Truncated Conjugated Gradient (TCG): an optimal strategy for the analytical evaluation of the many-body polarization energy and forces in molecular simulations

Félix Aviat, Antoine Levitt, Benjamin Stamm, Yvon Maday, Pengyu Ren, Jay W. Ponder, Louis Lagardere, Jean-Philip Piquemal

Truncated Conjugated Gradient (TCG): an optimal strategy for the analytical evaluation of the many-body polarization energy and forces in molecular simulations

Félix Aviat,[†] Antoine Levitt,[‡] Benjamin Stamm,[¶,§] Yvon Maday,[‖,⊥,#] Pengyu Ren,[@] Jay Ponder,[△] Louis Lagardère,[*,∇,†] and Jean-Philip Piquemal[*,†,⊥]

[†]*UPMC Univ. Paris 06, UMR 7617, Laboratoire de Chimie Théorique, F-75005, Paris, France*

[‡]*Inria Paris, F-75589 Paris Cedex 12, Université Paris-Est, CERMICS (ENPC), F-77455 Marne-la-Vallée*

[¶]*MATHCCES, Department of Mathematics, RWTH Aachen University, Schinkelstr. 2, D-52062 Aachen, Germany*

[§]*Computational Biomedicine, Institute for Advanced Simulation IAS-5 and Institute of Neuroscience and Medicine INM-9, Forschungszentrum Jülich, Germany*

[‖]*UPMC Univ. Paris 06, UMR 7598, Laboratoire Jacques-Louis Lions, F-75005, Paris, France*

[⊥]*Institut Universitaire de France*

[#]*Brown Univ, Division of Applied Maths, Providence, RI, USA*

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

[△]*Washington University in Saint Louis, Chemistry, Campus Box 1134, One Brookings Drive, Saint Louis, MO 63130*

[∇]*UPMC Univ. Paris 06, Institut du Calcul et de la Simulation, F-75005, Paris, France*

E-mail: louis.lagardere@upmc.fr; jpp@lct.jussieu.fr

Abstract

We introduce a new class of methods, denoted as Truncated Conjugate Gradient (TCG) methods, to solve the many-body polarization energy and its associated forces in molecular simulations encountered in molecular dynamics (MD) and Monte-Carlo techniques. The method consists of a fixed number of Conjugate Gradient (CG) iterations. TCG approaches provide a scalable solution to the polarization problem at a user-chosen cost and a corresponding optimal accuracy and complexity. The optimality of the CG-method guarantees that the number of the required matrix-vector products are reduced to a minimum compared to other iterative methods. This family of methods is non empirical, fully adaptive and provides analytical gradients, avoiding therefore any energy drift in MD as compared to popular iterative solvers. Besides speed, one great advantage of this class of approximate methods is that their accuracy is systematically improvable. Indeed, as the CG-method is a Krylov subspace method, the associated error is monotonically reduced at each iteration. On top of that, two improvements can be proposed at virtually no cost: (i) the use of preconditioners can be employed, which leads to the Truncated Preconditioned Conjugate Gradient (TPCG); (ii) since the residual of the final step of the CG-method is available, one additional Picard fixed point iteration ("peek"), equivalent to one step of Jacobi Over Relaxation (JOR) with relaxation parameter ω, can be made at almost no cost. This method is denoted by TCG-n(ω). Black box adaptive methods to find ω are provided and discussed. Results show that TPCG-3(ω) is converged to high accuracy for various types of systems including proteins and highly charged systems at the fixed cost of 4 matrix-vector products: (3 CG iterations+the initial CG descent direction) whereas T(P)CG-2(ω) provides robust results at a reduced cost (3 matrix-vector products) and offers new perspectives for long polarizable MD as a production algorithm. The T(P)CG-1(ω) level provides less accurate solutions for inhomogeneous systems, but its applicability to well-conditioned problems such as water is remarkable, with only two matrix-vector product evaluations.

1 Introduction

In recent years, the development of polarizable force field has lead to new methodologies incorporating more physics. Therefore higher accuracy in the evaluation of energies can be achieved.[1] Indeed, the explicit inclusion of the many-body polarization energy offers a better treatment of intermolecular interactions, with immediate applications in various fields of application ranging from biomolecular simulations to material science. However, adding polarization to a force field is associated to a significant increase of the overall computational cost. In that context, various strategies have been introduced, including Drude oscillators,[2] fluctuating charges,[3] Kriging methods[4] and induced dipoles.[1,5] Among them, the induced dipole approach has been shown to provide a good balance between accuracy and computational efficiency, and can be implemented in a scalable fashion.[6]

One issue with this approach is the mandatory resolution of a set of linear equations whose size depends on the number of atoms (or polarizable sites). In practice, for the large systems of interest of force fields methods, a direct matrix inversion approach using the LU or Cholesky decomposition is not computationally feasible because of its cubic cost in the number of atoms. Luckily, iterative methods provide a remedy. We showed in a recent paper[6,7] that techniques such as the Preconditioned Conjugated Gradient (PCG) or the Jacobi/Direct Inversion of the Iterative Subspace (JI/DIIS) were efficient for large scale simulations as they offer the possibility of a massively parallel implementation coupled to fast summation techniques such as the Smooth Particle Mesh Ewald (SPME).[8] The overall cost is then directly proportional to the number of iterations necessary to achieve a good convergence . In that context, predictor-corrector strategies have been introduced to reduce this number using the information of the previous steps.[9,10] Extended Lagrangian inspired from efficient ab initio methods has also been introduced in order to limit the computational cost but require additional thermostats.[11] In practice, iterative methods are now the standard but suffer from energy conservation issues due to their non-analytical evaluation of the forces. Moreover, force fields are optimized to reach a precision for 10^{-1} to 10^{-2} kcal/mol in

the polarization energy. Such a precision can easily be reached using a convergence threshold of 10^{-3} to 10^{-4} Debye on the induced dipoles. However, when using iterative schemes, one needs to enforce the quality of the non-analytical forces in order to guarantee the energy conservation. Hence, a tighter convergence criterion of 10^{-5} to 10^{-7} Debye must be used. This leads to a very significant increase of the number of iterations. Overall, this additional computational cost is not linked to the physical requirement on the polarization energy but only ensures the numerical stability of the MD scheme. In that context, in their 2005 seminal paper[12] (see also ref. 13), Wang and Skeel postulated that another strategy would be possible if one could offer a method allowing analytical derivatives and therefore avoiding by construction the risk of loss of energy conservation (i.e. the drift). Such a method would be associated to a fixed number of iterations and could extend the applicability of polarizable simulations. Wang explored such strategies based on modified Chebyshev polynomials but noticed that even if the intended analytical expression was obtained it offered little accuracy compared to fully converged iterated results. In that context, Simmonett *et al.*[14,15] recently proposed to revisit this assumption in the context of a perturbation approach evaluating an approximated polarization denoted as ExPT. They proposed a parametric equation offering analytical derivatives and a good accuracy for some class of systems. However, the parametric aspect of the approach limits its global applicability to some type of systems. The purpose of this paper is to introduce a non-empirical strategy based on the same goals: analytical derivatives in order to guaranty energy conservation, limited number of iterations and reasonable accuracy. We will first present the variational formulation of the polarization energy and the associated linear system. Then, we will look at the iterative methods that are commonly used to solve it and discuss how they can cause problems in molecular simulations. Following this, we will describe two classes of iterative methods, the fixed point methods and the Krylov methods, and see how one can compute the polarization energy and its associated forces analytically (therefore avoiding the energy drift mentioned above). Finally, we will show some numerical results and discuss the accuracy of the new methods.

2 Context and notations

In the context of forcefields, several techniques are used in order to take polarization into account. Everything that will be presented in this paper concerns the widely used induced dipoles model. In this model, each or some of the atomic sites are associated with a 3×3 polarizability tensor, so that induced dipoles appear on these sites because of the electric fields created by the permanent charge density and by the other induced dipoles.

2.1 Notations

In the rest of the paper, we will assume that the studied system consists of N atoms, each of them bearing a 3×3 polarizability tensor α_i. We will denote by $\vec{E}_i$ the electric field created by the permanent density of charge on site i, and by $\vec{\mu}_i$ the induced dipole on site i. The $3N$ vectors collecting these vectors will respectively be noted $\mathbf{E}$ and $\boldsymbol{\mu}$. Furthermore, for $i \neq j$, we will denote by T_{ij} the 3×3 tensor representing the interaction between the i-th and the j-th induced dipole, so that $T_{ij}\vec{\mu}_j$ is the (possibly damped) electric field created by $\vec{\mu}_j$ on site i. We are now able to define by blocks the so-called Polarization matrix of the system:

$$\mathbf{T} = \begin{pmatrix} \alpha_1^{-1} & -T_{12} & -T_{13} & \dots & -T_{1N} \\ -T_{21} & \alpha_2^{-1} & -T_{23} & \dots & -T_{2N} \\ -T_{31} & -T_{32} & \ddots & & \\ \vdots & \vdots & & & \vdots \\ -T_{N1} & -T_{N2} & & \dots & \alpha_N^{-1} \end{pmatrix}$$

This matrix is clearly symmetric and we assume that it is also positive definite (thanks to the damping of the electric fields at short range) so that the energy functional defined below has a minimum and "the polarization catastrophe" [16] is avoided.

2.2 Variation formulation of the polarization energy and the associated linear system

Given these notation, one can express the polarization energy of the studied system in the context of an induced dipole polarizable force field as follows :

$$E_{\text{pol}} = \frac{1}{2}\boldsymbol{\mu}^T \mathbf{T}\boldsymbol{\mu} - \boldsymbol{\mu}^T \mathbf{E} \tag{1}$$

The minimizing dipoles $\boldsymbol{\mu}$ of E_{pol} are determined by the first optimality condition in form of the following linear system:

$$\mathbf{T}\boldsymbol{\mu} = \mathbf{E} \tag{2}$$

so that finally:

$$E_{\text{pol}} = -\frac{1}{2}\boldsymbol{\mu}^T \mathbf{E} \tag{3}$$

for the minimizing dipoles $\boldsymbol{\mu}$. The linear system expressed above has to be solved at each time step of a MD trajectory, so that a computationally efficient technique has to be used to solve it. Two kinds of methods exist to solve a linear system, the direct ones and the iterative ones. The first approaches, such as the LU or Cholesky decomposition, yield exact results (up to round-off errors) but their computational cost grows proportionally to N^3 and their memory requirements proportionally to N^2, making them hardly usable for large systems of biological interest.

3 Iterative methods

By contrast, iterative techniques yield approximate results depending on a convergence criterion, but their computional cost is proportional to the number of iterations times the cost of one iteration (dominated by the cost of a matrix-vector product). This implies that the iterative techniques can be used in conjunction with an efficient matrix-vector multiplication

method such as the Smooth Particle Mesh Ewald or the Fast Multipoles[8,17] .

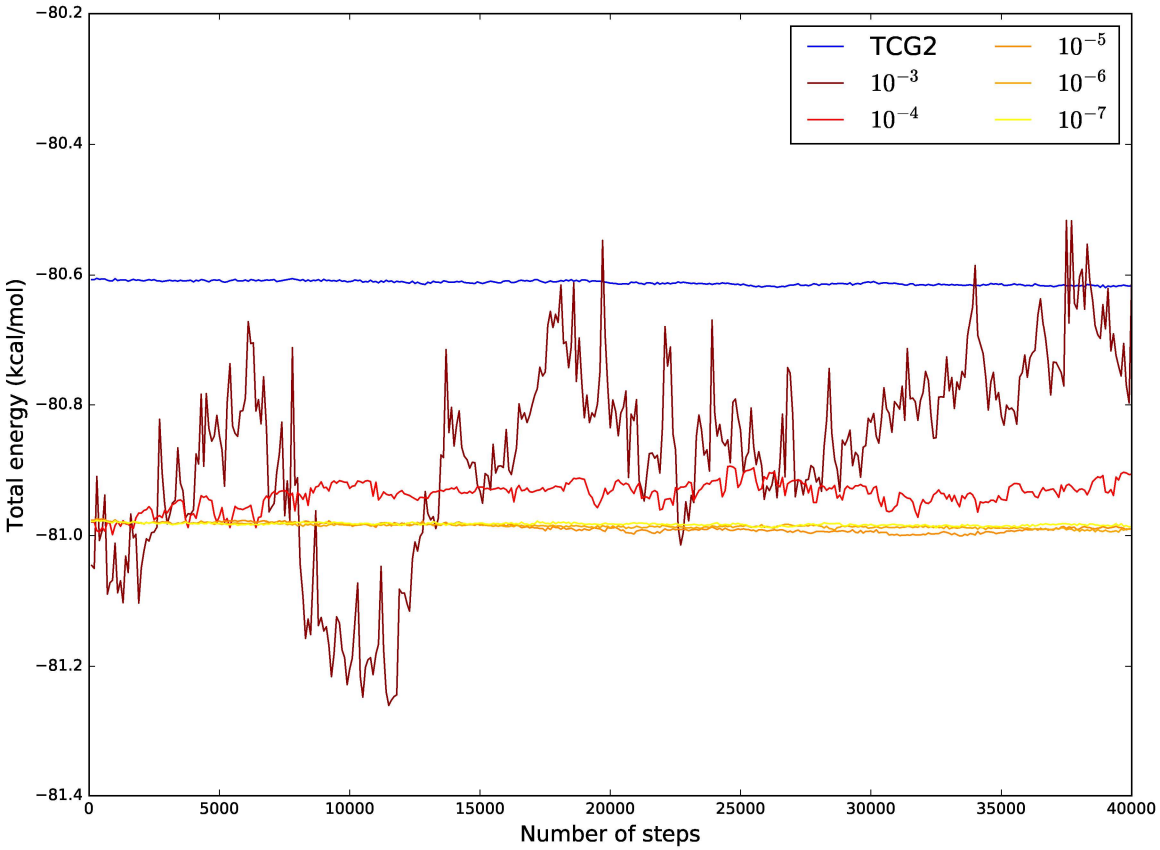
Several issues arise when using an iterative method to solve the polarization energy. The first one is related to the way the associated forces are computed. Indeed, the polarization energy is a function of the induced dipoles and of the atomic positions, so that one can rely on the chain rule to express the total derivative of this energy with respect to the atomic positions. The induced dipoles are then assumed to be completely minimizing E_{pol} so that $\frac{\partial E_{\text{pol}}}{\partial \boldsymbol{\mu}}$ is assumed to be zero, yielding the following expression (that is analogous to the Hellman-Feynman theorem in quantum mechanics):

$$\frac{dE_{\text{pol}}}{dr_i} = \frac{\partial E_{\text{pol}}}{\partial \boldsymbol{\mu}} \frac{\partial \boldsymbol{\mu}}{\partial r_i} + \frac{\partial E_{\text{pol}}}{\partial r_i} = \frac{\partial E_{\text{pol}}}{\partial r_i} \tag{4}$$

As the iterative method for the resolution of the induced dipoles is never perfectly converged, the previous assumption is never perfectly satisfied. Consequently, the forces calculated using this method are not exactly the negative of the first derivative of E_{pol} (eq. 3) with respect to the nuclear positions, potentially giving rise to an energy drift in a MD simulation. This is illustrated by the following graph (fig 1) representing the evolution of the total energy for a water box of 27 molecules, using the (diagonaly) Preconditioned Conjugate Gradient with different convergence threshold, namely $10^{-3}, 10^{-4}, 10^{-5}, 10^{-6}$ and 10^{-7}. An initial guess not issued from the past iterations was used, for a short MD simulation of 10 ps, using a time step of 0.25 fs. Such a small time step was used in order to minimize the numerical error coming from time-integration. One can directly observe the relation between the convergence threshold and the energy conservation.

The second issue is the computational cost of the iterative methods, proportional to the number of iterations times the cost of one iteration. Solving the polarization equations costs usually (depending on the settings of the simulation) more than 60% of the total cost of an MD step. It has already been shown that carefully choosing the iterative techique employed and taking an initial guess $\boldsymbol{\mu}_0$ using information from the past (by using predictor

Figure 1: Evolution of the total energy of a water box of 27 molecules computed with PCG and different convergence thresholds (AMOEBA), and with the TCG2 method.



guesses[9,10]) can yield an important reduction of the number of iterations required to reach a satisfactory convergence threshold. Nevertheless, some limitations exist due to the non perfect time reversibilty and volume preservation that they imply. Furthermore, the ability to parallelize the method efficiently also influences the choice of the optimal method[6,7] .

These issues motivate the derivation of a computationaly cheap analytical approximation of the polarization energy in polarizable MD. We aim also for such an approximation to be as close as possible to the exact results so that it would not require a reparametrization of the force fields. In order for the forces to be analytical, the computation of the induced dipoles have to be history free and should therefore avoid the use of predictors.

4 Fixed point methods and relation with ExPT

This first class of methods regroups the fixed point methods, also called stationary methods. In this case, one splits the matrix into two part in order to reexpress the solution of the linear system as a fixed point of a mapping associated to the splitting. For the polarization matrix one can reexpress $\mathbf{T}$ as the sum of its (block-)diagonal and off-diagonal part:

$$\mathbf{T} = \boldsymbol{\alpha}^{-1} - \mathcal{T} \tag{5}$$

yielding the following expression of the solution $\boldsymbol{\mu}$:

$$\boldsymbol{\mu} = \boldsymbol{\alpha}(\mathbf{E} + \mathcal{T}\boldsymbol{\mu}) \tag{6}$$

where $\boldsymbol{\mu}$ appears as the fixed point of a mapping. Then, Picard fixed point theorem[18] tells us that starting from some guess $\boldsymbol{\mu}_0$ and computing the following suite of dipoles (with $\mathbf{r}_n$ the residual associated with $\boldsymbol{\mu}_n$):

$$\boldsymbol{\mu}_{n+1} = \boldsymbol{\alpha}\mathbf{E} + \boldsymbol{\alpha}\mathcal{T}\boldsymbol{\mu}_n = \boldsymbol{\mu}_n + \boldsymbol{\alpha}\mathbf{r}_n \tag{7}$$

we converge towards the solution if $\rho(\boldsymbol{\alpha}\mathcal{T}) < 1$, with $\rho(\mathbf{M})$ the spectral radius of a given matrix $\mathbf{M}$. The method that was described above is indeed the Jacobi method and if we had split the matrix between its upper triangular part and the remaining terms, we would have obtained the Gauss-Seidel method.

A direct refinement of the Jacobi method consists in choosing a "relaxation" parameter ω and following the (relaxed) scheme:

$$\boldsymbol{\mu}_{n+1} = (1-\omega)\boldsymbol{\mu}_n + \omega(\boldsymbol{\mu}_n + \boldsymbol{\alpha}\mathbf{r}_n) = \boldsymbol{\mu}_n + \omega\boldsymbol{\alpha}\mathbf{r}_n \tag{8}$$

which is convergent if $\rho(I_d - \omega\boldsymbol{\alpha}\mathbf{T}) < 1$. In the rest of the text we will denote this method as JOR (Jacobi Over Relaxation)[19,20] .

One way to get analytical approximations of the polarization energy is to truncate one of these methods at a fixed order. One could for example choose to truncate the Jacobi method at some order n to obtain an analytical approximation to the solutions of the induced dipoles:

$$\boldsymbol{\mu}_n = \boldsymbol{\mu}_{(0)} + \boldsymbol{\mu}_{(1)} + ... + \boldsymbol{\mu}_{(n)} \tag{9}$$

with

$$\boldsymbol{\mu}_{(n)} = \boldsymbol{\alpha}(\mathcal{T}\boldsymbol{\alpha})^n\mathbf{E} \tag{10}$$

which is exactly the formulation of the perturbation theory (PT) method proposed by Simmonett *et al.*[14] , even if they follow another reasoning related to perturbation theory. The ExPT method they propose is then made by truncating this expansion at order two and by using a linear combination of $\boldsymbol{\mu}_1$ and $\boldsymbol{\mu}_3$:

$$\boldsymbol{\mu}_{\mathrm{ExPT}} = c_0\boldsymbol{\mu}_0 + c_1\boldsymbol{\mu}_3 \tag{11}$$

in order to provide the following expression for the approximation of the polarization energy:

$$E_{\text{pol,ExPT}} = -\frac{1}{2}\boldsymbol{\mu}_{\text{ExPT}}^{T}\mathbf{E} \tag{12}$$

The computational cost of this method is then equivalent to making three matrix-vector multiplication and its accuracy is good in many cases but it has the disadvantage of using two parameters that need to be fitted. Simmonett and coworkers recently extended the ExPT technique to higher-orders, giving the OPTn class of methods,[15] that lead to improved results but require more empirical parameters.

5 Krylov methods: Preconditioned Conjugate Gradient

The point of view of the Krylov methods is completely different[21] . It consists in minimizing at each iteration some energy functional over some growing subspaces.
Starting from some guess $\boldsymbol{\mu}_0$, let us define the associated residual:

$$\mathbf{r}_0 = \mathbf{E} - \mathbf{T}\boldsymbol{\mu}_0 \tag{13}$$

We are now able to define the so-called Krylov subspaces of order $p \in \mathbb{N}$:

$$K_p = span(\mathbf{r}_0, \mathbf{T}\mathbf{r}_0, ..., \mathbf{T}^{p-1}\mathbf{r}_0) \tag{14}$$

We clearly have the following hierarchical inclusion:

$$K_1 \subseteq K_2 \subseteq ... \tag{15}$$

Then μ_n is determined as the minimum of the energy functional over $\boldsymbol{\mu}_0 + K_p$. As one is minimizing at each iteration the energy functional over some increasing sequence of embedded spaces, the error (as measured by the functional) is necessarily decreasing. One can show that there exists a $p \leq 3N$ such that the exact solution $\boldsymbol{\mu}$ belongs to $\boldsymbol{\mu}_0 + K_p$, meaning that these methods always converge and even provide the exact solution after a finite number of steps, the worst case being when they converge in $3N$ iterations. In practice however, only very few iterations are needed to obtain accurate solutions.

The different Krylov methods are determined by the quantity that is minimized over the Krylov subspaces: if one minimizes E_{pol} then one obtains the conjugate gradient (given the assumption that $\mathbf{T}$ is symmetric definite positive), if ones minimizes $||\mathbf{r}_n||_{l^2}$ then one gets the GMRES method[21](which is equivalent to some version of the JI/DIIS[22]). But many others methods exist, such as the Minres method[23] or the BiCG method[21] for non symmetric matrices.

The conjugate gradient algorithm updates 3 vectors at each iteration: a descent direction, a dipole vector and the associated residual. These vectors are updated using 3 scalars that are obtained by making some scalar products over these vectors. After the following initialization (using here the direct field $\boldsymbol{\alpha}\mathbf{E}$ as a guess):

$$\begin{cases} \boldsymbol{\mu}_0 = \boldsymbol{\alpha}\mathbf{E} \\ \mathbf{r}_0 = \mathbf{E} - \mathbf{T}\boldsymbol{\mu}_0 \\ \mathbf{p}_0 = \mathbf{r}_0 \end{cases} \tag{16}$$

the algorithm reads as follows:

$$\begin{cases} \gamma_i = \frac{\mathbf{r}_i^{\mathbf{T}} \mathbf{r}_i}{{}^t\mathbf{p}_i \mathbf{T} \mathbf{p}_i} \\ \boldsymbol{\mu}_{i+1} = \boldsymbol{\mu}_i + \gamma_i \mathbf{p}_i \\ \mathbf{r}_{i+1} = \mathbf{r}_i - \gamma_i \mathbf{T} \mathbf{p}_i \\ \beta_{i+1} = \frac{\mathbf{r}_{i+1}^T \mathbf{r}_{i+1}}{\mathbf{r}_i^T \mathbf{r}_i} \\ \mathbf{p}_{i+1} = \mathbf{r}_{i+1} + \beta_{i+1} \mathbf{p}_i \end{cases} \tag{17}$$

where $\mathbf{p}_i$ is the descent direction at iteration i, $\boldsymbol{\mu}_i$ the associated dipole vector and $\mathbf{r}_i$ the associated residual. The magic of the conjugate gradient algorithm is that this simple recursion scheme still guarantees $\boldsymbol{\mu}_i$ to be the optimum over the entire Krylov-subspace of order i.

There are several techniques to accelerate the convergence of this algorithm. A widely used strategy is to use preconditioners. Indeed, one can show that the convergence rate of the conjugate gradient, and more generally of Krylov subspace methods, depends on the condition number of the matrix that is being inverted: the lower this number (it is always greater than one) , the faster the conjugate gradient will converge. In the case of symmetric positive definite (s. p. d.) matrices as the polarization matrix, this number can be expressed as:

$$\kappa(\mathbf{T}) = \frac{\lambda_{max}}{\lambda_{min}} \tag{18}$$

where λ_{max} and λ_{min} are the largest and smallest eigenvalues of the polarization matrix. A preconditioner is then a matrix P that is "close" to the inverse of $\mathbf{T}$, such that the matrix P is easily applied to a vector, $\kappa(P\mathbf{T}) \leq \kappa(\mathbf{T})$ and $\kappa(P\mathbf{T})$ is close to one. The conjugate gradient algorithm is then applied to the matrix $P\mathbf{T}$ with $P\mathbf{E}$ as a right hand side. We first chose to use one of the simplest form of preconditioner: the diagonal or Jacobi preconditioner, where P is equal to the inverse of the (block-)diagonal part of the polarization matrix. The

advantage of this choice in our context is that multiplying a matrix by a diagonal matrix is almost computationnaly free and that the diagonal of $\mathbf{T}$ does not depend on the positions of the atoms of the system studied. As a consequence, this choice does not complicate the expression of the gradients much. On the down side, the diagonal of $\mathbf{T}$ is of course not a perfect approximation of it, so that we don't expect the acceleration of convergence to be the highest among the possible choices of preconditioners. This is why we also considered a more efficient preconditioner designed for the polarization problem which we will present below. Wang and Skeel[12] start from the expression:

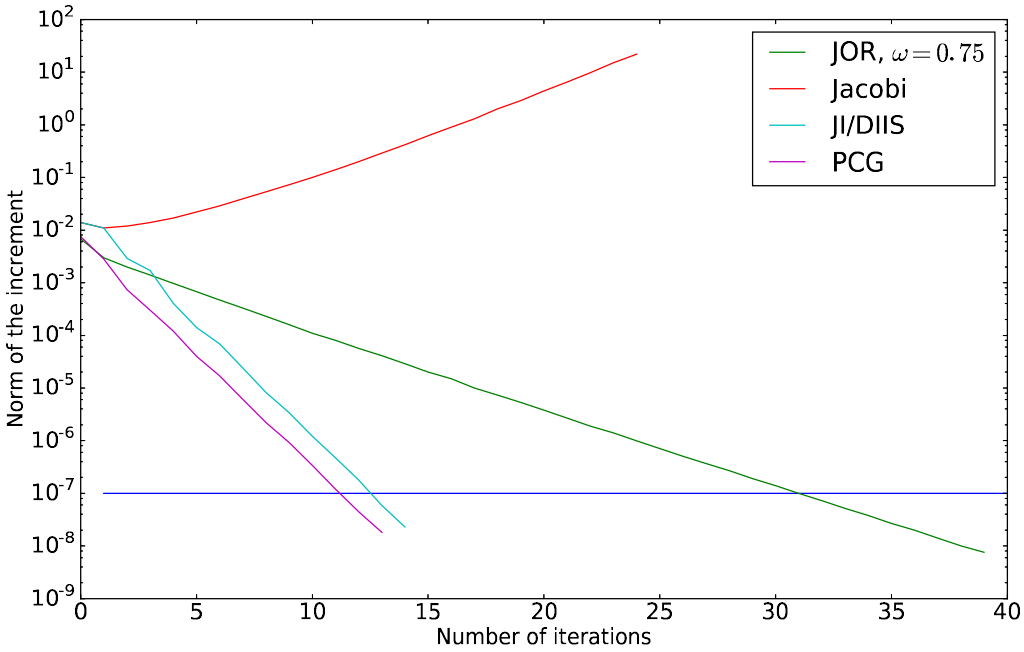
$$\mathbf{T}^{-1} = \boldsymbol{\alpha}(I_d - \boldsymbol{\alpha}\mathcal{T})^{-1} \tag{19}$$

giving the first approximation:

$$\mathbf{T}^{-1} \approx \boldsymbol{\alpha}(I_d + \boldsymbol{\alpha}\mathcal{T}) \tag{20}$$

which is in fact equivalent to one Jacobi iteration. A second approximation is then made by only considering the interactions within a certain cutoff in the matrix $\mathcal{T}$. A typical value for this cutoff is 3 Angstroms, value that we also used for our numerical tests presented below. This preconditioner has a bigger impact on reducing the condition number of the polarization matrix than the Jacobi one but it also has a higher computation cost per iteration. For the value we chose this cost is typically a bit less than half a real space matrix-vector product within a Particle Mesh Ewald simulation with usual settings (7 Angstroms cutoff). The parallel implementation of this preconditioner would require additional communications before and after the application of this preconditioner[6] . Finally, as it depends on the atoms positions, the expression of the gradients of the associated dipoles would be very involved.
To illustrate the different rates of convergence of these iterative methods we plotted in figure 2 their convergence as well as the one of JI/DIIS wich is described in ref. 7 (represented by the norm of the increment) as a function of the number of iterations in the context of the AMOEBA polarizable force field for the Ubiquitin protein in water. Note that the Jacobi

Figure 2: Norm of the increment as a function of the number of iterations for different iterative methods (AMOEBA), computed on ubiquitin.



iterations are not convergent in this case and that both the JI/DIIS and the Preconditioned Conjugate Gradient converge twice as fast as the JOR.
We will now explain how to truncate the preconditioned conjugate gradient to get analytical expressions that approximate the polarization energy.

6 Truncated Preconditioned Conjugate Gradient

Let us define $\boldsymbol{\mu}_{\mathrm{TCGn}}$, the approximation of the induced dipoles obtained by truncating the conjugate gradient at order n. We immediately have the result that $E_{\mathrm{pol}}(\boldsymbol{\mu}) \leq E_{\mathrm{pol}}(\boldsymbol{\mu}_{\mathrm{TCGn}}) \leq E_{\mathrm{pol}}(\boldsymbol{\mu}_{\mathrm{TCGm}})$ if $n \geq m$, with E_{pol} written as in equation 1 , and $\boldsymbol{\mu}$ the exact solution of the linear system. In other words, the quality of the approximation is systematically improvable. One can then unfold the algorithm at order one and two . Using the following notations:

$$\begin{aligned}
n_0 &= \mathbf{r}_0^T \mathbf{r}_0 \\
\mathbf{P}_1 &= \mathbf{T}\mathbf{r}_0 \\
t_1 &= \mathbf{r}_0^T \mathbf{P}_1 \\
t_2 &= \frac{n_0 ||\mathbf{P}_1||^2}{t_1^2} \\
\mathbf{P}_2 &= \mathbf{T}\mathbf{p}_1 \\
t_3 &= t_1 \mathbf{P}_1^T \mathbf{P}_2 \\
t_4 &= \frac{n_0}{t_1} \\
\gamma_1 &= \frac{t_1^2 - n_0 ||\mathbf{P}_1||^2}{t_3}
\end{aligned} \tag{21}$$

one obtains the cumbersome but analytical approximations for the dipoles corresponding to the conjugate gradient truncated at order one and two, thus allowing the derivation of

analytical forces that are the exact negative of the gradients of the energy:

$$\boldsymbol{\mu}_{\mathrm{TCG1}} = \boldsymbol{\mu}_0 + t_4\mathbf{r}_0 \tag{22}$$

$$\boldsymbol{\mu}_{\mathrm{TCG2}} = \boldsymbol{\mu}_0 + (t_4 + \gamma_1 t_2)\mathbf{r}_0 - \gamma_1 t_4 \mathbf{P}_1 \tag{23}$$

As in the ExPT approach, one can take the following expression as approximation of the polarization energy:

$$E_{\mathrm{pol,TCGn}} = -\frac{1}{2}\boldsymbol{\mu}_{\mathrm{TCGn}}^T\mathbf{E} \tag{24}$$

Note that both these expressions would be exact if the dipole vectors were exact and that the closer these vectors are to the fully converged dipoles, the closer these energies will be to the actual polarization energy. These energies are not the expression minimized over the Krylov subspaces at each iteration of the conjugate gradient (CG) algorithm (see equation 1), but they coincide at convergence which should almost be the case if our method is accurate.
These results are naturally extended to the preconditioned conjugate gradient (PCG).
One can of course also choose to truncate the (P)CG at a superior order and still be analytical to obtain a more accurate approximation, at the price however of additional computational time, and the analytical expression of the energy and its derivatives will be incrementely more complex, thus cumbersome to implement. In the following section, where numerical results are presented, we will limit ourselves to TCG3 as the highest order of truncation.
Moreover, having chosen an order of truncation of the (P)CG, one can exploit the residual (if it is computed) of the last iteration of the truncated algorithm in order to get closer to the converged value by computing one step of a fixed point iterative method. As Wang and Skeel,[12] we will call this operation a peek step. Indeed, many fixed point iterative methods as the Jacobi and more generally the Jacobi Over Relaxation (JOR) only require to know a starting value of the dipoles and the associated residual to be applied at each iteration. Note that the Jacobi method can be seen as a particular case of the JOR method with $\omega = 1$ and that this operation is not computationally expensive, as it only requires a matrix-vector

product with a diagonal matrix if the residual is known. As for any fixed-point method of a linear system, the asymptotic convergence of the JOR method depends on the spectral radius of the iteration matrix. More precisely, the condition:

$$\rho(I_d - \omega\boldsymbol{\alpha}\mathbf{T}) < 1 \tag{25}$$

guarantees that the JOR method is convergent. Asymptotically, the best convergence rate is obtained with the value of ω that minimizes this spectral radius. One can show that if $\mathbf{T}$ is symmetric positive definite, this value is:

$$\omega_{opt} = \frac{2}{\lambda_{min} + \lambda_{max}} \tag{26}$$

λ_{min} and λ_{max} being respectively the smallest and largest eigenvalue of $\boldsymbol{\alpha}\mathbf{T}$.

As these results are asymptotic, one can not necessarily expect the associated methods to give the best results if only the so-called peek step is applied, as this depends on the composition of the current approximation (which is in our case provided by the T(P)CG) in the eigenvector-basis of $\mathbf{T}$.

Since we can not rely on asymptotic results for one iteration, we also explored another strategy that can be of use in cases where one is particularly interested in the values of the energies, as for example in Monte-Carlo simulations for example. The ω_{opt} based on the spectrum intends to further optimize all the modes of the polarization matrix after the (P)CG steps (independently of the actual approximation) and should therefore asymptotically improve both the energy and the RMS. However, other values of ω that take into account the actual approximation can be used to further improve the accuracy. This explains why we tried, starting from one or two iteration of (P)CG, to choose the value of ω that gave the closest approximate polarization energy to the fully converged one. Since the optimal parameter (in this new sense) requires another matrix-vector multiplication, we tried to obtain values of this parameter on the fly by fitting one or several energies against the energies obtained

with the fully converged dipoles or a superior truncation of (P)CG. It will be called ω_{fit}. Starting for example from $\boldsymbol{\mu}_{TCG2}$, and noting:

$$\boldsymbol{\mu}_{TCG2,peek} = \boldsymbol{\mu}_{TCG2} + \omega\boldsymbol{\alpha}\mathbf{r}_2 \tag{27}$$

one can see this procedure as a line search: given the starting point $\boldsymbol{\mu}_2$, one further tries to optimize the energies along the parametrized line $\boldsymbol{\mu}_2 + \omega\boldsymbol{\alpha}\mathbf{r}_2$ for $\omega \in \mathbb{R}$.
Once one of these method is chosen, analytical expressions of the associated forces can be naturally obtained, thus ensuring that the forces are (up to round off errors) the opposite of the exact gradients of the polarization energy, and thus avoiding energy drift. Gradients of the energies have been derived and are presented in a technical appendix at the end of the manuscript.

7 Numerical Results

7.1 Energy conservation of the T(P)CGn approaches

Figure 1 displays an important result: the TCGn methods ensure total energy conservation as they embody analytical evaluation of the forces. All further refinements discussed in section 6 lead to the same behaviour, incremently closer to the reference energy.

7.2 Stability of the spectrum

Before presenting the complete numerical tests, we analyze here the spectrum of the polarization matrix during a MD simulation. Indeed, as pointed out in the theory section, some refinements of the TCG rely on the extreme eigenvalues of $\mathbf{T}$ and $\boldsymbol{\alpha}\mathbf{T}$. We followed the evolution of these eigenvalues during 100 picoseconds of MD. Those tests were made with a home version of the Lanczos method since all the matrices we are interested in are symmetric. Indeed, one great advantage of the Lanczos method is that it reduces the compu-

tational cost compared to direct methods (such as the one available in the Lapack library). That way, if direct eigenvalue solvers force the user to compute the full spectrum (i.e all the eigenvalues), Lanczos method allows rapid access to the extreme eigenvalues by constructing a much smaller tridiagonal matrix whose spectrum is really close to the one of the original matrix, leading to almost identical extreme eigenvalues that can then be obtained in a few iterations. We observed that in all cases these values are stable over the 100 picoseconds of the MD trajectories as pointed out by Skeel. This can be seen for S3 and the ubiquitin system in figure 3 and 4. This result motivated our choice to compute ω_{opt} and ω_{fit} for the first geometry of our equilibrated systems and to keep this value for all the others geometries. Both our Lanczos home version and the energy fitting procedure are fast enough to be used on the fly while being negligeable over a 100 picosecond MD simulation. In our tests, the adaptive reevaluation of the ωs was nevertheless never required.

Figure 3: Evolution of the extreme eigenvalues of $\boldsymbol{\alpha}\mathbf{T}$ for S3 and ubiquitin.

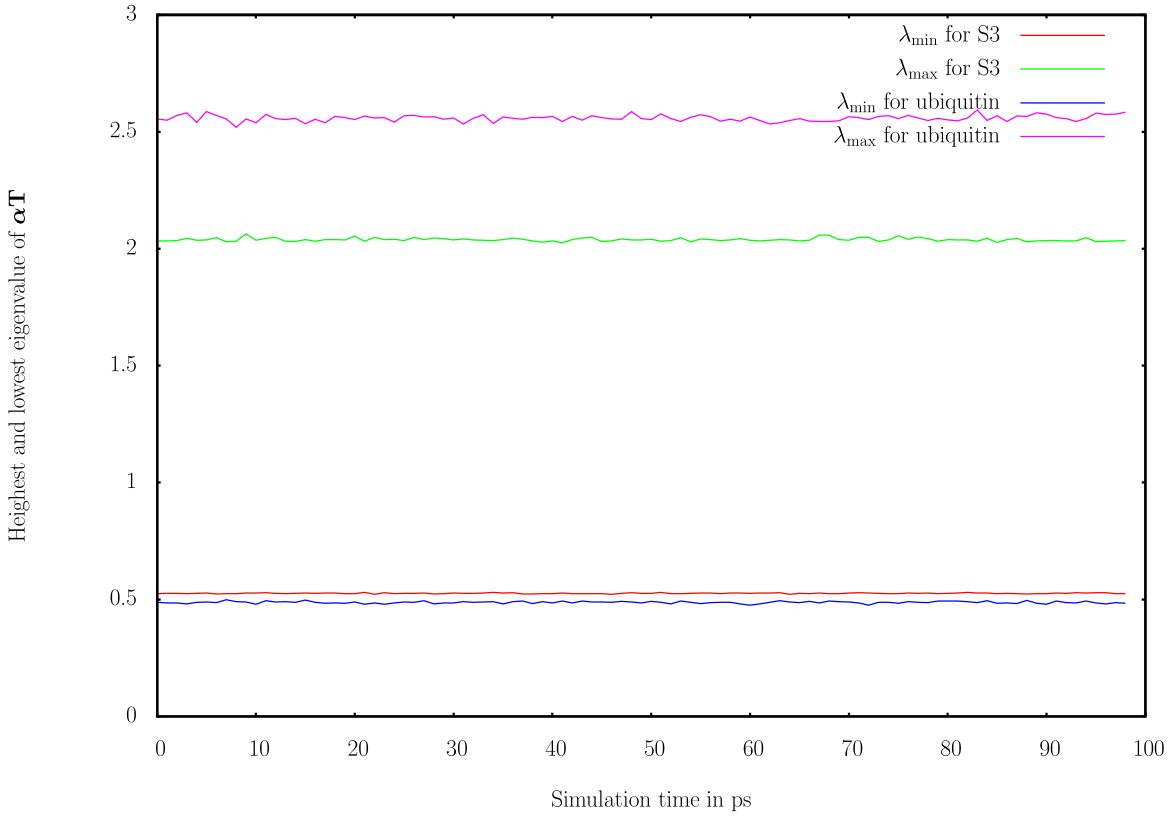
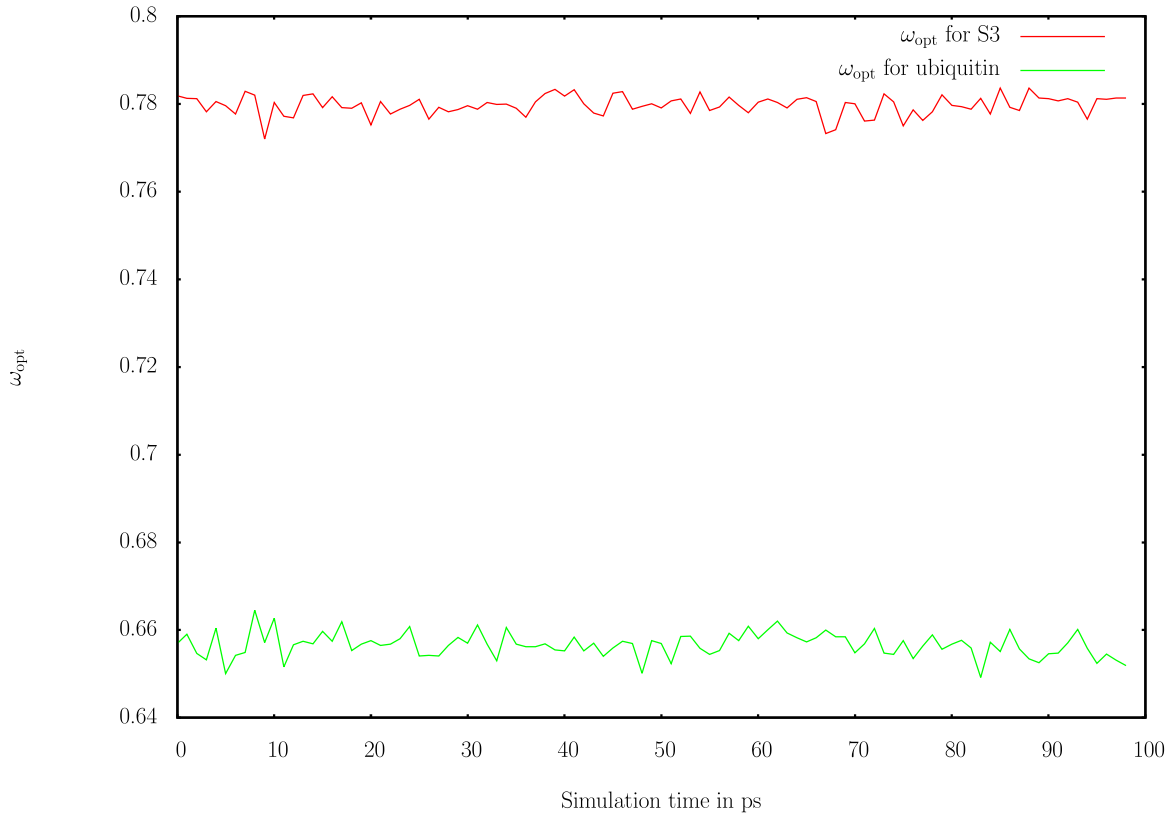


Figure 4: Evolution of ω_{opt} for S3 and ubiquitin.



7.3 Computational Details and Notations

In this section, we will present some numerical results about the methods presented above. All the tests presented here were made using the AMOEBA polarizable force field[24,25] implemented in our Tinker-HP code[26] . The proposed benchmarks deal with homogeneous and inhomogeneous systems: water clusters, protein in water droplets as well as an ionic liquid system. The three water systems will be called S1, S2 and S3 and contain respectively 27 molecules (81 atoms), 216 molecules (648 atoms) and 4000 molecules (12000 atoms). We chose difficult systems ranging from usual proteins to metalloproteins and highly charged ionic liquids[27] . The protein droplets are respectively: a metalloprotein containing two Zn(II) cations (nucleocapsid protein ncp7) with water (18515 atoms including counter ions), the

ubiquitin protein with water (9737 atoms) and the dhfr protein with water (23558 atoms). The ionic liquid system is made of [dmim+][Cl-] (1-3 dimethylimidazolium) ions, making it a highly charged system of 3672 atoms with a very different regime of polarization interactions. All the results presented in this section were averaged over 100 geometries that were extracted from a 100 picoseconds MD NVT trajectory (one geometry was saved every picosecond) at 300K for all systems, except the [dmim+][CL-] at 425K. The results, that will give indications about the accuracy of the approximate methods compared to the fully converged iterative results, will give two different and complementary aspects of this accuracy. We will first compare the polarization energies (in kcal/mol) obtained with dipoles converged with a very tight criterion (RMS of the residual under 10^{-9}) to the ones obtained with T(P)CG. We will then present the RMS of the difference between the fully converged dipole vectors and the approximate methods. This RMS is a good indicator of the quality of T(P)CG forces compared to the reference: the smaller this RMS is, the closer the approximated but analytical forces will be to the reference force.

Table 1 to 4 describe the water systems and table 5 to 8 describe the protein droplets and ionic liquid. We will denote by "Ref" the results obtained with dipoles converged up to 10^{-9} in the RMS of the residual; by ExPT the results obtained with the method of Simmonnet *et al.* presented in section 3; by TCGn (with n=1,2 and 3) the results obtained with the CG truncated at order 1,2 and 3; by TPCGn (P=diag) (with n=1,2 and 3) the results obtained with the preconditioned (with the diagonal) CG truncated at order 1,2 and 3; by TPCGn (P=Skeel) (with n=1,2 and 3) the results obtained with the preconditioned (using Wang and Skeel's preconditioner) CG truncated at order 1,2 and 3.

Furthermore, we will present results obtained with different kinds of peek steps. We will first denote by TCGn($\omega = 1$) (with n=1,2 and 3) the results obtained with the CG truncated at different orders when a Jacobi peek step is made after the last conjugate gradient iteration. We will also denote by TPCGn (P=diag) the results where the same peek step is made after different numbers of iterations of the PCG with a diagonal preconditioner.

We will also denote by TPCGn(P=diag)(ω_{opt}) (with n=1 and 2) the results obtained with 1 and 2 iterations of diagonally preconditioned CG and a JOR peek step with an "optimal" ω_{opt} in the sense of section 4.2 (that depend whether a preconditioner is used or not).
As explained in the previous section we also explored a strategy where the damping parameter of the JOR step is fitted to reproduce energy values. In the following tables, the damping parameter will be denoted by ω_{fit}.

7.4 Water Boxes, protein droplets and others

Table 1: Polarization Energies of water systems.

Water Box	S1	S2	S3
Ref	-81.03	-803.33	-15229.87
ExPT	-69.54	-660.95	-12822.79
TCG1	-73.50	-728.73	-13814.35
TCG2	-80.69	-800.32	-15173.15
TCG3	-81.24	-805.20	-15265.65
TPCG1 (P=diag)	-74.98	-741.91	-14028.18
TPCG2 (P=diag)	-80.81	-801.61	-15194.87
TPCG3 (P=diag)	-81.20	-805.26	-15268.43
TPCG1 (P=Skeel)	-78.63	-779.17	-14743.48
TPCG2 (P=Skeel)	-81.03	-803.11	-15222.53
TPCG3 (P=Skeel)	-81.06	-803.64	-15236.03

Table 2: Polarization Energies of water systems, using a peek-step.

Water Box	S1	S2	S3
Ref	-81.03	-803.33	-15229.87
TCG1($\omega = 1$)	-81.41	-806.83	-15315.13
TCG2($\omega = 1$)	-80.23	-794.49	-15061.22
TCG3($\omega = 1$)	-80.78	-800.83	-15181.55
TPCG1 (P=diag)($\omega = 1$)	-79.88	-791.51	-15001.40
TPCG2 (P=diag)($\omega = 1$)	-80.98	-802.74	-15218.27
TPCG3 (P=diag)($\omega = 1$)	-81.03	-803.27	-15228.74
TPCG1 (P=diag)(ω_{opt})	-78.98	-780.94	-14789.04
TPCG2 (P=diag)(ω_{opt})	-80.95	-802.50	-15213.17
TPCG1 (P=diag)(ω_{fit})	-81.06	-803.42	-15230.10
TPCG2 (P=diag)(ω_{fit})	-81.02	-803.06	-15231.14

Table 3: RMS of the dipole vector compared to the reference for water systems.

Water Box	S1	S2	S3
ExPT	1.4×10^{-2}	2.5×10^{-2}	2.6×10^{-2}
TCG1	6.3×10^{-3}	7.0×10^{-3}	7.1×10^{-3}
TCG2	1.7×10^{-3}	1.9×10^{-3}	1.9×10^{-3}
TCG3	4.7×10^{-4}	5.4×10^{-4}	5.5×10^{-4}
TPCG1 (P=diag)	4.9×10^{-3}	5.6×10^{-3}	5.8×10^{-3}
TPCG2 (P=diag)	9.2×10^{-4}	1.1×10^{-3}	1.1×10^{-3}
TPCG3 (P=diag)	3.8×10^{-4}	3.8×10^{-4}	3.9×10^{-4}
TPCG1(P=Skeel)	2.2×10^{-3}	2.6×10^{-3}	2.7×10^{-3}
TPCG2 (P=Skeel)	3.0×10^{-4}	3.9×10^{-4}	4.2×10^{-4}
TPCG3 (P=Skeel)	6.6×10^{-5}	9.5×10^{-4}	1.0×10^{-4}

Table 4: RMS of the dipole vector compared to the reference for water systems, using a peek-step.

Water Box	S1	S2	S3
TCG1($\omega = 1$)	3.6×10^{-3}	3.9×10^{-3}	3.7×10^{-3}
TCG2($\omega = 1$)	1.5×10^{-3}	1.7×10^{-3}	1.8×10^{-3}
TCG3($\omega = 1$)	4.6×10^{-4}	4.9×10^{-4}	4.8×10^{-4}
TPCG1(P=diag)($\omega = 1$)	2.2×10^{-3}	2.6×10^{-3}	2.7×10^{-3}
TPCG2 (P=diag)($\omega = 1$)	4.1×10^{-4}	5.0×10^{-4}	5.2×10^{-4}
TPCG3 (P=diag)($\omega = 1$)	1.3×10^{-4}	1.5×10^{-4}	1.6×10^{-4}
TPCG1 (P=diag)(ω_{opt})	2.3×10^{-3}	2.7×10^{-3}	2.8×10^{-3}
TPCG2 (P=diag)(ω_{opt})	3.9×10^{-4}	4.6×10^{-4}	4.7×10^{-4}
TPCG1 (P=diag)(ω_{fit})	2.6×10^{-3}	3.0×10^{-3}	3.0×10^{-3}
TPCG2 (P=diag)(ω_{fit})	5.3×10^{-4}	7.0×10^{-4}	1.0×10^{-3}

Table 5: Polarization Energies of protein droplet and ionic liquids.

System	ncp7	ubiquitin	dhfr	[dmim+][Cl-]
Ref	-24202.54	-11154.87	-28759.01	-1476.79
ExPT	-27362.70	-10919.77	-28076.62	-5841.73
TCG1	-21733.63	-9897.22	-25583.50	-1428.35
TCG2	-23922.79	-11031.67	-28463.51	-1420.00
TCG3	-24262.87	-11174.93	-28812.99	-1450.22
TPCG1 (P=diag)	-21438.14	-9907.09	-25588.07	-1465.66
TPCG2 (P=diag)	-23613.31	-10948.32	-28206.73	-1462.22
TPCG3 (P=diag)	-24219.49	-11164.62	-28775.53	-1469.89
TPCG1 (P=Skeel)	-22489.55	-10458.44	-27030.86	-1424.49
TPCG2 (P=Skeel)	-24056.53	-11090.36	-28637.35	-1469.05
TPCG3 (P=Skeel)	-24208.22	-11144.53	-28763.55	-1477.02

Table 6: Polarization Energies of protein droplet and ionic liquids, using a peek-step.

System	ncp7	ubiquitin	dhfr	[dmim+][Cl-]
Ref	-24202.54	-11154.87	-28759.01	-1476.79
TCG1($\omega = 1$)	-24481.14	-11231.35	-28986.08	-1477.08
TCG2($\omega = 1$)	-23965.96	-11009.06	-28384.49	-1465.73
TCG3($\omega = 1$)	-24121.02	-11105.78	-28635.73	-1441.95
TPCG1 (P=diag)($\omega = 1$)	-23532.73	-10829.84	-27972.41	-1493.58
TPCG2 (P=diag)($\omega = 1$)	-24123.65	-11128.14	-28683.52	-1471.34
TPCG3 (P=diag)($\omega = 1$)	-24194.37	-11150.95	-28749.68	-1478.83
TPCG1 (P=diag)(ω_{opt})	-22773.65	-10513.24	-27079.47	-1484.24
TPCG2 (P=diag)(ω_{opt})	-23938.70	-11066.44	-28504.96	-1468.29
TPCG1 (P=diag)(ω_{fit})	-24161.11	-11162.02	-28766.40	-1479.06
TPCG2 (P=diag)(ω_{fit})	-24205.30	-11154.21	-28753.60	-1475.08

Table 7: RMS of the dipole vector compared to the reference for protein droplets and ionic liquids.

Water Box	ncp7	ubiquitin	dhfr	[dmim+][Cl-]
ExPT	8.1×10^{-2}	5.2×10^{-2}	5.4×10^{-2}	1.3×10^{-1}
TCG1	8.9×10^{-3}	8.8×10^{-3}	8.8×10^{-3}	1.1×10^{-2}
TCG2	3.5×10^{-3}	3.2×10^{-3}	3.2×10^{-3}	7.2×10^{-3}
TCG3	2.1×10^{-3}	1.7×10^{-3}	1.7×10^{-3}	5.3×10^{-3}
TPCG1 (P=diag)	8.6×10^{-3}	8.0×10^{-3}	8.1×10^{-3}	6.9×10^{-3}
TPCG2 (P=diag)	2.5×10^{-3}	2.0×10^{-3}	2.2×10^{-3}	3.4×10^{-3}
TPCG3 (P=diag)	7.1×10^{-4}	6.5×10^{-4}	7.2×10^{-4}	7.9×10^{-4}
TPCG1 (P=Skeel)	5.5×10^{-3}	4.4×10^{-3}	4.5×10^{-3}	5.6×10^{-3}
TPCG2 (P=Skeel)	9.0×10^{-4}	7.7×10^{-4}	7.8×10^{-4}	1.5×10^{-3}
TPCG3 (P=Skeel)	2.1×10^{-4}	1.8×10^{-4}	1.9×10^{-4}	3.2×10^{-4}

Table 8: RMS of the dipole vector compared to the reference for protein droplets and ionic liquids, using a peek-step.

Water Box	ncp7	ubiquitin	dhfr	[dmim+][Cl-]
TCG1($\omega = 1$)	4.6×10^{-3}	4.4×10^{-3}	4.5×10^{-3}	7.0×10^{-3}
TCG2($\omega = 1$)	2.9×10^{-3}	2.5×10^{-3}	2.5×10^{-3}	5.5×10^{-3}
TCG3($\omega = 1$)	1.6×10^{-3}	1.1×10^{-3}	1.1×10^{-3}	4.1×10^{-3}
TPCG1 (P=diag)($\omega = 1$)	4.4×10^{-3}	3.9×10^{-3}	4.1×10^{-3}	3.2×10^{-3}
TPCG2 (P=diag)($\omega = 1$)	1.7×10^{-3}	1.4×10^{-3}	1.7×10^{-3}	1.6×10^{-3}
TPCG3 (P=diag)($\omega = 1$)	4.3×10^{-4}	3.8×10^{-4}	4.8×10^{-4}	4.5×10^{-4}
TPCG1 (P=diag)(ω_{opt})	5.1×10^{-3}	4.7×10^{-3}	4.8×10^{-3}	3.8×10^{-3}
TPCG2 (P=diag)(ω_{opt})	1.3×10^{-3}	1.0×10^{-3}	1.1×10^{-3}	1.9×10^{-3}
TPCG1 (Jacobi)(ω_{fit})	4.9×10^{-3}	4.5×10^{-3}	4.6×10^{-3}	4.5×10^{-3}
TPCG2 (Jacobi)(ω_{fit})	2.2×10^{-3}	1.7×10^{-3}	2.1×10^{-3}	2.0×10^{-3}

A first observation to make is that given a particular matrix (preconditioned or not) and with or whithout a JOR peek step, the results are always better in terms of energy and RMS when one performs more matrix-vector products, *i.e.* going to a higher order of truncation. This is naturally explained in the context of the Krylov methods: an additional matrix-vector product increases the dimension of the Krylov subspace on which the polarization functional (see equation 1) is minimized, and thus systematically improves the associated results. We should also recall here that the functional that is minimized over growing subspaces is not exactly the same as the one we are taking as the polarization energy and that this explains the non variationality of some of our results: there are many cases where the energy associated TCG3 is slightly lower than the one associated with the fully converged dipoles (see discussion on section 6).

We can also see on the numerical tests that using a preconditioner systematically reduces the associated RMS. Concerning the energy, the improvement is less systematic and depends on the type of preconditioner: the diagonal is less accurate than the one described by Wang *et al.*,[12] a result that was anticipated.

Nevertheless, preconditioning is important when coupled with a peek step: a combination

of any preconditioner with the peek is better than the peek alone. However, concerning the peek itself, one observes a systematic improvement of both RMS and energy with and without preconditioning. In particular this is the case when $\omega = 1$ (Jacobi peek step).

As the spectrum is stable (see section 7.2), one can use an adaptive ω_{opt} coefficient computed on one geometry using a few iterations of the Lanczos method. In that case, the energies are slighlty less accurate than the ones obtained with $\omega = 1$. Concerning the RMS, we observe a systematic reduction by a factor 2 for TPCG2 and TPCG3 but not for TPCG1. This is due to the fact that if the asymptotic coefficient ω_{opt} is the same, the starting point of the peek step is different and is significantely better for TPCG2 and TPCG3 as additional matrix-vector products have been computed.

The results obtained with ω_{fit} after 1,2 or 3 iterations of PCG show that it is possible to stay close to the converged value of the polarization energy with only one or two matrix-vector products and a ω parameter that is only fitted once during a 100 picoseconds dynamic . But we can also see that this is made at the cost of slightly degrading the RMS compared to the results obtained with ω_{opt} or with $\omega = 1$. Overall, these RMS are of the same order of magnitude than the ones obtained with ω_{opt} and $\omega = 1$. This balance between RMS and energy depending on the choice of ω as the relaxation parameter for a JOR peek step can be seen as the choice to favor the minimization of the error along some modes of the polarization matrix: the energy is more sensitive to modes corresponding to large eigenvalues whereas the RMS is sensitive to all of them. Overall, a $\omega = 1$ Jacobi peek step tends to improve both RMS and the energy whereas ω_{opt} favors RMS and ω_{fit} favors energies. As we showed, TPCG1 should not be used with a ω_{opt} peek step but with one corresponding to $\omega = 1$ and ω_{fit}, but all options are open for TPCG2 and TPCG3.

A choice can then be made depending on the simulation that one wants to run. For a Monte-Carlo simulation it is essential to have accurate energies : the strategy of using an adaptative parameter (refittable at a negligeable cost) that allows to reproduce correctly the energies with only one or two iterations of the (P)CG would hence produce excellent

results. On the other side, during a MD simulation, one wants to get the dynamic right; in this case, choosing the method that minimizes the RMS and thus the error made on the forces may produce improved results. For example, using TPCG2(P=diag)(ω_{opt}) is a good strategy to fulfill this purpose. However, the procedure leading to ω_{fit} only slightly degrades the RMS and provides RMS far beyond the usual values for which the force field models are parametrized. One has also to keep in mind that other source of errors exist in MD, such as the ones due to the PME discretization or van der Waals cutoffs, that are larger than the error discussed in this section. Nevertheless, none of the refinements will compete with a full additional matrix-vector product because an additional CG step is optimal. We see clearly that TPCG3(ω_{fit}) reaches high accuracy on both RMS and energies.

Concerning preconditioning, we confirm the very good behavior of the Skeel preconditioner. However, its cost is non negligible in terms of computations, in terms of necessary communications arising when running in parallel and in terms of complexity of implementation. We recommend therefore the use of the simpler yet efficient diagonal preconditioner. Overall, possibilities of tayloring TCG approaches are infinite. In practice, one could design more adapted preconditioners combining accuracy and low computational cost.

To conclude, a striking result is obtained for well conditioned systems such as water: computations show that they will require a smaller order of truncation than the proteins to obtain the same level of accuracy.

8 Conclusion

We proposed a general way to derive an analytical expression of the many-body polarization energy that approximates the inverse of $\mathbf{T}$ using a truncated preconditioned conjugated gradient approach. The general method gives analytical forces, guaranteeing they are the opposite of the exact gradients of the energies, parameter free, and can replace the usual many-body polarization solvers in popular codes with little effort. The proposed technique

allows by construction a true energy conservation as it embodies analytical derivatives. The method minimizes the energy over the (preconditioned) Krylov space which leads to superior accuracy than fixedpoint inspired method such as ExPT and associated methods. It is not using any history of the previous steps and is therefore fully time reversible and is compatible with multi-timestep integrators[28] . The best compromise between accuracy and speed appears to be the TPCG-2 approach that consists in 2 iterations of PCG with a computational cost of 3 matrix vector multiplications for the energy (one for the direction descent plus 2 for the iterations). The analytical derivatives have a cost equivalent to an additional matrix vector product. The overall computational cost is therefore identical to the ExPT's one. We showed that the method allows computing of potential energy surfaces very close to the exact ones and that it is systematically improvable using a final peek step. Strategies for adaptative JOR coefficients have been discussed and allows improving the desired quantities at a negligeable cost. Overall, among all the derived methods, TPCG3(ω_{fit}) provides high accuracy in both energy and RMS. Concerning the accuracy future improvements of the accuracy of the method, one could find dedicated preconditionners improving the efficiency of the CG steps. Nevertheless, the final choice of ingredients will be a tradeoff between accuracy, computational cost and communication cost when running in parallel. We will address this issue in the context of the Tinker-HP package. The TPCG-n approaches will be coupled to a domain decomposition infrastructure with linear scaling capabilities, thanks to a SPME[8] implementation, which is straightforward in link with our previous work on PCG. Future work will then include validation of the methods by comparing condensed-phase properties obtained using different orders of TCG. Given the level of accuracy already obtained on induced dipoles and energies, we expect the majority of these properties to be conserved by using T(P)CG2 and higher-order methods.

9 Technical Appendix

9.1 Analytical gradients and polarization energies for TCG

In this section, we will present the analytical derivatives of the polarization energies associated with the polarization energies $E_{\text{pol,TCG1}}$ and $E_{\text{pol,TCG2}}$ with respect to the positions of the atoms of the system. The extension to $E_{\text{pol,P(=diag)TCG1}}$ and $E_{\text{pol,(P=diag)TCG2}}$ is straightforward, as well as the expressions including a final JOR peek step. We don't report here the expression of the analytical gradients of $E_{\text{pol,TCG3}}$ as it follows the same logic but is just incrementely complex.

These gradients have been validated against the ones obtained with finite differences of the energies and an implementation of these equations will be accessible through the Tinker-HP software public distribution.

Since we are in the context of the AMOEBA force field, we will consider that each atom site embodies a permanent multipole expansion up to quadrupoles. For site i, the components of this expansion will be denoted by $q_i, \vec{\mu}_{p,i}, \theta_i$.

Furthermore, since the permanent dipoles and quadrupoles are expressed in a local frame that depends on the positions of neighboring atoms, they are rotated in the lab frame with rotation matrices depending on these positions, so that we now have to deal with partial derivatives of the dipole and quadrupole components : the "torques". Therefore, the derivative of the polarization energy ϵ, written as $\frac{1}{2}\boldsymbol{\mu}^T\mathbf{E}$ for $\boldsymbol{\mu} = \boldsymbol{\mu}_{TCG1}$ or $\boldsymbol{\mu}_{TCG2}$, with respect to the β-component of the k-th site is given by:

$$\frac{d\epsilon}{dr_k^\beta} = \frac{\partial\epsilon}{\partial r_k^\beta} + \sum_{i=1,N}\sum_{\alpha=1,3}\sum_{\gamma=1,3}\frac{\partial\epsilon}{\partial\theta_{p,i}^{\alpha,\gamma}}\frac{\partial\theta_{p,i}^{\alpha,\gamma}}{\partial r_k^\beta} + \sum_{i=1,N}\sum_{\alpha=1,3}\frac{\partial\epsilon}{\partial\mu_{p,i}^\alpha}\frac{\partial\mu_{p,i}^\alpha}{\partial r_{k^\beta}} \tag{28}$$

Formally, these derivatives can be written:

$$\epsilon' = -\frac{1}{2}(\boldsymbol{\mu}'^T\mathbf{E} + \boldsymbol{\mu}^T\mathbf{E}') \tag{29}$$

Hence different types of derivatives are involved :

- the derivatives of the rotated permanent multipoles;
- the derivatives of the permanent electric field with respect to the spatial components ;
- the derivatives of the permanent electric field with respect to the permanent multipoles;
- the derivatives of the induced dipole vector ($\boldsymbol{\mu}$) with respect to spatial components;
- the derivatives of the induced dipole vector with respect to the permanent multipole components.

As these quantities are standard except for the ones concerning the approximate dipole vector, these are the only one we will express here.

Using the same notation as before we have :

$$
\begin{aligned}
\mathbf{r}_0 &= \mathbf{E} - \mathbf{T}\boldsymbol{\mu}_0 \\
\mathbf{p}_0 &= \mathbf{r}_0 \\
n_0 &= \mathbf{r}_0^T\mathbf{r}_0 \\
\mathbf{P}_1 &= \mathbf{T}\mathbf{r}_0 \\
t_1 &= \mathbf{r}_0^T\mathbf{P}_1 \\
t_2 &= \frac{n_0||\mathbf{P}_1||^2}{t_1^2} \\
\mathbf{P}_2 &= \mathbf{T}\mathbf{p}_1 \\
t_3 &= t_1\mathbf{P}_1^T\mathbf{P}_2 \\
t_4 &= \frac{n_0}{t_1} \\
\gamma_1 &= \frac{t_1^2 - n_0||\mathbf{P}_1||^2}{t_3} \\
t_5 &= \mathbf{P}_1^T\mathbf{P}_2 \\
\beta_2 &= \frac{n_0 + t_4^2||\mathbf{P}_1||^2 + \gamma_1^2||\mathbf{P}_2||^2 - 2t_1t_4 - 2\gamma_1t_4||\mathbf{P}_1||^2 + 2\gamma_1t_4t_5}{(t_2-1)n_0} \\
\mathbf{P}_3 &= (1+\beta_2t_2)\mathbf{T}\mathbf{r}_0 - (t_4+\beta_2t_4)\mathbf{T}\mathbf{P}_1 - \gamma_1\mathbf{T}\mathbf{P}_2 \\
\gamma_2 &= \frac{n_0 + t_4^2||\mathbf{P}_1||^2 + \gamma_1^2||\mathbf{P}_2||^2 - 2t_1t_4 - 2\gamma_1t_4||\mathbf{P}_1||^2 + 2\gamma_1t_4t_5}{(1+\beta_2t_2)\mathbf{r}_0^T\mathbf{P}_3 - (t_4+\beta_2t_4)\mathbf{P}_1^T\mathbf{P}_3 + \gamma_1\mathbf{P}_2^T\mathbf{P}_3}
\end{aligned}
\tag{30}
$$

So that:

$$\boldsymbol{\mu}_{TCG1} = \boldsymbol{\mu}_0 + t_4\mathbf{r}_0 \tag{31}$$

$$\boldsymbol{\mu}_{TCG2} = \boldsymbol{\mu}_0 + (\gamma_1t_2 + t_4)\mathbf{r}_0 - \gamma_1t_4\mathbf{P}_1 \tag{32}$$

$$\boldsymbol{\mu}_{TCG3} = \boldsymbol{\mu}_0 + (t_4 + \gamma_1t_2 + \gamma_2 + \gamma_2\beta_2t_2)\mathbf{r}_0 - (\gamma_1t_4 + \gamma_2t_4 + \gamma_2\beta_2t_4)\mathbf{P}_1 - \gamma_1\gamma_2\mathbf{P}_2 \tag{33}$$

We then need to differentiate these expressions with respect to space and multipole compo-

nents respectively. Using the following formal development for the spatial derivative:

$$\begin{aligned}
\mathbf{r}'_0 &= \mathbf{E}' - \mathbf{T}'\boldsymbol{\mu}_0 - \mathbf{T}\boldsymbol{\mu}'_0 \\
n'_0 &= 2\mathbf{r}_0^T\mathbf{r}'_0 \\
\mathbf{P}'_1 &= \mathbf{T}'\mathbf{r}_0 + \mathbf{T}\mathbf{r}'_0 \\
(||\mathbf{P}_1||^2)' &= 2\mathbf{P}_1^T\mathbf{P}_1 \\
t'_1 &= \mathbf{r}_0^T\mathbf{P}'_1 + \mathbf{P}_1^T\mathbf{r}'_0 \\
t'_2 &= \frac{(n'_0||\mathbf{P}_1||^2 + n_0(||\mathbf{P}_1||^2)')t_1^2 - (n_0||\mathbf{P}_1||^2)2t_1t'_1}{t_1^4} \\
\mathbf{P}'_2 &= \mathbf{T}'\mathbf{P}_1 + \mathbf{T}\mathbf{P}'_1 \\
t'_3 &= t'_1\mathbf{P}_1^T\mathbf{P}_2 + t_1\mathbf{P}_2^T\mathbf{P}'_1 + t_1\mathbf{P}_1^T\mathbf{P}'_2 \\
t'_4 &= \frac{n'_0t_1 - n_0t'_1}{t_1^2} \\
\gamma'_1 &= \frac{1}{t_3^2}\left((2t_1t'_1 - n'_0||\mathbf{P}_1||^2 - n_0(||\mathbf{P}_1||^2)')t_3 - (t_1^2 - n_0||\mathbf{P}_1||^2)t'_3\right)
\end{aligned} \tag{34}$$

we obtain

$$\boldsymbol{\mu}'_{TCG1} = \boldsymbol{\mu}'_0 + t_4\mathbf{r}'_0 + t'_4\mathbf{r}_0 \tag{35}$$

$$\boldsymbol{\mu}'_{TCG2} = \boldsymbol{\mu}'_0 + (t_4 + \gamma_1t_2)\mathbf{r}'_0 + (t'_4 + \gamma'_1t_2 + \gamma_1t'_2)\mathbf{r}_0 + \gamma'_1t_4\mathbf{P}_1 + \gamma_1t'_4\mathbf{P}_1 + \gamma_1t_4\mathbf{P}'_1 \tag{36}$$

9.2 Acknowledgments

This work was supported in part by French state funds managed by CalSimLab and the ANR within the Investissements d'Avenir program under reference ANR-11-IDEX-0004-02. Funding from French CNRS through a PICS grant between UPMC and UT Austin is acknowledged. Jean-Philip Piquemal is grateful for support by the Direction Générale de

l'Armement (DGA) Maitrise NRBC of the French Ministry of Defense. JWP and PR thank support by National Institutes of Health (R01GM106137 and R01GM114237).

References

(1) Gresh, N.; Cisneros, G. A.; Darden, T. A.; Piquemal, J.-P. *J. Chem. Theory Comput.* **2007**, *3*, 1960–1986.

(2) Lopes, P. E.; Huang, J.; Shim, J.; Luo, Y.; Li, H.; Roux, B.; MacKerell Jr, A. D. *J. Chem. Theory Comput.* **2013**, *9*, 5430–5449.

(3) Rick, S. W.; Stuart, S. J.; Berne, B. J. *J. Chem. Phys.* **1994**, *101*, 6141–6156.

(4) Mills, M. J.; Popelier, P. L. *Comput. Theor. Chem.* **2011**, *975*, 42–51.

(5) Ren, P.; Ponder, J. W. *J. Phys. Chem. B* **2003**, *107*, 5933–5947.

(6) Lagardère, L.; Lipparini, F.; Polack, É.; Stamm, B.; Cancès, É.; Schnieders, M.; Ren, P.; Maday, Y.; Piquemal, J.-P. *J. Chem. Theory Comput.* **2014**, *11*, 2589–2599.

(7) Lipparini, F.; Lagardère, L.; Stamm, B.; Cancès, E.; Schnieders, M.; Ren, P.; Maday, Y.; Piquemal, J.-P. *J. Chem. Theory Comput.* **2014**, *10*, 1638–1651.

(8) Essmann, U.; Perera, L.; Berkowitz, M. L.; Darden, T.; Lee, H.; Pedersen, L. G. *J. Chem. Phys* **1995**, *103*, 8577–8593.

(9) Kolafa, J. *J. Comput. Chem.* **2004**, *25*, 335–342.

(10) Gear, C. W. *Commun. ACM* **1971**, *14*, 176–179.

(11) Albaugh, A.; Demerdash, O.; Head-Gordon, T. *J. Chem. Phys.* **2015**, *143*, 174104.

(12) Wang, W.; Skeel, R. D. *J. Chem. Phys.* **2005**, *123*, 164107.

(13) Wang, W. Fast Polarizable Force Field Computation in Biomolecular Simulations. Ph.D. thesis, University of Illinois at Urbana-Champaign, 2013.

(14) Simmonett, A. C.; Pickard IV, F. C.; Shao, Y.; Cheatham III, T. E.; Brooks, B. R. *J. Chem. Phys.* **2015**, *143*, 074115.

(15) Simmonett, A. C.; Pickard IV, F. C.; Ponder, J. W.; Brooks, B. R. *The Journal of Chemical Physics* **2016**, *145*, 164101.

(16) Thole, B. T. *J. Chem. Phys.* **1981**, *59*, 341–350.

(17) Cheng, H.; Greengard, L.; Rokhlin, V. *J. Comput. Phys.* **1999**, *155*, 468 – 498.

(18) Picard, E. *J. Math. Pures Appl.* **1890**, *6*, 145–210.

(19) Ryaben'kii, V. S.; Tsynkov, S. V. *A theoretical introduction to numerical analysis*; CRC Press, 2006.

(20) Quarteroni, A.; Sacco, R.; Saleri, F. *In Numerical mathematics*; Springer Science & Business Media, 2010; Vol. 37.

(21) Saad, Y. *In Iterative methods for sparse linear systems*; Siam, 2003.

(22) Rohwedder, T.; Schneider, R. *J. Math. Chem.* **2011**, *49*, 1889–1914.

(23) Paige, C. C.; Saunders, M. A. *SIAM journal on numerical analysis* **1975**, *12*, 617–629.

(24) 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.

(25) Shi, Y.; Xia, Z.; Zhang, J.; Best, R.; Wu, C.; Ponder, J. W.; Ren, P. *J. Chem. Theory Comput.* **2013**, *9*, 4046–4063.

(26) `http://www.ip2ct.upmc.fr/tinkerHP/`.

(27) Starovoytov, O. N.; Torabifard, H.; Cisneros, G. A. s. *J. Phys. Chem. B* **2014**, *118*, 7156–7166.

(28) Tuckerman, M.; Berne, B. J.; Martyna, G. J. *J. Chem. Phys.* **1992**, *97*, 1990–2001.