Chapter 11

Polarizable Continuum Models for (Bio)Molecular Electrostatics: Basic Theory and Recent Developments for Macromolecules and Simulations

John M. Herbert[a] **and Adrian W. Lange**[a,b]
[a] *Department of Chemistry and Biochemistry,*
The Ohio State University, Columbus, OH, USA
[b] *Present address: Apple, Inc., Cupertino, CA, USA*
herbert@chemistry.ohio-state.edu

11.1 Overview

The topic of this chapter is the solution of a simple and well-defined model problem, namely, the molecular electrostatics problem for one or more molecules immersed in a homogeneous dielectric medium characterized by a dielectric constant, ε. The interface between the atomistic region (the solute) and the continuum solvent is defined by a molecule-shaped cavity such as the ones depicted in Figs. 11.1(a) and 11.1(b). In practice, this cavity is often constructed from atom-centered spheres, although more

---

Many-Body Effects and Electrostatics in Biomolecules
Edited by Qiang Cui, Markus Meuwly, and Pengyu Ren

ISBN 978-981-4613-92-7 (Hardcover), 978-981-4613-93-4 (eBook)
www.panstanford.com

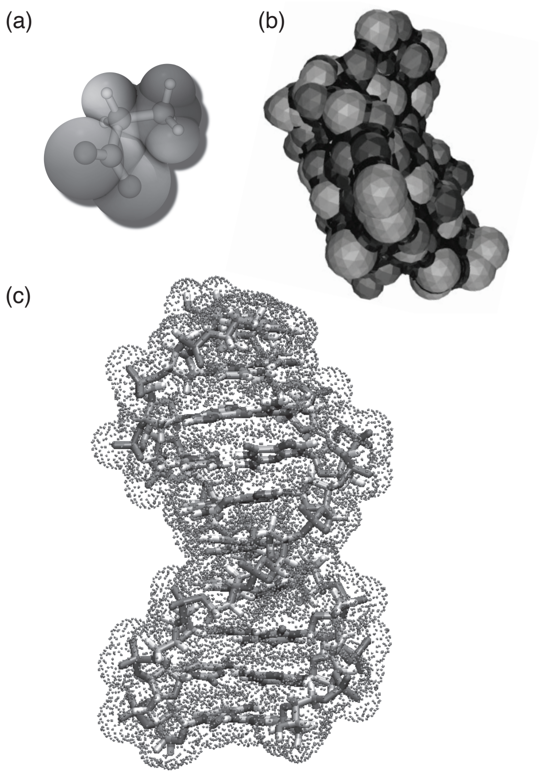

Figure 11.1 (a) Pictorial depiction of a cavity, constructed from atom-centered spheres, that defines the interface between the atomistic region and the continuum. (b) Triangular tessellation of the atom-centered spheres that define the surface of the protein 3U7T (crambin). (c) Cavity surface for a segment of double-stranded DNA, discretized with atom-centered Lebedev grids. Panel (b) is reprinted from Ref. [25]; copyright 2002 John Wiley and Sons.

complicated constructions have been considered [21]. Atomistic electrostatics is used for the solute, often with $\varepsilon = 1$ inside of the cavity, although this choice is not required by the theory and other values have been employed, e.g., in an attempt to incorporate a protein dielectric "constant". In any case, there is a sharp discontinuity in $\varepsilon(\mathbf{r})$ at the cavity surface. The atomistic region can be described at various levels of complexity: quantum-mechanically, in terms of an electron density, or classically in terms of a set of point charges and/or higher-order multipoles, be they static or polarizable.

Given a solute charge distribution $\rho(\mathbf{r})$ whose corresponding electrostatic potential we denote by $\phi^{\rho}(\mathbf{r})$, the solution to the aforementioned model problem consists in determining the total (solute + continuum) electrostatic potential

$$\phi(\mathbf{r}) = \phi^{\rho}(\mathbf{r}) + \phi_{\text{rxn}}(\mathbf{r}), \tag{11.1}$$

which includes a *reaction-field potential*, $\phi_{\text{rxn}}(\mathbf{r})$, that arises from polarization of the medium. The total potential $\phi(\mathbf{r})$ is obtained by solution of *Poisson's equation* [7, 85]. For a sharp dielectric boundary, this equation reads

$$\hat{\nabla}^2\phi(\mathbf{r}) = -4\pi\rho(\mathbf{r}) \times \begin{cases} 1/\varepsilon_{\text{inside}} \\ 1/\varepsilon_{\text{outside}} \end{cases}, \tag{11.2}$$

expressed here in unrationalized CGS units [85]. [For a solute described by classical multipoles, the definition of $\rho(\mathbf{r})$ in Eq. (11.2) might be considered problematic, but the methods discussed below actually require only the electrostatic potential $\phi^{\rho}(\mathbf{r})$ generated by these multipoles.] Having determined $\phi(\mathbf{r})$, the total electrostatic (or polarization) free energy is

$$G_{\text{pol}} = \frac{1}{2}\int_{\mathbb{R}^3} d\mathbf{r}\ \rho(\mathbf{r})\ \phi(\mathbf{r}), \tag{11.3}$$

where the factor of 1/2 accounts for the reversible work done in polarizing the medium (hence why G_{pol} is a *free* energy) [7].

Equation (11.2) is a partial differential equation in three dimensions, subject to boundary conditions such that $\phi(\mathbf{r})$ is continuous across the cavity surface but must decay faster than r^{-2} as $r \to \infty$ [77]. This equation can be solved using grid-based finite-difference techniques [5, 35, 54], though this requires discretizing the whole of three-dimensional space, including the infinite continuum region. For a macromolecular solute described using a classical force field, such methods form the basis of much of modern biomolecular electrostatics calculations [4, 54]. (In practice, the equation that is usually solved in biomolecular applications is the *Poisson–Boltzmann equation* [4, 33], which includes the effects of a thermal distribution of dissolved ions; this will be considered in Section 11.3.1). Such methods are useful for producing an electrostatic map of the surface of the macromolecular solute, but their finite-difference nature means that forces obtained from such algorithms are inherently discontinuous. Although progress is being made to reduce this problem [80–83], the discontinuities pose a fundamental problem for the use of finite-difference solvers in molecular dynamics (MD) simulations. Moreover, the requirement to discretize all of three-dimensional space (or at least sufficiently far into the continuum so that $\phi(\mathbf{r})$ has decayed to zero) means that the size of the discretized linear systems becomes extremely large for macromolecules. The matrices involved are sparse; nevertheless, only highly parallelized approaches are tractable.

This chapter explores an alternative category of methods aimed at solving the same continuum electrostatics problem using an *apparent surface charge* (ASC), $\sigma(\mathbf{s})$, induced at the cavity surface by polarization of the medium. Here, we use $\mathbf{s} \in \mathbf{\Gamma}$ to denote a point on the cavity surface, $\mathbf{\Gamma}$, whereas $\mathbf{r} \in \mathbb{R}^3$. The quantity $\sigma(\mathbf{s})$ is determined from $\rho(\mathbf{r})$ as described in Section 11.2 but exists only on $\mathbf{\Gamma}$. Thus

$$G_{\mathrm{pol}} = \frac{1}{2} \int_{\mathbf{\Gamma}} d\mathbf{s}\; \rho(\mathbf{s})\; \sigma(\mathbf{s}). \tag{11.4}$$

Relative to finite-difference Poisson–Boltzmann approaches, such methods have the advantage that only the two-dimensional cavity surface must be discretized.

Methods based on an ASC have a long history in quantum-mechanical (QM) calculations with continuum solvent [60, 61, 77], where they are generally known as *polarizable continuum models* (PCMs). However, PCMs have seen little use in the area of biomolecular electrostatics, for reasons that are unclear to us. In the QM context, such methods are inherently approximate, even with respect to the model problem defined by Poisson's equation, owing to the *volume polarization* that results from the tail of the QM electron density that penetrates beyond the cavity and into the continuum [13, 14, 89]. The effects of volume polarization can be treated only approximately within the ASC formalism [14, 15, 89]. For a classical solute, however, there is no such tail and certain methods in the PCM family *do* afford a numerically exact solution of Poisson's equation, up to discretization errors that are systematically eliminable. Moreover, ASC methods have been generalized to solve the linearized Poisson–Boltzmann equation [17, 22, 43, 58], and to inhomogeneous dielectrics where the scalar ε is replaced by a dielectric tensor [8, 57, 77]. Long-standing problems with discontinuities engendered by discretization have been overcome [41, 42, 70, 76, 87], so that intrinsically smooth forces are available for MD simulations. Finally, linear-scaling implementations of the PCM algorithm render such methods amenable to macromolecular solutes [25, 69]. Such developments are potentially useful not only for traditional biomolecular electrostatics calculations, but also for QM/MM/PCM calculations, in which the PCM serves as a boundary condition for a QM/MM calculation (replacing periodic boundary conditions), but where the size of the large MM region dictates the dimensionality of the linear equations, that must be solved to obtain the ASC. For many QM/MM/PCM calculations, the cost of solving the PCM equations would exceed the QM cost, were it not for linear-scaling implementations of the PCM algorithm.

The goal of this chapter is to draw attention to some of these developments, with the aim of popularizing PCMs beyond small-molecule QM applications. We do not have the space here for a comprehensive review (and several recent ones can be found [56, 77]) but will focus mainly on our own work [41–46]. Some knowledge of basic continuum electrostatics is assumed; see Ref. [7] for an excellent pedagogical introduction. This chapter will focus mostly on the advantages of the PCM formulation of the electrostatics problem, with an emphasis on methods that might ultimately replace finite-difference Poisson–Boltzmann solvers. In addition, details of our linear-scaling implementation of the PCM algorithm are presented here for the first time, although this algorithm has been available for some time as part of the Q-Chem software [39].

11.2 Theoretical Background

11.2.1 *Continuum Electrostatics*

The basic setup of the continuum electrostatics problem has been outlined above. The ASC formalism is based on an *ansatz* in which the exact reaction-field potential—which includes the effects of volume polarization and is defined throughout three-dimensional space—is nevertheless generated by a charge distribution $\widetilde{\sigma}$ that exists only on the cavity surface:

$$\phi_{\text{rxn}}(\mathbf{r}) = \int_{\Gamma} d\mathbf{s}\, \frac{\widetilde{\sigma}(\mathbf{s})}{|\mathbf{s}-\mathbf{r}|}. \tag{11.5}$$

The *apparent* surface charge, $\widetilde{\sigma}$, should be distinguished from the *actual* surface charge that forms at any dielectric boundary [85]. The latter is given by

$$\sigma(\mathbf{s}) = \left(\frac{\varepsilon - 1}{4\pi}\right) (\mathbf{n_s} \cdot \hat{\nabla})\phi(\mathbf{s})\Big|_{\mathbf{s}=\mathbf{s}^+}. \tag{11.6}$$

Here, $\mathbf{n_s}$ represents the outward-pointing surface normal vector located at the point $\mathbf{s}$, so that the derivative in Eq. (11.6) represents the outward-pointing normal component of the electric field. (The notation $\mathbf{s} = \mathbf{s}^+$ indicates that this derivative should be evaluated infinitesimally outside of the cavity.) The normal electric field is discontinuous at a dielectric boundary, and satisfies a “jump” boundary condition [7, 85],

$$\varepsilon_{\text{outside}}\, (\mathbf{n_s} \cdot \hat{\nabla})\phi(\mathbf{s})\Big|_{\mathbf{s}^+} = \varepsilon_{\text{inside}}\, (\mathbf{n_s} \cdot \hat{\nabla})\phi(\mathbf{s})\Big|_{\mathbf{s}^-}. \tag{11.7}$$

This comes from the fact that the electric displacement (= ε × electric field) is *continuous* across the dielectric boundary.

Equation (11.7) can be used to eliminate the exterior derivative of ϕ from Eq. (11.6). Then, given some initial approximation for ϕ (perhaps just ϕ^{ρ}, which is known once the solute’s wave function has been computed), one could compute the surface charge, and thus the reaction-field potential, without the need to perform any calculations outside of the solute cavity. For a QM solute, this procedure must then be iterated to self-consistency. The original PCM of Miertuš, Scrocco, and Tomasi [60, 61] used precisely this approach; this model is now known as D-PCM. It is less desirable than more modern PCMs, owing to the need to compute the normal electric field, which may be subject to increased numerical noise relative to later formulations that involve only electrostatic potentials [77]. Perhaps more significantly, the formulation of this model *has* conflated the apparent and actual surface charge distributions, and corresponds to a neglect of volume polarization [13].

A key point in the elementary theory of dielectric materials is that the polarization vector can be replaced by an appropriate charge distribution, which consists of both a surface charge distribution at the dielectric boundaries [Eq. (11.6)] and a volume charge distribution in the dielectric material itself [7, 85]. The latter was ignored in the early development of PCMs [13, 59], but was finally treated carefully in the late 1990s by Chipman [13–15, 89]. Generalizing Chipman's treatment to an arbitrary value of $\varepsilon_{\text{inside}}$, we note that in the absence of the medium, the solute's electrostatic potential would satisfy the Poisson equation $\hat{\nabla}^2\phi^\rho = -4\pi\rho/\varepsilon_{\text{inside}}$ throughout all space. On the other hand, the total potential $\phi = \phi^\rho + \phi_{\text{rxn}}$ must satisfy Eq. (11.2); hence, the reaction-field potential must satisfy the equation

$$\hat{\nabla}^2\phi_{\text{rxn}}(\mathbf{r}) = \begin{cases} 0 & \text{for } \mathbf{r} \text{ inside of } \mathbf{\Gamma} \\ 4\pi\left(\varepsilon_{\text{inside}}^{-1} - \varepsilon_{\text{outside}}^{-1}\right)\rho(\mathbf{r}) & \text{for } \mathbf{r} \text{ outside of } \mathbf{\Gamma} \end{cases}. \tag{11.8}$$

This can be accomplished by invoking an apparent volume charge

$$\beta(\mathbf{r}) = \begin{cases} 0 & \text{for } \mathbf{r} \text{ inside of } \mathbf{\Gamma} \\ \left(\varepsilon_{\text{outside}}^{-1} - \varepsilon_{\text{inside}}^{-1}\right)\rho(\mathbf{r}) & \text{for } \mathbf{r} \text{ outside of } \mathbf{\Gamma} \end{cases} \tag{11.9}$$

that satisfies a Poisson equation

$$\hat{\nabla}^2\phi^\beta = -4\pi\beta. \tag{11.10}$$

As such, an *exact* treatment of volume polarization [13, 18, 89], which is not considered here, requires discretization of three-dimensional space in order to solve Eq. (11.10).

If ϕ^β is known, then according to Eq. (11.6) the proper surface charge should be [14]

$$\sigma(\mathbf{s}) = \left(\frac{\varepsilon - 1}{4\pi}\right)\left[\hat{\partial}_{\mathbf{s}}\,\phi^\rho(\mathbf{s}) + \hat{\partial}_{\mathbf{s}}\,\phi^\beta(\mathbf{s}) + \hat{\partial}_{\mathbf{s}}\,\phi^\sigma(\mathbf{s}^+)\right], \tag{11.11}$$

where the notation $\hat{\partial}_{\mathbf{s}}\,\phi = (\mathbf{n}_{\mathbf{s}} \cdot \hat{\nabla})\phi$ has been introduced, and $\hat{\partial}_{\mathbf{s}}\,\phi^\sigma(\mathbf{s}^+)$ is the contribution arising from the dielectric boundary. Infinitesimal displacements in ϕ^ρ and ϕ^β are not necessary, as these potentials are continuous across the cavity surface [14].

An approximate treatment is obtained by noting that the (actual) surface charge is obtainable directly from the discontinuity in the normal electric field [13],

$$\sigma(\mathbf{s}) = \frac{1}{4\pi}\left(\hat{\partial}_{\mathbf{s}}\,\phi\big|_{\mathbf{s}=\mathbf{s}^-} - \hat{\partial}_{\mathbf{s}}\,\phi\big|_{\mathbf{s}=\mathbf{s}^+}\right), \tag{11.12}$$

The potential ϕ includes the reaction-field part, which depends upon both the surface and volume charge distributions; hence, σ implicitly appears on both sides of Eq. (11.12). Combining this result with Eq. (11.7), and setting $\varepsilon_{\text{inside}} = 1$ and $\varepsilon_{\text{outside}} = \varepsilon$ for the remainder of this chapter, one obtains

$$\sigma(\mathbf{s}) = \frac{f_\varepsilon}{2\pi}\,\hat{\partial}_{\mathbf{s}}\left[\phi^\rho(\mathbf{s}) + \phi^\sigma(\mathbf{s}) + \phi^\beta(\mathbf{s})\right], \tag{11.13}$$

where $f_\varepsilon = (\varepsilon - 1)/(\varepsilon + 1)$ and the normal derivative $\hat{\partial}_{\mathbf{s}}\,\phi^\sigma(\mathbf{s})$ is now evaluated *at* the cavity surface, rather than an infinitesimal displacement away. For pedagogical reasons we rewrite Eq. (11.13) in the form

$$\left(\hat{1} - \frac{f_\varepsilon}{2\pi}\hat{\partial}_{\mathbf{s}}\right)\sigma(\mathbf{s}) = \frac{f_\varepsilon}{2\pi}\,\hat{\partial}_{\mathbf{s}}\left[\phi^\rho(\mathbf{s}) + \phi^\beta(\mathbf{s})\right]. \tag{11.14}$$

This is as far as one can go with an exact formulation, unless one is willing to solve Eq. (11.10) in three dimensions. However, the effect of volume polarization can be *approximated* by introducing an additional surface charge, $\alpha(\mathbf{s})$, that is defined such that its electrostatic potential at the cavity surface is identical to that generated by ϕ^β. Let us define an operator $\hat{S}$ that acts on functions $f(\mathbf{s})$ defined on $\boldsymbol{\Gamma}$, generating the corresponding electrostatic potential:

$$\hat{S}f(\mathbf{s}) = \int_{\boldsymbol{\Gamma}} d\mathbf{s}'\,\frac{f(\mathbf{s}')}{|\mathbf{s} - \mathbf{s}'|}, \tag{11.15}$$

We therefore insist that [15]

$$\hat{S}\alpha(\mathbf{s}) = \phi^\beta(\mathbf{s}), \tag{11.16}$$

and set $\widetilde{\sigma} = \sigma + \alpha$. This approximation allows for the elimination of ϕ^β in Eq. (11.14), affording an equation that requires only surface integration [15]:

$$\underbrace{\hat{S}\left(\hat{1} - \frac{f_\varepsilon}{2\pi}\hat{D}^\dagger\right)}_{\hat{K}}\widetilde{\sigma}(\mathbf{s}) = \underbrace{f_\varepsilon\left(\frac{1}{2\pi}\hat{D} - \hat{1}\right)}_{\hat{R}}\phi^\rho(\mathbf{s}). \tag{11.17}$$

Here, the operator $\hat{D}^\dagger$ generates the negative of the outward-pointing normal electric field [13, 15],

$$\hat{D}^\dagger f(\mathbf{s}) = -\int_\Gamma d\mathbf{s}'\ f(\mathbf{s}')\frac{\partial}{\partial \mathbf{n}_\mathbf{s}}|\mathbf{s}-\mathbf{s}'|^{-1}, \tag{11.18}$$

and its adjoint is defined such that

$$\hat{D} f(\mathbf{s}) = \int_\Gamma d\mathbf{s}'\ f(\mathbf{s}')\frac{\partial}{\partial \mathbf{n}_{\mathbf{s}'}}|\mathbf{s}-\mathbf{s}'|^{-1}. \tag{11.19}$$

The origin of Eq. (11.17), or at least its left side, is evident from Eq. (11.14). Equation (11.17) also indicates the notation that we will henceforth use for this equation:

$$\hat{K}\,\widetilde{\sigma}(\mathbf{s}) = \hat{R}\,\phi^\rho(\mathbf{s}). \tag{11.20}$$

Equation (11.20) is the primary PCM equation. It must be discretized for actual computation (see Section 11.2.2), but then given the solute's electrostatic potential evaluated at the surface discretization points, this equation can be solved for the induced surface charge at those points (i.e., the discretized $\widetilde{\sigma}$). In an MM/PCM calculation, the electrostatic solvation energy is then immediately available via a discretized version of Eq. (11.3), although in QM applications the surface charge must be included in the next self-consistent field (SCF) iteration, and the SCF procedure is iterated until both the electron density and the surface charge have reached mutual self-consistency.

For QM solutes, volume polarization is treated approximately (but accurately [89]) by Eq. (11.17), and Chipman has called this approach *surface and simulation of volume polarization for electrostatics* [SS(V)PE] [15]. An equivalent form of Eq. (11.17) was actually derived prior to Chipman's work, where it was called the *integral equation formalism* (IEF) [10, 58]. The equivalence is not obvious, as the original IEF requires the solute's electric field as an input in addition to its electrostatic potential, but it was later shown that the former could be eliminated in order to obtain Eq. (11.17) [9]. The operator $\hat{K}$ can similarly be manipulated into different forms, by means of the identity [15]

$$\hat{D}\hat{S} = \hat{S}\hat{D}^\dagger. \tag{11.21}$$

However, this identity is not preserved upon discretization, and different implementations of SS(V)PE/IEF-PCM are therefore possible, not all of which perform equally well in practice [44]. This is discussed below.

Finally, it is worth emphasizing that for classical solutes $\phi^{\beta} \equiv 0$ and Eq. (11.17) represents an *exact* solution to the continuum electrostatics problem. To emphasize this point, we have performed numerical comparisons of MM/PCM calculations versus results obtained from the "adaptive Poisson–Boltzmann solver" (APBS) [5], which represents a recent implementation of the three-dimensional finite-difference approach. (The solvent's ionic strength was set to zero in the APBS calculations.) Results for amino acids, plotted in Fig. 11.2, show sub-kcal/mol differences in most cases, and differences of < 0.1 kcal/mol for the "$\mathbf{X} = \mathbf{DAS}$" version of SS(V)PE

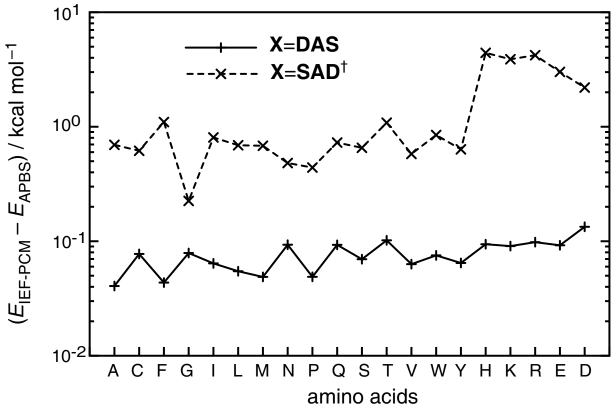

Figure 11.2 Comparison of total energies (on a logarithmic scale) for aqueous amino acids, where the solute is described using the AMBER99 force field and the solvent is a dielectric continuum. The continuum electrostatics problem is solved either by finite-difference solution of Poisson's equation using the APBS software [5], or else using two different forms of IEF-PCM ($\mathbf{X} = \mathbf{DAS}$ or $\mathbf{X} = \mathbf{SAD}^{\dagger}$, as described in Section 11.2.2.1). What is plotted is the difference $E_{\text{IEF-PCM}} - E_{\text{APBS}}$ between these two solutions. The APBS and IEF-PCM solute cavities are identical. APBS calculations used a 193 × 193 × 193 grid with a grid resolution of 0.1 Å, whereas IEF-PCM calculations used 590 Lebedev points per atomic sphere with Gaussian blurring.

that is our preferred implementation of this model, for reasons discussed below.

11.2.2 *Practical Considerations*

11.2.2.1 Matrix equations

Specific choices for how to construct and discretize the solute cavity are discussed below, but for now let us assume that this has been done, so that $\mathbf{\Gamma}$ has been turned into a discrete set of points $\mathbf{s}_i$, each with a well-defined surface area, a_i. The continuous surface charge $\widetilde{\sigma}(\mathbf{s})$ is thus replaced with a set of point charges q_i and Eq. (11.20) is converted into a set of linear equations

$$\mathbf{Kq} = \mathbf{Rv} \tag{11.22}$$

for the vector $\mathbf{q}$ of surface charges, with $v_i = \phi^\rho(\mathbf{s}_i)$. The matrices $\mathbf{K}$ and $\mathbf{R}$ depend upon the matrix representations of the operators $\hat{S}$, $\hat{D}$, and $\hat{D}^\dagger$.

Since $\hat{S}$ generates the electrostatic potential [Eq. (11.15)], it is clear that $S_{ij} = |\mathbf{s}_i - \mathbf{s}_j|^{-1}$ (in atomic units) for $i \neq j$, because then the quantity $S_{ij}q_j$ is the electrostatic potential due to q_j, evaluated at the point $\mathbf{s}_i$. The diagonal elements S_{ii} could in principle be obtained by evaluating the surface Coulomb integral in Eq. (11.15) over the area $a_i \subset \mathbf{\Gamma}$. For efficiency, however, the expression

$$S_{ii} = C\left(\frac{4\pi}{a_i}\right)^{1/2} f_i^{\mathrm{shape}} \tag{11.23}$$

is widely used, where $C \approx 1.06$ and f_i^{shape} is an (often omitted) shape factor [19]. This choice is based on the exact result $S_{ii} = (4\pi/a_i)^{1/2}$ for a uniform spherical surface grid.

The integral operator $\hat{D}$ is replaced by a matrix product $\mathbf{DA}$, where $\mathbf{A}$ is a diagonal matrix containing the areas a_i. The matrix elements of $\mathbf{D}$ are typically defined as [19]

$$D_{ij} = \begin{cases} -(2\pi + \sum_{k\neq i} D_{ik}a_k)/a_i & \text{for } i = j \\ -\mathbf{n}_j \cdot (\mathbf{s}_j - \mathbf{s}_i)|\mathbf{s}_j - \mathbf{s}_i|^{-3} & \text{for } i \neq j \end{cases}. \tag{11.24}$$

The off-diagonal matrix element is recognizable from the integrand in Eq. (11.19), whereas the diagonal elements are based upon a sum rule derived in Ref. [68]. (This sum rule proves to be problematic in modern, smooth discretization schemes, and the definitions of D_{ii} and S_{ii} will be modified below.)

One complication with the discretized PCM equation is that the discretized analogue of Eq. (11.21), which would read $\mathbf{DAS} = \mathbf{SAD}^\dagger$, is in general *not* satisfied, except in the special case of a spherical cavity. The discretized form of Eq. (11.17) is therefore ambiguous, because the operator $\hat{S}\hat{D}^\dagger$ appearing in that equation could be replaced by any linear combination $\hat{X} = a\hat{S}\hat{D}^\dagger + b\hat{D}\hat{S}$ so long as $a + b = 1$, but the corresponding matrix $\mathbf{X} = a\mathbf{SAD}^\dagger + b\mathbf{DAS}$ leads to different matrix equations for each choice of coefficients. In Chipman's original work on SS(V)PE [15], the choice $a = b = 1/2$ was suggested, as this leads to a symmetric matrix $\mathbf{K}$ and thus more efficient solution of Eq. (11.22). However, IEF-PCM calculations using the other two "obvious" choices ($a = 0$ and $b = 1$, or *vice versa*) have also been reported [24, 26, 51]. Only recently have the consequences of these choices been recognized [44]. In particular, for realistic molecular cavities, only the choice $\mathbf{X} = \mathbf{DAS}$ achieves the correct conductor limit ($\varepsilon \to \infty$), whereas $\mathbf{X} = \mathbf{SAD}^\dagger$ does not, nor does the symmetrized version $\mathbf{X} = (\mathbf{DAS} + \mathbf{SAD}^\dagger)/2$. A particular example is shown in Fig. 11.3, and an analytic proof is provided in Ref. [44].

As a result, we choose $\mathbf{X} = \mathbf{DAS}$ to define the $\mathbf{K}$ matrix. For definiteness, the forms of $\mathbf{K}$ and $\mathbf{R}$ for this version of SS(V)PE/IEF-

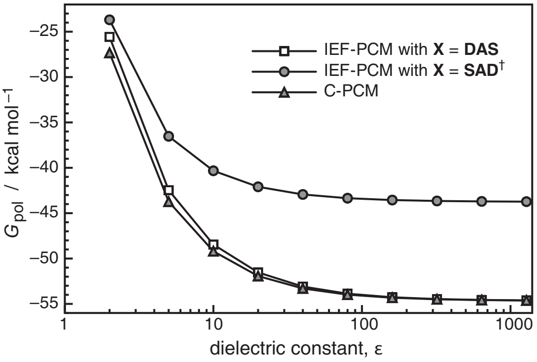

Figure 11.3 Electrostatic solvation energy for classical histidine as a function of dielectric constant. The C-PCM approach is free of the matrix $\mathbf{D}$ and achieves the correct conductor limit as $\varepsilon \to \infty$. Reprinted from Ref. [44]; copyright 2011 Elsevier.

Table 11.1 Matrices used in the equation $\mathbf{Kq} = \mathbf{Rv}$ for several different PCMs[a]

Method	Matrix **K**	Matrix **R**
SS(V)PE/ IEF-PCM	$\mathbf{S} - (f_\varepsilon/2\pi)\mathbf{DAS}$	$-f_\varepsilon(\mathbf{1} - \frac{1}{2\pi}\mathbf{DA})$
C-PCM/ GCOSMO	$\mathbf{S}$	$-\left(\frac{\varepsilon-1}{\varepsilon}\right)\mathbf{1}$
DESMO	$\mathbf{S}$	$-\mathbf{1} + \frac{1}{\varepsilon}\mathbf{M}$

[a] The factor $f_\varepsilon = (\varepsilon - 1)/(\varepsilon + 1)$ and the matrix $\mathbf{M}$ has elements $M_{ij} = \delta_{ij}\phi_\kappa^\rho(\mathbf{s}_i)/\phi_0^\rho(\mathbf{s}_i)$.

PCM are explicated in Table 11.1. Also listed in this table are the forms of **K** and **R** for the so-called *conductor-like* model, C-PCM [25]. This model is considerably simpler in that the matrix **D** is absent. C-PCM is identical to the *generalized conductor-like screening model* (GCOSMO) [78], and almost identical to the original COSMO [37]. (G)COSMO was introduced prior to SS(V)PE/IEF-PCM, based on *ad hoc* arguments and designed to achieve the correct $\varepsilon \to \infty$ limit. We will show below that this model differs from SS(V)PE/IEF-PCM only by terms of order ε^{-1}. Due to its simplicity, C-PCM is therefore our preferred model for high-dielectric solvents such as water.

11.2.2.2 Cavity construction and discretization

In order to obtain the matrix equations above, one must decide how to construct, and subsequently discretize, the cavity surface. The most widely used methods take the cavity to be a union of atom-centered spheres [77], as suggested in Fig. 11.1(a). The electrostatic solvation energy is quite sensitive to the radii of these spheres (it varies as $\sim R^{-1}$ in the Born ion model), and highly parameterized constructions that exploit information about the bonding topology [6] or the charge states of the atoms [31] are sometimes employed. The details of these parameterizations are beyond the scope of the present work, especially given that careful reconsideration of these parameters is probably necessary for classical biomolecular electrostatics calculations.

Having selected a set of atomic radii, these must next be turned into a discrete set of surface grid points. In QM/PCM calculations, the most popular approach has been the *generating polyhedra* (GEPOL) algorithm [1], which tessellates the surface of each sphere into a collection of small triangles or *tesserae*, using a 60-sided regular polygon. [An example is depicted in Fig. 11.1(b).] A discretization charge q_i is placed in the center of each tessera. One difficulty with this procedure is the complicated geometry of how these triangles should change as a function of the atomic coordinates, which significantly complicates the formulation of analytic energy derivatives [23]. Furthermore, the GEPOL discretization has only a limited degree of systematic improvability [1].

A more appealing procedure is to use atom-centered Lebedev angular quadrature grids [42, 70, 87], which are designed as exact quadratures through a given order in spherical harmonic functions and are therefore systematically improvable [42]. Figure 11.1(c) shows an example of double-stranded DNA, discretized using 50 Lebedev points per atomic sphere.

For QM/PCM applications, an appealing alternative to carefully parametrized atomic radii is a one-parameter cavity construction in which the cavity is defined as an isocontour of the QM electron density [12, 19, 30]. Unfortunately, analytic energy gradients have never been reported for such a construction (they are complicated by the fact that $\mathbf{n}_\mathbf{s}$ becomes density-dependent [19, 30]), and in any case such an approach is not possible in MM/PCM or QM/MM/PCM calculations.

In the context of generalized Born models, Friesner and coworkers [88] have experimented with a cavity defined as an isosurface of a *pseudo*-density, $d(\mathbf{r})$, that is expressed as a sum of atom-centered Gaussians:

$$d(\mathbf{r}) = \sum_{K}^{N_{\text{atoms}}} \exp\left(-B|\mathbf{r}-\mathbf{r}_K|^2/R_K^2\right). \qquad (11.25)$$

The parameter B controls what we term the “blobbiness” of the surface, and R_K is a Gaussian width parameter for the atom centered at $\mathbf{r}_K$. An isosurface contour value $d = e^{-B}$ ensures that the isosurface coincides with the radius R_K for a single, isolated atom (Born ion model). A discretization grid for the isosurface can be obtained using the *marching cubes algorithm* [52], arriving at a tessellated surface grid made up of triangles. This construction will be used in macromolecular PCM calculations presented later in this chapter.

An issue with all of these discretization schemes—except possibly the genuine isodensity surface that is not considered in this work—is the fact that the solvation energy is a *discontinuous* function of the atomic coordinates, because discretization points appear and disappear as the overlap between atomic spheres changes. (In principle, the energy also loses rotational invariance upon discretization, but we fund that this problem is not serious [42]). The discontinuity problem, which is shared by finite-difference Poisson–Boltzmann solvers, has recently been resolved in the context of PCMs, with the development of intrinsically smooth discretization algorithms [42, 70, 76, 87]. These are discussed in Section 11.4.1.

11.2.2.3 Beyond electrostatics

This chapter is devoted strictly to a discussion of electrostatic interactions between solute and continuum solvent; non-electrostatic interactions are not discussed beyond a brief mention here. Such interactions include the *cavitation energy* (a destabilizing interaction representing the energy required to carve a molecule-shaped void out of the continuum); *dispersion* (the stabilizing van der Waals interaction); specific interactions such as hydrogen bonding; and changes to the solvent structure upon insertion of the solute. To some extent, these effects can be captured (especially in QM/PCM calculations) by including one or more explicit solvent molecules in the atomistic region, albeit at increased cost.

Simple corrections for non-electrostatic interactions have been suggested, wherein atomic-specific parameters are used to describe cavitation, Pauli repulsion, and dispersion [2, 3, 20, 84]. These non-electrostatic interactions are then added to G_{pol} to obtain the total solvation energy. The most successful examples of this approach are the so-called SM*x* models of Cramer and Truhlar [27], most of which are not actually PCMs per se but rather generalized Born models. However, one such model ("SMD") has recently been parameterized for use with IEF-PCM electrostatics [55] and exhibits mean errors of $\lesssim$ 1 kcal/mol as compared to experimental solvation energies for small neutral molecules, although the mean error for ions is 4 kcal/mol. More recently, Pomogaeva and Chipman [67] suggested “more ab initio” forms for the various non-electrostatic interactions, and demonstrated performance equal to or exceeding that of the best SM*x* models, for aqueous solvation, with fewer empirical parameters.

All of the aforementioned examples were developed in the context of QM/PCM calculations and would undoubtedly need to be reconsidered, or at least re-parameterized, for classical solutes.

11.3 New Models and Insights

In this section we discuss new *theory*, as opposed to the new algorithms that are discussed in Section 11.4. Recent theoretical developments include new methods for incorporating salt effects into PCMs (Section 11.3.1) and new connections between PCM and generalized Born models (Section 11.3.2), which may help to improve the latter.

11.3.1 *Generalized Debye–Hückel Theory*

The discussion of continuum electrostatics in Section 11.2.1 was limited to solution of Poisson’s equation, which can be achieved exactly (for classical solutes) or to a good approximation (for QM solutes) using PCMs. In biomolecular applications, however, the objective is usually solution of the Poisson–Boltzmann equation [4, 33]. For low concentrations of dissolved ions, the latter is often replaced by the *linearized* Poisson–Boltzmann equation (LPBE),

$$\begin{cases} \hat{\nabla}^2\phi(\mathbf{r}) = -4\pi\rho(\mathbf{r}) & \text{for } \mathbf{r} \text{ inside of } \mathbf{\Gamma} \\ (\hat{\nabla}^2 - \kappa^2)\phi(\mathbf{r}) = -4\pi\rho(\mathbf{r})/\varepsilon & \text{for } \mathbf{r} \text{ outside of } \mathbf{\Gamma}. \end{cases} \tag{11.26}$$

Here, $\kappa = 8\pi e^2\mathcal{I}/\varepsilon k_B T$ is the inverse Debye length, for a solution whose ionic strength is $\mathcal{I}$. The LPBE was derived by Debye and Hückel [28], and its analytic solution for a spherical cavity forms the basis of the eponymous theory. In this section, we discuss how PCMs can be modified to solve the LPBE, but first we present an alternative derivation of GCOSMO that will be useful in this respect.

11.3.1.1 Alternative derivation of C-PCM/GCOSMO

The original derivation of COSMO was based on taking $\varepsilon \to \infty$, in which case $\mathbf{Sq} = -\mathbf{v}$ is the exact solution to the molecular electrostatics problem, then rescaling the solution for finite ε [37, 78]. Recently, we presented a much more satisfying derivation [43]. Our approach starts from an *ansatz*

$$\phi(\mathbf{r}) = \begin{cases} \phi_0^\rho(\mathbf{r}) + \phi_0^\sigma(\mathbf{r}) & \text{for } \mathbf{r} \text{ inside of } \boldsymbol{\Gamma} \\ \phi_0^\rho(\mathbf{r})/\varepsilon & \text{for } \mathbf{r} \text{ outside of } \boldsymbol{\Gamma} \end{cases} \tag{11.27}$$

for the electrostatic potential, consisting of a solute contribution ϕ_0^ρ and a second contribution ϕ_0^σ arising from the induced surface charge. The subscripts on these quantities indicate that $\kappa = 0$, which will later be replaced by nonzero κ in the event of dissolved ions. Enforcing the condition that $\phi(\mathbf{r})$ must be continuous across $\boldsymbol{\Gamma}$, Eq. (11.27) immediately affords the C-PCM/GCOSMO equation in Table 11.1 [43]. On the other hand, this *ansatz* cannot be made to satisfy the jump boundary condition in Eq. (11.7).

Noting that $\hat{\partial}_{\mathbf{s}}\,\phi_0^\rho$ is continuous across $\boldsymbol{\Gamma}$, the reaction-field potential must be solely responsible for the jump in the electric field [16]. This condition can be expressed as [43]

$$\hat{\partial}_{\mathbf{s}}\,\phi_{\text{rxn}}(\mathbf{s}^+) = -\left(\frac{\varepsilon - 1}{\varepsilon}\right)\hat{\partial}_{\mathbf{s}}\,\phi_0^\rho(\mathbf{s}) + \frac{1}{\varepsilon}\,\hat{\partial}_{\mathbf{s}}\,\phi_{\text{rxn}}(\mathbf{s}^-). \tag{11.28}$$

The normal derivative of the *ansatz* in Eq. (11.27) lacks the second term in Eq. (11.28); hence, C-PCM/GCOSMO engenders errors of order ε^{-1}, as compared to an exact treatment of classical electrostatics. Such errors are negligible in water [44], as seen in Fig. 11.3.

11.3.1.2 DESMO and ion exclusion

The above derivation of GCOSMO immediately suggests how this model can be generalized to solvents with non-zero ionic strength, using a modified *ansatz* of the form

$$\phi(\mathbf{r}) = \begin{cases} \phi_0^\rho(\mathbf{r}) + \phi_0^\sigma(\mathbf{r}) & \text{for } \mathbf{r} \text{ inside of } \boldsymbol{\Gamma} \\ \phi_\kappa^\rho(\mathbf{r})/\varepsilon & \text{for } \mathbf{r} \text{ outside of } \boldsymbol{\Gamma} \end{cases} \tag{11.29}$$

and enforcing continuity of ϕ at the cavity surface but neglecting the jump boundary condition for the electric field [43]. In homage to (G)COSMO, we have called the resulting PCM the *Debye–Hückel-like screening model* (DESMO), and its basic working equation is given in Table 11.1. The only change, relative to GCOSMO, is the need to compute the screened electrostatic potential ϕ_κ^ρ at the cavity surface, in addition to the unscreened potential, ϕ_0^ρ. (The screened form simply uses the Yukawa potential $e^{-\kappa r}/\varepsilon r$ in place of the Coulomb potential $1/\varepsilon r$ that is used in the unscreened form.) DESMO represents the leading-order (in $1/\varepsilon$) approximation to the "screened" SS(V)PE and IEF-PCM models that have been developed to solve the LPBE [17, 22, 58]. Working equations for the latter models are more complicated and can be found in Ref. [43]. In high-dielectric solvents, however, DESMO incurs negligible error with respect to those models but retains the simplicity of (G)COSMO.

On the other hand, the screened SS(V)PE [17] and IEF-PCM [22, 58] treatments of the LPBE lack one important feature of the original Debye–Hückel theory, namely, a correction for the finite size of the dissolved ions. To understand this, let us recall the model problem considered by Debye and Hückel [28], which consists of a point charge q centered in a spherical cavity of radius R_{cav}, outside of which is the dielectric medium. The dissolved ions are assumed to have a finite radius R_{ion}, and their centers therefore cannot approach the charge q closer than a distance $R_{\text{cav}} + R_{\text{ion}}$. This manifests as an *ion exclusion layer* (Stern layer) for $R_{\text{cav}} \leq r \leq R_{\text{cav}} + R_{\text{ion}}$, and a long-range electrostatic potential (for $r > R_{\text{cav}} + R_{\text{ion}}$) of the form

$$\phi^{\text{DH}}_{\text{long-range}}(r) = q\left(\frac{e^{-\kappa r}}{\varepsilon r}\right)\underbrace{\frac{e^{\kappa(R_{\text{cav}}+R_{\text{ion}})}}{1+\kappa(R_{\text{cav}}+R_{\text{ion}})}}_{\gamma}. \tag{11.30}$$

This potential has the form of the charge q times a screened Coulomb potential (Yukawa potential, $e^{-\kappa r}/\varepsilon r$) multiplied by what we have termed an *ion exclusion factor*, γ [43]. This suggests that ion exclusion might be incorporated into DESMO using an *ansatz* of the form [43]

$$\phi_\kappa^\rho(\mathbf{s}_i) = \underbrace{\frac{e^{\kappa(R_{\text{ion}}+R_I)}}{1+\kappa(R_{\text{ion}}+R_I)}}_{\gamma_I}\int_{\mathbb{R}^3} d\mathbf{r}\,\rho(\mathbf{r})\frac{e^{-\kappa|\mathbf{s}_i-\mathbf{r}|}}{|\mathbf{s}_i-\mathbf{r}|} \tag{11.31}$$

for discretization points $\mathbf{s}_i$ located on the Ith atomic sphere.

Table 11.2 Solvation energies[a] for a model consisting of 25 disjoint spheres with a point charge in each[b]

ε	κ^{-1}/ Å	G_{pol}/ kcal mol^{-1}	Error / kcal mol^{-1} DESMO with γ_I	DESMO without γ_I	SS(V)PE/ IEF-PCM
4	∞	−2899.49	−0.01	−0.01	0.98
20	∞	−3672.69	0.00	0.00	0.27
80	∞	−3817.67	0.00	0.00	0.07
4	25	−2914.09	−0.26	−1.39	0.70
20	25	−3675.61	−0.05	−0.28	0.21
80	25	−3818.40	−0.01	−0.07	0.06
4	5	−3035.50	−0.41	−20.92	0.37
20	5	−3699.90	−0.08	−4.18	0.13
80	5	−3824.47	−0.02	−1.04	0.04
4	3	−3122.65	−0.22	−44.47	0.30
20	3	−3717.32	−0.05	−8.90	0.09
80	3	−3828.83	−0.01	−2.22	0.03

[a] Computed from an exact analytic solution of the LPBE,[53] for various values of ε and κ with $R_{\text{ion}} = 0$.

[b] Reprinted from Ref. [43]; copyright 2011 American Institute of Physics.

Table 11.2 presents some results for a simple model consisting of 25 disjoint spheres immersed in a salty dielectric, with a point charge centered in each sphere but with mobile ions of zero size ($R_{\text{ion}} = 0$). The LPBE can be solved analytically for this toy problem [53], which is intended to explore how continuum methods might perform for modeling protein–protein interactions in solution. Solvation energies obtained from the LPBE are on the order of −3000 kcal/mol or more and DESMO, with ion exclusion factors γ_I as suggested in Eq. (11.31), reproduces these energies to within 0.4 kcal/mol in each case. Without the ion exclusion factors, however, very large errors can result. Interestingly, DESMO with ion exclusion is actually slightly more accurate than the versions of SS(V)PE and IEF-PCM that have been suggested for use with the LPBE (and which are equivalent for a classical solute). Errors in SS(V)PE/IEF-PCM are a reflection of the fact that this method is fundamentally approximate in the presence of outlying charge, which arises here not from tails of a wave function but rather from the presence of disjoint solute cavities [43].

In the future, DESMO should be tested with finite ion size and compared to numerical solution of the LPBE using a cavity surface (defined by the van der Waals radii R_I) that does not coincide with the ion exclusion surface (defined by $R_I + R_{\text{ion}}$). Finite ion size has incorporated into Generalized Born models, however, via the ion exclusion factors in Eq. (11.31) [46]. These models are discussed in the next section.

11.3.2 *Connections to Generalized Born Models*

The most widely used implicit solvation models in biomolecular simulations are probably the *Generalized Born* (GB) models [63, 79], because they are computationally inexpensive and amenable to analytic forces. GB models posit that the electrostatic solvation energy can be expressed in the form

$$G_{\text{pol}}^{\text{GB}} = -\frac{1}{2}\left(\frac{\varepsilon - 1}{\varepsilon}\right)\sum_{i,j}\frac{q_i q_j}{f_{ij}}, \tag{11.32}$$

where q_i and q_j are the MM point charges on atoms i and j, and the quantity f_{ij}^{-1} is an effective Coulomb potential. In the case of a spherical cavity, f_{ij} has two known limits: [34, 45]

$$\begin{array}{ll} f_{ij}^{\text{sphere}} \to r_{ij} & \text{as } r_{ij} \to \infty \\ f_{ij}^{\text{sphere}} \to \left(r_{ij}^2 + R_i^{\text{perf}} R_j^{\text{perf}}\right)^{1/2} & \text{as } r_{ij} \to 0 \end{array}. \tag{11.33}$$

(We use atomic units in this discussion, so r_{ij}^{-1} is the Coulomb potential between charges q_i and q_j.) The limit $r_{ij} \to \infty$ corresponds to the solvable model problem of two Born ions in non-overlapping spherical cavities, while the limit $r_{ij} \to 0$ becomes valid when q_i and q_j occupy the same spherical cavity [34].

The quantity R_i^{perf} in Eq. (11.33) denotes the “perfect” effective Born radius for q_i [64], the efficient and accurate computation of which is a major part of the development of GB models. To define R_i^{perf}, let $G_{\text{pol},ii}^{\text{PE}}$ denote the exact polarization energy (obtained by solving Poisson’s equation) for the atomic charge q_i in a cavity representative of the entire molecule. (That is, we turn off all charges

$q_{j\neq i}$ but leave the cavity unchanged.) Then the definition

$$R_i^{\mathrm{perf}} = -\frac{1}{2}\left(\frac{\varepsilon - 1}{\varepsilon}\right)\frac{q_i^2}{G_{\mathrm{pol},ii}^{\mathrm{PE}}} \tag{11.34}$$

assures the correct Born ion limit in Eq. (11.32) [45].

Equation (11.34) is not a practical construction of the perfect radii, because it requires solving Poisson's equation for the entire molecule, once per atom. Computationally tractable approximations have been proposed and tested [48, 62], but will not be discussed here. Instead, we discuss a formal connection that was discovered recently between PCMs and the GB *ansatz* [45], and propose PCMs as a means to generate benchmark data for testing the various approximations that go into GB models.

The key breakthrough is to recognize that when the solute is composed of point charges (or higher-order classical multipoles [45]), each solute charge's contribution to the ASC can be treated individually within the PCM formalism. As a result, the total electrostatic solvation energy assumes a pairwise-additive form. Equating this energy with Eq. (11.32) affords a formal expression for the *exact* effective Coulomb operator for GB theory: [45]

$$\frac{1}{f_{ij}} = \frac{1}{q_i q_j}\int_{\Gamma} d\mathbf{s}\, \phi_i^{\rho}(\mathbf{s})\, \hat{C}_{\varepsilon}^{-1}\, \phi_j^{\rho}(\mathbf{s}). \tag{11.35}$$

Exact perfect radii are given by $R_i^{\mathrm{perf}} = f_{ii}^{-1}$. The quantity $\phi_i^{\rho}(\mathbf{s})$ in Eq. (11.35) denotes the electrostatic potential at the point $\mathbf{s}$ that is generated by the solute charge q_i, and the operator $\hat{C}_{\varepsilon}$ can be expressed in terms of the PCM operators $\hat{S}$ and $\hat{D}$ introduced above [45]. The subscript in $\hat{C}_{\varepsilon}$ is intended as a reminder that this operator depends explicitly on the dielectric constant, so that f_{ij}^{-1} *cannot* be independent of ε, as is assumed in most (though not all [72]) GB models. We have argued [45] that the only reasonable, ε-independent choice is the $\varepsilon \to \infty$ limit (especially given the importance of aqueous solvation), which has the added benefit of simplifying the operator $\hat{C}_{\varepsilon}$, since IEF-PCM reduces to C-PCM in that limit.

Thus, we have demonstrated a formal equivalence between PCM and GB calculations, wherein perfect radii and exact values of f_{ij} can be computed from PCM calculations. Exact f_{ij} values are defined only in a pairwise way, for each pair of atoms in a macromolecule, and the key to an accurate GB model is to pick an analytic functional form that can interpolate between the two limits in Eq. (11.33) while fitting the pairwise f_{ij} data. A commonly used form for the analytic interpolating function is [65]

$$f_{ij} = \sqrt{r_{ij}^2 + R_i R_j \Psi_{ij}}, \tag{11.36}$$

where Ψ_{ij} is some function of r_{ij} and the atomic radii R_i and R_j. (The latter are generally some approximations to the perfect radii.) The form

$$\Psi_{ij}^{\mathrm{Still}} = \exp(-r_{ij}^2/cR_i R_j) \tag{11.37}$$

is often used, with $c = 4$ in the original GB model of Still et al. [74]

We have used C-PCM calculations to obtain a data set of f_{ij} values and perfect radii for a small collection of proteins, using Eq. (11.35) [45]. Figure 11.4 plots the pairwise Ψ_{ij} data obtained for one of these proteins, which consists of 515 atoms for a total of 132,355 values of $\Psi_{ij}^{\mathrm{C\text{-}PCM}}$ with $i \neq j$. Although the functional form originally proposed by Still et al. [74] grossly conforms to the contours of the data, there appears to be room for improvement.

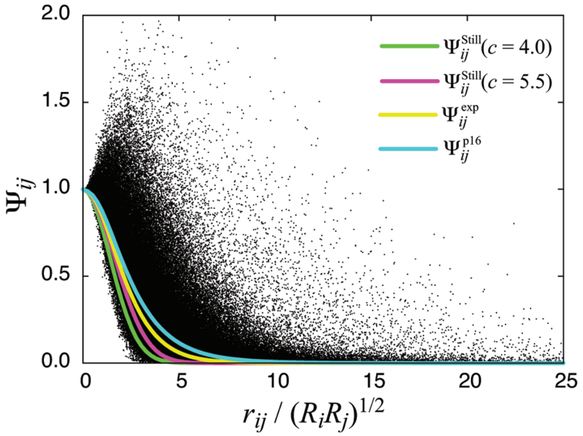

Figure 11.4 Exact values of Ψ_{ij} (black dots) for all pairs of atoms in the protein 1AJJ, obtained from C-PCM calculations with perfect radii R_i and R_j. The colored curves depict various analytic interpolating functions.

Setting

$$\Psi_{ij} = \frac{2\,\Omega_{ij} r_{ij}}{(R_i R_j)^{1/2}} + \Omega_{ij}^2 \tag{11.38}$$

in Eq. (11.36) affords $f_{ij} = r_{ij} + \Omega_{ij}(R_i R_j)^{1/2}$, where Ω_{ij} is a new pairwise interpolating function having limits

$$\begin{aligned} \Omega_{ij} &\to 1 \ \text{ as } r_{ij}/R_i R_j \to 0 \\ \Omega_{ij} &\to 0 \ \text{ as } r_{ij}/R_i R_j \to \infty. \end{aligned} \tag{11.39}$$

The quantities $(R_i R_j)^{1/2}$ that are needed to obtain Ω_{ij} and f_{ij} can be computed outside of the pairwise GB loop, and one may seek a form for Ω_{ij} that does not require calls to the exponential or square root functions, both of which are required when the "canonical" form of Ψ_{ij} [Eq. (11.37)] is used. An example is the function

$$\Omega_{ij}^{\mathrm{p16}} = \left[1 + \frac{\xi\, r_{ij}}{16(R_i R_j)^{1/2}}\right]^{-16}, \tag{11.40}$$

which is a 16th-order approximation to the function in Eq. (11.37), but which can be evaluated using only a small number of floating point operations [45]. In Ref. [45], we fit the parameter ξ (along with the truncation order, $p = 16$) to a protein training set. Visual inspection of the various interpolating functions that are plotted in Fig. 11.4 suggests that the function Ψ_{ij}^{p16} obtained from Eq. (11.40) does indeed fit the data better than the function suggested by Still et al. [74], although the enormous number of data points in the figure somewhat obscures the true spread of the data. In any case, an exact data set of Ψ_{ij} values has never been available before, so the utility of PCMs in re-parameterizing GB models seems clear.

Figure 11.5(a) shows contour plots of two-dimension histograms that count the number of $\Psi_{ij}^{\mathrm{C\text{-}PCM}}$ data points, as a function of the value of $\Psi_{ij}^{\mathrm{C\text{-}PCM}}$ and the dimensionless distance $r_{ij}/(R_i R_j)^{1/2}$. Various analytic interpolating functions Ψ_{ij} are superimposed on top of these contours. Much more so than the function $\Psi_{ij}^{\mathrm{Still}}$, the interpolating functions Ψ_{ij}^{exp} and Ψ_{ij}^{p16} that we suggested in Ref. [45] cut a path through where the number of data points is peaked. On the other hand, Fig. 11.5(b) superimposes these functions on top of two-dimensional histograms of the pairwise GB energy contributions,

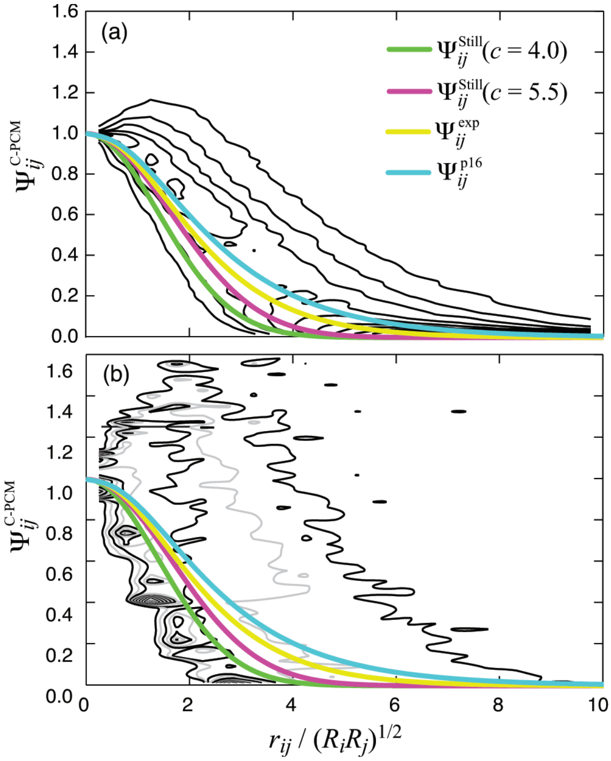

Figure 11.5 Analytic interpolating functions Ψ_{ij} superimposed on top of contours that represent (a) the total number of exact $\Psi_{ij}^{\mathrm{C\text{-}PCM}}$ data points for protein 1AJJ in the scatter plot of Fig. 11.4, and (b) the pairwise energies $|G_{\mathrm{pol},ij}^{\mathrm{GB}}| = |q_i q_j / 2 f_{ij}|$ associated with each data point. In (a), the contours are shown in black with the outermost contour representing 100 data points per bin and subsequent contours increasing in increments of 100 data points per bin. In (b), the contours are shown in alternating black and gray, with the outermost (rightmost) contour representing $G_{\mathrm{pol},ij}$ = 1 kcal/mol and subsequent contours increasing in increments of 1 kcal/mol. Bin widths are 0.5 and 0.026, respectively, in the dimensionless quantities $r_{ij}/(R_i R_j)^{1/2}$ and Ψ_{ij}.

$|G_{\mathrm{pol},ij}^{\mathrm{GB}}| = |q_i q_j|/2 f_{ij}$ [see Eq. (11.32)]. This figure seems to suggest that the new interpolating functions push Ψ_{ij} away from the energetically most important data points, so further improvement in the effective Coulomb operator may be possible.

However, error statistics confirm that the interpolating functions suggested in Ref. [45] do fit the $|G^{\mathrm{GB}}_{\mathrm{pol},ij}|$ data better than $\Psi^{\mathrm{Still}}_{ij}$. In fact, the function Ψ^{p16}_{ij} [Eq. (11.40)] actually *reduces* the errors in GB solvation energies while simultaneously *accelerating* the calculations [45]. To wit, when quasi-perfect "R6*" radii [62], which can be computed cheaply, are used in place of the perfect radii that are only available in benchmark calculations, the mean absolute error in $G^{\mathrm{GB}}_{\mathrm{pol}}$ as compared to C-PCM benchmarks is reduced from 8.7 kcal/mol for the canonical GB operator $\Psi^{\mathrm{Still}}_{ij}$ to 5.1 kcal/mol for the interpolating function Ψ^{p16}_{ij}. At the same time, use of Ψ^{p16}_{ij} reduces the cost by a factor of three relative to the canonical GB model based on $\Psi^{\mathrm{Still}}_{ij}$ [45]. The new interpolating function can be "dropped in" to existing MD codes with GB capability, and given the sizable speed-ups that we have observed, we advocate extensive further testing of the GB kernel in Eq. (11.40) and related functions.

Finally, let us briefly mention the incorporation of salt effects into GB models. It is recognized that standard GB models tend to exaggerate the importance of the salt, likely due to neglect of the ion exclusion layer [73]. Empirical scaling of κ has been suggested as a remedy [71, 73]. Alternatively, however, the DESMO ion exclusion factors [γ_I in Eq. (11.31)], in conjunction with the formal connection between PCMs and the GB *ansatz*, can be used to suggest "first principles" corrections to GB models that incorporate salt effects [46]. Several new GB models that incorporate salt effects were suggested in Ref. [46], based on formal connections to DESMO, and shown to be only slightly less accurate than methods that use an empirical scaling factor for κ. As such, these new models may serve as starting points for future development of salty GB models.

11.4 Advances in Algorithms

In this section we focus on technical rather than theoretical developments, but ones that are absolutely essential if PCMs are going to be brought to bear on macromolecules.

11.4.1 *Intrinsically Smooth Discretization*

An issue with the PCM formalism introduced in Section 11.2.2.1 is that the electrostatic energy is in general a discontinuous function as the solute atoms are displaced, because the number and size of the surface tesserae may change as a function of solute geometry. A similar problem is suffered by finite-difference Poisson–Boltzmann solvers, and the "solution" in those cases (in order to achieve stable forces for MD simulations, for example) is tight thresholding and/or some kind of interpolation between grid points [80–83].

The situation is simpler in the case of PCMs, where only the cavity surface (and not the whole of three-dimensional space) needs to be discretized. A switching function of the form

$$F_i = \prod_{K,\, i \notin K}^{\text{atoms}} f(\mathbf{s}_i, \mathbf{r}_K) \tag{11.41}$$

can be used to attenuate the contribution to the PCM equations from the ith surface grid point $\mathbf{s}_i$, as that point passes through a narrow buffer region around the solute cavity surface, which is defined in terms of spheres centered at the atoms $\mathbf{r}_K$. The quantity f in Eq. (11.41) is some function that changes smoothly from 0 to 1 across this buffer region [41, 42, 87].

This simple procedure, however, leads to problems for certain cavity surface definitions [41, 44]. In particular, while the switching function can provide a potential energy surface that is rigorously smooth in the mathematical sense of having continuous derivatives, those derivatives may fluctuate wildly as a function of the solute coordinates [41]. These oscillations are actually *exacerbated* by the switching function, which allows the surface charges to approach one another more closely than would be the case if they were simply turned on or off discontinuously at the cavity surface boundary. The result can be sharp singularities in the energy along a geometry optimization [41] or MD trajectory [42]. For example, Fig. 11.6 shows harmonic vibrational spectra for a relatively large system (adenine with 52 explicit water molecules, all embedded within a polarizable continuum) computed using the *fixed points with variable areas* (FIXPVA) discretization algorithm of Ref. [75]. The FIXPVA approach achieves a smooth potential surface by applying

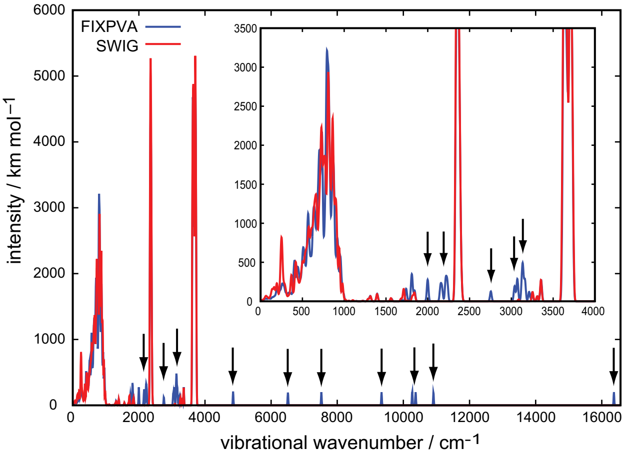

Figure 11.6 Harmonic vibrational spectra for (adenine)$(H_2O)_{52}$ computed at the MM/PCM level, using two different smooth implementations of the C-PCM solvation model. Harmonic frequencies were computed via finite difference of analytic energy gradients and convolved with 10 cm^{-1} Gaussians. Arrows indicate FIXPVA peaks with no obvious SWIG analogues, and the inset blows up the region of the spectrum < 4000 cm^{-1}. Reprinted from Ref. [41]; copyright 2010 American Chemical Society.

a switching function to the surface areas a_i, scaling them smoothly to zero as the point $\mathbf{s}_i$ passes through the buffer region and into the cavity. However, sharp fluctuations in the FIXPVA energy gradient (which, we emphasize, is a continuous function) manifest as anomalously large vibrational frequencies of up to 16,000 cm^{-1}!

A solution to this problem is to use Gaussian blurring of the surface charges [41, 87], in which each discretization charge q_i is replaced by a Gaussian function

$$g_i(\mathbf{r}) = q_i \left(\frac{\zeta_i^2}{\pi}\right)^{3/2} \exp\left(-\zeta_i^2 |\mathbf{r} - \mathbf{s}_i|^2\right). \qquad (11.42)$$

The width parameters ζ_i are chosen so as to approximate a uniform surface charge in the case of a single point charge centered in a spherical cavity [87], and are fixed parameters once the number of Lebedev discretization points per sphere is specified. The matrix elements of **S** are then

$$S_{ij} = \begin{cases} \zeta_i (2/\pi)^{1/2} F_i^{-1} & i = j \\ \mathrm{erf}(\zeta_{ij} s_{ij})/s_{ij} & i \neq j \end{cases} \tag{11.43}$$

where $s_{ij} = |\mathbf{s}_i - \mathbf{s}_j|$ and $\zeta_{ij} = \zeta_i \zeta_j (\zeta_i^2 + \zeta_j^2)^{1/2}$. The off-diagonal element is simply the Coulomb interaction between two Gaussians, while the diagonal element S_{ii} consists of the $\mathbf{s}_j \to \mathbf{s}_i$ limit of that Coulomb interaction, multiplied by F_i^{-1}. The latter factor guarantees a smooth potential surface by ensuring that Eq. (11.22) has a null space corresponding to grid points for which $F_i = 0$ [42]. As such, it is safe to discard points for which F_i falls below a given threshold, thus reducing the dimension of the linear system in Eq. (11.22).

The matrix elements of **D** require some care. Off-diagonal elements can be computed from S_{ij} according to [42]

$$D_{ij} = \mathbf{n}_j \cdot \frac{\partial S_{ij}}{\partial \mathbf{s}_i}. \tag{11.44}$$

Diagonal elements are often computed by means of a sum rule [see Eq. (11.24)] [68], but this relationship is no longer rigorously valid in the presence of attenuated grid points that may actually reside *within* the cavity. This can lead to serious numerical problems in the context of smooth PCMs [44]. Instead, we take $D_{ii} = S_{ii}/2R_I$, where R_I is the radius of the atomic sphere on which the point $\mathbf{s}_i$ resides [42]. This formula is correct for a spherical surface of radius R_I [58].

The combination of switching function, Gaussian blurring, and Lebedev discretization, with these choices for **S** and **D**, constitutes what we have called the *switching/Gaussian* (SWIG) discretization approach [42]. All of the required matrix elements are analytically differentiable functions of the atomic coordinates, and the derivatives are rigorously continuous and free of unphysical oscillations. As compared to the FIXPVA approach, spurious vibrational frequencies are absent (see Fig. 11.6). In QM/PCM calculations, SWIG discretization preserves the variational property of the original operator formalism, namely, that the solute/continuum electrostatic interaction necessarily lowers the SCF energy [42]. When the $\mathbf{X} = \mathbf{DAS}$ version of the **K** matrix is employed in IEF-PCM (or when C-PCM is used instead), SWIG discretization yields the same solvation energies, to very high accuracy, as compared to traditional discretizations [44].

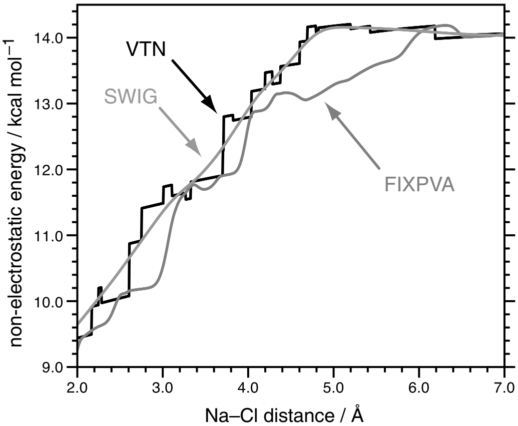

Figure 11.7 Non-electrostatic solvation energy for a QM/PCM calculation of aqueous NaCl, as a function of the distance between the two atoms. The model assumes that the non-electrostatic energy is proportional to the exposed cavity surface area, which jumps in discontinuous steps for VTN discretization. The SWIG discretization smoothly interpolates through these steps, whereas the FIXPVA discretization achieves smoothness by scaling the tesserae surface areas, leading it to underestimate the total surface area. Adapted from Ref. [41]; copyright 2010 American Chemical Society.

Within the SWIG-PCM approach, the exposed cavity surface area is also a rigorously smooth function of the atomic coordinates, even though the "seams" between atomic spheres are no longer sharp cusps, and discretization points within the buffer zone do contribute to the total surface area, albeit with some attenuation. The fact that surface areas are smooth is important because the non-electrostatic energy is often parameterized in terms of the cavity surface area [2, 3, 20, 47, 84]. In biomolecular applications, so-called MM/PBSA methods [36, 38] also use the cavity surface area to obtain the non-electrostatic part of the solvation energy. Figure 11.7 plots the total surface area, obtained using various discretization methods, as two atomic spheres are pulled apart. The *variable tesserae number* (VTN) scheme [49] amounts to a discontinuous throwing away of grid points as they enter the solute cavity, and serves as a control experiment. As one would expect for such an approach, the VTN surface area consists of a sequence of discrete steps corresponding to addition or loss of grid points. Due to the simplicity of the model, these steps should represent an accurate (if discontinuous) approximation to the cavity surface area, and the SWIG surface area smoothly interpolates through these steps. FIXPVA discretization, while it does afford a rigorously smooth surface area, tends to underestimate the VTN surface area. We have shown that the lack of Coulomb regularization in FIXPVA necessitates a more aggressive switching function in order to avoid singularities [41, 44], with the result that many grid points are attenuated completely away, leading to "holes" in the cavity surface [41].

A major advantage of the inherently smooth SWIG-PCM approach, as compared to a grid-based finite-difference solution of Poisson's equation, is that stable forces for MD simulations are obtained by straightforward differentiation of Eq. (11.22) [42], even for fairly coarse discretization grids. In the finite-difference approach, one must resort to very fine grids, or else interpolation or other tricks, in order to render discontinuities small enough so that energy-conserving MD can be achieved [80–83]. Figure 11.8 plots the electrostatic solvation energy, G_{pol}, from an ab initio MD trajectory of glycine in implicit water [42]. In this simulation, the solute is initialized in its carboxylic acid tautomer, whereas the zwitterionic tautomer is more stable in aqueous solution. As such, the molecule spontaneously undergoes intramolecular proton transfer, evident in Fig. 11.8 by the dramatic change in G_{pol}. Close examination, however, reveals that G_{pol} is a perfectly smooth function of time, even during the course of this radical change in cavity geometry.

Stable forces are also obtained in MM/PCM simulations, as shown in Fig. 11.9 for a classical MD simulation of a segment of DNA bound to a histone protein. Here, the energy fluctuations amount to an acceptable $\sim 0.0001\%$ of the total energy.

11.4.2 *Linear Scaling and Parallelization*

The MM/PCM simulation in Fig. 11.9 consists of $\sim$124,000 surface discretization points (those for which $F_i > 10^{-6}$). As such, solution of Eq. (11.22) by matrix inversion or other $\mathcal{O}(N^3)$ methods is clearly infeasible, and a linear-scaling approach (in both memory and CPU

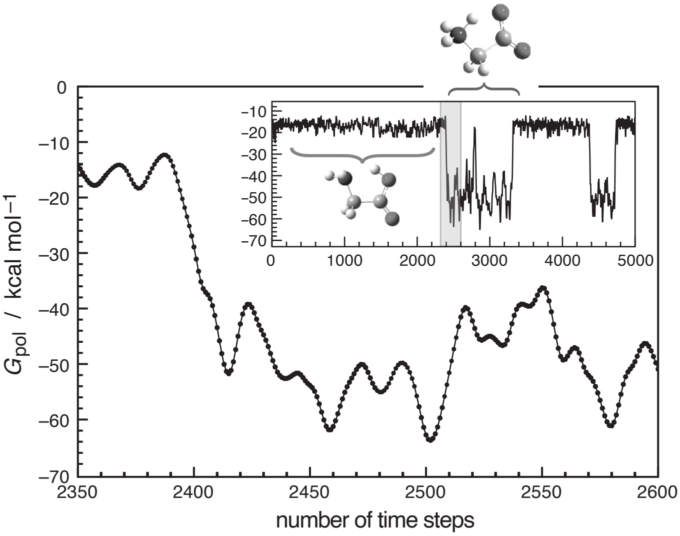

Figure 11.8 Ab initio (PBE0/6-31+G*) QM/PCM MD simulation of intramolecular proton transfer in aqueous glycine. The inset shows the total electrostatic solvation energy (G_{pol}), which is much larger for the zwitterionic tautomer than for the carboxylic acid tautomer. The shaded region has been enlarged in the main part of the figure, in order to demonstrate that the solvation energy is a smooth function of time despite the radical change in cavity shape upon proton transfer. The time step is ≈ 1 fs. Adapted from Ref. [42]; copyright 2010 American Institute of Physics.

time) is required. Such algorithms have been reported [25, 69], and parallelization has been discussed as well [29]. Our approach is described here for the first time, although versions of it have actually been available in the Q-CHEM software [39] since v. 3.2. Our strategy involves (bi)conjugate gradient solvers for linear equations, which do not require explicit formation of matrices such as **S** and **D**; a treecode version [50] of the fast multipole method [32]; and parallelization using both OpenMP and MPI. The discussion below pays particular attention to scalability and to the parallelization strategy, focusing on practical considerations as the system size is scaled up, and on how appropriate choices can be made for optimal efficiency at different points along the way.

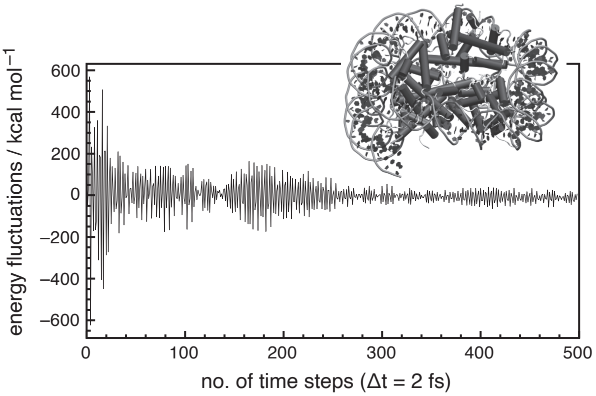

Figure 11.9 Energy fluctuations in an MM/PCM MD simulation of a segment of DNA bound to a histone protein. The solute (DNA + protein) consists of 21,734 AMBER99 atoms and the cavity surface is discretized using ≈ 124,000 surface charges. After the initial equilibration period, energy fluctuations amount to ≈ 0.0001% of the total energy.

11.4.2.1 Conjugate gradient solvers

The straightforward way to solve Eq. (11.22) is by constructing the matrix $\mathbf{K}^{-1}\mathbf{R}$, or more realistically by some equivalent factorization procedure such as LU decomposition. Even in an iterative SCF procedure, this needs to be done just once per molecular geometry. Nevertheless, this operation scales as $O(N_{\text{grid}}^3)$ and becomes the bottleneck surprisingly quickly in QM/PCM calculations, especially for dense discretization grids. The SWIG discretization procedure exacerbates the cost of QM/PCM calculations, both by increasing the number of grid points (we retain all $\mathbf{s}_i$ for which $F_i > 10^{-6}$) and also because it requires evaluation of three-index Gaussian integrals, $(g_i|\mu\nu)$. For example, in Hartree–Fock/6-31G* calculations on linear alkanes, using 302 Lebedev points per atom (which is sufficient to converge the electrostatics to within ≈ 0.3 kcal/mol [42]), the PCM cost exceeds the QM cost starting around octane. For QM/MM/PCM calculations, the $O(N_{\text{grid}}^2)$ cost in memory can also become a serious limitation.

A solution is to use Krylov subspace methods, such as the conjugate gradient (CG) method, the biconjugate gradient (BiCG) method, or other variants. The former is appropriate for C-PCM and DESMO, where the matrix **K** is symmetric, and the BiCG algorithm can be used for the non-symmetric SS(V)PE/IEF-PCM case. The cost of these algorithms is dominated by matrix-vector products that scale as $\mathcal{O}(N_{\text{grid}}^2)$, although matrix-matrix multiplication is required for SS(V)PE/IEF-PCM, which brings that method's scaling back up to $\mathcal{O}(N_{\text{grid}}^3)$ if matrices are constructed explicitly.

These matrix multiplications can be avoided using a combination of the CG and BiCG algorithms to bypass construction of **K** [19]. In the first stage, the BiCG algorithm is used to solve the equation

$$\left(\mathbf{I} - \frac{f_\varepsilon}{2\pi}\mathbf{DA}\right)\mathbf{w} = -f_\varepsilon\left(\mathbf{I} - \frac{1}{2\pi}\mathbf{DA}\right)\mathbf{v} \tag{11.45}$$

for **w**. Following that, the equation $\mathbf{Sq} = \mathbf{w}$ is next solved for **q**, using the CG algorithm. The cost of this two-stage approach scales as $\mathcal{O}(N_{\text{grid}}^2)$.

The CG and BiCG algorithms are complicated by the presence of an inverse switching function in the definition of S_{ii} [Eq. (11.43)], which causes $S_{ii} \to \infty$ as $F_i \to 0$. Although in practice these matrix elements are discarded when F_i is smaller than some predetermined threshold, values of F_i just above threshold tend to inflate the condition number of **S**, which can cause numerical instabilities or slow convergence in CG/BiCG algorithms. Large condition numbers can be avoided by appropriate factorizations, for example

$$\mathbf{S} = \mathbf{S}_{\text{diag}}^{1/2}\left(\mathbf{1} + \mathbf{S}_{\text{diag}}^{-1/2}\mathbf{S}_{\text{off}}\mathbf{S}_{\text{diag}}^{-1/2}\right)\mathbf{S}_{\text{diag}}^{1/2}\,, \tag{11.46}$$

where $\mathbf{S}_{\text{diag}}$ and $\mathbf{S}_{\text{off}}$ represent the diagonal and off-diagonal parts of **S**, respectively. The factor in parentheses is symmetric and thus amenable to a CG approach, and ought to have a significantly smaller condition number than **S** because small F_i appear in the *numerator* of $\mathbf{S}_{\text{diag}}^{-1/2}$. For C-PCM, Eq. (11.46) can be used to obtain an intermediate equation

$$\left(\mathbf{1} + \mathbf{S}_{\text{diag}}^{-1/2}\mathbf{S}_{\text{off}}\mathbf{S}_{\text{diag}}^{-1/2}\right)\widetilde{\mathbf{q}} = -f_\varepsilon\,\mathbf{S}_{\text{diag}}^{-1/2}\mathbf{v} \tag{11.47}$$

that is solvable by CG techniques. Having solved this equation for $\widetilde{\mathbf{q}}$, the final C-PCM charges are $\mathbf{q} = \mathbf{S}_{\text{diag}}^{-1/2}\,\widetilde{\mathbf{q}}$. This strategy can also be used in the two-stage CG/BiCG calculation for SS(V)PE/IEF-PCM, as outlined above. The cost remains $\mathcal{O}(N_{\text{grid}}^2)$ in CPU time.

Pre-conditioning improves both the rate of convergence and the stability of CG algorithms, but its effectiveness depends sensitively on the nature of the pre-conditioner matrix, **M**. For solving linear equations $\mathbf{Ax} = \mathbf{b}$, the pre-conditioner should be selected such that $\mathbf{M}^{-1}\mathbf{A}$ has a smaller condition number than **A**, which usually implies that $\mathbf{M}^{-1}$ is some approximation to $\mathbf{A}^{-1}$. A common choice is to set **M** equal to the diagonal of **A**, but if the C-PCM equations are factored according to Eq. (11.47), this would make **M** a unit matrix and therefore pointless. We find that factorization according to Eq. (11.47), without pre-conditioning, exhibits superior convergence properties as compared to pre-conditioning using the diagonal of **A**.

Block-diagonal pre-conditioners, which can be easily diagonalized and stored in core memory, are also popular, and this is the approach that we take. The fast multipole method [32, 50] (FMM) that is described below is used to partition the surface discretization charges, and this partition suggests a natural block structure for **M**. We define the blocks of **M** to be equal to sub-blocks of **S** consisting of the "leaf boxes," which are the smallest partitions in the "octree" data structure of the FMM (see below). The maximum number of grid points in one of these leaf boxes is a controllable threshold (N_{thresh}) in the FMM procedure, and keeping $N_{\text{thresh}} \lesssim 200$ ensures that **M** can be rapidly diagonalized and efficiently stored. We use this pre-conditioner *without* the factorization in Eq. (11.47), and find that convergence is accelerated by about 20% for large systems, relative to other methods discussed here (Fig. 11.10).

11.4.2.2 Fast multipole method

Given that the matrix elements of both **S** and **D** are essentially just particle–particle interactions, the FMM algorithm [32] can be used to improve the scaling to either $\mathcal{O}(N)$ or $\mathcal{O}(N \log N)$, depending on the precise details. Our implementation (in Q-Chem [39]) is based on the octree Cartesian FMM developed by Krasny and co-workers [50], which recursively sub-divides space into eight cubes of equal size and computes a multipole expansion of the charges contained in each box. In our implementation, these sub-divisions cease when the number of particles in a box falls below a given threshold (N_{thresh}),

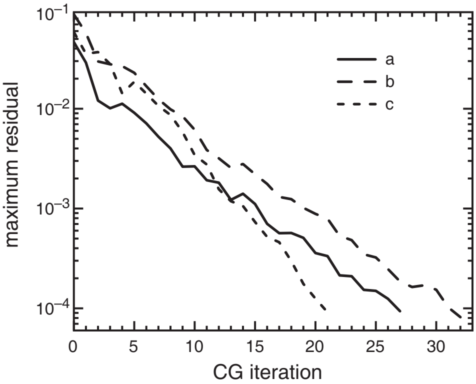

Figure 11.10 Convergence of AMBER99/C-PCM calculations for $(\text{alanine})_{1000}$ using SWIG discretization with 110 Lebedev points per atom. Method (a) uses the factorization in Eq. (11.47) with no pre-conditioning; method (b) uses a diagonal pre-conditioner; and method (c) uses a block-diagonal pre-conditioner. The convergence threshold was set to a maximum residual of 10^{-4} and electrostatics were computed exactly (no FMM).

or when the distance between the center and the vertices of the cube falls below another threshold (R_{thresh}).

Following construction of the octree data structure, the Coulomb interaction between a given particle (surface grid point) and the $N-1$ other particles is computed by traversing the octree, starting at the root box and then traveling downward into each box containing the given particle's coordinates, until the particle reaches a terminal ("leaf") box of the tree. At each level in this traversal, we sum the multipole interactions between the given particle and each "child" box that is within a certain multipole acceptance criterion (MAC), θ_{MAC} [50]. The criterion for using multipoles rather than explicit particle–particle interactions is

$$R_{c,\text{box}}/r_{ic} \leq \theta_{\text{MAC}}, \qquad (11.48)$$

where $R_{c,\text{box}}$ is the radius of the cth box (or "cluster" in the language of Ref. [50]), and r_{ic} is the distance between the ith tree-traversing particle and the center of the cth box. If the inequality in Eq. (11.48) is met, then the multipole expansion for the cth box is used for its interaction with tree-traversing particle i, otherwise the pairwise particle–particle interactions are computed explicitly. (The limit

$\theta_{\text{MAC}} = 0$ is equivalent to never accepting multipoles, and the limit $\theta_{\text{MAC}} = 1$ accepts all multipoles except for those inside of a box, so that particle–particle interactions are computed explicitly in the leaf boxes.) Tree traversal terminates when the particle reaches a leaf box, and in the leaf box, the particle computes its explicit pairwise interactions within the leaf box as well as any remaining boxes on the same level that are not within the MAC. At the end, one will have summed the pairwise interactions between particle i and all other particles, in $\mathcal{O}(\log N)$ effort. Repeating this for all N values of i leads to overall $\mathcal{O}(N \log N)$ scaling. No matrices are stored, so memory usage is $\mathcal{O}(N)$.

This FMM procedure readily interfaces with the CG solver, replacing all matrix-vector products involving $\mathbf{S}$ with the FMM using the Coulomb kernel r_{ij}^{-1} for point charges [50]. The matrix $\mathbf{D}$ can similarly be replaced by computing the electric field via FMM and appropriately multiplying the normal vectors. However, the Gaussian blurring used in SWIG discretization complicates the situation, because it involves a modified pairwise kernel, $\text{erf}(\zeta_{ij} r_{ij})/r_{ij}$. Although variants of FMM that are appropriate for Gaussian basis sets have been developed [86], we take a simpler approach, allowing Gaussian charges to interact as if they were point charges outside of a certain distance, R_{error}. Note that the pairwise Coulomb kernel S_{ii} reduces to the point-charge kernel in the limit $\zeta_{ij} \to \infty$. For a given PCM grid, we therefore select a minimum Gaussian width, $\zeta_i^{\min}$, whence the minimum possible value of ζ_{ij} is $\zeta_i^{\min}/\sqrt{2}$. The maximum error in the pairwise Gaussian charge interactions is then equal to $1 - \text{erf}(\zeta_i^{\min} R_{\text{error}}/2^{1/2})$, so R_{error} can be tuned to achieve a desired accuracy. The minimum box "radius", R_{thresh}, serves as a stopping criterion during the subdivision recursion, such that the minimum distance for meeting the MAC is $R_{\text{thresh}}/\theta_{\text{MAC}}$. Then, in order to ensure that we only accept multipoles beyond R_{error}, one must simply ensure that

$$R_{\text{thresh}}/\theta_{\text{MAC}} \leq R_{\text{error}}. \tag{11.49}$$

This applies to explicit charge–charge interactions, while the error in multipoles remains controlled by the order at which the multipole expansions are truncated. The explicit pairwise interactions between Gaussian charges in the leaf boxes are still computed explicitly using the $\text{erf}(\zeta_{ij} r_{ij})/r_{ij}$ kernel.

11.4.2.3 Parallelization strategies

The methods described above can be applied to any PCM discussed herein, including those based on SWIG discretization, and scale as $\mathcal{O}(N \log N)$ in CPU time and $\mathcal{O}(N)$ in memory. Here, we discuss a hybrid OpenMP/MPI paradigm that we have used to parallelize these calculations. Our focus is on C-PCM and its DESMO extension for salt effects, as these models are simpler and cheaper than SS(V)PE/IEF-PCM, and provide nearly identical results in water. Although parallel implementations of the FMM algorithm for MD simulations have been reported before [11, 40], achieving good scalability for the two-dimensional PCM electrostatics problem (with a cavity surface that is changing dynamically as the simulation evolves) may pose different challenges as compared to three-dimensional MD simulations. Indeed, our preliminary implementation suffers from some load-balancing issues, as discussed below, that have not yet been resolved.

Most of the work in an FMM implementation of C-PCM goes into computing the matrix-vector products in each CG iteration; these operations are a good target for multithreading with OpenMP. We store the entire FMM octree data structure in shared memory so that each thread can access the octree in a parallel fork. We do not store the Taylor series multipole expansion coefficients for the FMM but instead compute them on-the-fly as needed, via bootstrapping through recursion, whenever the MAC is met. This maintains a low memory footprint and also seems to benefit the performance with more cache hits within the function stack memory than by otherwise fetching the coefficients more slowly from heap memory. When the FMM is called, the multipole moments for each box in the octree are updated with the provided charges. We multithread this loop over boxes, providing it with the "guided" OpenMP threading schedule to account somewhat for load imbalance in the boxes. The FMM can then proceed to compute the electrostatic potential for each particle. We multithread the loop over the particle tree traversals, which must be done once for each discretization point, and each particle accumulates its potential into the shared memory vector that stores the result of the matrix-vector product. For t OpenMP threads, the ideal scaling for each CG iteration is $\mathcal{O}[(N \log N)/t]$.

We reserve distributed MPI parallelism for a different purpose. The Cartesian space of the surface grid is partitioned into separate regions and each region is owned by a single MPI rank that is responsible for storing the surface grid data (Cartesian coordinates, normal vectors, charges, etc.) for that region. In so doing, the memory storage for the surface grid is distributed, scaling roughly as $\mathcal{O}(N/p)$ for p MPI ranks, assuming an even load balance of the grid points. The regions can be determined in a number of ways, either automatically or fixed ahead of the calculation. The boundaries of the regions should not overlap, as this may degrade the accuracy of the FMM. The reason for this restriction is that we let each MPI rank build a distinct FMM tree for its region. This provides parallelization over the number of regions, which is in addition to the tree traversal parallelization provided by the OpenMP multithreading.

Because the surface grid is distributed, a given grid point is available to only one MPI rank at first. In order to compute electrostatic interactions between grid points belonging to different MPI ranks, it is therefore necessary to communicate grid information between MPI ranks. To do so, we establish a communication ring for all MPI ranks, wherein each rank has a neighbor rank to the "left" and also one to the "right," forming a closed circle. To carry out the distributed FMM, we first let each MPI rank compute in parallel its local electrostatic interactions (i.e., interactions with the grid points that comprise the given MPI rank's FMM tree). Next, each MPI rank sends its list of grid-point information and the corresponding (incomplete) electrostatic potential vector to its neighboring MPI rank to the right, while simultaneously each MPI rank receives incoming grid and potential information from its neighbor to the left. The incoming grid points, which are only stored temporarily, are then allowed to traverse the MPI rank's FMM tree, accumulating the interactions in the incoming electrostatic potential vector. After tree traversal has been performed for all temporary grid points, the grid information and potential are again passed along to the neighboring MPI rank. This compute-and-pass procedure continues until the grid information makes a complete cycle around the ring, which takes p steps of communication. Upon completion of the cycle, the electrostatic potential that has been passed around will have traversed the FMM tree of each MPI rank, and it will have returned to the MPI rank to which it belongs. This communication pattern can benefit from using non-blocking MPI sends and receives, allowing computation and communication to overlap to some extent.

A further optimization that is possible within the distributed FMM scheme is to impose a certain cutoff distance for interactions between grid points belonging to different MPI ranks. When a grid point from rank p_i interacts with those from rank p_j, the MAC criteria are tested for p_i's grid point against the level-zero box from p_j's FMM tree. So long as p_i and p_j are sufficiently distant, the MAC will always be met for points in p_i, and there is no need for these points to traverse p_j's octree. The criterion to let all of p_i's grid points interact with the level-zero multipole expansion of p_j's grid is

$$r_{ij} - R_{i,\mathrm{box}} - R_{j,\mathrm{box}} \geq R_{\mathrm{cut\text{-}box}}, \tag{11.50}$$

where r_{ij} is the distance between the centroids of trees p_i and p_j, $R_{i,\mathrm{box}}$ is the radius of p_i's level-zero box, and $R_{\mathrm{cut\text{-}box}}$ is a pre-determined cutoff. If the inequality in Eq. (11.50) is satisfied, then we only compute level-zero multipoles for p_j's grid points, which affords some savings.

In the distributed FMM scheme, it is also beneficial to use the same FMM trees to compute the electrostatic potential on the surface grid. This involves a procedure similar to that described above, in which all solute point charges traverse the distributed surface grid. In our implementation, we let each MPI rank store all atomic coordinates and related information, so that each MPI rank can independently compute its portion of $\mathbf{v}$ in Eq. (11.22), without communication. Allowing all MPI ranks to store the global atomic coordinates and charges is usually feasible because there are far fewer atoms than surface grid points.

What we have described above applies only to solute point charges, whereas an electron density must be treated differently. A simple procedure for the latter is to let the QM charge density interact explicitly (rather than through the distributed FMM trees) with all surface grid points within a certain pre-determined cutoff distance, which maintains a nearly constant amount of CPU time for a fixed QM region. The electrostatic potential thus computed can then be communicated as needed to the appropriate MPI ranks.

For QM/MM/PCM jobs, this means that the FMM trees are used to compute interactions between the surface charges themselves, and between the surface charges and the MM charges, but not for interactions that involve the QM region. While this approach clearly could be improved, we often find that the purely classical steps are the bottlenecks when the MM region is large.

11.4.2.4 Surface construction strategies

The distributed FMM scheme can also be used to accelerate construction of the PCM surface grid. Note that SWIG discretization requires the evaluation of $\mathcal{O}(N_{\text{atoms}}^2 N_{\text{Lebedev}})$ switching functions, although this number could be reduced somewhat by atom–atom pairwise distance cutoffs. The FMM octree spatial partitioning provides further acceleration, by constructing an octree for all the atoms of a given solute, similar to what was described above for the surface grid electrostatic interactions. Then, each atom traverses the octree using a switching function acceptance criterion (instead of a MAC) to determine if the tree-traversing atom needs to compute its switching function with the atoms of a neighboring octree box or not. The switching function acceptance criterion that we use is

$$r_{ic} - R_{c,\text{box}} \leq R_{\text{cut-switch}}, \tag{11.51}$$

where $R_{\text{cut-switch}}$ is a pre-determined cutoff distance, selected so that atom–atom distances larger than this cutoff will not alter the switching function for the ith atom's grid points. The quantity r_{ic} is the distance from the ith tree-traversing atom to the center of the cth octree box whose radius is $R_{c,\text{box}}$. Only if the inequality in Eq. (11.51) is valid do we compute the explicit atom–atom pairwise switching functions. This procedure scales roughly as $\mathcal{O}(N_{\text{Lebedev}} N_{\text{atoms}} \log N_{\text{atoms}})$ and can afford significant savings for macromolecular solutes.

Furthermore, this octree switching function procedure can be parallelized within the distributed surface scheme. In our implementation, each MPI rank is assigned a set of atoms whose switching functions it will compute. Each MPI rank independently constructs the switching function octree from the entire global set of solute atoms (stored in each MPI rank) and then the chosen subset of atoms for each MPI rank traverses the switching-function octree. In the end, each MPI rank will have evaluated and constructed the PCM surface grid for its set of atoms. It is this surface, then, from which the surface octree is constructed and used in the distributed FMM scheme. This is a preliminary implementation and there are undoubtedly load-balance issues associated with this approach. In a globular solute, for example, one MPI rank may be assigned only interior atoms and end up with no surface grid because the switching functions all evaluate to zero. A load-balancing scheme that can dynamically respond to protein conformational changes would be preferable, but remains to be developed.

Alternatively, one could abandon the switching functions of SWIG discretization and construct a smooth isodensity surface, either using the actual electron density [12, 19, 30] (in QM/PCM calculations) or else some pseudo-density, as discussed in Section 11.2.2.2. A pseudo-density isosurface can be constructed using the marching cubes algorithm [52], which is trivially parallelizable by multithreading the "marching" loop over all cubes and partitioning the array of cubes across MPI ranks. We perform this partition before the calculation begins, by assigning a number of MPI ranks to each of the x, y, and z Cartesian dimensions. As with the SWIG octree approach, this procedure is vulnerable to poor load balance if some MPI process receives a set of solvent-inaccessible atoms. In practice, we are often able to achieve reasonably good load balance by examining the geometry of the solute and assigning a greater number of MPI processes to the larger Cartesian dimensions, but for MD applications a dynamical load-balancing scheme is probably required. As with the switching function octree, the resulting distributed surface grid is reused in the FMM scheme for solving the PCM equations.

11.4.2.5 Scalability tests

We next consider some examples to demonstrate the scalability of the algorithms described above, focusing on AMBER99/C-PCM jobs. We set $\varepsilon = 78.4$ and (for DESMO calculations) $\kappa^{-1} = 3.0$ Å, which equates to a fairly large ionic strength of about 1 mol/L for water at 25°C. The solute cavity is constructed as a pseudo-density

Table 11.3 Dimensionless width parameters for pseudo-density isosurfaces.

Δx/Å[a]	ζ[b]
≥ 0.4	5.9
0.3	5.5
0.2	5.2
0.1	4.9

[a] Marching cubes grid resolution.
[b] Gaussian widths ζ_i in Eq. (11.42) are given by $\zeta_i = \zeta a_i^{-1/2}$.

isosurface (see Section 11.2.2), with $B = 2.5$ Å as in Ref. [88]. Our implementation of this cavity construction uses Gaussian blurring (Section 11.4.1) to avoid numerical issues related to Coulomb singularities. As such, a set of Gaussian widths is required. We determine these by minimizing the error in the Born ion solvation energy for a spherical cavity of radius 2 Å, at various grid resolutions. The width parameter ζ obtained for each grid resolution is listed in Table 11.3, and the Gaussian width parameters ζ_i in Eq. (11.42) are taken to be $\zeta_i = \zeta a_i^{-1/2}$. A marching cubes grid resolution of 0.4 Å was employed in all calculations.

Parameters for the CG-FMM algorithm were selected based on test calculations for (alanine)$_{20}$, in order to obtain a solvation energy that is within 1 kcal/mol of that obtained by explicit matrix inversion. Multipoles up to $\ell = 5$ were included for computing interactions between the surface charges, using $R_{\text{thresh}} = 2.0$ Å, $N_{\text{thresh}} = 200$, and $\theta_{\text{MAC}} = 0.7$. For interactions between the solute charges and the surface charges, multipoles up to $\ell = 4$ were included, with $R_{\text{thresh}} = 2.0$ Å, $N_{\text{thresh}} = 50$, and $\theta_{\text{MAC}} = 0.5$. The CG algorithm was considered converged when the maximum residual fell below a threshold of 10^{-3}. All calculations were performed on a cluster of 12-core HP Intel Xeon x5650 processors with 48 Gb of RAM per node, using a locally modified version of Q-CHEM [39].

A quasi-linear solute is a best-case scenario for scalability, so we first examine unfolded alanine polypeptides, (Ala)$_n$. Table 11.4 is a strong-scaling analysis for a fixed solute size, (Ala)$_{250}$, with a surface grid consisting of $\approx 350,000$ points, well beyond the

Table 11.4 Strong-scaling data for CG-FMM applied to $(\mathrm{Ala})_{250}$[a]

Nodes	Threads	Cores	Wall time/sec	Parallel efficiency
1	1	1	171.0	1.00
1	2	2	88.0	0.97
1	4	4	46.0	0.93
1	8	8	26.4	0.81
1	12	12	19.5	0.73
2	12	24	10.2	0.70
4	12	48	6.9	0.52
8	12	96	4.1	0.43
16	12	192	2.9	0.31

[a] Surface grid consists of 349,797 points.

feasible memory limits for matrix inversion. The multithreaded CG-FMM approach scales quite well across all 12 cores of one node, with a parallel efficiency of 73% that greatly exceeds that of recent multi-threaded FMM algorithms [11, 66]. However, the use of additional nodes at 12 cores/node scales only moderately well for a few additional nodes, and leads to diminishing returns as the amount of work/node becomes small and communication becomes a significant fraction of cost. Nevertheless, this fairly significant single-point calculation can be performed in just 10 seconds on 2 × 12 cores, with 70% parallel efficiency.

Next we investigate weak-scaling parallelism with $(\mathrm{Ala})_n$ polymers, increasing n in proportion to the number of MPI ranks (Table 11.5). Although the parallel efficiency is not great, calculations

Table 11.5 Weak-scaling data for CG-FMM applied to $(\mathrm{Ala})_n$[a]

Nodes	Cores	Wall time/sec	Parallel efficiency	n	No. grid points
1	12	19.5	1.00	250	349,797
2	24	23.8	0.82	500	698,589
4	48	32.4	0.60	1000	1,397,704
8	96	38.7	0.50	2000	2,793,018
16	192	46.2	0.42	4000	5,583,607

[a] All calculations use 12 cores/node, and the parallel efficiency is defined relative to the single-node performance.

Table 11.6 Strong scaling data for the protein 1LXL[a]

Nodes	Threads	Cores	COSMO wall time/sec	COSMO parallel efficiency	DESMO wall time/sec	DESMO parallel efficiency
1	1	1	344.1	1.00	351.7	1.00
1	2	2	175.6	0.98	180.0	0.98
1	4	4	90.6	0.95	92.9	0.95
1	8	8	50.9	0.85	52.1	0.84
1	12	12	37.4	0.77	37.9	0.77
2	12	24	25.9	0.55	25.0	0.59
4	12	48	15.4	0.47	15.7	0.47
8	12	96	11.2	0.32	12.1	0.30
16	12	192	11.0	0.17	11.2	0.16

[a] Surface grid consists of 285,446 points.

with several million grid points can be performed in less than a minute for systems as large as $(\mathrm{Ala})_{4000}$. The calculations in Table 11.5 represent the largest PCM calculations of which we are aware.

Finally, we present strong-scaling tests for an irregularly shaped protein (PDB code 1LXL) in Table 11.6, using both C-PCM and DESMO. As with the quasi-linear alanine chains, scaling remains good across one node but drops noticeably across multiple nodes. Note, however, that the DESMO method, which incorporates salt effects, incurs negligible overhead as compared to C-PCM. The extra overhead in a DESMO MM/PCM calculation is simply the need to compute the screened electrostatic potential on the surface grid one time, and this can be accomplished using an adaptation of the FMM algorithm of Krasny and co-workers [50].

Clearly, it is desirable to improve the MPI aspect of the parallelization strategy, which is presently bottlenecked by communication of grid information around the ring of MPI ranks. One possible way to accomplish this would be to define neighboring MPI ranks based on their FMM boxes. Non-neighbors could either be ignored beyond some cutoff, or (preferably) they could broadcast only their FMM level-zero multipoles, rather than the larger quantity of grid information that is passed in our present implementation. In this modified algorithm, grid information needs only to be passed between neighboring MPI ranks, rather than the full ring, which may lead to a quantity of communication that is nearly constant with respect to system size.

Alternatively, one could introduce another layer of parallelism between the distributed-grid MPI ranks and the tree-traversing OpenMP threads. In this layer, one would allow a certain number of MPI ranks to perform tree-traversal, with each of possessing a copy of the grid information that it needs. In this master/slave setup, each MPI master rank builds the appropriate portion of the grid (as every single MPI rank does in our present implementation), then passes the grid information to its set of MPI slaves. In addition, each of the MPI slaves would also exploit OpenMP multi-threading (as in our current implementation) to assist with tree traversal. Such an algorithm does not directly reduce the communication problem, but by sub-dividing the work of tree-traversal this approach allows the use of larger boxes at the level of the "master" MPI ranks. This should reduce the communication in an indirect way, since fewer master ranks will be required, and these are the only ones that must communicate grid information around the ring.

11.5 Summary and Future Directions

The prospects for the use of PCMs in macromolecular electrostatics calculations seem bright. The accuracy is (and theoretically speaking, should be) comparable to that achievable using finite-difference solution of Poisson's equation [44], but the computational cost is greatly reduced since only the molecular cavity surface, and not the whole of three-dimensional space, need be discretized. Problems with discontinuous forces are entirely eliminated by recently developed smooth discretization schemes [41, 42, 70, 76, 87]. Reported here for the first time is our multithreaded OpenMP implementation of a conjugate gradient/fast multipole PCM solver, whose cost is $\mathcal{O}(N)$ in memory and $\mathcal{O}(N \log N)$ in CPU time. This approach shows good scalability across all 12 cores of one node, with a parallel efficiency exceeding that of other multi-threaded FMM algorithms, although the present implementation does not scale well beyond one or two nodes.

To improve the accuracy of implicit-solvent potential energy surfaces, non-electrostatic interactions *must* be included, although such interactions have received only a brief mention here. The smooth, linear-scaling PCM technology that is discussed here is immediately ready for use in MM/PBSA applications [36, 38], as a replacement for finite-difference electrostatics. Other formulas for the non-electrostatic interactions [47] can also be used in PCM calculations, possibly after some re-parameterization. In general these non-electrostatic interaction formulas depend in some way on the cavity surface area, which is smooth and easily calculable by means of the PCM algorithms discussed herein.

Particular attention should be paid to the DESMO method [43], as this model appears to be suitable for use with solvents containing dissolved ions that are described by the linearized Poisson–Boltzmann equation. DESMO shows promising accuracy with respect to benchmark LPBE calculations. This includes an analytically solvable model problem consisting of multiple solute cavities, as would be encountered in a study of protein–protein interactions in implicit solvent.

Finally, PCMs are useful for creating a data set of perfect radii and effective pairwise Coulomb interactions that can be used to parameterize novel generalized Born models. Several improved GB models, having slightly better accuracy and significantly lower cost, have been suggested based on comparisons to PCM benchmarks [45]. These new GB models are ready to be “dropped in” to existing MD codes. Comparison to DESMO suggests new ways to incorporate salt effects into GB models [46], which warrant further exploration as well.

Acknowledgments

The authors’ work on PCMs has been supported by the National Science Foundation (grant nos. CHE-0748448 and CHE-1300603, to J.M.H.), through an Ohio State Presidential Fellowship (to A.W.L.), and through generous allocations of computing time from the Ohio Supercomputer Center (project nos. PAS-0291 and PAA-0003).

References

1. Ahuir, J. L. P., and Silla, E. (1990). GEPOL: An improved description of molecular surfaces. I. Building the spherical surface set, *J. Comput. Chem.* **11**, pp. 1047–1060.
2. Amovilli, C., and Mennucci, B. (1997). Self-consistent-field calculation of Pauli repulsion and dispersion contributions to the solvation free energy in the polarizable continuum model, *J. Phys. Chem. B* **101**, pp. 1051–1057.
3. Bachs, M., Luque, F. J., and Orozco, M. (1994). Optimization of solute cavities and van der Waals parameters in ab initio MST-SCRF calculations of neutral molecules, *J. Comput. Chem.* **15**, pp. 446–454.
4. Baker, N. A. (2004). Poisson–Boltzmann methods for biomolecular electrostatics, *Method. Enzymol.* **383**, pp. 94–118.
5. Baker, N. A., Sept, D., Joseph, S., Holst, M. J., and McCammon, J. A. (2001). Electrostatics of nanosystems: Application to microtubules and the ribosome, *Proc. Natl. Acad. Sci. USA* **98**, pp. 10037–10041.
6. Barone, V., Cossi, M., and Tomasi, J. (1997). A new definition of cavities for the computation of solvation free energies by the polarizable continuum model, *J. Chem. Phys.* **107**, pp. 3210–3221.
7. Bottcher, C. J. (1976). *Theory of Electric Polarization*, 2nd ed. (Elsevier).
8. Cancès, E., and Mennucci, B. (1998). New applications of integral equations methods for solvation continuum models: Ionic solutions and liquid crystals, *J. Math. Chem.* **23**, pp. 309–326.
9. Cancès, E., and Mennucci, B. (2001). Comment on "Reaction field treatment of charge penetration" [*J. Chem. Phys*. **112**, 5558 (2000)], *J. Chem. Phys.* **114**, pp. 4744–4745.
10. Cancés, E., Mennucci, B., and Tomasi, J. (1997). A new integral equation formalism for the polarizable continuum model: Theoretical background and applications to isotropic and anisotropic dielectrics, *J. Chem. Phys.* **107**, pp. 3032–3041.
11. Chau, N. H. (2013). Parallelization of the fast multipole method for molecular dynamics simulations on multicore computers, in *Advanced Computational Methods for Knowledge Engineering*, *Studies in Computational Intelligence*, vol. 479 (Springer), pp. 209–224.
12. Chen, F., and Chipman, D. M. (2003). Boundary element methods for dielectric cavity construction and integration, *J. Chem. Phys.* **119**, pp. 10289–10297.

13. Chipman, D. M. (1997). Charge penetration in dielectric models of solvation, *J. Chem. Phys.* **106**, pp. 10194–10206.
14. Chipman, D. M. (1999). Simulation of volume polarization in reaction field theory, *J. Chem. Phys.* **110**, pp. 8012–8018.
15. Chipman, D. M. (2000). Reaction field treatment of charge penetration, *J. Chem. Phys.* **112**, pp. 5558–5565.
16. Chipman, D. M. (2002). Comparison of solvent reaction field representations, *Theor. Chem. Acc.* **107**, pp. 80–89.
17. Chipman, D. M. (2004). Solution of the linearized Poisson–Boltzmann equation, *J. Chem. Phys.* **120**, pp. 5566–5575.
18. Chipman, D. M. (2006). New formulation and implementation for volume polarization in dielectric continuum theory, *J. Chem. Phys.* **124**, pp. 224111:1–10.
19. Chipman, D. M., and Dupuis, M. (2002). Implementation of solvent reaction fields for electronic structure, *Theor. Chem. Acc.* **107**, pp. 90–102.
20. Colominas, C., Luque, F. J., Teixidó, J., and Orozco, M. (1999). Cavitation contribution to the free energy of solvation. Comparison of different formalisms in the context of MST calculations, *Chem. Phys.* **240**, pp. 253–264.
21. Connolly, M. L. (1983). Solvent-accessible surfaces of proteins and nucleic acids, *Science* **221**, pp. 709–713.
22. Cossi, M., Barone, V., Mennucci, B., and Tomasi, J. (1998). Ab initio study of ionic solutions by a polarizable continuum dielectric model, *Chem. Phys. Lett.* **286**, pp. 253–260.
23. Cossi, M., Mennucci, B., and Cammi, R. (1996). Analytical first derivatives of molecular surfaces with respect to nuclear coordinates, *J. Comput. Chem.* **17**, pp. 57–73.
24. Cossi, M., Rega, N., Scalmani, G., and Barone, V. (2001). Polarizable dielectric continuum model of solvation with inclusion of charge penetration effects, *J. Chem. Phys.* **114**, pp. 5691–5701.
25. Cossi, M., Rega, N., Scalmani, G., and Barone, V. (2003). Energies, structures, and electronic properties of molecules in solution with the C-PCM solvation model, *J. Comput. Chem.* **24**, pp. 669–681.
26. Cossi, M., Scalmani, G., Rega, N., and Barone, V. (2002). New developments in the polarizable continuum model for quantum mechanical and classical calculations on molecules in solution, *J. Chem. Phys.* **117**, pp. 43–54.

27. Cramer, C. J., and Truhlar, D. G. (2008). A universal approach to solvation modeling, *Acc. Chem. Res.* **41**, pp. 760–768.
28. Debye, P., and Hückel, E. (1954). On the theory of electrolytes. I. Freezing point depression and related phenomena, in *Collected Papers of Peter J. W. Debye* (Interscience Publishers, Inc.), pp. 217–263.
29. Ferrighi, L., Frediani, L., Fossgaard, E., and Ruud, K. (2006). Parallelization of the integral equation formulation of the polarizable continuum model for higher-order response functions, *J. Chem. Phys.* **125**, p. 154112.
30. Foresman, J. B., Keith, T. A., Wiberg, K. B., Snoonian, J., and Frisch, M. J. (1996). Solvent effects 5. Influence of cavity shape, truncation of electrostatics, and electron correlation on ab initio reaction field calculations, *J. Phys. Chem.* **100**, pp. 16098–16104.
31. Ginovska, B., Camaioni, D. M., Dupuis, M., Schwerdtfeger, C. A., and Gil, Q. (2008). Charge-dependent cavity radii for an accurate dielectric continuum model of solvation with emphasis on ions: Aqueous solutes with oxo, hydroxo, amino, methyl, chloro, bromo, and fluoro functionalities, *J. Phys. Chem. A* **112**, pp. 10604–10613.
32. Greengard, L., and Rokhlin, V. (1987). A fast algorithm for particle simulations, *J. Comput. Phys.* **73**, pp. 325–348.
33. Grochowski, P., and Trylska, J. (2008). Continuum molecular electrostatics, salt effects, and counterion binding—A review of the Poisson–Boltzmann theory and its modifications, *Biopolymers* **89**, pp. 93–113.
34. Grycuk, T. (2003). Deficiency of the Coulomb-field approximation in the generalized Born model: An improved formula for Born radii evaluation, *J. Chem. Phys.* **119**, pp. 4817–4826.
35. Holst, M. J., and Saied, F. (1995). Numerical solution of the nonlinear Poisson–Boltzmann equation: Developing more robust and efficient methods, *J. Comput. Chem.* **16**, pp. 337–364.
36. Homeyer, N., and Gohlke, H. (2012). Free energy calculations by the molecular mechanics Poisson–Boltzmann surface area method, *Mol. Inf.* **31**, pp. 114–122.
37. Klamt, A., and Schüürmann, G. (1993). COSMO: A new approach to dielectric screening in solvents with explicit expressions for the screening energy and its gradient, *J. Chem. Soc. Perkin Trans. 2* , pp. 799–805.
38. Kollman, P. A., Massova, I., Reyes, C., Kuhn, B., Huo, S., Chong, L., Lee, M., Lee, T., Duan, Y., Wang, W., Donini, O., Cieplak, P., Srinivasan, J., Case, D. A., and Cheatham III, T. E. (2000). Calculating structures and free energies

of complex molecules: Combining molecular mechanics and continuum models, *Acc. Chem. Res.* **33**, pp. 889–897.

39. Krylov, A. I., and Gill, P. M. W. (2013). Q-Chem: An engine for innovation, *WIREs Comput. Mol. Sci.* **3**, pp. 317–326.
40. Kurzak, J., and Pettitt, B. M. (2006). Fast multipole methods for particle dynamics, *Mol. Simul.* **32**, pp. 775–790.
41. Lange, A. W., and Herbert, J. M. (2010). Polarizable continuum reaction-field solvation models affording smooth potential energy surfaces, *J. Phys. Chem. Lett.* **1**, pp. 556–561.
42. Lange, A. W., and Herbert, J. M. (2010). A smooth, non-singular, and faithful discretization scheme for polarizable continuum models: The switching/Gaussian approach, *J. Chem. Phys.* **133**, pp. 244111:1–18.
43. Lange, A. W., and Herbert, J. M. (2011). A simple polarizable continuum solvation model for electrolyte solutions, *J. Chem. Phys.* **134**, pp. 204110:1–15.
44. Lange, A. W., and Herbert, J. M. (2011). Symmetric versus asymmetric discretization of the integral equations in polarizable continuum solvation models, *Chem. Phys. Lett.* **509**, pp. 77–87.
45. Lange, A. W., and Herbert, J. M. (2012). Improving generalized Born models by exploiting connections to polarizable continuum models. I. An improved effective Coulomb operator, *J. Chem. Theory Comput.* **8**, pp. 1999–2011.
46. Lange, A. W., and Herbert, J. M. (2012). Improving generalized Born models by exploiting connections to polarizable continuum models. II. Corrections for salt effects, *J. Chem. Theory Comput.* **8**, pp. 4381–4392.
47. Lee, M. S., and Olson, M. A. (2013). Comparison of volume and surface area nonpolar solvation free energy terms for implicit solvent simulations, *J. Chem. Phys.* **139**, pp. 044119:1–6.
48. Lee, M. S., Salsbury, Jr., F. R., and Brooks III, C. L. (2002). Novel generalized Born methods, *J. Chem. Phys.* **116**, pp. 10606–10614.
49. Li, H., and Jensen, J. H. (2004). Improving the efficiency and convergence of geometry optimization with the polarizable continuum model: New energy gradients and molecular surface tessellation, *J. Comput. Chem.* **25**, pp. 1449–1462.
50. Li, P., Johnston, H., and Krasny, R. (2009). A Cartesian treecode for screened Coulomb interactions, *J. Comput. Phys.* **228**, pp. 3858–3868.
51. Lipparini, F., Scalmani, G., Mennucci, B., Cancès, E., Caricato, M., and Frisch, M. J. (2010). A variational formulation of the polarizable continuum model, *J. Chem. Phys.* **133**, pp. 014106–1–11.

52. Lorensen, W. E., and Cline, H. E. (1987). Marching cubes: A high resolution 3D surface construction algorithm, *Comp. Graph.* **21**, pp. 163–169.
53. Lotan, I., and Head-Gordon, T. (2006). An analytical electrostatic model for salt screened interactions between multiple proteins, *J. Chem. Theory Comput.* **2**, pp. 541–555.
54. Lu, B. Z., Zhou, Y. C., Holst, M. J., and McCammon, J. A. (2008). Recent progress in numerical methods for the Poisson–Boltzmann equation in biophysical applications, *Commun. Comput. Phys.* **3**, pp. 973–1009.
55. Marenich, A. V., Cramer, C. J., and Truhlar, D. G. (2009). Universal solvation model based on solute electron density and on a continuum model of the solvent defined by the bulk dielectric constant and atomic surface tensions, *J. Phys. Chem. B* **113**, pp. 6378–6396.
56. Mennucci, B. (2012). Polarizable continuum model, *WIREs Comput. Mol. Sci.* **2**, pp. 386–404.
57. Mennucci, B., and Cammi, R. (2003). Ab initio model to predict NMR shielding tensors for solutes in liquid crystals, *Int. J. Quantum Chem.* **93**, pp. 121–130.
58. Mennucci, B., Cancés, E., and Tomasi, J. (1997). Evaluation of solvent effects in isotropic and anisotropic dielectrics and in ionic solutions with a unified integral equation method: Theoretical bases, computational implementation, and numerical applications, *J. Phys. Chem. B* **101**, pp. 10506–10517.
59. Mennucci, B., and Tomasi, J. (1997). Continuum solvation models: A new approach to the problem of solute's charge distribution and cavity boundaries, *J. Chem. Phys.* **106**, pp. 5151–5158.
60. Miertuš, S., Scrocco, E., and Tomasi, J. (1981). Electrostatic interaction of a solute with a continuum. a direct utilization of ab initio molecular potentials for the prevision of solvent effects, *Chem. Phys.* **55**, pp. 117–129.
61. Miertuš, S., and Tomasi, J. (1982). Approximate evaluations of the electrostatic free energy and internal energy changes in solution processes, *Chem. Phys.* **65**, pp. 239–245.
62. Mongan, J., Svrcek-Seiler, W. A., and Onufriev, A. (2007). Analysis of integral expressions for effective Born radii, *J. Chem. Phys.* **127**, pp. 185101:1–10.
63. Onufriev, A. (2010). Continuum electrostatics solvent modeling with the generalized Born model, in M. Feig (ed.), *Modeling Solvent Environments: Applications to Simulations of Biomolecules*, Chapter 6 (Wiley-VCH, Hoboken, NJ), pp. 127–165.

64. Onufriev, A., Case, D. A., and Bashford, D. (2002). Effective Born radii in the generalized Born model approximation: The importance of being perfect, *J. Comput. Chem.* **23**, 14, pp. 1297–1304.
65. Onufriev, A. V., and Sigalov, G. (2011). A strategy for reducing gross errors in the generalized Born models of implicit solvation, *J. Chem. Phys.* **134**, pp. 164104:1–15.
66. Pan, X. M., Pi, W. C., and Sheng, X. Q. (2011). An OpenMP parallelization of the multilevel fast multipole algorithm, *Prog. Electromag. Res.* **112**, pp. 199–213.
67. Pomogaeva, A., and Chipman, D. M. (2014). Hydration energy from a composite method for implicit representation of the solvent, *J. Chem. Theory Comput.* **10**, pp. 211–219.
68. Purisima, E. O., and Nilar, S. H. (1995). A simple yet accurate boundary element method for continuum dielectric calculations, *J. Comput. Chem.* **16**, pp. 681–689.
69. Scalmani, G., Barone, V., Kudin, K. N., Pomelli, C. S., Scuseria, G. E., and Frisch, M. J. (2004). Achieving linear-scaling computational cost for the polarizable continuum model of solvation, *Theor. Chem. Acc.* **111**, pp. 90–100.
70. Scalmani, G., and Frisch, M. J. (2010). Continuous surface charge polarizable continuum models of solvation. I. General formalism, *J. Chem. Phys.* **132**, pp. 114110:1–15.
71. Sigalov, G., Fenley, A., and Onufriev, A. (2006). Analytical electrostatics for biomolecules: Beyond the generalized Born approximation, *J. Chem. Phys.* **124**, pp. 124902:1–14.
72. Sigalov, G., Scheffel, P., and Onufriev, A. (2005). Incorporating variable dielectric environments into the generalized Born model, *J. Chem. Phys.* **122**, pp. 094511:1–15.
73. Srinivasan, J., Trevathan, M. W., Beroza, P., and Case, D. A. (1999). Application of a pairwise generalized Born model to proteins and nucleic acids: Inclusion of salt effects, *Theor. Chem. Acc.* **101**, pp. 426–434.
74. Still, W. C., Tempczyk, A., Hawley, R. C., and Hendrickson, T. (1990). Semianalytical treatment of solvation for molecular mechanics and dynamics, *J. Am. Chem. Soc.* **112**, pp. 6127–6129.
75. Su, P., and Li, H. (2009). Continuous and smooth potential energy surface for conductor-like screening solvation model using fixed points with variable areas, *J. Chem. Phys.* **130**, pp. 074109:1–13.

76. Thellamurege, N. M., and Li, H. (2012). Note: FixSol solvation model and FIXPVA2 tessellation scheme, *J. Chem. Phys.* **137**, pp. 246101:1–2.
77. Tomasi, J., Mennucci, B., and Cammi, R. (2005). Quantum mechanical continuum solvation models, *Chem. Rev.* **105**, pp. 2999–3093.
78. Truong, T. N., Nguyen, U. N., and Stefanovich, E. V. (1996). Generalized conductor-like screening model (GCOSMO) for solvation: An assessment of its accuracy and applicability, *Int. J. Quantum Chem. Symp.* **60**, pp. 1615–1622.
79. Tsui, V., and Case, D. A. (2001). Theory and applications of the generalized Born solvation model in macromolecular simulations, *Biopolymers (Nucl. Acid Sci.)* **56**, pp. 275–291.
80. Wang, J., Cai, Q., Li, Z.-L., Zhao, H.-K., and Luo, R. (2009). Achieving energy conservation in Poisson–Boltzmann molecular dynamics: Accuracy and precision with finite-difference algorithms, *Chem. Phys. Lett.* **468**, pp. 112–118.
81. Wang, J., Cai, Q., Xiang, Y., and Luo, R. (2012). Reducing grid dependence in finite-difference Poisson–Boltzmann calculations, *J. Chem. Theory Comput.* **8**, pp. 2741–2751.
82. Wang, J., Tan, C., Chanco, E., and Luo, R. (2010). Quantitative analysis of Poisson–Boltzmann implicit solvent in molecular dynamics, *Phys. Chem. Chem. Phys.* **12**, pp. 1194–1202.
83. Wang, J., Tan, C., Tan, Y.-H., Lu, Q., and Luo, R. (2008). Poisson–Boltzmann solvents in molecular dynamics simulations, *Commun. Comput. Phys.* **3**, pp. 1010–1031.
84. Wang, Y., and Li, H. (2009). Smooth potential energy surface for cavitation, dispersion and repulsion energies in continuum solvation model, *J. Chem. Phys.* **131**, pp. 206101:1–2.
85. Wangsness, R. K. (1986). *Electromagnetic Fields*, 2nd ed. (Wiley).
86. White, C. A., Johnson, B. G., Gill, P. M. W., and Head-Gordon, M. (1994). The continuous fast multipole method, *Chem. Phys. Lett.* **230**, pp. 8–16.
87. York, D. M., and Karplus, M. (1999). Smooth solvation potential based on the conductor-like screening model, *J. Phys. Chem. A* **103**, pp. 11060–11079.
88. Yu, Z., Jacobson, M. P., and Friesner, R. (2005). What role do surfaces play in GB models? A new-generation of surface-generalized Born model based on a novel Gaussian surface for biomolecules, *J. Comput. Chem.* **27**, pp. 72–89.
89. Zhan, C.-G., Bentley, J., and Chipman, D. M. (1998). Volume polarization in reaction field theory, *J. Chem. Phys.* **108**, pp. 177–192.