Theor Chem Acc (2004) 111:90–100
DOI 10.1007/s00214-003-0527-2

Theoretical
Chemistry Accounts

Regular article

Achieving linear-scaling computational cost for the polarizable continuum model of solvation

Giovanni Scalmani[1], Vincenzo Barone[1], Konstantin N. Kudin[2], Christian S. Pomelli[2], Gustavo E. Scuseria[2], Michael J. Frisch[3]

[1] Dipartimento di Chimica, Università di Napoli "Federico II", Complesso Universitario di Monte S. Angelo via Cintia, 80126 Naples, Italy
[2] Department of Chemistry and Center for Nanoscale Science and Technology, Mail Stop 60, Rice University, Houston, TX 77005-1892, USA
[3] Gaussian, Inc., 140 Washington Avenue North Haven, CT 06473, USA
Permanent address: C.S. Pomelli, Dipartimento di Chimica e Chimica Industriale, Università di Pisa, Via Risorgimento 35, 56100 Pisa, Italy

Received: 10 March 2003 / Accepted: 30 April 2003 / Published online: 3 February 2004

Abstract. This work describes a new and low-scaling implementation of the polarizable continuum model (PCM) for computing the self-consistent solvent reaction field. The PCM approach is both general and accurate. It is applicable in the framework of both quantum and classical calculations, and also to hybrid quantum/classical methods. In order to further extend the range of applicability of PCM we addressed the problem of its computational cost. The generation of the finite-elements molecular cavity has been reviewed and reimplemented, achieving linear scaling for systems containing up to 500 atoms. Linear scaling behavior has been achieved also for the iterative solution of the PCM equations, by exploiting the fast multipole method (FMM) for computing electrostatic interactions. Numerical results for large (both linear and globular) chemical systems are discussed.

Keywords: Continuum solvent model – Finite-elements molecular surface – Linear-scaling fast multipoles method

1 Introduction

In order to achieve quantitative accuracy in the computational simulation of molecules and chemical processes, the key importance of solvent effects is nowadays established [1, 2, 3, 4]. Solute–solvent interactions modify the energy, the structure, the properties and thus the overall behavior of molecules. Both in the case of classical simulations and in the case of quantum methods, an accurate model for the description of solvent effects should be part of any computational strategy.

Both discrete [5, 6, 7] and continuum [8, 9, 10] solvent models have been devised. Among them the polarizable continuum model (PCM) [11, 12] is characterized by a general and robust formalism, great flexibility and good accuracy. According to this model, the solvent is represented by a continuum dielectric medium within which a cavity is dug to host the solute molecule. The mutual polarization of the solute and the dielectric is computed and the solvent reaction field is then represented by a number of apparent solvation charges (ASCs) placed on the solute–solvent boundary (i.e. the cavity surface). The solution of the required electrostatic problem is achieved by a finite-elements approach: the cavity surface is partitioned in small tiles, the tesserae, where the ASCs are located.

The flexibility of the PCM lies in the fact that it can be easily cast both in the framework of classical calculations, such as molecular mechanics (MM) or molecular dynamics simulations, and in the case of quantum methods (Hartree–Fock, density functional theory, perturbative and variational correlated methods) [13]. Given this generality, the PCM has also been introduced to describe solvent effects in hybrid quantum/classical approaches and mixed methods, such as the own *N*-layered integrated molecular orbital method (ONIOM) and MM method.

Another important aspect of the PCM is that it can be used to reduce the complexity of a discrete description of the solvent. This is particularly useful when the solute establishes specific interactions with few solvent molecules and these interactions are responsible for a significant change in the solute properties. In such cases a discrete description of the solvent is unavoidable, but it can be limited to the few molecules specifically interacting with the solute, while the effect of the solvent

Contribution to the Jacopo Tomasi Honorary Issue

Correspondence to: G. Scalmani
e-mail: giovanni@lsdm.dichi.unina.it

"bulk" can be accounted for using the PCM approach [15]. The exact locations of the explicit solvent molecules around the solute can be found simply by energy minimization or can be extracted from a dynamics simulation.

From the algorithmic point of view, the PCM reaction field is computed from the solution of a linear system of equations of dimension equal to the number of finite elements (number of tesserae NTs) in which the cavity surface has been partitioned. The coefficient matrix of the linear system depends on the geometric features of the cavity (location, surface area and normal vector of all tesserae), the unknowns are the polarization charges (i.e. the ASCs) and the right-hand side involves either the total electrostatic potential or the electric field generated by the solute on the tesserae. The PCM linear system can be solved either by matrix inversion or iteratively. The cost of computing the PCM reaction field used to be usually negligible if compared to other steps in an ab-initio calculation (e.g. Fock matrix building and diagonalization). Nowadays this is no longer true.

On the one hand, in the field of ab initio methods, fast algorithms whose cost grows linearly with the system size (e.g. the fast multipole method (FMM)) have been developed and are beginning to be widely available and applied [16]. On the other hand, the significant reasearch effort being devoted to hybrid quantum/classical approaches [17] is driving the PCM towards the handling of large (partly) classical solutes. Note that, in this context, any solvent model has always to deal with the whole real system, which could easily involve thousands of atoms.

Given the methodological and algorithmic advances cited already, the cost of calculating the PCM reaction field could easily become the computational bottleneck. Indeed, whenever fast and linear scaling algorithms, such as the FMM, are used to compute the electrostatic potential/field on the tesserae, the cost of evaluating the interaction of the polarization charges among themselves will grow more steeply and will rapidly become dominant as the number of tesserae increases. Moreover, the definition of the solute–solvent boundary of the finite-elements (the cavity) can also represent a very costly task. In particular, in order to include the proper cavity deformation contribution into the energy gradient, such a boundary surface needs to be fully analytically defined and differentiable with respect to the atomic coordinates.

In conclusion, an efficient and low-scaling reformulation of the whole PCM procedure (beside the calculation of the solute electrostatic potential/field at the tesserae) has to be sought and implemented. This will be equally useful for classical and quantum solutes, as the cost of the cavity generation and of the solution of the PCM linear system no longer depends on the nature of the solute once its electrostatic potential/field has been computed. This paper describes such a new PCM implementation, which meets the efficiency requirements just described and that will be available in the upcoming release of the Gaussian package. In this contribution we consider only classical solutes since this allows us to scale the structures up to thousands of atoms to properly investigate the computational cost of the various steps in the procedure on a broad range of systems size. The nature of the solute–solvent boundary is described in Sect. 2, while the equations which define the PCM reaction fields for both classical and quantum solutes are described in Sect. 3. In Sect. 4 we describe in some detail the linear-scaling iterative solver we have developed. Finally, in Sect. 5 we use two sets of large chemical structures to asses the performances of the new cavity code and of the iterative solver.

2 Solute–solvent boundary

The definition of the solute–solvent boundary, i.e. the cavity surface, is one of the key steps in PCM calculations. The solute (a single molecule or a cluster containing some explicit solvent molecules) needs first to be represented by a set of interlocking spheres centered on atoms or atomic groups. Many different choices are available for the radii of such spheres. Nonetheless, it is important to select a consistent set of radii since the solvent effects computed on energies and properties depend critically on the size of the cavity.

For "all-atoms" representations we recommend the use of the universal force field (UFF) set of radii [18], mainly because of its generality (it is defined for the whole periodic system), although other choices are possible, such as Pauling's [19] or Bondi's [20] set of radii. On the other hand, for "united-atom" models, we have developed the so-called united atom topological model (UATM) [21] in which hydrogens are assimilated within the sphere of the heavy atom to which they are bound. The radii of the spheres are then computed according to a set of rules based on the atomic number, the hybridization, the formal charge of the atom and the nature of its first neighbours. The UATM model is particularly recommended for solvation energy calculations using ab initio methods.

Once the initial set of spheres is defined, the cavity surface is smoothed by adding spheres whose positions and radii are computed by a new version of the GePol algorithm [22, 23, 24]. The topological boundary of the final set of interlocking spheres is thus an accurate approximation of the so-called solvent excluding surface (SES) [25], but it is made only by convex patches. This is obviously one of the possible definitions of the solute–solvent boundary, but it has been proven to be the best choice for the calculation of the mutual solute–solvent-electrostatic interaction. Two other widely used definitions of molecular cavity are the van der Waals (VdW) surface, which is just the topological boundary of the initial set of interlocking spheres, and the solvent accessible surface (SAS) [26], which is defined as the VdW surface once all the radii of the initial spheres have been augmented by the radius of the sphere representing a solvent molecule. Both the VdW and the SA surfaces are relevant in the framework of the PCM since they are used to compute the nonelectrostatic contributions to the free energy of solvation, according to a well-established parameterization of such solute–solvent interactions.

The surface finite elements (tesserae) are then generated by inscribing a polyhedron with faces of suitable size and number within each sphere, projecting the faces onto the spherical surface and discarding those which are fully inside an intersecting sphere. If a polyhedron face happens to be partially inside a neighbouring sphere, the exposed portion is analytically cut by adding edges to the tessera being generated. All the tesserae (the finite elements) are thus defined in terms of connected arc segments, and their surface areas (the integration weights) can be analytically computed using the Gauss–Bonnet formula [23]. Moreover, this method of adding extra spheres to smooth the cavity surface, together with the definition of the tesserae in terms of connected arc segments, allows the calculation of the analytical derivatives, with respect to the atomic coordinates, of both the centroid and the surface area of all the tesserae. These are indeed required for a correct calculation of the solute energy gradient which involves a term related to the change in the molecular surface and thus in the reaction field.

Recently the algorithm for generating the whole set of spheres and tesserae and for computing their derivatives has been reviewed and extended [24]. The computer code which implements such an algorithm has also been rewritten from scratch. The aim of this effort was the development of a robust and efficient code suitable for applications to very large systems (up to tens of thousands of atoms and to hundreds of thousands of tesserae). In order to do that without prohibitively increasing the computational cost, we focused on the use and development of linear scaling procedures starting from the simple form of the GePol algorithm available in Gaussian98 [27], which showed cubic cost and limited robustness, already for solutes made by few tens of atoms.

The most important features of the new implementation of the cavity code are the following:

1. The position and the shape of all the tesserae are constrained to have the same symmetry properties of the solute. To do that, we devised a simple approach in which the polyhedra inscribed in the spheres are chosen and oriented so that the ensemble of their faces conforms to the molecular point group. This feature is particularly important in ab initio calculations to preserve the symmetry of the density matrix once the PCM correction is added to the solute Hamiltonian.
2. The whole algorithm has been generalized to handle periodic boundary conditions. Translational symmetry in one, two or three dimensions can be imposed and also the presence of screw axes or glide planes can be handled. These features are important for the study of the solvent effects in periodic systems like polymer chains or wet surfaces.
3. Given the initial set of interlocking spheres, the sphere's neighbor list (NL) should be obtained with linear-scaling computational cost. The NL lists, for each sphere, the nearby spheres that intersect it. The spheres NL is particularly important since it allows us to set an upper limit to the length of loops during the generation of the tesserae. The generation of the sphere's NL it is a well-known problem in computational geometry, whose trivial solution scales quadratically with the number of spheres, and in the new cavity code we implemented two efficient low-scaling algorithms to carry out this task.
4. An effective prescreening algorithm has been developed to reduce the number of polyhedron faces which must be examined during the tesserae generation. This approach allows us to discard very fastly all the faces that are either completely buried inside a sphere or completely exposed to the solvent.
5. The computational cost of the algorithm to create the extra spheres required by the SES, in its trivial implementation, increases cubically with the number of spheres; thus, a major effort has been devoted to reduce this scaling. Using various techniques to speed up the collision check that the candidate new sphere need to pass, we reduced this scaling behavior from cubic to weakly quadratic.

3 Reaction field

Within the PCM framework, the solvent reaction field due to an isotropic solvent is expressed in terms of a polarization charge density, $\sigma(\mathbf{s})$, spread on the solute–solvent boundary surface, which is the solution of the following integral equation:

$$\left(\frac{\epsilon+1}{\epsilon-1}-\frac{1}{2\pi}\hat{D}^*\right)\sigma(\mathbf{s})=-\frac{1}{2\pi}E_\perp(\mathbf{s}) \ , \qquad (1)$$

where ϵ is the solvent dielectric constant, $E_\perp(\mathbf{s})$ is the normal component of the electric field generated by the solute at the cavity surface and $\hat{D}^*$ is an operator that accounts for the electric field generated by σ itself. The equation is exact when all the solute charge distribution is strictly enclosed by the cavity, as in the case of a classical solute. On the other hand, when the solute is represented by a quantum mechanical method or, more generally, it is described by a mixed quantum/classical approach, one has to deal with the so-called outlying charge, which is the (small) fraction of the solute charge density which lies outside the cavity because of the tails of the electronic wavefunction.

Recently, the PCM formalism has been reviewed [12, 13] and it is now understood that the effect of the "outlying charge" can be taken into account implicitly, thus avoiding any a posteriori correction of the reaction field to fulfill Gauss' law. In particular, Eq. (1) is generalized to the following form

$$\left(\frac{\epsilon+1}{\epsilon-1}\hat{S}-\frac{1}{2\pi}\hat{S}\hat{D}^*\right)\sigma(\mathbf{s})=\left(-1+\frac{1}{2\pi}\hat{D}\right)V(\mathbf{s}) \ , \qquad (2)$$

where $V(\mathbf{s})$ is the total electrostatic potential generated by the solute at point $\mathbf{s}$ on the cavity surface and the operator $\hat{S}$ is related to the surface charge potential. Note that in this formalism the polarization charge density $\sigma(\mathbf{s})$ depends on the solute potential and not on the field as in Eq. (1).

A detailed derivation and description of the formalism have been already published [13] and they are thus omitted here. Also the finite-elements formulation of the method and the corresponding working equations are available elsewhere. However, in order to develop the iterative solution of the PCM problem in the next section, we need to recall

1. The definitions of the **S**, $\mathbf{D}^*$ and **D** matrices (which are the finite-elements representation of the $\hat{S}$, $\hat{D}^*$ and $\hat{D}$ operators):

$$\begin{cases} S_{ii} = 1.0694\sqrt{\dfrac{4\pi}{a_i}} \\ S_{ij} = \dfrac{1}{|s_i - s_j|} \end{cases} , \tag{3}$$

$$\begin{cases} D^*_{ii} = -\left(2\pi + \displaystyle\sum_{j\neq i} D^*_{ji} a_j\right)\dfrac{1}{a_i} \\ D^*_i j = -\dfrac{(\mathbf{s}_i - \mathbf{s}_j)\cdot \hat{n}_i}{|s_i - s_j|^3} \end{cases} , \tag{4}$$

$$\begin{cases} D_{ii} = -\left(2\pi + \displaystyle\sum_{j\neq i} D_{ij} a_j\right)\dfrac{1}{a_i} \\ D_{ij} = \dfrac{(\mathbf{s}_i - \mathbf{s}_j)\cdot \hat{n}_j}{|s_i - s_j|^3} \end{cases} , \tag{5}$$

where a_i is the area of the ith tessera, $\mathbf{s}_i$ is its position vector and $\hat{n}_i$ is the unit vector normal to the cavity surface at $\mathbf{s}_i$.

2. The matrix equations that correspond to Eq. (1), which hold for the special case of classical solutes:

$$\left(2\pi\frac{\epsilon+1}{\epsilon-1}\mathbf{A}^{-1} - \mathbf{D}^*\right)\mathbf{q} = -\mathbf{E}_\perp \ , \tag{6}$$

where **A** is a diagonal matrix collecting the tesserae surface areas, **q** is the polarization charges array and $\mathbf{E}_\perp$ collects the values of the solute normal electric field at the tesserae.

3. The matrix equations that correspond to Eq. (2), which hold in general for quantum and classical solutes:

$$\mathbf{Tq} = \mathbf{RV} \ , \tag{7}$$

where **q** is the polarization charges array, **V** collects the values of the solute potential at the tesserae and the **T** and **R** matrices are defined as follows

$$\mathbf{T} = \frac{\epsilon+1}{\epsilon-1}\mathbf{S} - \frac{1}{2\pi}\mathbf{DAS} \tag{8}$$

and

$$\mathbf{R} = -\mathbf{I} + \frac{1}{2\pi}\mathbf{DA} \tag{9}$$

using the previous definitions of the **A**, **S**, $\mathbf{D}^*$ and **D** matrices.

4 Iterative formulation

Before describing the iterative procedures required for the solution of the PCM equations, for both classical and quantum solutes, we define some useful quantities. These are not only used to cast the subsequent expressions in a more compact form, but also provide a clear way to identify the computational kernels which need to be carried out in order to obtain overall linear scaling behavior.

The first quantity is the vector function $\mathbf{y}[\mathbf{x}]$, whose elements are defined as

$$y_i[x_j] = \sum_{j\neq i} S_{ij} x_j \ , \tag{10}$$

where S_{ij} are the elements of the "potential" matrix defined in Eq. (3). We also define $\mathbf{z}^*[\mathbf{x}]$ and $\mathbf{z}[\mathbf{x}]$ as

$$z^*_i[x_j] = \sum_{j\neq i} D^*_{ij} x_j \tag{11}$$

and

$$z_i[x_j] = \sum_{j\neq i} D_{ij} x_j \ , \tag{12}$$

where D^*_{ij} and D_{ij} are the elements of the "field" (Eq. 4) and "normal field" (Eq. 5) matrices, respectively.

In the case of classical solutes, the polarization charges are obtained by solving system 6. Recalling the definition of the D^* and D matrices, the i-th equation of such system can be written as

$$\left[\frac{1}{a_i}\left(\frac{4\pi\epsilon}{\epsilon-1} + z_i[a_j]\right)\right] q_i - z^*_i[q_j] = -(\mathbf{E}_\perp)_i \ , \tag{13}$$

which is solved iteratively setting

$$q^{(n)}_i = \left[\frac{1}{a_i}\left(\frac{4\pi\epsilon}{\epsilon-1} + z_i[a_j]\right)\right]^{-1}\left[-(\mathbf{E}_\perp)_i + z^*_i[q^{(n-1)}_j]\right] \ , \tag{14}$$

where $q^{(n)}_i$ and $q^{(n-1)}_i$ are the charges at the nth and $(n-1)$th iteration, respectively.

When the solute charge distribution extends outside the cavity (typically in the case of ab initio calculations), one has to compute the polarization weights, **w**, as

$$\mathbf{w} = \frac{\mathbf{q} + \mathbf{q}^*}{2} \ , \tag{15}$$

where **q** are the polarization charges and $\mathbf{q}^*$ can be defined as "transposed" polarization charges. The **q** are the solutions of the linear systems of equations

$$\mathbf{Tq} = \mathbf{b} = \mathbf{RV} \ , \tag{16}$$

while the $\mathbf{q}^*$ can be obtained as

$$\mathbf{q}^* = (\mathbf{T}^{-1}\mathbf{R})^\dagger \mathbf{V} = \mathbf{R}^\dagger (\mathbf{T}^\dagger)^{-1}\mathbf{V} \ , \tag{17}$$

where, setting

$$\mathbf{c} = (\mathbf{R}^\dagger)^{-1}\mathbf{q}^* \tag{18}$$

and solving the linear system

$$\mathbf{T}^{\dagger}\mathbf{c} = \mathbf{V} \ , \tag{19}$$

one just need to carry out the matrix–vector product

$$\mathbf{q}^{*} = \mathbf{R}^{\dagger}\mathbf{c} \tag{20}$$

to obtain the $\mathbf{q}^{*}$.

In more detail, we first consider $\mathbf{b} = \mathbf{RV}$, which is the right-hand-side of system 16. Recalling the definition of the $\mathbf{R}$ matrix (Eq. 9), the ith element of $\mathbf{b}$ can be written as

$$\begin{aligned} b_i &= -V_i + \frac{1}{2\pi}\sum_j D_{ij}a_j V_j \\ &= -V_i\left(2 + \frac{1}{2\pi}z_i[a_j]\right) + \frac{1}{2\pi}z_i[a_j V_j] \end{aligned} \tag{21}$$

and the ith equation of system 16 is

$$\sum_j \left(\frac{\epsilon+1}{\epsilon-1}\right)S_{ij}q_j \ - \frac{1}{2\pi}\sum_{jk} D_{ik}a_k S_{kj}q_j = b_i \ , \tag{22}$$

which, by recalling the definition of the D matrix diagonal elements (Eq. 5) and performing some manipulations, becomes

$$\begin{aligned} &\left(f(\epsilon) + \frac{1}{2\pi}z_i[a_j]\right)(S_{ii}q_i + y_i[q_j]) \\ &\quad - \frac{1}{2\pi}z_i[a_j(S_{jj}q_j + y_j[q_k])] = b_i \ , \end{aligned} \tag{23}$$

where $f(\epsilon) = (\epsilon+1)/(\epsilon-1) + 1$. Equation (23) can be solved iteratively expressing the charge q_i at iteration n as

$$\begin{aligned} q_i^{(n)} = &\left[\left(f(\epsilon) + \frac{1}{2\pi}z_i[a_j]\right)S_{ii}\right]^{-1}\left[b_i - \left(f(\epsilon) + \frac{1}{2\pi}z_i[a_j]\right)\right. \\ &\left.\times y_i[q_j^{(n-1)}] + \frac{1}{2\pi}z_i[a_j(S_{jj}q_j^{(n-1)} + y_j[q_k^{(n-1)}])]\right] \ . \end{aligned} \tag{24}$$

On the other hand, the ith equation of system 19 is

$$\sum_j \left(\frac{\epsilon+1}{\epsilon-1}\right)S_{ij}c_j \ - \frac{1}{2\pi}\sum_{jk} S_{ik}a_k D^{*}_{kj}c_j = V_i \ , \tag{25}$$

which can be manipulated by inserting the definition of the D^{*} matrix diagonal elements (Eq. 4) and using the fact that D is the transpose of D^{*}, leading to the following expression

$$\begin{aligned} &\left(f(\epsilon) + \frac{1}{2\pi}z_i[a_j]\right)S_{ii}c_i \ + y_i\left[\left(f(\epsilon) + \frac{1}{2\pi}z_j[a_k]\right)c_j\right] \\ &\quad - \frac{1}{2\pi}\left(S_{ii}a_i z_i^{*}[c_j] + y_i[a_j z_j^{*}[c_k]]\right) \ . \end{aligned} \tag{26}$$

The iterative solution of these equations is achieved by expressing the c_i elements as

$$\begin{aligned} c_i^{(n)} = &\left[\left(f(\epsilon) + \frac{1}{2\pi}z_i[a_j]\right)S_{ii}\right]^{-1}\left[b_i - y_i\left[\left(f(\epsilon) + \frac{1}{2\pi}z_j[a_k]\right)\right.\right. \\ &\left.\left.\times c_j^{(n-1)}\right] + \frac{1}{2\pi}\left(S_{ii}a_i z_i^{*}[c_j^{(n-1)}] + y_i[a_j z_j^{*}[c_k^{(n-1)}]]\right)\right] , \end{aligned} \tag{27}$$

and, after system 19 has been solved, the "transposed charges" are obtained as $\mathbf{q}^{*} = \mathbf{R}^{\dagger}\mathbf{c}$, where q_i^{*} is

$$\begin{aligned} q_i^{*} &= -c_i + \frac{a_i}{2\pi}\sum_j D^{*}_{ij}c_j \\ &= -c_i\left(2 + \frac{1}{2\pi}z_i[a_j]\right) + \frac{1}{2\pi}a_i z_i^{*}[c_j] \ . \end{aligned} \tag{28}$$

4.1 Iterative solvers and preconditioning

We have implemented three different iterative algorithms to solve systems 6, 16 and 19. We also chose tight convergence (10^{-9}) criteria on both the root mean square and maximum residue. While this is probably not required to achieve microhartree accuracy on the free energy, it is necessary in order to compute an accurate gradient of the energy with respect to, for example, atomic degrees of freedom. Indeed, analytical gradients are available for both the PCM models discussed in this paper [13] as well as for the conductor-like PCM model [28, 29].

The first algorithm exploits the expression for the ith unknown at iteration n as a function of the other unknowns at the previous iteration. Each iteration involves one matrix–vector multiplication where the diagonal of the matrix is missing. The diagonal elements, i.e. the first terms on the right-hand side of Eqs. (14), (24) and (27), do not depend on the unknowns and are thus precomputed and reused at each iteration. To accelerate the convergence we use the direct inversion of the iterative subspace (DIIS) method [30], which turned out to be mandatory in order to obtain convergence in a reasonable number of iterations. Thus, in the following we identify this first algorithm simply as DIIS. The storage requirement scales linearly with the number of iterations involved in the DIIS extrapolation as the intermediate residues and unknown vectors need to be saved.

The second and third algorithms are standard iterative solvers for nonsymmetric matrices [31]. The conjugate gradient squared (CGS) and biconjugate gradient stabilized (BiCGStab) methods are both derived from the simple biconjugate gradient method, but do not involve the product by the transposed matrix and should provide faster and/or smoother convergence behavior. The cost per iteration is roughly twice the cost of a DIIS iteration since two matrix–vector multiplications are involved in this case. On the other hand, the storage requirement is just a constant number of times the size of the unknown vector.

In order to accelerate the convergence of both the CGS and BiCGStab methods, we implemented two preconditioning schemes. The first one (type 1) is simply the so-called Jacobi diagonal preconditioner where the full matrix is approximated by its diagonal. As previously said, the diagonal elements do not depend on the unknowns and are thus precomputed and reused at each iteration. The second scheme (type 2) is a sphere-based nonoverlapping block-diagonal preconditioner. In this case the full matrix is approximated by a block-diagonal one where the blocks correspond to the tesserae

belonging to the same atomic or added sphere. The linear systems defined by these blocks are solved exactly by LU decomposition with partial pivoting.

4.2 Linear scaling

By inspecting Eqs. (13), (14), (23), (24), (26) and (27) it is easy to realize that if the computational cost associated to the quantities **y**[**x**], **z**[**x**] and **z***[**x**] increases linearly with the number of tesserae, then the whole cost per iteration will also grow linearly.

The definitions of **y**[**x**], **z***[**x**] and **z**[**x**] (Eqs. 10, 11, 12) state that these "kernels" cost formally as the square of `NTs`. However, taking a closer look at those expressions, it is easy to realize that all three quantities can be computed in the framework of the FMM algorithm [13, 32], thus achieving linear-scaling computational cost. In particular, **y**[**x**] corresponds to the electrostatic potential at the ith tessera generated by the (pseudo)charges x_j located at the other `NTs`-1 tesserae, i.e.

$$y_i[x_j] = \sum_{j \neq i} S_{ij} x_j = \sum_{j \neq i} \frac{1}{|s_i - s_j|} x_j \ . \tag{29}$$

For the same arguments, **z***[**x**] is the normal component of the electric field at the ith tessera generated by the (pseudo)charges x_j located at the other `NTs`-1 tesserae,

$$z_i^*[x_j] = \sum_{j \neq i} D_{ij}^* x_j = \left(- \sum_{j \neq i} \frac{(\mathbf{s}_i - \mathbf{s}_j)}{|s_i - s_j|^3} x_j \right) \cdot \hat{n}_i \ . \tag{30}$$

Finally, **z**[**x**] is the sum of the "normal fields" generated by the (pseudo)charges x_j at the other `NTs`-1 tesserae, i.e.

$$z_i[x_j] = \sum_{j \neq i} D_{ij} x_j = - \sum_{j \neq i} \frac{(\mathbf{s}_i - \mathbf{s}_j) \cdot \hat{n}_j}{|s_i - s_j|^3} x_j \ , \tag{31}$$

where the presence of the $\hat{n}_j$ vectors prevents the use of the usual FMM approach "as is". However, setting $s_{ij} = \mathbf{s}_i - \mathbf{s}_j$ and separating the components of the scalar product, we end up with

$$\begin{aligned} z_i[x_j] = & - \sum_{j \neq i} \frac{(s_{ij})_x}{|s_{ij}|^3} (\hat{n}_j)_x x_j - \sum_{j \neq i} \frac{(s_{ij})_y}{|s_{ij}|^3} (\hat{n}_j)_y x_j \\ & - \sum_{j \neq i} \frac{(s_{ij})_z}{|s_{ij}|^3} (\hat{n}_j)_z x_j \ , \end{aligned} \tag{32}$$

where each summation can be obtained as the c component of the electric field generated by the pseudocharges $\{(\hat{n}_j)_c \ x_j\}$ for $c = x, y, z$. Thus, a generalization of the FMM procedure has been implemented which carries out three far-field calculations using pseudocharges, skipping two field components each time, while the near-field contribution includes explicitely the $\hat{n}_j$ vectors.

A final note about linear scaling concerns the speed/accuracy tradeoff in the application of the FMM. The cost of FMM is controlled mainly by two parameters: the box length (`boxlen`), which defines what is within the near-field of a given point in space, and the highest degree of the multipole expansion (`lmax`) used to represent the local potential generated by a set of charges. Since the cost of FMM formally increases as $O(p^3)$ where p is the length of the multipole expansion, it is very important to find the lowest value of `lmax` which ensures at least microhartree accuracy. On the other hand, as `boxlen` is increased, the cost of the near-field part increases while the cost of the far-field term decreases. We tested various combinations of `lmax` and `boxlen` values on the chemical systems discussed in the following section, using both the PCM formulations outlined in Eqs. (6) and (7), and our conclusion was that `lmax` $= 6$ and `boxlen` $= 6.0$ Å strictly ensure less than microhartree accuracy in the final free energy. Such a choice is rather conservative, but it is required if energy derivatives are to be computed.

5 Numerical applications

5.1 Performance of the new cavity code

Some performance results for the new cavity code are shown in Fig. 1. All the calculations were carried out

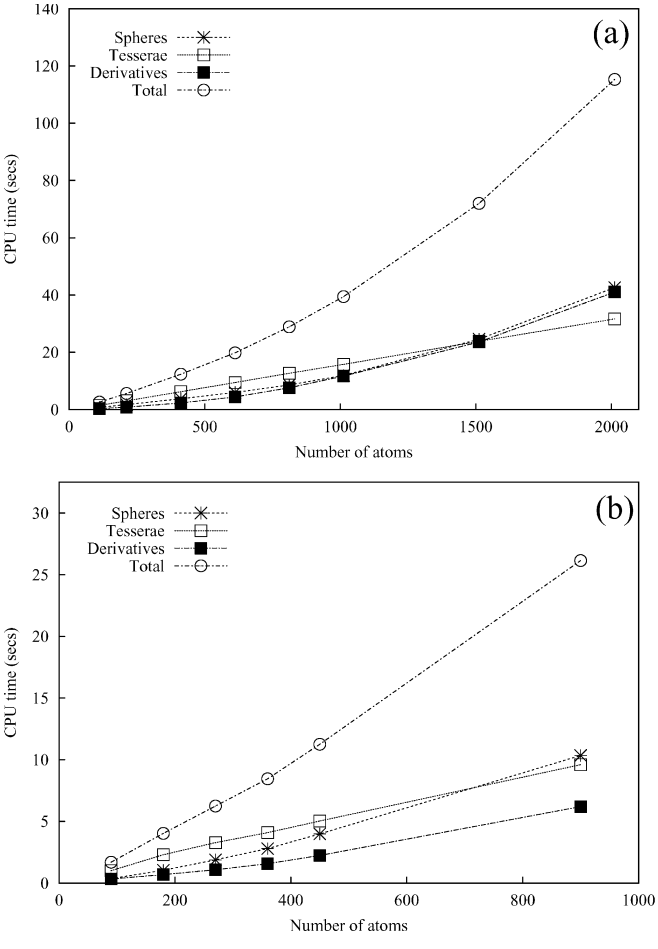

Fig. 1. Computation time (*CPU*) (seconds, with Athlon MP 1900+) to build the cavity and to compute the analytical derivatives of the tesserae position and surface area with respect to the nuclear coordinates: **a** α-helix polyalanine from 10 to 200 residues and **b** water clusters from 30 to 300 water molecules. In both cases the average area of the tesserae was set to 0.05 Å^2

with an AMD AthlonMP 1900+ machine. Two kinds of chemical systems were considered: (1) a series of polyalanines Ac-$(Ala)_n$-NMe (Ac = acetyl, NMe = *N*-methylammino) in the α-helix conformation ($\psi = -40.0^\circ$ and $\phi = -60.0^\circ$) ranging from 10 to 200 alanine residues, i.e. from 112 to 2012 atoms and (2) a set of water clusters of increasing size from 30 to 300 molecules. We chose these two sets of structures in order to carefully assess the scaling behavior of the computation time. Indeed, while the shape of the polyalanines is highly anisotropic in one direction, the shape of the water clusters is much more isotropic. Linear structures like the polyalanines could lead to a biased evaluation in favor of the low scaling of a given algorithm, while globular ones represent a much tough and unbaised benchmark.

The SES cavity was generated in all cases and the average surface area of the tesserae was set to 0.05 $Å^2$, which is an extremely small value if compared to the one recommended for ab initio calculations (0.2–0.4 $Å^2$). The number of tesserae for the largest water cluster was about 75000, while for Ac-$(Ala)_{200}$-NMe it was over 322000. The timings corresponding to the three major computational tasks have been reported separately. The ''spheres'' time includes both the generation of the extra spheres required to smooth the cavity surface and the production of the sphere's NL on the resulting set of spheres (the latter always being negligible with respect to the former). The tesserae time is the time taken to define all the spherical patches from the polyhedron faces, to discard the faces completely buried and to properly cut the ones partially exposed. The ''derivatives'' time corresponds to the calculations of the analytical derivatives with respect to the atomic coordinates of the position and the radius of all the added spheres and the position and the surface area of all the tesserae.

Our results confirm that in all cases the time required to generate the tesserae (given the set of spheres and the sphere's NL) increases linearly with the number of atoms. Both the spheres and derivatives steps show a weak quadratic scaling which appears beyond 500 atoms. The effectiveness of the algorithms being used and of their implementation is confirmed by the fact that the scaling behavior of all three steps is the same for both linear and globular molecular structures. Moreover, the residual quadratic scaling is due to the creation of the added spheres required by the SES, and is likely to affect also the derivatives time. Thus, a completely linear scaling procedure is available for the generation of VdW and solvent-accessible surfaces.

5.2 Polyalanines

In the case of a classical solute, the PCM reaction field computed by Eqs. (6) and (7) should be identical since for such solutes there is no ''outlying charge''. To verify this conclusion, we performed a series of PCM/MM calculations of the relative stability (ΔE) in aqueous solution, of polyalanines Ac-$(Ala)_n$-NMe at two different conformations: the α-helix conformation ($\psi = -40.0^\circ$ and $\phi = -60.0^\circ$) and the ''extended'' conformation ($\psi = 180.0^\circ$ and $\phi = 180.0^\circ$). The Amber force field was used throughout and we computed the difference in the relative stability ($\Delta\Delta E$) of the two conformations as predicted by the two PCM approaches of Eqs. (6) and (7). In Table 1 are collected the ΔE and the $\Delta\Delta E$ computed for a series of Ac-$(Ala)_n$-NMe with n starting from 10 and growing up to 200, using tesserae with an average surface area of 0.4 $Å^2$. The discrepancy between the relative stability computed by the two equations grows with the size of the system, but remains under 1% of the ΔE. Although this result substantially confirms the equivalence of the two PCM formulations in the case of classical solutes, we investigated this problem further. First, we recomputed all the energy differences in Table 1 enforcing the Gauss' law, i.e. scaling the polarization charges so that their sum exactly matches the sum of the solute charge. Including such a correction, the discrepancy between the results of Eqs. (6) and (7) did not change. Thus, we focused on the smallest system, i.e.

Table 1. Polarizable continuum model (*PCM*)/molecular mechanics (*MM*) calculations of the relative stability of the α-helix and extended conformations of polyalanines of increasing size. All energies are in kilocalories per mole. The convergence threshold on the polarization charges is 10^{-9}

Number of alanine units	Energy		ΔE	$\Delta\Delta E$
	α-helix	Extended		
PCM model as in Eq. (6)				
10	−49.85	4.96	−54.81	
20	−71.94	45.27	−117.20	
40	−116.30	125.90	−242.20	
80	−204.93	287.18	−492.11	
100	−248.99	368.65	−617.64	
150	−360.69	570.63	−931.32	
200	−471.79	772.45	−1244.24	
PCM model as in Eq. (7)				
10	−49.81	4.80	−54.61	0.20 (0.37%)
20	−72.07	44.64	−116.71	0.49 (0.42%)
40	−116.38	124.33	−240.72	1.49 (0.62%)
80	−204.99	283.73	−488.72	3.39 (0.69%)
100	−249.24	363.74	−612.98	4.66 (0.76%)
150	−359.95	563.17	−923.11	8.21 (0.89%)
200	−470.98	762.52	−1233.49	10.75 (0.87%)

Table 2. PCM/MM calculations of the relative stability of the α-helix and extended conformations of Ac-$(Ala)_{10}$-NMe at increasing number of tesserae. The convergence threshold on the polarization charges is 10^{-9}

Average size of tesserae (Å^2)	α-helix conformation		Extended conformation		ΔE (kcal mol^{-1})	$\Delta\Delta E$ (kcal mol^{-1})
	Number of tesserae	Energy (kcal mol^{-1})	Number of tesserae	Energy (kcal mol^{-1})		
PCM model as in Eq. (6)						
0.40	3016	−49.94	4009	4.96	−54.90	
0.20	5133	−50.19	6756	4.40	−54.59	
0.10	9440	−50.19	12232	4.33	−54.52	
0.05	17121	−50.21	21642	4.15	−54.37	
0.02	39597	−50.22	50106	4.13	−54.36	
PCM model as in Eq. (7)						
0.40	3016	−49.93	4009	4.80	−54.73	−0.17 (0.32%)
0.20	5133	−50.13	6756	4.34	−54.46	−0.13 (0.24%)
0.10	9440	−50.19	12232	4.28	−54.47	−0.06 (0.10%)
0.05	17121	−50.23	21642	4.19	−54.42	0.05 (0.10%)
0.02	39597	−50.27	50106	4.12	−54.39	0.03 (0.05%)

Table 3. PCM/MM calculations of the solvation energy of α-helix polyalanine of increasing size. The number of iterations and the total computation time (seconds, with Athlon MP 1900+) are reported for various solution methods. The convergence threshold on the polarization charges is 10^{-9}

Number of atoms	Number of tesserae	DIIS[a]		CGS				BiCGStab			
				PC type 1[b]		PC type 2[c]		PC type 1[b]		PC type 2[c]	
PCM model as in Eq. (6)											
112	3090	15	4.69	15	8.76	13	7.98	15	8.62	13	8.12
212	5584	17	9.78	14	15.33	16	19.54	14	15.64	16	19.81
412	10550	18	20.94	15	33.41	16	40.15	13	28.99	15	36.92
612	15515	16	29.39	15	50.47	17	65.76	14	47.50	17	67.44
812	20655	16	40.05	16	72.62	19	101.88	14	65.55	17	92.60
1012	25531	18	55.01	17	95.64	18	124.16	17	95.69	18	122.16
1512	38009	26	123.83	21	188.75	22	254.69	19	172.66	21	239.15
2012	50523	20	130.95	16	193.20	20	322.36	17	203.04	18	295.55
PCM model as in Eq. (7)											
112	3090	75	33.6	49	43.6	27	82.3	42	36.7	28	88.2
212	5584	160	147.3	75	136.4	43	242.3	62	113.4	41	237.7
412	10550	309	551.9	93	334.7	51	534.4	87	313.4	52	546.1
612	15515	317	858.5	140	752.4	61	938.4	112	609.4	63	963.4
812	20655		[d]	227	1669.4	104	2147.7	281	2059.5	127	2612.4
1012	25531		[d]	252	2297.1	102	2617.9	194	1769.0	113	2890.4
1512	38009		[d]		[d]	229	9461.2		[d]	319	13120.0
2012	50523		[d]	392	7528.8	207	11200.8		[d]	248	13430.2

[a] DIIS extrapolation restarted every 20 iterations or near linear dependence
[b] Type 1 preconditioning (see text), i.e. Jacobi diagonal preconditioning
[c] Type 2 preconditioning (see text), i.e. sphere-based block-diagonal preconditioning
[d] Convergence not reached within 400 iterations

$Ac\text{-}(Ala)_{10}$-NMe, and we recomputed the energy differences by increasing the accuracy of the finite-elements integration. We reduced the average surface area of the tesserae from 0.4 to 0.02 Å^2, increasing their number up to about 40000 for the α-helix conformation and 50000 for the "extended" structure. As shown by the results collected in Table 2, by increasing the surface integration accuracy the energy differences computed by Eqs. (6) and (7) become even closer. Thus we conclude that, for a classical solute, the two formulations of the PCM equations described in Sect.3 give essentially the same result as expected. The residual discrepancy can be significantly reduced by increasing the surface integration accuracy.

We proceed now to discuss the performances of the iterative solver for the set of α-polyalanines.The computation times required to solve the PCM equations by means of different algorithms are collected in Table 3: DIIS and two flavors of the conjugate gradient method for nonsymmetric matrices, i.e. CGS and BiCGStab, each using two preconditioning strategies (see Sect. 4). In all cases we require a very tight convergence on the polarization charges (10^{-9}) on both their root mean square and maximum changes. All the calculations were carried out

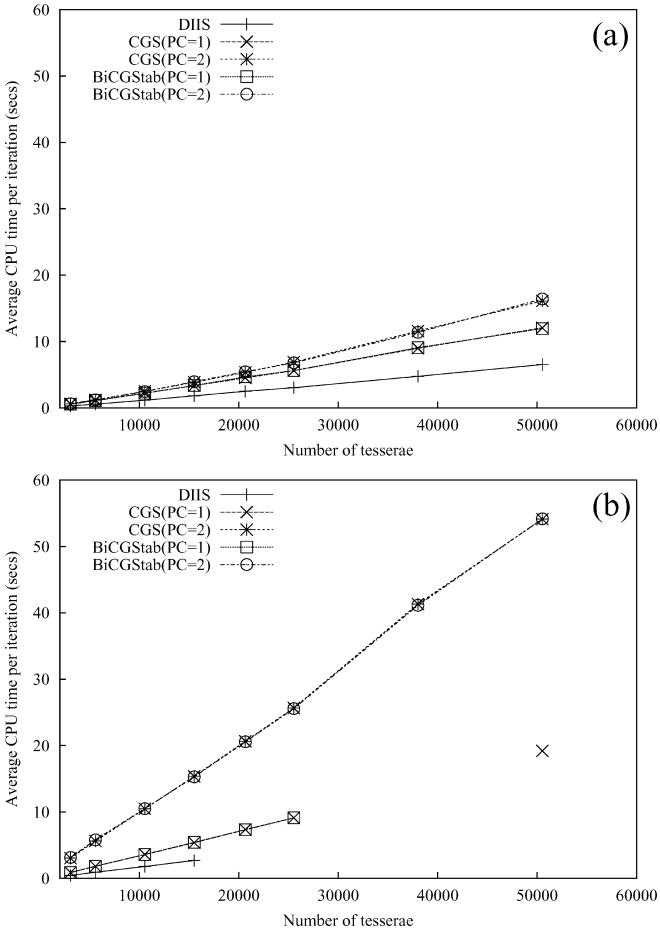

Fig. 2. Polarizable continuum model (PCM)/Molecular mechanics (*MM*) calculations of the solvation energy of α-helix polyalanine of growing size. **a** PCM model as in Eq. (6); **b** PCM model as in Eq. (7). Average CPU time per iteration (seconds, with Athlon MP 1900+) using different solution algorithms

using the developmental version of the Gaussian package [33] on an Athlon 1900+ processor. The PCM model as in Eq. (6) is not only computationally simpler than the one of Eq. (7), it is also different in its physical bases since it involves only an r^{-3} dependence on the separation of the polarization charges. Thus, the polarization charges are "more decoupled" and the solution of the linear system is found in a smaller number of iterations. On the other hand, Eq. (7) contains terms that go as r^{-1} and couple more tightly the polarization charges, making the iterative solutions more difficult and slower. In this case, the DIIS approach fails to converge for medium and large systems and it takes an excessive number of iterations for the smaller ones. The two conjugate gradient schemes are both more effective and ensure convergence, provided that a good preconditioning is used. The convergence rate (number of iterations) of CGS and BiCGStab is substantially equivalent in the solution of Eq. (6), while for Eq. (7) the CGS algorithm is faster, especially using the sphere-based block-diagonal preconditioning (type 2). The average time per iteration is reported in Fig.2 as a function of the system size. All solvers show linear-scaling computational cost, confirming the effectiveness of the FMM code for computing the quantities defined in Eqs. (10), (11) and (12). This linear scaling behavior is confirmed if the computation time per iteration is plotted with respect to the number of atoms since in the case of linear structures the average surface area per atom (i.e. the average number of tesserae per atom) is constant. Finally, we note that the cost per iteration includes the preconditioning: while the cost of the diagonal preconditioning (type 1) is linear scaling by construction, it is significant that also the sphere-based block-diagonal approach (type 2) shows linear-scaling behavior. The time per iteration required by the DIIS solver is significantly shorter if compared to the conjugate gradient ones. The costs of CGS and BiCGStab are very similar when the same preconditioning scheme is used, although the sphere-based preconditioning turns out to be computationally more expensive, especially in the solution of Eq. (7).

5.3 Water clusters

Most of the comments made on the performance of the different solution schemes in the case of α-polyalanines hold also for water clusters. Generally speaking, the water clusters we considered are smaller systems with respect to the set of α-polyalanines, the largest one including 300 molecules. Moreover, considering the number of tesserae, i.e. the total surface area of the solute–solvent boundary, the water cluster of 300 molecules should be compared to the Ac-$(Ala)_{60}$-NMe system, which is made by about 600 atoms. Thus, the differences in the behavior of the various solution algorithms can be related to either the smaller size of the system or the different shape of the structure (linear *vs* globular) or to both these factors. The computation times required to solve the PCM equations for a set of water clusters from 30 to 300 molecules are collected in Table 4. As previously said, since most of the comments made for α-polyalanines hold also here, we rather point out the differences that exist between the results obtained for the two sets of structures. The conjugate gradient methods using the diagonal preconditioning never fail to converge for this set of water clusters, but show a strongly irregular behavior in the solution of Eq. (7) using CGS. The rate of convergence of CGS and BiCGStab, using sphere-based preconditioning, is in this case the same not only for Eq. (6), but also for Eq. (7). Comparing the results for the largest water cluster with the ones obtained for α-polyalanine with approximately the same number of tesserae, it turns out that the solution of Eq. (6) shows basically the same trends, while as far as Eq. (7) is concerned, the convergence rate is slower and the use of the sphere-based rather than the diagonal preconditioning scheme appears to be less effective for globular structures than for linear ones. The linear growth of the computation time per iteration is shown in Fig. 3 with respect to the system size. We point out that this linear scaling behavior is perfectly consistent with the use of the FMM to compute the interactions among the polarization charges. Moreover, in the case of water clusters, and of globular structures in general, if the computation time per iteration is plotted with respect to the number of atoms, a sublinear scaling

Table 4. PCM/MM calculations of the solvation energy of water clusters of increasing size. The number of iterations and the total computation time (seconds, with Athlon MP 1900+) are reported for various solution methods. The convergence threshold on the polarization charges is 10^{-9}

Number of atoms	Number of tesserae	DIIS[a]		CGS				BiCGStab			
				PC type 1[b]		PC type 2[c]		PC type 1[b]		PC type 2[c]	
PCM model as in Eq. (6)											
90	3144	19	6.6	13	8.6	15	10.2	13	8.4	16	11.4
180	5011	15	9.0	14	15.5	15	18.0	13	14.1	15	18.3
270	6893	18	15.7	16	24.9	15	27.5	14	23.0	15	26.6
360	8257	22	21.7	14	26.8	17	37.5	15	28.4	17	36.0
450	9821	26	30.0	19	42.3	19	50.6	19	41.3	17	46.2
900	15179	20	37.2	17	58.2	19	89.3	16	55.0	18	82.8
PCM model as in Eq. (7)											
90	3144	195	91.1	55	52.0	33	60.1	44	41.5	35	62.6
180	5011	128	112.7	67	117.2	33	99.1	59	104.0	35	103.6
270	6893		[d]	102	259.2	52	216.1	161	406.3	67	265.5
360	8257		[d]	106	331.9	53	261.2	87	275.3	55	269.2
450	9821		[d]	173	659.3	99	582.9	225	858.3	120	703.5
900	15179		[d]	190	1097.5	117	1097.6	161	929.3	108	1010.6

[a] DIIS extrapolation restarted every 20 iterations or near linear dependence
[b] Type 1 preconditioning (see text), i.e. Jacobi diagonal preconditioning
[c] Type 2 preconditioning (see text), i.e. sphere-based block-diagonal preconditioning
[d] Convergence not reached within 400 iterations

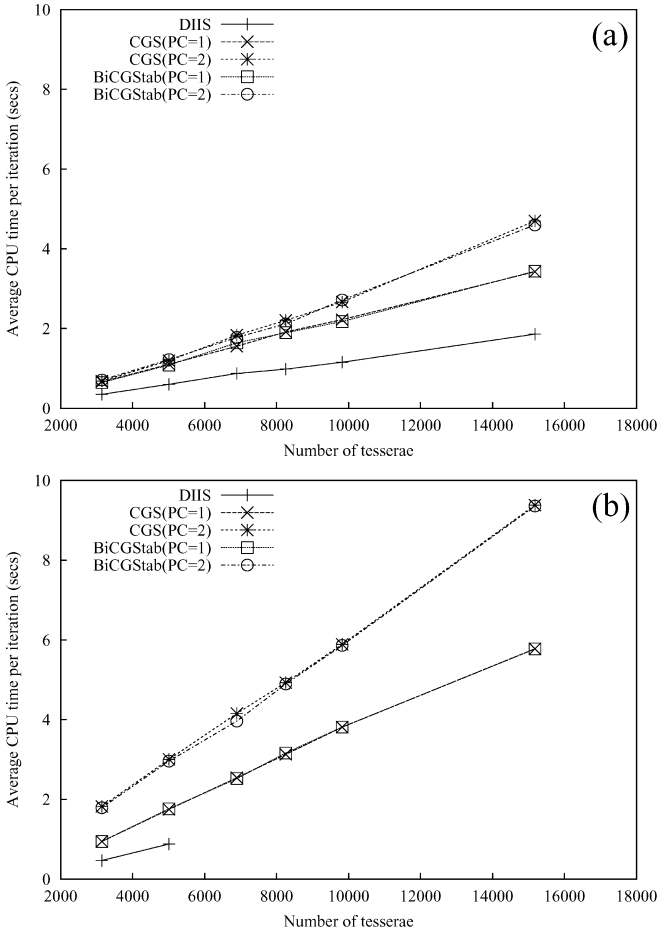

Fig. 3. PCM/MM calculations of the solvation energy of water clusters of increasing size. **a** PCM model as in Eq. (6); **b** PCM model as in Eq. (7). Average CPU time per iteration (seconds, with Athlon MP 1900+) using different solution algorithms

is achieved as the average number of tesserae per atom decreases with the increase of the size of the system.

6 Conclusion

In this contribution we reported a new and low-scaling implementation of the PCM method for computing the solvent reaction field. This approach is applicable to classical, quantum and hybrid quantum/classical solutes. In our analysis of the computational cost we did not consider the cost of computing the solute electrostatic potential or electric field on the tesserae since this is usually negligible in the case of classical solutes, while it largely depends on the advances in the technology for computing one-electron integrals in the case of quantum solutes. The cost of generating the tesserae and computing their derivatives scales now linearly for system sizes up to about 500 atoms, while for larger systems a weak quadratic dependence is observed. On the other hand, the cost per iteration of the iterative solver shows perfect linear scaling even for globular solutes. This is due to the use of the FMM to compute the electrostatic interaction of the polarization charges among themselves. The performances of different iterative solvers (DIIS, CGS and BiCGStab) and preconditioning strategies have been assessed. We demonstrated the superiority of the DIIS approach in the special PCM formulation for classical solutes, while for the general form of the PCM equations the preconditioned CGS method is recommended.

Acknowledgements. The authors are happy to contribute to the special issue of *Theoretical Chemistry Accounts* in honor of the career of Jacopo Tomasi, whose contribution to the development of theoretical and computational chemistry can hardly be overemphasized. We thank the University of Naples Centro Interdipartimentale di Metodologie Chimico-Fisiche for providing computing resources. Support from Gaussian, Inc. is also gratefully acknowledged.

References

1. Reichardt C (1990) Solvents and solvent effects in organic chemistry, 2nd edn. (VCH, Weinhein)
2. Rivail JL, Rinaldi D, Ruiz-Lopez MF (1996) In: Leszcynsk J.(ed) Computational chemistry: review of current trends. World Scientific, Singapore, p 139
3. Cramer CJ, Truhlar DG (1996) In: Tapia O, Bertra'n J (eds) Solvent effects and chemical reactivity. Kluwer, Dordrecht, p1
4. Adamo C, Cossi M, Rega N, Barone V (2001) In: Erikson LA (ed) Theoretical biochemistry: processes and properties of biological systems. Elsevier, Amsterdam, p 467
5. (a) Zeng J, Craw JS, Hush NS, Reimers JR (1993) J Chem Phys 99: 1482; (b) Zeng J, Hush NS, Reimers JR (1993) J Chem Phys 99: 1508; (c) Zeng J, Woywod C, Hush NS, Reimers JR (1995) J Am Chem Soc 117: 8618
6. (a) Coutinho K, Canuto S (2000) J Chem Phys 113: 9132; (b) Mennucci B, Martinez JM, Tomasi J (2001) J Phys Chem A 105: 7287
7. (a) Chesnut DB, Rusiloski BE (1994) J Mol Struct (Theochem) 314: 19; (b) Pecul M, Sadlej J (1998) Chem Phys 234: 111
8. Tomasi J, Persico M (1994) Chem Rev 94: 2027
9. Cramer CJ, Truhlar DG (1999) Chem Rev 99: 2161
10. (a) Li J, Zhu T, Cramer CJ, Truhlar DG (2000) J Phys Chem A 104: 2178; (b) Dolney DM, Hawkins GD, Winget P, Liotard DA, Cramer CJ, Truhlar DG (2000) J Comput Chem 21: 340
11. (a) Miertuš S, Scrocco E, Tomasi J (1981) Chem Phys 55: 117; (b) Cammi R, Tomasi J (1995) Comput Chem 16: 1449; (c) Cossi M, Barone V, Cammi R, Tomasi J (1996) Chem Phys Lett 255: 327; (d) Amovilli C, Barone V, Cammi R, Cancès E, Cossi M, Mennucci B, Pomelli CS, Tomasi J (1998) Adv Quantum Chem 32: 227
12. (a) Cancès E, Mennucci B, Tomasi J (1997) J Chem Phys 107: 3032; (b) Mennucci B, Cancès E, Tomasi J (1997) J Phys Chem B 101: 10506; (c) Cossi M, Barone V (1998) Chem Phys 109: 6246; (d) Mennucci B, Cammi R, Tomasi J (1999) J Chem Phys 110: 6858; (e) Cammi R, Mennucci B (1999) J Chem Phys 110: 9877; (f) Cossi M, Barone V (2001) J Chem Phys 115: 4708; (g) Cossi M, Rega N, Scalmani G, Barone V (2001) J Chem Phys 114: 5691
13. Cossi M, Scalmani G, Rega N, Barone V (2002) J Chem Phys 117: 43
14. Vreven T, Mennucci B, da Silva CO, Morokuma K, Tomasi J (2001) J Chem Phys 115: 62
15. (a) Langella E, Improta R, Barone V (2002) J Am Chem Soc 124: 11531; (b) Mennucci B, Martínez JM, Tomasi J (2001) J Phys Chem A 105: 7287; (c) Cossi M, Crescenzi O J Chem Phys (in press.)
16. (a) Greengard L, Rokhlin V (1987) J Comput Phys 73: 325; (b) Strain MC, Scuseria GE, Frisch MJ (1996) Science 271: 51
17. (a) Singh UC, Kollman PA (1986) J Comput Chem 7: 718; (b) Gao J (1996) In: Libkowitz KB, Boyd DB (eds) Reviews in computational chemistry, vol 7. VCH, Weinhein, pp 119–185; (c) Dapprich S, Komaromi I, Byun KS, Morokuma K, Frisch MJ (1999) J Mol Struct (THEOCHEM), 461–462: 1
18. Rappè AK, Casewit CJ, Colwell KS, Goddard WA, III Skiff WM (1992) J Am Chem Soc 114: 10024
19. Lide DR (ed) (2001) Handbook of chemistry and physics. CRC, Cleveland, OH
20. Bondi A (1964) J Phys Chem 68: 441
21. Barone V, Cossi M, Tomasi J (1997) J Chem Phys 107: 3210
22. Pascual-Ahuir J.-L, Silla E, Tuñon I (1994) J Comput Chem 15: 1127
23. Cossi M, Mennucci B, Cammi R (1996) J Comput Chem 17: 57
24. Scalmani G, Rega N, Cossi M, Barone V (2002) J Comput Methods Sci Eng 2: 159
25. (a) Greer J, Bush BL (1978) Proc Natl Acad Sci USA 75: 303; (b) Connolly ML (1983) J Appl Crystallogr 16: 548
26. Lee B, Richards FM (1971) J Mol Biol 55: 379
27. Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR, Zakrzewski VG, Montgomery JA, Jr Stratmann RE, Burant JC, Dapprich S, Millam JM, Daniels AD, Kudin KN, Strain MC, Farkas O, Tomasi J, Barone V, Cossi M, Cammi R, Mennucci B, Pomelli C, Adamo C, Clifford S, Ochterski J, Petersson GA, Ayala PY, Cui Q, Morokuma K, Salvador P, Dannenberg JJ, Malick DK, Rabuck AD, Raghavachari K, Foresman JB, Cioslowski J, Ortiz JV, Baboul AG, Stefanov BB, Liu G, Liashenko A, Piskorz P, Komaromi I, Gomperts R, Martin RL, Fox DJ, Keith T, Al-Laham MA, Peng CY, Nanayakkara A, Challacombe M, Gill PMW, Johnson B, Chen W, Wong MW, Andres JL, Gonzalez C, Head-Gordon M, Replogle ES, Pople JA (2001) Gaussian 98, (revision A.11). Gaussian, Pittsburgh, PA
28. Barone V, Cossi M (1998) J Phys Chem A 102: 1995
29. Cossi M, Scalmani G, Rega N, Barone V J Comput Chem (in press.)
30. (a) Pulay P (1980) Chem Phys Lett 73: 393; (b) Pomelli CS, Tomasi J, Barone V (2001) Theor Chem Acc 105: 446
31. Barrett R, Berry M, Chan TF, Demmel J, Donato J, Dongarra J, Eijkhout V, Pozo R, Romine C, Van der Vorst H (1994) Templates for the solution of linear systems: building blocks for iterative methods, 2nd edn. SIAM, Philadelphia (also available at www.netlib.org)
32. Kudin KN and Scuseria GE, unpublished.
33. Frisch MJ, Trucks GW, Schlegel HB, Scuseria GE, Robb MA, Cheeseman JR, Zakrzewski VG, Montgomery JA, Jr. Kudin KN, Burant JC, Millam JM, Stratmann RE, Tomasi J, Barone V, Mennucci B, Cossi M, Scalmani G, Rega N, Iyengar S, Petersson GA, Ehara M, Toyota K, Nakatsuji H, Adamo C, Jaramillo J, Cammi R, Pomelli C, Ochterski J, Ayala PY, Morokuma K, Salvador P, Dannenberg JJ, Dapprich S, Daniels AD, Strain MC, Farkas O, Malick DK, Rabuck AD, Raghavachari K, Foresman JB, Ortiz JV, Cui Q, Baboul AG, Clifford S, Cioslowski J, Stefanov BB, Liu G, Liashenko A, Piskorz P, Komaromi I, Gomperts R, Martin RL, Fox DJ, Keith T, Al-Laham MA, Peng CY, Nanayakkara A, Challacombe M, Gill PMW, Johnson B, Chen W, Wong MW, Andres JL, Gonzalez C, Head-Gordon M, Replogle ES, Pople JA (2001) Gaussian 01, development version (revision B.01). Gaussian, Pittsburgh, PA