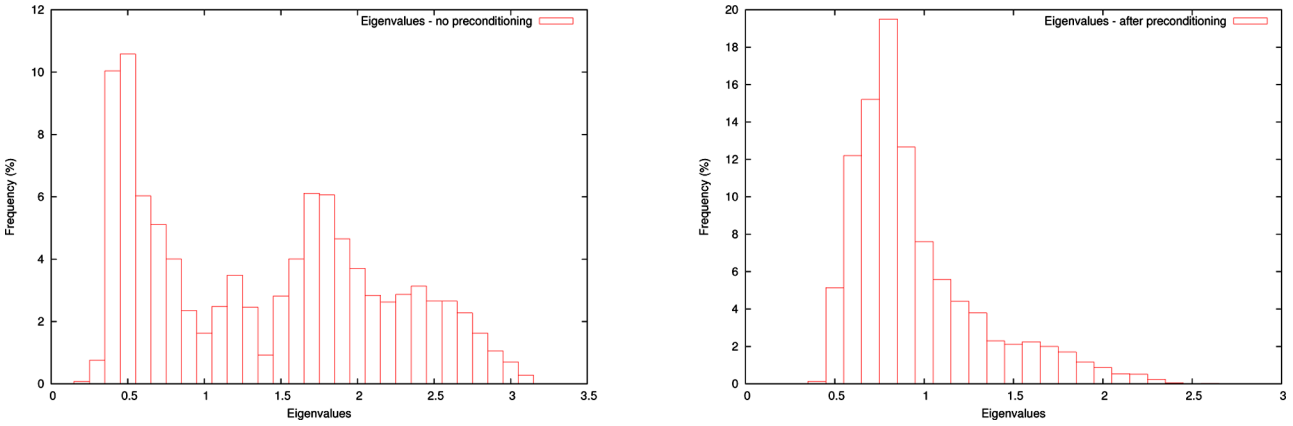


Figure 1. Eigenvalues of the polarization matrix for Ubiquitin, using the AMOEBAbio09 parameters (including damping), without (left) and with (right) preconditioning.

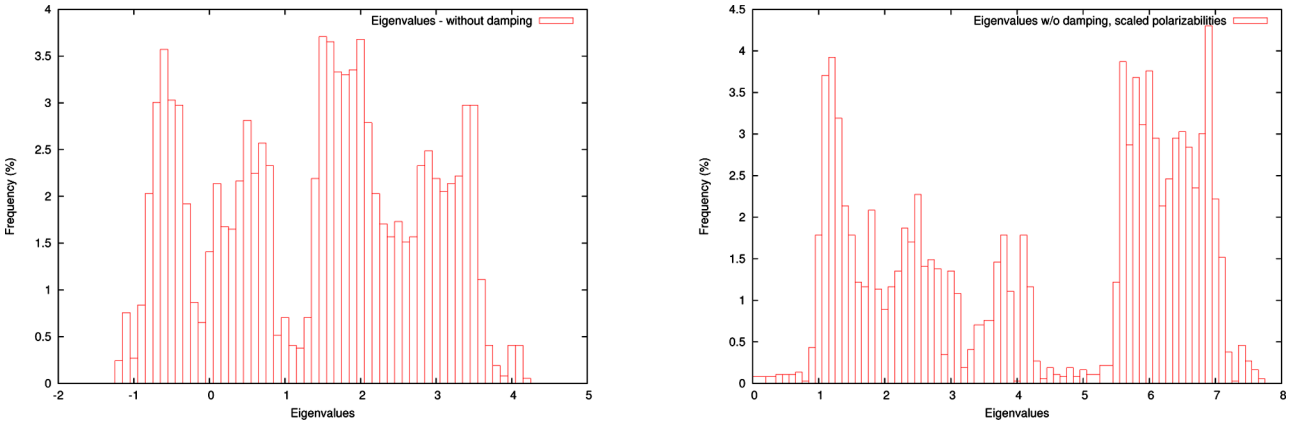


Figure 2. Eigenvalues of the polarization matrix for Ubiquitin without Thole's damping, using the AMOEBAbio09 parameters (left) or with the AMOEBAbio09 polarizabilities scaled by 1/3 (right).

subject to numerical errors such that its symmetry could be spoiled, the CG algorithm might not be the optimal choice. In those cases, CG (and PCG) are generally replaced by GMRES iterative algorithms that recover the good convergences properties. The CG method is easy to implement and requires one to compute one matrix−vector product per iteration, plus one matrix−vector product to compute the starting direction. Its convergence rate depends on the matrix condition number $\kappa(\mathbf{T})$, i.e., the ratio between its largest and smallest eigenvalues, according to

$$\|\mathrm{x}^{[n]} - \mathrm{x}\|_T \leq 2\left(\frac{\sqrt{\kappa(\mathbf{T})} - 1}{\sqrt{\kappa(\mathbf{T})} + 1}\right)^n \|\mathrm{x}^{[0]} - \mathrm{x}\|_T \tag{20}$$

where $\mathbf{x}$ is the exact solution and

$$\|\mathbf{v}\|_T = \sqrt{\mathbf{v}^\dagger \mathbf{T}\mathbf{v}}$$

is the energy norm. The CG method exhibits nonoscillating (monotonic) convergence, meaning that the energy norm of the solution will decrease monotonically at each iteration. If the condition number is large, it is recommended to use a preconditioner to improve convergence. The preconditioned conjugate gradient (PCG) method solves the modified linear system

$$\mathbf{P}^{-1}\mathbf{T}\mathrm{x} = \mathbf{P}^{-1}\mathbf{b} \tag{21}$$

where the matrix $\mathbf{P}^{-1}$ should be close to $\mathbf{T}^{-1}$ but easy to compute. The simplest choice for the preconditioner is to use the inverse of the diagonal elements of $\mathbf{T}$. This diagonal preconditioner has a negligible impact on the overall computational cost and can already be quite effective in clustering the eigenvalues and improving the convergence properties. Of course, *ad hoc* preconditioners can be proposed that better take into account the understanding of the phenomenon under consideration. However, for the polarization problem, the simple generic choice of the diagonal turns out to give sufficiently good results.

As a final numerical procedure to solve a linear system of equations, we report the use of Pulay's DIIS[43,44] to accelerate (or even get) the convergence of JI. The JI/DIIS method can be considered as a Krylov subspace method,[42] where the solution at iteration n is extrapolated as the linear combination of all the solutions at the previous m steps; the coefficients of the extrapolation being determined according to Pulay's procedure and using the residual as an error. The choice of m is in general critical, as large values correspond to better convergence properties but also require more storage and more CPU work for the extrapolation. In the specific case, however, convergence is usually reached in a number of iterations small enough to keep all the iterations in memory (we use $m = 20$, usually convergence is achieved in a smaller number of iterations). In particular, Rohwedder and Schneider[42] established a connection between the JI/DIIS method used for linear

dx.doi.org/10.1021/ct401096t | J. Chem. Theory Comput. 2014, 10, 1638−1651

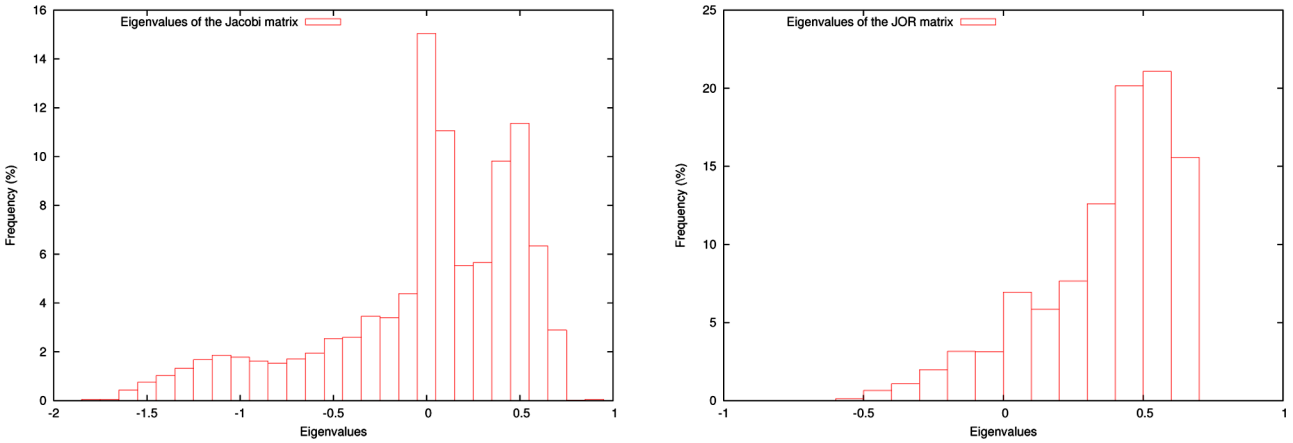


Figure 3. Eigenvalues of the Jacobi iteration matrix (left) and of the JOR matrix (right) with the optimal damping parameter. Notice that the Jacobi matrix coincides with the JOR matrix for $\omega = 1$.

systems and the popular GMRES method, of which the JI/DIIS inherits the nice convergence behavior. In all our numerical experiments, the JI/DIIS algorithm showed convergence rates that are mostly identical to the ones of the preconditioned CG; on the other hand, such a methodology is much less sensitive to small errors that might spoil the symmetry of the matrix and maintains the simplicity of the JI scheme. However, the DIIS extrapolation has a small cost, which scales as $O(m^3)$ as one has to solve a linear system to compute the m extrapolation coefficients, and for $2Nm$, more memory is required.

3.2. Application to the Polarization Problem. In order to choose a convenient algorithm to solve the polarization equations, it is fundamental to better understand the properties of the polarization matrix $\mathbf{T}$. In Figure 1, we show the distribution of the eigenvalues of the polarization matrix $\mathbf{T}$ for a small protein, Ubiquitin (1232 atoms), as obtained by using the AMOEBAbio09[45] force field and Thole's damping, with and without the diagonal preconditioning. The structure of Ubiquitin can be found in the Supporting Information. We also report in Figure 2 the eigenvalues of such a matrix without Thole's damping, either with the AMOEBAbio09 polarizabilities and with the same polarizabilities scaled by a factor of 1/3. Furthermore, in Figure 3, we report the eigenvalues of the JI matrix $\mathbf{D}^{-1}\mathbf{O}$ and of the JOR matrix $\mathbf{B}_w = \omega\mathbf{D}^{-1}\mathbf{O} + (1 - \omega)\mathbf{I}$, where $\mathbf{D}$ is the (block) diagonal matrix of the atomic polarizabilities, $\mathbf{O} = \mathbf{D} - \mathbf{T}$, and ω is the optimal relaxation parameter (Figure 4). The spectral radius of the JOR matrix as a function of the relaxation parameter is also reported in Figure 4. From the left panel in Figure 1, it is possible to see that the matrix produced with the AMOEBAbio09 parameters and Thole's damping has no negative eigenvalue and is therefore positive definitive. Moreover, the eigenvalues are clustered in a small region, and the condition number is relatively small (κ = 13.64). The beneficial effect of the preconditioner is apparent from the right panel of the same figure, as the eigenvalues of the preconditioned matrix are clustered in a smaller interval and the condition number is reduced to κ = 5.47 together with a better repartition of the spectrum. CG convergence is expected to be much faster thanks to the preconditioner [It follows indeed from eq 20 that the reduction factor for the CG algorithm given by the formula $\rho = (\kappa(\mathbf{T}))^{1/2} - 1/(\kappa(\mathbf{T}))^{1/2} + 1$ gives ρ_{cg} = 0.57, while for the PCG, the same expression is ρ_{PCG} = 0.4.]. The effect of the interaction model and of the parametrization are sketched in Figure 2. The polarization matrix loses its

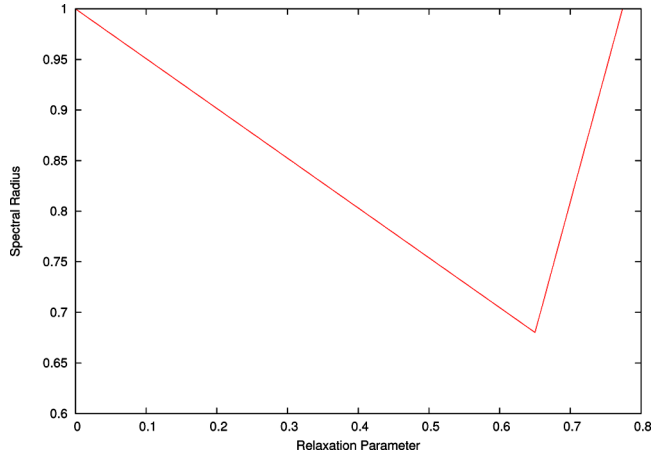


Figure 4. Spectral radius of the JOR matrix as a function of the relaxation parameter.

positivity if Thole's damping is suppressed, making the CG method unsuitable for the solution of the linear equations. A scaling of the polarizabilities of a factor of roughly 1/3 is necessary to restore the positivity. Notice, nevertheless, that the eigenvalues are more scattered and that they cover a more widespread range. The condition number for this matrix is indeed 45.11, much larger than the condition number of the nonpreconditioned Thole's matrix.

The eigenvalues of the Jacobi Iteration matrix, reported in the left panel of Figure 3, show that for the system chosen as an example the JI would not converge. The spectral radius of the matrix is indeed larger than 1 (ρ = 1.58). The effect of the relaxation is apparent from the right panel of the same figure. The eigenvalues are scaled and shifted to be between −1 and 1. The choice of the relaxation parameter is critical to achieve or improve convergence; the spectral radius of the matrix depends sharply on it as shown in Figure 4. Notice how the spectral radius remains high ($\rho(\mathbf{B}_\omega) \simeq 0.7$), even for the optimal ω.

4. NUMERICAL RESULTS

Computational Details. The various algorithms proposed have been implemented in a local version of the TINKER package (MPI, OpenMP, and hybrid implementations) and in a local version of FFX (OpenMP only). All the results shown have been produced with TINKER. All the following numerical

dx.doi.org/10.1021/ct401096t | J. Chem. Theory Comput. 2014, 10, 1638−1651

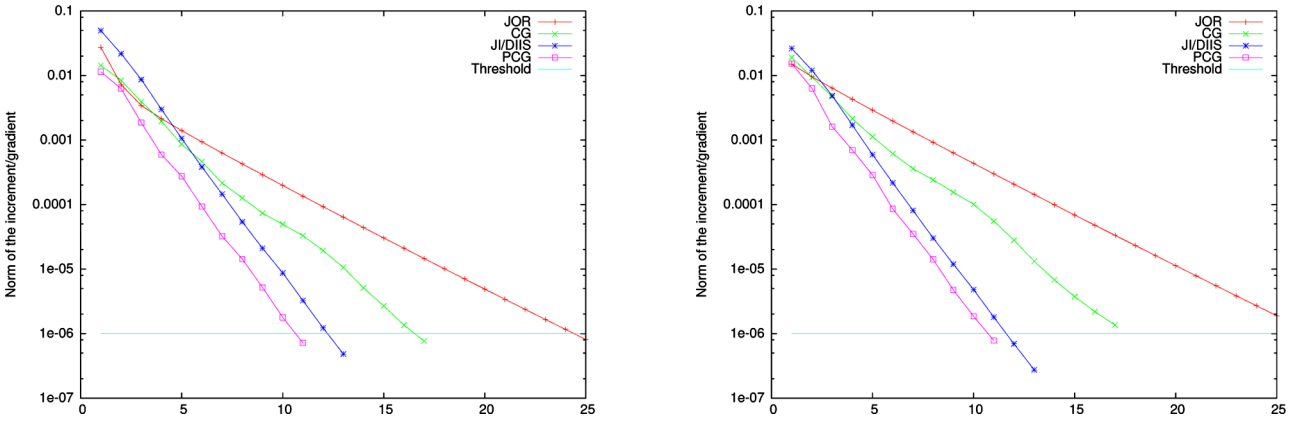


Figure 5. Norm of the increment (or residual) and norm of the error along the iterations for Ubiquitin using various methods.

results were obtained on the SGI UV 2000 supercomputer of the ICS (Institut du Calcul et de la Simulation, SU/UPMC Paris-06), which is mainly made of 64 computational nodes, each one of them being equipped with two Intel Xeon E5−4650L CPUs with eight cores at 2.6 Ghz, associated with 256 GB of DDR3 DIMM RAM, reaching a total of 1024 cores. Furthermore, the shared memory system of the machine uses the SGI NUMAlink interconnect, making transfers at a peak bandwidth of 12.5 Gb/s possible.

4.1. Single Node Comparison of the Various Algorithms. In order to compare the performances of various iterative solutions, we report in Figure 5 the root-mean-square (RMS) norm of the appropriate vector (the increment for JOR and JI/DIIS, the gradient for CG) with respect to the number of iterations, as well as the norm of the error, i.e., the difference between the various solutions at each iteration and the "exact" solution, computed by matrix inversion (using the Cholesky factorization). Ubiquitin is used as a test case, and convergence is reached when the RMS of the increment (or gradient) is smaller than 10^{-6} and the following guess

$$\vec{\mu}_i^{[0]} = \alpha_i \vec{E}_i \tag{22}$$

is common to each iterative solution. We recall that the α_i are the polarizability tensors given as parameters, and the $\vec{E}_i$ are the electric field values created by the static multipoles at the polarizable site i (eq 3). To avoid the quadratic storage requirements, the matrix−vector products are always computed on-the-fly. The polarization matrix $\mathbf{T}$ is never assembled and stored. The Jacobi method, as is apparent from Figure 3, does not converge for Ubiquitin and is therefore not reported. The low performance of the JOR solver is consistent with the considerations made in Section 3. Even by choosing an optimal or close-to-optimal relaxation parameter, the spectral radius of the iteration matrix remains close to 1. Let $\mathbf{e}^{[0]} = \boldsymbol{\mu}^{[0]} - \boldsymbol{\mu}^{[0]}$ be the error made with the initial guess. Any component of such a vector lying along the eigenvectors of the JOR iteration matrix that correspond to a close-to-one eigenvalue will be difficult to annihilate, which explains the slow convergence (25 iterations for residual of 10^{-6}). A straightforward conjugate gradient solver is enough to get better (17 iterations for a similar accuracy) but still not satisfying performances. From the analysis made in Section 3, we expect the effect of the preconditioner to be highly beneficial and in particular to cut the number of iterations by roughly 50%. This is confirmed in this numerical experiment, as convergence is achieved in 11 iterations [It should be noticed that with the values previously computed $\rho_{\mathrm{CG}}^{17} \simeq \rho_{\mathrm{PCG}}^{11}$.]. In addition, it is shown in the same figure that the asymptotic rate of the PCG is much better (actually almost twice better), and that is a consequence of the better repartition of the spectrum as noticed in Figure 1.

Finally, the JI/DIIS solver performs almost as well as the preconditioned conjugate gradient, achieving convergence in 13 iterations. It is our experience that such a solver usually takes one or two iterations more than the preconditioned conjugate gradient to achieve convergence, which makes it a competitive and solid choice. We recall here that while both the JI/DIIS and the CG methodologies need one (expensive) matrix−vector multiplication per iteration, the CG also requires a matrix vector multiplication to compute the starting direction. The overall cost of the iterative solution is therefore roughly the same for the two methods.

4.2. Parallel Implementation. Both the PCG and JI/DIIS solvers are easily parallelized on shared memory architectures via the OpenMP paradigm. We report the performance gain as a function of the number of cores for both solvers and for the computation of the forces in Figure 6, as computed on a SGI

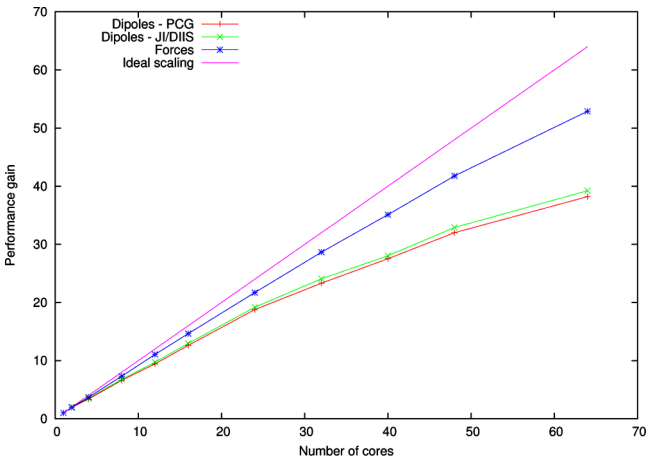


Figure 6. Parallel (shared memory) performance of the PCG and JI/DIIS solvers and of the force evaluation on a NUMA architecture.

UV 2000 machine. A cluster made of Ubiquitin and 2835 water molecules (for a total of 9737 atoms, the structure can be found in the Supporting Information) has been used for the following computations. The scaling is particularly good in the "single-node" region, i.e., within 16 processors, but it is still fair up to 48−64 cores. The Message Passing Interface (MPI) paradigm

dx.doi.org/10.1021/ct401096t | J. Chem. Theory Comput. 2014, 10, 1638−1651

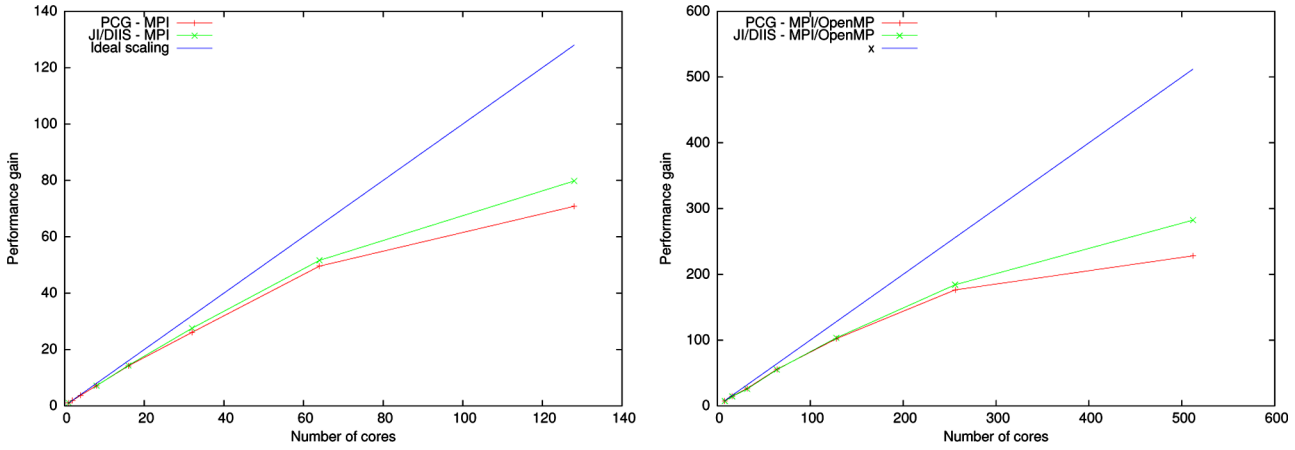


Figure 7. Parallel performance of the PCG and JI/DIIS solvers in a pure MPI (left panel, up to 128 cores) or hybrid MPI/OpenMP (right panel, up to 512) fashion. The Hybrid computation has been performed by assigning eight threads per MPI process.

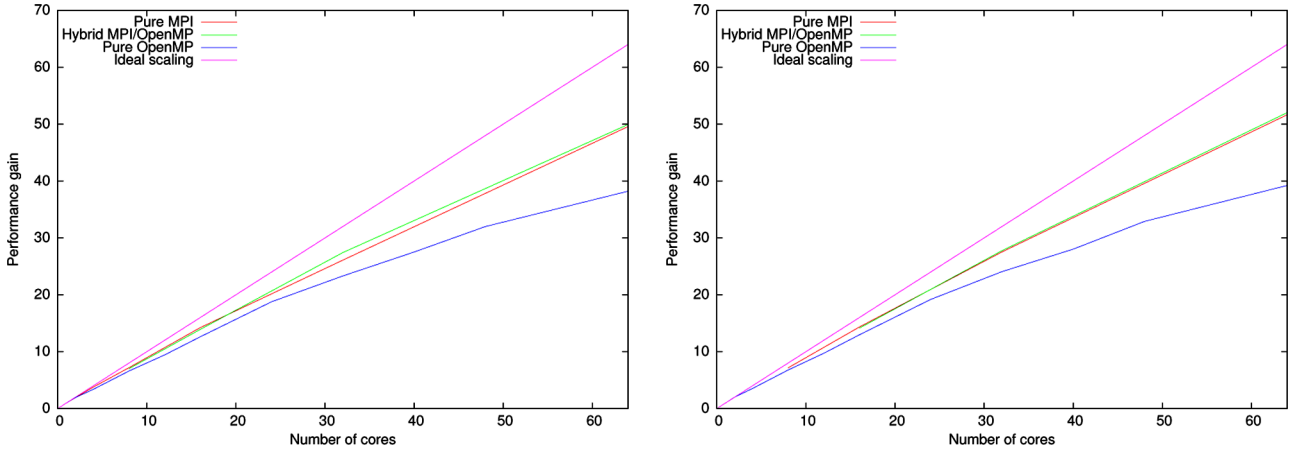


Figure 8. Pure MPI, Pure OpenMP, and Hybrid MPI/OpenMP implementations compared for both the PCG (left) and JI/DIIS (right) solvers.

can be exploited to achieve a more general parallel implementation, which can be exploited also on cluster architectures. The MPI implementation is straightforward. The dipoles are uniformly distributed among the processes, and at each step of the iterative solver (PCG or JI/DIIS), a certain number of communications and reduction operations are needed. For the PCG, once the initial direction is computed, two vectors need to be sent to all the other processes per step: the new solution of the linear system and the descent direction. Moreover, a few very fast reduction operations are needed. The implementation is even simpler for the JI/DIIS method because only one vector, the updated solution, is to be sent at each iteration; again, a few reduction operations are needed to compute the increment and the JI/DIIS matrix.

In both cases, the computational bottleneck lies in the communications, as they grow quadratically with the number of processes. Notice that the good performances of the OpenMP code can be combined with the more general MPI one in a hybrid fashion, which with respect to the pure MPI implementation can exploit the same number of cores while reducing the number of MPI processes and hence the number of communications. This is a particularly good strategy when a cluster architecture is available, as it allows one to minimize the number of (slow) communications through the network while still exploiting all the cores of each node. We report in Figure 7 the scaling performances of both the pure MPI implementation and the hybrid MPI/OpenMP one. The same Ubiquitin−water cluster has been used as a test case. The performances of the MPI implementation are overall good; as expected, the JI/DIIS solver performs slightly better than the PCG one, especially when the hybrid MPI/OpenMP code is used. With respect to the pure OpenMP implementation, where the breakdown is observed after 64 cores, the scalability of the code is overall better, and the number of exploited cores can be doubled. Our tests show the breakdown after 128 cores for the MPI implementation and to 512 cores for the hybrid MPI/OpenMP one; this latter result is particularly promising as the size of the system used as a test case begins to become small with respect to the number of atoms. The overall scaling properties of the code are remarkable. A comparison between the OpenMP, Hybrid, and pure MPI implementations in the 1−64 cores regime is reported inFigure 8. While in the "single node" region, the codes perform in a similar way, the superiority of the MPI and hybrid implementations becomes apparent beyond 32 cores.

4.3. Extension to Molecular Dynamics. In a classical molecular dynamics simulation, a classical trajectory is obtained by integrating the classical equations of motion for a set of atoms moving on the potential energy surface provided by the force field. Let $U(\mathbf{r})$ be such a potential, where the vector $\mathbf{r}$ collects the coordinates of all atoms. The dynamics of the system is described by Newton's equation of motion

dx.doi.org/10.1021/ct401096t | J. Chem. Theory Comput. 2014, 10, 1638−1651

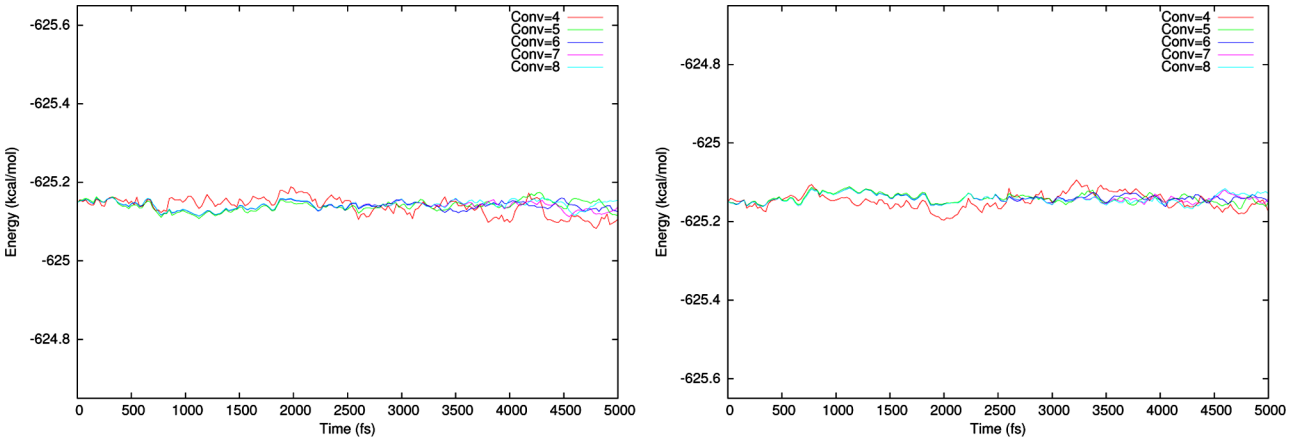


Figure 9. Energy profile of a short MD simulation of Ubiquitin using the direct guess for both the PCG (left) and JI/DIIS (right) solvers for increasing convergence thresholds (threshold is 10^{-x}, with x being reported in the key).

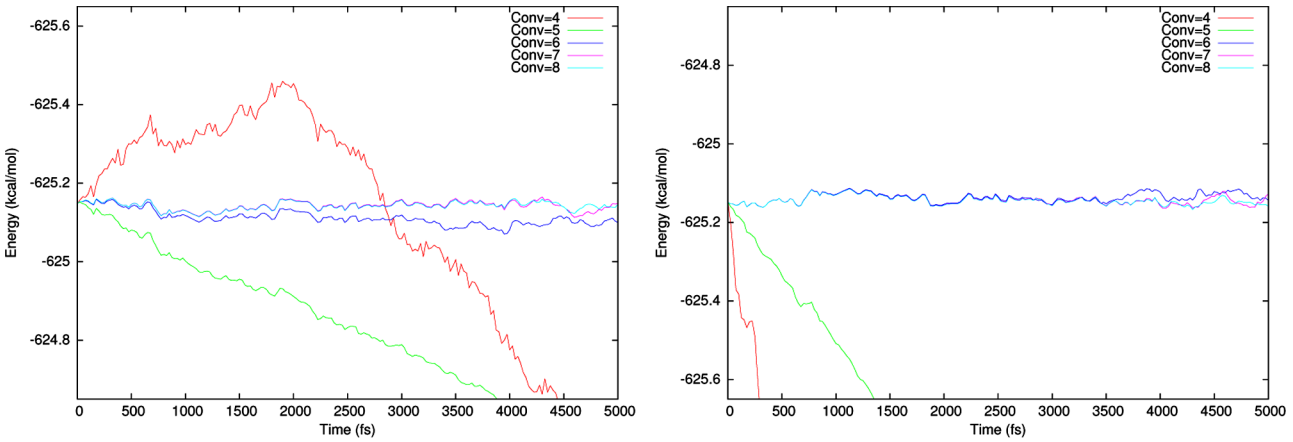


Figure 10. Energy profile of a short MD simulation of Ubiquitin using the previous guess for both the PCG (left) and JI/DIIS (right) solvers for increasing convergence thresholds (threshold is 10^{-x}, with x being reported in the key).

$$\forall\ i \quad m_i \ddot{\mathbf{r}}_i = -\nabla_i U(\mathbf{r}) \tag{23}$$

where we use Newton's notation for time derivatives, and m_i is the mass of the i-th particle. Such equations are integrated by means of the Velocity Verlet[46,47] algorithm, which introduces a discretization of the time variable into finite time steps. For an isolated system, i.e., in the microcanonical ensemble, the total mechanical energy, defined as

$$E = E_{kin}(\dot{\mathbf{r}}) + U(\mathbf{r})$$

is conserved. The quality of the numerical integration can be judged by how well such a requirement is enforced by the approximate trajectory. Clearly, energy conservation is affected by the quality of the solution to the polarization equations. Several parameters of the MD simulation can affect the energy conservation, the most influential being the time step used for the numerical integration. If a polarizable force field is used, the quality of the solution to the polarization equations plays an important role too. In the following, we investigate how the initial guess for the induced dipoles and the convergence threshold used for the iterative solution affect the energy conservation. Among the various guesses, we examine (i) the use of eq 22 ("direct guess"), (ii) the use of the solution to the polarization equations at the previous step ("previous guess"), and (iii) the use of the predictor step of Kolafa's always stable predictor−corrector algorithm[48] ("predictor guess"). The use of a predictor has been suggested also by Wang and Skeel.[18] Notice that we are not using Kolafa's integrator but only his predictor in order to provide a guess for the iterative solvers. For each guess, we perform several short MD simulations (20,000 steps) with different convergence thresholds, and we compare the total energy profiles, using both the PCG and the JI/DIIS solvers. We use a small time step (0.25 fs) to limit its effects on the simulation and focus on the effects of the quality of the dipoles. All the simulations were performed using Ubiquitin as a test case.

The results for the direct guess are reported in Figure 9. The energy conservation is very good even for the sleaziest (10^{-4}) threshold for the RMS norm of the residual for both solvers.

The direct guess, however, is not an optimal choice, as it does not exploit the information that is available during the simulation to provide a good approximation to the solution. The previous guess is already a better approximation, as it is reasonable that the induced dipoles do not change abruptly at each simulation step. The energy conservation results for the previous guess are reported in Figure 10. While convergence is achieved quicker than with the direct guess, the stability of the simulation is affected, and the sleaziest convergence thresholds produce an unsatisfactory energy conservation. At least a 10^{-6} threshold is required to produce a stable simulation. Notice that the two solvers behave similarly, with the JI/DIIS being slightly more stable. The predictor guess uses the information of the

dx.doi.org/10.1021/ct401096t | J. Chem. Theory Comput. 2014, 10, 1638−1651

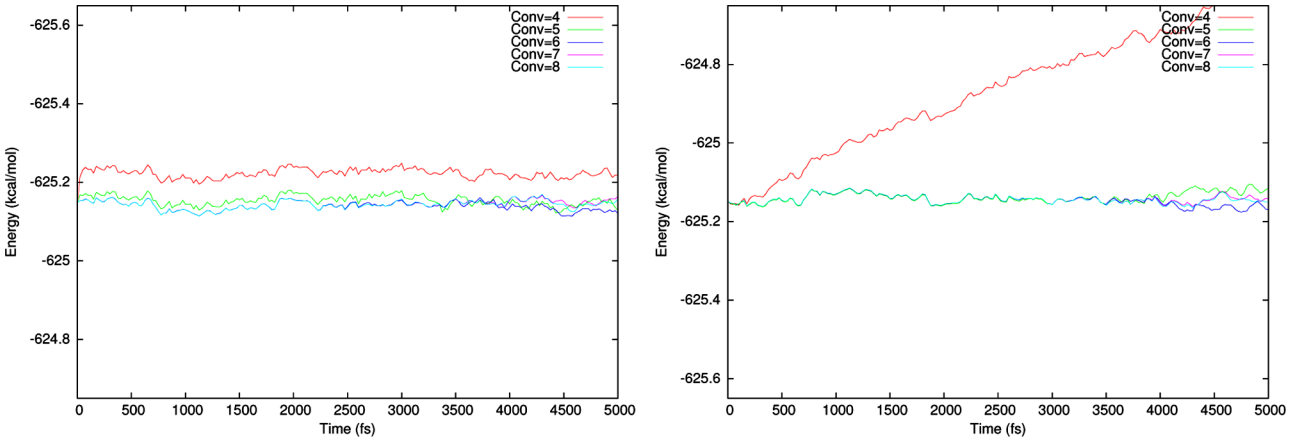


Figure 11. Energy profile of a short MD simulation of Ubiquitin using the predictor guess for both the PCG (left) and JI/DIIS (right) solvers for increasing convergence thresholds (threshold is 10^{-x}, with x being reported in the key).

previous six steps to extrapolate a good approximation of the dipoles. The energy conservation results for such a guess are reported in Figure 11. The simulation becomes stable for a convergence threshold of 10^{-5}. This last guess represents an optimal choice, as it requires a negligible computational effort—the storage of six dipole vectors and their linear combination— while ensuring a fast convergence and a nice stability. Both solvers behave remarkably, and the tighter convergence required to achieve stability with respect to the one required by the direct guess is compensated by the smaller number of iterations required to solve the linear system. To better appreciate the effect of the guess on the number of iterations, we report in Table 1 the number of iterations needed to reach convergence as a function of the threshold for the three guesses, both for the PCG and JI/DIIS solvers. The simulation uses a more realistic time step (1 fs), and represents therefore a "real-life" situation. From the previous analysis, it emerged that the direct guess always produces stable trajectories. A convergence threshold of 10^{-4}, which corresponds to six iterations for the PCG solver and eight for JI/DIIS is sufficient. A 10^{-5} threshold, which corresponds to two additional iterations for either solver, can be used for a better energy conservation. With the previous guess, a 10^{-6} threshold, corresponding to eight iterations for the PCG and nine for JI/DIIS is required. For the PCG, a convergence threshold of 10^{-7} can be used for better energy conservation. Such a guess does not represent a competitive choice with respect to the direct one, especially for the PCG solver. The situation is different if the predictor guess is employed. The simulation becomes stable with a 10^{-5} convergence threshold (3−4 iterations for the PCG, 5 for the JI/DIIS), which might be increased to 10^{-6} for greater precision using the PCG solver (6 iterations). The gain produced by the convergence acceleration amply compensates the tighter convergence requirements, making the computation of the dipoles significantly (30% or more) faster with respect to the direct guess, producing a trajectory of the same quality. This is especially true if the JI/DIIS solver is used, as its behavior is more stable with respect to the choice of the initial guess. Finally, long-term stability is ensured by the use of a symplectic integrator and when the predictor guess is used by its time-reversibility properties.[48] Such a behavior was confirmed numerically, with long simulations showing no apparent total energy drift and oscillation within ±1 kcal/mol.

Table 1. Average Number of Iterations Needed To Reach Convergence for the Direct, Previous, and Predictor Guesses, as a Function of the Convergence Threshold, for Both the PCG and JI/DIIS Solvers

	PCG			JI/DIIS		
threshold	direct	previous	predictor	direct	previous	predictor
10^{-4}	6	3.0	1.9	8	4.4	3.0
10^{-5}	8	5.0	3.5	10	7.0	5.0
10^{-6}	11	8.0	6.0	12	9.0	7.1
10^{-7}	13	10.0	8.0	15	11.9	9.9
10^{-8}	16	12.1	10.0	17	14.0	12.0

5. CONCLUSIONS AND PERSPECTIVES

In this paper, we demonstrate that polarizable force field computations can scale efficiently concerning the direct space evaluation of the polarization energy and forces. Various algorithms have been studied and compared and their numerical stability discussed. If the JOR approach does not ensure convergence without knowing the spectrum and should not be retained being also relatively slow, both the PCG and JI/DIIS approaches have shown to be mathematical sound offering fast and guaranteed convergence. The two strategies appear also viable for parallelization using both the Open-MP and MPI paradigms (and associated hybrid implementation). However, the JI/DIIS involves significantly less communication between processes and appears more suitable for massively parallel implementation. Moreover, in the context of a direct space strategy presented in this paper, the simplicity of the further coupling with a polarizable continuum method would grant the capability of the polarizable point dipole approach to perform multiscale QM/MM/continuum computations. Overall, with a parallel implementation using the JI/DIIS approach, two orders of magnitude in time have been gained compared to the serial JOR initial implementation. To give an order of magnitude, the time to compute the solution to the polarization equations for the Ubiquitin−water cluster goes from more than 1 min with the serial initial JOR code to roughly half a second with the parallel JI/DIIS code using 128 cores, which is appropriate for the size of the system. The time is further reduced roughly by a factor of two during a MD simulation thanks to the use of the predictor guess. An additional, non-negligible gain can be obtained by using Kolafa's predictor to compute a guess for the

dx.doi.org/10.1021/ct401096t | J. Chem. Theory Comput. 2014, 10, 1638−1651

iterative solution. In a practical point of view, the careful study of the matrix **T** also should motivate chemists to derive new force fields with parameters that ensure the proper variational behavior of the polarization energy (resulting in no negative eigenvalue in the matrix **T**!), in link with a meaningful choice of the damping scheme, which profoundly modifies the polarization matrix.

In that context, upcoming works will deal with several aspects. First, we will extend such machinery to more complex force fields such as SIBFA[3] or GEM,[25,49] still in the context of the Tinker and FFX packages. Second, we will study the behavior of the PCG and JI/DIIS in the context of periodic boundary conditions where a massively parallel implementation of the PME is required. Last, we will extend further the scalability of the present approach in the context of our newly developed dd-COSMO toward large-scale polarizable MD simulations on peta- and hexa-scale computers, which will also require the use of state-of-art linear scaling techniques to compute matrix−vector products.

A. APPENDIX: EXPLICIT EXPRESSIONS OF THE DAMPED COULOMB TENSORS

We report here for completeness the expression of the damped Coulomb tensors, which are needed to compute the energy and derivatives of the polarization energy. The (damped) dipole−dipole and dipole−quadrupole interaction tensors are obtained by differentiating once and twice eq 5

$$\mathcal{T}_{ij}^{\alpha\beta} = -\frac{\delta_{\alpha\beta}}{r_{ij}^3}\lambda_3(r_{ij}) + 3\frac{r_{ij}^{\alpha} r_{ij}^{\beta}}{r_{ij}^5}\lambda_5(r_{ij}) \tag{24}$$

$$\mathcal{T}_{ij}^{\alpha\beta\gamma} = 3\frac{r_{ij}^{\alpha}\delta_{\beta\gamma} + r_{ij}^{\beta}\delta_{\alpha\gamma} + r_{ij}^{\gamma}\delta_{\alpha\beta}}{r_{ij}^5}\lambda_5(r_{ij}) - 15\frac{r_{ij}^{\alpha} r_{ij}^{\beta} r_{ij}^{\gamma}}{r_{ij}^7}\lambda_7(r_{ij}) \tag{25}$$

where the damping functions are

$$\lambda_5(u) = 1 - (1 + au^3)e^{-au^3} \tag{26}$$

$$\lambda_7(u) = 1 - \left(1 + au^3 + \frac{3}{5}a^2u^6\right)e^{-au^3} \tag{27}$$

Finally, to compute the derivatives of the electric field, a further differentiation step is needed

$$\begin{aligned}\mathcal{T}_{ij}^{\alpha\beta\gamma\epsilon} = \frac{\partial}{\partial r_{ij}^{\alpha}}\mathcal{T}_{ij}^{\beta\gamma\epsilon} &= 3\frac{\delta_{\alpha\beta}\delta\gamma\epsilon + \delta_{\alpha\gamma}\delta_{\beta\epsilon} + \delta_{\alpha\epsilon}\delta_{\beta\gamma}}{r_{ij}^5}\lambda_5(r_{ij}) \\ &+ 15\frac{r_{ij}^{\alpha} r_{ij}^{\beta}\delta_{\gamma\epsilon} + r_{ij}^{\alpha} r_{ij}^{\gamma}\delta_{\beta\epsilon} + r_{ij}^{\alpha} r_{ij}^{\epsilon}\delta_{\beta\epsilon} + r_{ij}^{\beta} r_{ij}^{\gamma}\delta_{\alpha\epsilon} + r_{ij}^{\beta} r_{ij}^{\epsilon}\delta_{\alpha\gamma} + r_{ij}^{\gamma} r_{ij}^{\epsilon}\delta_{\alpha\beta}}{r_{ij}^7} \\ &\times \lambda_7(r_{ij}) \\ &+ 105\frac{r_{ij}^{\alpha} r_{ij}^{\beta} r_{ij}^{\gamma} r_{ij}^{\delta}}{r_{ij}^9}\lambda_9(r_{ij})\end{aligned} \tag{28}$$

where

$$\lambda_9(u) = 1 - \left(1 + au^3 + \frac{18a^2u^6 + 9a^3u^9}{35}\right)e^{-au^3} \tag{29}$$

B. APPENDIX: ROTATION MATRICES AND THEIR DERIVATIVES

In this appendix, we will detail the definition of three common local frames used to rotate the atomic multipoles in the lab frame. Troughout the appendix, we will use bold font to denote vectors and standard font to denote the length of such vector, so that $v = |\mathbf{v}|$, for instance. Let i be the index of the atoms where the multipoles are placed. Let also *iz*, *ix*, and *iy* be the indexes of the atoms used to define the local frame with respect to the global one. We will denote

$$\xi = \mathbf{r}_{iz} - \mathbf{r}_i$$

$$\eta = \mathbf{r}_{ix} - \mathbf{r}_i$$

$$\varsigma = \mathbf{r}_{iy} - \mathbf{r}_i$$

and characterize the unit vectors that define the local frame in the global frame basis through these three vectors that point, from the center where the multipoles are to two or three neighboring atoms. For any two nonparallel vectors **u** and **v**, we will introduce the orthogonal basis vectors $\mathbf{e}_z = e_z(\mathbf{u})$, $\mathbf{e}_x = \mathrm{e}_x(\mathrm{e}_z,\mathbf{v})$, and $\mathrm{e}_y = \mathrm{e}_y(\mathrm{e}_z,\mathrm{e}_x)$ by

$$\mathbf{e}_z = \frac{\mathbf{u}}{u},\ \mathbf{e}_x = \frac{\mathbf{v} - (\mathbf{v}\cdot\mathbf{e}_z)\mathbf{e}_z}{\left[v^2 - (\mathbf{v}\cdot\mathbf{e}_z)^2\right]^{1/2}},\ \mathbf{e}_y = \mathbf{e}_z \times \mathbf{e}_x \tag{30}$$

A sketch of the definition of all the aforementioned quantities can be found in Figure 12. In the following, we present four ways to introduce a local frame by different choices of **u** and **v** as a function of $\boldsymbol{\xi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\zeta}$. Then, we will be able to write the derivatives by exploiting a suitable chain rule. Notice that the **u** vector defines the z direction, while the x axis lies on the plane spanned by **u** and **v** (in particular, it is obtained by Gram−Schmidt orthogonalization of **v** with respect to **u**), and the y direction is the one orthogonal to the x,z plane.

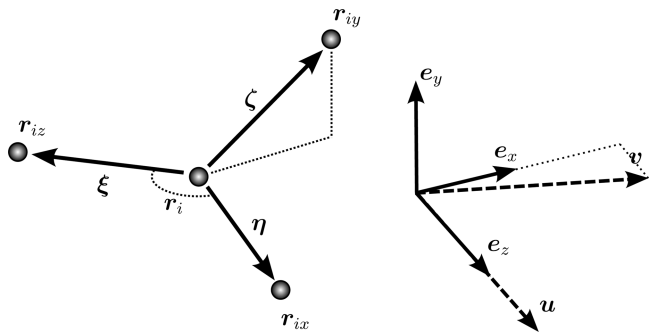


Figure 12. Definition of the $\boldsymbol{\xi}$, $\boldsymbol{\eta}$, and $\boldsymbol{\zeta}$ vectors (left panel) and of the **u** and **v** vectors (right panel). Notice how the local frame is defined in terms of **u** and **V**.

- In the **Z-then-X** scheme (Figure 13), for each atom i, two other atoms are used to define the local frame. The first one, labeled *iz*, defines the direction of the z axis. The second one, labeled *ix*, is used to define the x−z plane, and the x axis is defined by Gram−Schmidt orthogonalization. More in detail, it consists of the above procedure with the choice of $\mathbf{u} = \boldsymbol{\xi}$ and $\mathbf{v} = \boldsymbol{\eta}$, so that

$$\mathbf{e}_z = \frac{\boldsymbol{\xi}}{\xi}$$

dx.doi.org/10.1021/ct401096t | J. Chem. Theory Comput. 2014, 10, 1638−1651

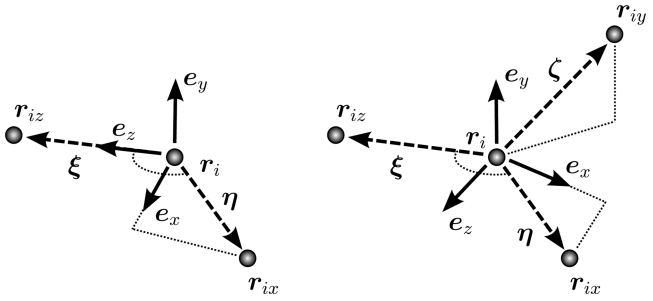


Figure 13. Definition of the Z-then-X (left) and Bisector (right) schemes.

$$\mathbf{e}_x = \frac{\boldsymbol{\eta} - (\boldsymbol{\eta}\cdot\mathbf{e}_z)\mathbf{e}_z}{|\boldsymbol{\eta} - (\boldsymbol{\eta}\cdot\mathbf{e}_z)\mathbf{e}_z|} = \frac{\boldsymbol{\eta} - (\boldsymbol{\eta}\cdot\mathbf{e}_z)\mathbf{e}_z}{[\eta^2 - (\boldsymbol{\eta}\cdot\mathbf{e}_z)^2]^{1/2}}$$

- The **Bisector** scheme (Figure 13) defines the direction of the z axis, given a central atom i and two atoms iz, ix, as the bisector of the $iz{-}i{-}ix$ angle. The $x{-}z$ plane is the defined using the ix atom, and the x axis is defined as usual by orthogonalization. This consists of the above procedure with $\mathbf{u} = \eta\boldsymbol{\xi} + \xi\boldsymbol{\eta}$ and $\mathbf{v} = \boldsymbol{\eta}$, so that

$$\mathbf{e}_z = \frac{\eta\boldsymbol{\xi} + \xi\boldsymbol{\eta}}{|\eta\boldsymbol{\xi} + \xi\boldsymbol{\eta}|} = \frac{\eta\boldsymbol{\xi} + \xi\boldsymbol{\eta}}{[2\xi^2\eta^2 + 2\xi\eta(\boldsymbol{\xi}\cdot\boldsymbol{\eta})]^{1/2}}$$

$$\mathbf{e}_x = \frac{\boldsymbol{\eta} - (\boldsymbol{\eta}\cdot\mathbf{e}_z)\mathbf{e}_z}{|\boldsymbol{\eta} - (\boldsymbol{\eta}\cdot\mathbf{e}_z)\mathbf{e}_z|} = \frac{\boldsymbol{\eta} - (\boldsymbol{\eta}\cdot\mathbf{e}_z)\mathbf{e}_z}{[\eta^2 - (\boldsymbol{\eta}\cdot\mathbf{e}_z)^2]^{1/2}}$$

- The **Z-Bisector** scheme (Figure 14) uses the atom iz to identify the z axis and the bisector of the $ix{-}i{-}iy$ atoms to

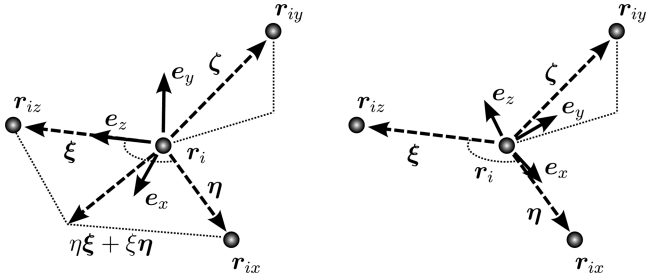


Figure 14. Definition of the Z-Bisector (left) and 3-Fold (right) schemes.

define the xz plane. The x axis is always obtained by orthogonalization. Again, $\mathbf{u} = \boldsymbol{\xi}$ and $\mathbf{v} = \eta\boldsymbol{\zeta} + \zeta\boldsymbol{\eta}$, so that

$$\mathbf{e}_z = \frac{\boldsymbol{\xi}}{\xi}$$

$$\mathbf{e}_x = \frac{\eta\boldsymbol{\zeta} + \zeta\boldsymbol{\eta} - [(\eta\boldsymbol{\zeta} + \zeta\boldsymbol{\eta})\cdot\mathbf{e}_z]\mathbf{e}_z}{|\eta\boldsymbol{\zeta} + \zeta\boldsymbol{\eta} - [(\eta\boldsymbol{\zeta} + \zeta\boldsymbol{\eta})\cdot\mathbf{e}_z]\mathbf{e}_z|}$$

- Finally, the **3-Fold** scheme (Figure 14) puts the z direction along the sum of the vectors going from the central atom to ix, iy, and iz, respectively. The x,z plane is then identified by such vector and the one pointing from i to ix. Again, $\mathbf{u} = \eta\zeta\boldsymbol{\xi} + \xi\zeta\boldsymbol{\eta} + \xi\eta\boldsymbol{\zeta}$ and $\mathbf{v} = \boldsymbol{\eta}$, so that

$$e_z = \frac{\eta\zeta\boldsymbol{\xi} + \xi\zeta\boldsymbol{\eta} + \xi\eta\boldsymbol{\zeta}}{|\eta\zeta\boldsymbol{\xi} + \xi\zeta\boldsymbol{\eta} + \xi\eta\boldsymbol{\zeta}|} = \frac{\eta\zeta\boldsymbol{\xi} + \xi\zeta\boldsymbol{\eta} + \xi\eta\boldsymbol{\zeta}}{[3\xi^2\eta^2\zeta^2 + 2\xi^2\eta\zeta(\boldsymbol{\eta}\cdot\boldsymbol{\zeta}) + 2\eta^2\xi\zeta(\boldsymbol{\xi}\cdot\boldsymbol{\zeta}) + 2\zeta^2\xi\eta(\boldsymbol{\xi}\cdot\boldsymbol{\eta})]^{1/2}}$$

$$\mathbf{e}_x = \frac{\boldsymbol{\eta} - (\boldsymbol{\eta}\cdot\mathbf{e}_z)\mathbf{e}_z}{|\boldsymbol{\eta} - (\boldsymbol{\eta}\cdot\mathbf{e}_z)\mathbf{e}_z|} = \frac{\boldsymbol{\eta} - (\boldsymbol{\eta}\cdot\mathbf{e}_z)\mathbf{e}_z}{[\eta^2 - (\boldsymbol{\eta}\cdot\mathbf{e}_z)^2]^{1/2}}$$

Here, we put $\mathbf{u} = \eta\zeta\boldsymbol{\xi} + \xi\zeta\eta + \xi\eta\boldsymbol{\zeta}$ and $\mathbf{v} = \boldsymbol{\eta}$.

B.1. Analytical Derivatives of the Rotation Matrices

We will here proceed to compute the analytical derivatives of the three orthogonal vectors that define the rotation matrix. In order to do so, we will exploit the following chain rules

$$\frac{\partial e_z^\alpha}{\partial r_j^\alpha} = \sum_\gamma \frac{\partial e_z^\alpha}{\partial u^\gamma}\frac{\partial u^\gamma}{\partial r_j^\alpha}$$

$$\frac{\partial e_x^\alpha}{\partial r_j^\alpha} = \sum_\gamma \left(\frac{\partial e_x^\alpha}{\partial v^\gamma}\frac{\partial u^\gamma}{\partial r_j^\alpha} + \frac{\partial e_x^\alpha}{\partial e_z^\gamma}\frac{\partial e_z^\gamma}{\partial r_j^\alpha}\right)$$

$$\frac{\partial e_y^\alpha}{\partial r_j^\alpha} = \sum_\gamma \left(\frac{\partial e_y^\alpha}{\partial e_x^\gamma}\frac{\partial e_x^\gamma}{\partial r_j^\alpha} + \frac{\partial e_y^\alpha}{\partial e_z^\gamma}\frac{\partial e_z^\gamma}{\partial r_j^\alpha}\right)$$

where $j = i,iz,ix,iy$. Each derivative consists of a term that does not depend on the specific choice of the local frame, which is to be contracted with the proper specific term. The general contributions are, respectively

$$\frac{\partial e_z^\alpha}{\partial u^\gamma} = \frac{\delta_{\alpha\gamma}}{u} - \frac{u^\alpha u^\gamma}{u^3}$$

$$\frac{\partial e_x^\alpha}{\partial v^\gamma} = \frac{\delta_{\alpha\gamma} - e_z^\alpha e_z^\gamma}{[v^2 - (\mathbf{v}\cdot\mathbf{e}_z)^2]^{1/2}} - \frac{(v^\alpha - (\mathbf{v}\cdot\mathbf{e}_z)e_z^\alpha)(v^\gamma - (\mathbf{v}\cdot\mathbf{e}_z)e_z^\gamma)}{[v^2 - (\mathbf{v}\cdot\mathbf{e}_z)^2]^{3/2}}$$

$$\frac{\partial e_x^\alpha}{\partial e_z^\gamma} = \frac{(v^\alpha - (\mathbf{v}\cdot\mathbf{e}_z)e_z^\alpha)(\mathbf{v}\cdot\mathbf{e}_z)v^\gamma}{[v^2 - (\mathbf{v}\cdot\mathbf{e}_z)^2]^{3/2}} - \frac{v^\gamma e_z^\alpha + (\mathbf{v}\cdot\mathbf{e}_z)\delta_{\alpha\gamma}}{[v^2 - (\mathbf{v}\cdot\mathbf{e}_z)^2]^{1/2}}$$

$$\frac{\partial e_y^\alpha}{\partial e_x^\gamma} = \epsilon_{\alpha\sigma\tau}e_z^\sigma\delta_{\gamma\tau}$$

$$\frac{\partial e_y^\alpha}{\partial e_z^\gamma} = \epsilon_{\alpha\sigma\tau}\delta_{\gamma\sigma}e_x^\tau$$

where $\epsilon_{\alpha\beta\gamma}$ is the Levi–Civita symbol. The specific terms follow for the various definitions of the local frame

- Z-then-X

$$\frac{\partial u^\gamma}{\partial r_i^\beta} = \delta_{\gamma\beta},\ \frac{\partial u^\gamma}{\partial r_{iz}^\beta} = -\delta_{\gamma\beta}$$

$$\frac{\partial v^\gamma}{\partial r_i^\beta} = \delta_{\gamma\beta},\ \frac{\partial v^\gamma}{\partial r_{ix}^\beta} = -\delta_{\gamma\beta}$$

- Bisector

$$\frac{\partial u^\gamma}{\partial r_i^\beta} = (\xi + \eta)\delta_{\gamma\beta} + \frac{\eta^\gamma\xi^\beta}{\xi} + \frac{\xi^\gamma\eta^\beta}{\eta}$$

$$\frac{\partial u^\gamma}{\partial r_{iz}^\beta} = -\eta\delta_{\gamma\beta} - \frac{\eta^\gamma\xi^\beta}{\xi}$$

$$\frac{\partial u^\gamma}{\partial r_{ix}^\beta} = -\xi\delta_{\gamma\beta} - \frac{\xi^\gamma\eta^\beta}{\eta}$$

dx.doi.org/10.1021/ct401096t | J. Chem. Theory Comput. 2014, 10, 1638−1651

$$\frac{\partial v^{\gamma}}{\partial r_{i}^{\beta}} = \delta_{\gamma\beta}, \frac{\partial v^{\gamma}}{\partial r_{ix}^{\beta}} = -\delta_{\gamma\beta}$$

- Z-bisect

$$\frac{\partial u^{\gamma}}{\partial r_{i}^{\beta}} = \delta_{\gamma\beta}, \frac{\partial u^{\gamma}}{\partial r_{iz}^{\beta}} = -\delta_{\gamma\beta}$$

$$\frac{\partial v^{\gamma}}{\partial r_{i}^{\beta}} = (\eta + \zeta)\delta_{\gamma\beta} + \frac{\eta^{\gamma}\zeta^{\beta}}{\zeta} + \frac{\zeta^{\gamma}\eta^{\beta}}{\eta}$$

$$\frac{\partial v^{\gamma}}{\partial r_{ix}^{\beta}} = -\zeta\delta_{\gamma\beta} - \frac{\zeta^{\gamma}\eta^{\beta}}{\eta}, \frac{\partial v^{\gamma}}{\partial r_{iy}^{\beta}} = -\eta\delta_{\gamma\beta} - \frac{\eta^{\gamma}\zeta^{\beta}}{\zeta}$$

- 3-Fold

$$\frac{\partial u^{\gamma}}{\partial r_{i}^{\beta}} = \eta\zeta\delta_{\gamma\beta} + \frac{\zeta}{\xi}\eta^{\gamma}\xi^{\beta} + \frac{\eta}{\xi}\zeta^{\gamma}\xi^{\beta} + \xi\zeta\delta_{\gamma\beta} + \frac{\zeta}{\eta}\xi^{\gamma}\eta^{\beta} + \frac{\xi}{\eta}\zeta^{\gamma}\eta^{\beta} + \xi\eta\delta_{\gamma\beta} + \frac{\eta}{\zeta}\xi^{\gamma}\zeta^{\beta} + \frac{\xi}{\zeta}\eta^{\gamma}\zeta^{\beta}$$

$$\frac{\partial u^{\gamma}}{\partial r_{iz}^{\beta}} = \eta\zeta\delta_{\gamma\beta} + \frac{\zeta}{\xi}\eta^{\gamma}\xi^{\beta} + \frac{\eta}{\xi}\zeta^{\gamma}\xi^{\beta}$$

$$\frac{\partial u^{\gamma}}{\partial r_{ix}^{\beta}} = \xi\zeta\delta_{\gamma\beta} + \frac{\zeta}{\eta}\xi^{\gamma}\eta^{\beta} + \frac{\xi}{\eta}\zeta^{\gamma}\eta^{\beta}$$

$$\frac{\partial u^{\gamma}}{\partial r_{iy}^{\beta}} = \xi\eta\delta_{\gamma\beta} + \frac{\eta}{\zeta}\xi^{\gamma}\zeta^{\beta} + \frac{\xi}{\zeta}\eta^{\gamma}\zeta^{\beta}$$

$$\frac{\partial v^{\gamma}}{\partial r_{i}^{\beta}} = \delta_{\gamma\beta}, \frac{\partial v^{\gamma}}{\partial r_{ix}^{\beta}} = -\delta_{\gamma\beta}$$

■ ASSOCIATED CONTENT

Ⓢ Supporting Information

A documented Fortran subroutine to compute the rotation matrices and their derivatives and structures of the systems used for the tests. This material is available free of charge via the Internet at http://pubs.acs.org.

■ AUTHOR INFORMATION

Corresponding Authors

*E-mail: filippo.lipparini@courriel.upmc.fr (F.L.).
*E-mail: maday@ann.jussieu.fr (Y.M.).
*E-mail: jpp@lct.jussieu.fr (J.-P.P.).

Notes

The authors declare no competing financial interest.

■ ACKNOWLEDGMENTS

This work was supported in part by the ANR Manif and the French state funds managed by CALSIMLAB and the ANR within the Investissements d'Avenir program under reference ANR-11-IDEX-0004-02. This work was granted access to the HPC resources of The Institute for Scientific Computing and Simulation financed by the project Equip@Meso (reference ANR-10-EQPX-29-01) of the programme Investissements d'Avenir supervised by the ANR. Support from CNRS (PICS grant) is also acknowledged.

■ REFERENCES

(1) Jorgensen, W. L. *J. Chem. Theory Comput.* **2007**, *3*, 1877−1877 and references therein.
(2) Piquemal, J.-P.; Jordan, K. D. *Theor. Chem. Acc.* **2012**, *131*, 1−2.
(3) Gresh, N.; Cisneros, G. A.; Darden, T. A.; Piquemal, J.-P. J. *Chem. Theory Comput.* **2007**, *3*, 1960−1986.
(4) Warshel, A.; Kato, M.; Pisliakov, A. V. *J. Chem. Theory Comput.* **2007**, *3*, 2034−2045.
(5) Rick, S. W.; Stuart, S. J.; Berne, B. J. *J. Chem. Phys.* **1994**, *101*, 6141−6156.
(6) Rick, S. W.; Stuart, S. J.; Berne, B. J. *J. Chem. Phys.* **1994**, *101*, 6141−6156.
(7) Verstraelen, T.; Speybroeck, V. V.; Waroquier, M. *J. Chem. Phys.* **2009**, *131*, 044127.
(8) Lipparini, F.; Barone, V. *J. Chem. Theory Comput.* **2011**, *7*, 3711−3724.
(9) Piquemal, J.-P.; Chelli, R.; Procacci, P.; Gresh, N. *J. Phys. Chem. A* **2007**, *111*, 8170−8176.
(10) Lamoureux, G.; Roux, B. *J. Chem. Phys.* **2003**, *119*, 3025−3039.
(11) Boulanger, E.; Thiel, W. *J. Chem. Theory Comput.* **2012**, *8*, 4527−4538.
(12) Gordon, M. S.; Smith, Q. A.; Xu, P.; Slipchenko, L. V. *Annu. Rev. Phys. Chem.* **2013**, *64*, 553−578.
(13) Ponder, J. W.; Wu, C.; Ren, P.; Pande, V. S.; Chodera, J. D.; Schnieders, M. J.; Haque, I.; Mobley, D. L.; Lambrecht, D. S.; DiStasio, R. A.; Head-Gordon, M.; Clark, G. N. I.; Johnson, M. E.; Head-Gordon, T. *J. Phys. Chem. B* **2010**, *114*, 2549−2564.
(14) Ren, P.; Ponder, J. W. *J. Phys. Chem. B* **2003**, *107*, 5933−5947.
(15) Applequist, J.; Carl, J. R.; Fung, K.-K. *J. Am. Chem. Soc.* **1972**, *94*, 2952−2960.
(16) Thole, B. *Chem. Phys.* **1981**, *59*, 341−350.
(17) Sala, J.; Guàrdia, E.; Masia, M. *J. Chem. Phys.* **2010**, *133*, 234101.
(18) Wang, W.; Skeel, R. D. *J. Chem. Phys.* **2005**, *123*, 164107.
(19) Sagui, C.; Pedersen, L. G.; Darden, T. A. *J. Chem. Phys.* **2004**, *120*, 73−87.
(20) Lipparini, F.; Cappelli, C.; Barone, V. *J. Chem. Theory Comput.* **2012**, *8*, 153−4165.
(21) Lipparini, F.; Cappelli, C.; Scalmani, G.; De Mitri, N.; Barone, V. *J. Chem. Theory Comput.* **2012**, *8*, 4270−4278.
(22) Lipparini, F.; Cappelli, C.; Barone, V. *J. Chem. Phys.* **2013**, *138*, 234108.
(23) Curutchet, C.; Muñoz-Losa, A.; Monti, S.; Kongsted, J.; Scholes, G. D.; Mennucci, B. *J. Chem. Theory Comput.* **2009**, *5*, 1838−1848.
(24) Caprasecca, S.; Curutchet, C.; Mennucci, B. *J. Chem. Theory Comput.* **2012**, *8*, 4462−4473.
(25) Cisneros, G. A.; Piquemal, J.-P.; Darden, T. A. *J. Phys. Chem. B* **2006**, *110*, 13682−13684.
(26) Ponder, J. W. TINKER Molecular Modeling, Package V 6.2. http://dasher.wustl.edu/tinker/ (accessed July 14, 2013).
(27) Schnieders, M. J.; Fenn, T. D.; Pande, V. S. *J. Chem. Theory Comput.* **2011**, *7*, 1141−1156.
(28) Force Field X. http://ffx.eng.uiowa.edu/ (accessed November 14, 2013).
(29) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. *J. Chem. Phys.* **1995**, *103*, 8577−8593.
(30) Cisneros, G. A.; Piquemal, J.-P.; Darden, T. A. *J. Chem. Phys.* **2006**, *125*, 184101.
(31) Tomasi, J.; Mennucci, B.; Cammi, R. *Chem. Rev.* **2005**, *105*, 2999−3093.
(32) Cramer, C. J.; Truhlar, D. G. *Chem. Rev.* **1999**, *99*, 2161−2200.
(33) Still, W. C.; Tempczyk, A.; Hawley, R. C.; Hendrickson, T. *J. Am. Chem. Soc.* **1990**, *112*, 6127−6129.
(34) Brancato, G.; Rega, N.; Barone, V. *J. Chem. Phys.* **2006**, *124*, 214505.
(35) Rega, N.; Brancato, G.; Barone, V. *Chem. Phys. Lett.* **2006**, *422*, 367−371.
(36) Klamt, A.; Schuurmann, G. *J. Chem. Soc., Perkin Trans. 2* **1993**, 799−805.

(37) Cancès, E.; Maday, Y.; Stamm, B. *J. Chem. Phys.* **2013**, *139*, 054111.
(38) Lipparini, F.; Stamm, B.; Cancès, E.; Maday, Y.; Mennucci, B. *J. Chem. Theory Comput.* **2013**, *9*, 3637−3648.
(39) Lipparini, F.; Scalmani, G.; Mennucci, B.; Cancès, E.; Caricato, M.; Frisch, M. J. *J. Chem. Phys.* **2010**, *133*, 014106.
(40) Greengard, L.; Rokhlin, V. *J. Comput. Phys.* **1987**, *73*, 325−348.
(41) Cheng, H.; Greengard, L.; Rokhlin, V. *J. Comput. Phys.* **1999**, *155*, 468−498.
(42) Rohwedder, T.; Schneider, R. *J. Math. Chem.* **2011**, *49*, 1889−1914.
(43) Pulay, P. *Chem. Phys. Lett.* **1980**, *73*, 393−398.
(44) Pulay, P. *J. Comput. Chem.* **1982**, *3*, 556−560.
(45) Shi, Y.; Xia, Z.; Zhang, J.; Best, R.; Wu, C.; Ponder, J. W.; Ren, P. *J. Chem. Theory Comput.* **2013**, *9*, 4046−4063.
(46) Verlet, L. *Phys. Rev.* **1967**, *159*, 98−103.
(47) Swope, W. C.; Andersen, H. C.; Berens, P. H.; Wilson, K. R. *J. Chem. Phys.* **1982**, *76*, 637−649.
(48) Kolafa, J. *J. Comput. Chem.* **2004**, *25*, 335−342.
(49) Piquemal, J.-P.; Cisneros, G. A.; Reinhardt, P.; Gresh, N.; Darden, T. A. *J. Chem. Phys.* **2006**, *124*, 104101.