SANDIA REPORT

SAND2014-0355
Unlimited Release
Printed January 2014

Multilevel summation methods for efficient evaluation of long-range pairwise interactions in atomistic and coarse-grained molecular simulation

Stephen D. Bond

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.

Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online ordering: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2014-0355
Unlimited Release
Printed January 2014

Multilevel summation methods for efficient evaluation of long-range pairwise interactions in atomistic and coarse-grained molecular simulation

Stephen D. Bond
Multiphysics Simulation Technologies
Sandia National Laboratories
P.O. Box 5800, MS 1318
Albuquerque, NM 87185-1318

Abstract

The availability of efficient algorithms for long-range pairwise interactions is central to the success of numerous applications, ranging in scale from atomic-level modeling of materials to astrophysics. This report focuses on the implementation and analysis of the multilevel summation method for approximating long-range pairwise interactions. The computational cost of the multilevel summation method is proportional to the number of particles, N, which is an improvement over FFT-based methods who's cost is asymptotically proportional to $N \log N$. In addition to approximating electrostatic forces, the multilevel summation method can be use to efficiently approximate convolutions with long-range kernels. As an application, we apply the multilevel summation method to a discretized integral equation formulation of the regularized generalized Poisson equation. Numerical results are presented using an implementation of the multilevel summation method in the LAMMPS software package. Preliminary results show that the computational cost of the method scales as expected, but there is still a need for further optimization.

Acknowledgment

I would like to acknowledge the Sandia National Laboratories early career LDRD program for supporting this work. This work was funded under LDRD Project Number 157688 and Title "Multilevel Summation Methods for Efficient Evaluation of Long-Range Pairwise Interactions in Atomistic and Coarse-Grained Molecular Simulation". I would like to thank my early career LDRD mentor, Steve Plimpton, for his advice and guidance. The implementation of a parallel version of the multilevel summation method in the LAMMPS software package was completed primarily by Stan Moore and Paul Crozier. Without Stan and Paul's work on the LAMMPS parallel implementation, I would have been able to generate only a small fraction of the results presented in this report. I would like to thank David Hardy, the author of the NAMD-Lite implementation of the multilevel summation method. David has been helpful in providing guidance concerning parameter selection, error analysis, and software implementation. David's thesis, "Multilevel Summation for the Fast Evaluation of Forces for the Simulation of Biomolecules", was an invaluable reference for the research in this project.

The work on the application of the multilevel summation method to the Generalized Poisson Equation was completed in collaboration with Bob Skeel and Dmytro Yershov. I would like to thank both Bob and Dmytro for their work on this important application, and the National Science Foundation (CCF 08-30578) who initially supported this research.

Finally, I would like to thank Andrew Baczewski, Eric Cyr, Bruce Hendrickson, Jeremy Lechman, Rich Lehoucq, Randy Summers, and Aidan Thompson for their advice, support, and guidance during the completion of this project.

Contents

1 Introduction **9**

2 Problem formulation **11**

Pairwise interactions 11

Generalized Poisson Equation 13

3 Approximation methods **17**

Overview 17

Multilevel Summation 18

Laplacian-centered Generalized Poisson Equation 21

4 Numerical results **25**

5 Conclusion **27**

References **28**

Appendix

A Application to the Generalized Poisson Equation **37**

List of Figures

2.1 (a) 6 particles with the 21 non-zero interactions (dashed lines) for a symmetric kernel with no self-interactions. (b) The 5 interactions with the middle particle. . . . 11

2.2 6 particles with periodic boundary conditions in two dimensions represented by an infinite lattice (only the middle 9 cells are shown). Note that the central unshaded particle interacts with all other particles, including its self-images. 12

2.3 Ewald splitting with $\beta_e = 1$ 13

2.4 Subdomains for a simple chain molecule 14

3.1 An unbalanced octtree with two-levels 17

4.1 Per-step simulation time for 10 steps of molecular dynamics using the multilevel summation method applied to a box of water with N = 17496, 139968, 1119744, and 8957952 atoms. The plot on the right is the same as the left, but with a logarithmic scale. 25

List of Tables

4.1 Performance comparison for 10 MD steps 26

This page intentionally left blank.

Chapter 1

Introduction

The availability of efficient algorithms for long-range pairwise interactions is central to the success of numerous applications, ranging in scale from atomic-level modeling of materials to astrophysics. Molecular dynamics (MD), in particular, can require months of supercomputer time, due to the expense of the large number of force evaluations required. The challenge is to design reliable, efficient, portable, scalable algorithms for calculating long-range interactions in large systems. Scalability and portability are of particular concern for modern exascale supercomputers with hybrid architectures and massive numbers of processors.

A diverse set of methods has evolved for rapid approximation of long-range interactions, including fast-multipole methods and Fourier-based particle-mesh Ewald methods. Multipole methods excel when applied to systems with large variations in density (e.g., astrophysics), but have generally been considered less competitive for more uniform systems (e.g., molecular dynamics). As a result, state-of-the-art MD codes like NAMD and LAMMPS, use particle-mesh Ewald.

Due to the use of the Fast Fourier Transform (FFT), particle-mesh Ewald methods do not scale well as the system size is increased, with a computational cost proportional to $N \log N$, where N is the number of atoms. The FFT also has a large communication overhead, due to the parallel scalability problems associated with the matrix transpose.

This report focuses on the implementation and analysis of the multilevel summation method (MSM), which is a relatively new algorithm for computing pairwise interactions. Preliminary studies have found that it has a computational cost proportional to N, rather than $N \log N$, and relatively low communication overhead (uses nested grids instead of the FFT). Further development and implementation of this method has the potential to dramatically improve the efficiency of MD software used for predictive simulation of materials.

An interesting application for this method is to the solution of the Generalized Poisson Equation (GPE), either as a stand-alone solver, or as a preconditioner for a method for solving the Poisson Boltzmann Equation. We derive an integral formulation for the GPE, and demonstrate how the MSM can be applied as a fast method for computing the matrix-vector products in the discrete formulation.

The remainder of this report is organized as follows. In Chapter 2, we provide a brief overview of long-range pairwise interaction problem, focusing on the Coulomb problem and the Generalized Poisson Equation. In Chapter 3, we discuss popular methods for approximating long-range

pairwise interactions, and provide a review of the multilevel summation method. At the end of Chapter 3, we review how the multilevel summation method can be applied to an integral formulation of the Generalized Poisson Equation. In Chapter 4, we present preliminary numerical results for the multilevel summation method implemented in the LAMMPS software package. A complete report on the application of the multilevel summation method to the Generalized Poisson Equation is included in the appendix.

Chapter 2

Problem formulation

Pairwise interactions

We begin by considering the general problem of computing the pairwise particle interactions of N particles in 3-dimensional space. Let $\vec{r}_i$ be the position of the ith particle in $\mathbb{R}^3$. The position of all N particles is denoted collectively by $x = [\vec{r}_1^{\mathsf{T}}, \vec{r}_2^{\mathsf{T}}, \ldots, \vec{r}_N^{\mathsf{T}}]^{\mathsf{T}}$. Let q_i be some property of the ith particle (e.g., charge), and $q = [q_1, q_2, \ldots, q_N]^{\mathsf{T}}$. In the simple case, the primary computational task is to approximate sums and derivatives of sums of the form

$$E(x) = \frac{1}{2} C \sum_{i=1}^{N} {\sum_{j=1}^{N}}' q_i q_j k(\vec{r}_i, \vec{r}_j), \tag{2.1}$$

where C is a constant. The primed sum omits certain values of j which excludes non-interacting particles.

(a) **(b)**

Figure 2.1. (a) 6 particles with the 21 non-zero interactions (dashed lines) for a symmetric kernel with no self-interactions. (b) The 5 interactions with the middle particle.

The function $k(\vec{r}, \vec{r}')$ is the interaction kernel. For the applications considered in this report, we assume that k is a radial function of the form $f(|\vec{r}' - \vec{r}|)$. For the gravitational N-body problem, $f(r) = 1/r$, and q_i is the mass of the ith particle. For the Coulomb problem, $f(r) = 1/r$, and q_i is a partial charge. The primed sum omits $j = i$ as well as values of j corresponding to covalently bonded atoms. Other examples of kernels include $1/r^6$ for van der Waals attractions and the $1/(r^2 + \delta^2)$ potential used in smoothed particle hydrodynamics.

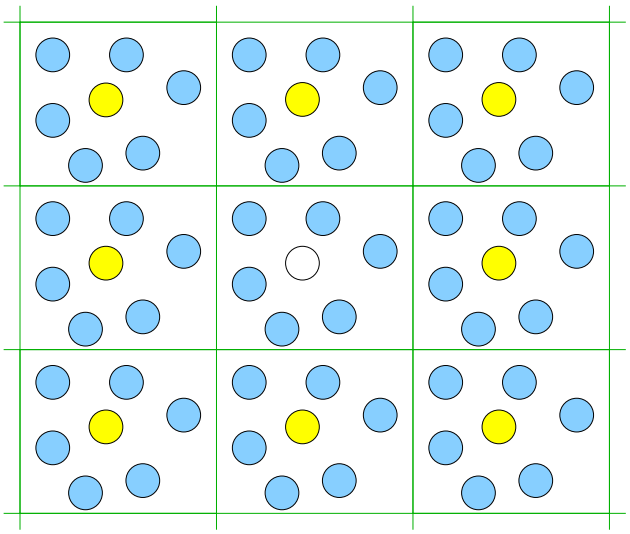


Figure 2.2. 6 particles with periodic boundary conditions in two dimensions represented by an infinite lattice (only the middle 9 cells are shown). Note that the central unshaded particle interacts with all other particles, including its self-images.

Note that the double sum can be written using a matrix formulation,

$$E(x) = \frac{1}{2} q^{\mathsf{T}} K(x) q \qquad \text{where} \qquad K_{ij}(x) = \begin{cases} 0, & i,j \text{ excluded,} \\ f(|\vec{r}_j - \vec{r}_i|), & \text{otherwise.} \end{cases} \tag{2.2}$$

The computational task is essentially equivalent to computing a matrix–vector product to approximate $K(x)q$, and derivatives of $K(x)q$. Once this product is obtained, a simple dot–product with q is used to compute the double sum.

For systems with periodic boundary conditions, the double sum must be extended to include periodic images. This is achieved by extending the finite sum to an infinite sum on a lattice. For example, for a periodically tiled rectangular cell with dimensions $\vec{L}_x$, $\vec{L}_y$, and $\vec{L}_z$, the kernel is replaced by

$$k(\vec{r},\vec{r}') = \sum_{\vec{m}\in S} f(|\vec{r}' - \vec{r} + \vec{m}|)$$

where S is the set of offsets of all lattice images,

$$S = \left\{ [n_{\mathrm{x}} L_x, n_{\mathrm{y}} L_y, n_{\mathrm{z}} L_z]^{\mathsf{T}} \in \mathbb{R}^3 \,\middle|\, n_{\mathrm{x}}, n_{\mathrm{y}}, n_{\mathrm{z}} \in \mathbb{Z} \right\}. \tag{2.3}$$

We should note that this sum may not converge, or may be only conditionally convergent if the potential decays slowly, and one must be careful in defining the summation order. For example, for Coulomb potentials, the kernel is proportional to $1/r$, and the derivative (used to obtain the force) is proportional to $1/r^2$. In three dimensions, the sum is conditionally convergent, and divergent if the system is not charge neutral. To address these issues, the convention is to use the summation order prescribed by the Ewald sum, and to use a neutralizing background charge.

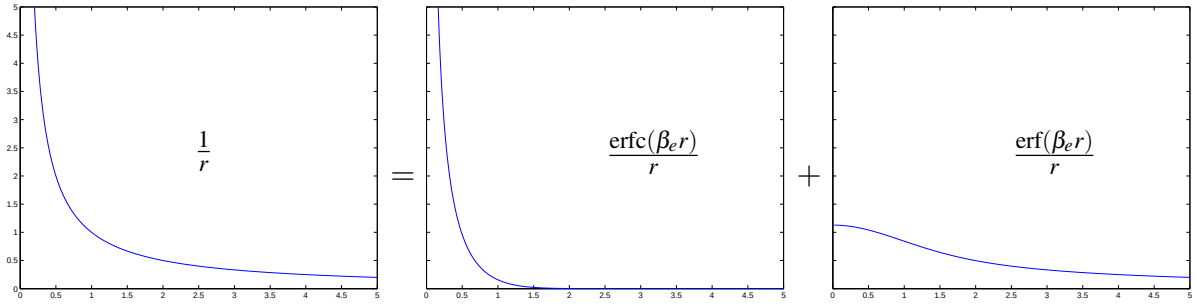


Figure 2.3. Ewald splitting with $\beta_e = 1$

In the Ewald sum, the kernel is split into two kernels,

$$\frac{1}{r} = \frac{\operatorname{erfc}(\beta_e r)}{r} + \frac{\operatorname{erf}(\beta_e r)}{r},$$

where β_e is a constant, $\operatorname{erf}(r)$ is the error function,

$$\operatorname{erf}(r) = \frac{2}{\sqrt{\pi}} \int_0^r \exp\left(-x^2\right) dx,$$

and $\operatorname{erfc}(r)$ is the complimentary error function,

$$\operatorname{erfc}(r) = 1 - \operatorname{erf}(r).$$

The two terms in the split kernel are called the "real space" and "reciprocal space" terms respectively. The real space term is singular and decays very rapidly, and the corresponding sum is absolutely convergent for any constant value of β_e. The reciprocal space term is non-singular, but decays slowly, asymptotically approaching $1/r$ for large r. However, due to the particular functional form, the Fourier transformation of the reciprocal space term decays rapidly, the corresponding sum is absolutely convergent in Fourier space.

Generalized Poisson Equation

Consider the following generalized Poisson equation (GPE) with zero-at-infinity boundary conditions,

$$-\nabla \cdot (\varepsilon(\vec{r}) \nabla \Phi(\vec{r})) = \rho_c(\vec{r}), \qquad \vec{r} \in \mathbb{R}^3,$$

with $\Phi(\vec{r}) \to 0$ as $\|\vec{r}\| \to \infty$. Here, ε is the spatially dependent electric permittivity, ρ_c is an explicit charge distribution, and Φ is the unknown electrostatic potential. The GPE is a second-order elliptic partial differential equation used to model the electrostatic potential corresponding to a charged molecule immersed in an implicit solvent.

In this report, we assume that the fixed charge distribution is the linear combination of three-dimensional delta distributions, corresponding to fixed point charges,

$$\rho_c(\vec{r}) = \sum_{p=1}^{N} q_{c,p} \delta^3 \left(\vec{r} - \vec{r}_{c,p}\right).$$

Here, $q_{c,p}$ is the signed magnitude of the pth point charge at position $\vec{r}_{c,p}$, and $\delta^3(\vec{r})$ is the three-dimensional delta distribution centered at $\vec{r}$.

When $\varepsilon(\vec{r}) = \varepsilon_m$ is a constant, the generalized Poisson equation reduces to the Poisson equation,

$$-\varepsilon_m \nabla \cdot \nabla \Phi_c(\vec{r}) = \sum_{p=1}^{N} q_{c,p} \delta^3 \left(\vec{r} - \vec{r}_{c,p}\right),$$

and the solution is the linear combination of free-space Green's functions,

$$\Phi_c(\vec{r}) = \frac{1}{\varepsilon_m} \sum_{p=1}^{N} \frac{q_{c,p}}{4\pi |\vec{r} - \vec{r}_{c,p}|}.$$

Now, consider the case of a solute molecule immersed in an unbounded solvent. Let Ω_m be the region of the domain corresponding to the molecule, Ω_s the unbounded region corresponding to the solvent, and Ω_b a narrow boundary layer separating Ω_m from Ω_s (see Figure 2.4). In some models, the boundary layer is assumed to be a boundary surface, but here we assume that it has a small, but finite volume. Note that these three subdomains partition $\Omega = \mathbb{R}^3$, with $\Omega = \overline{\Omega_m \cup \Omega_b \cup \Omega_s}$.

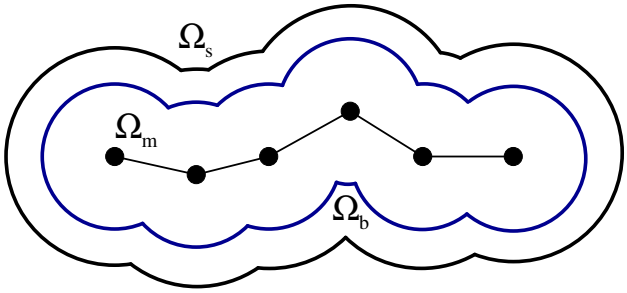


Figure 2.4. Subdomains for a simple chain molecule

For simplicity, we will assume that ε is piecewise constant in the molecular and solvent domains, with $\varepsilon(\vec{r}) = \varepsilon_m$ when $\vec{r} \in \Omega_m$ and $\varepsilon(\vec{r}) = \varepsilon_s$ when $\vec{r} \in \Omega_s$. In the boundary layer, Ω_b, we assume that ε transitions between ε_m and ε_s, so that ε is a twice continuously differentiable on Ω.

Due to the use of a point-charge source term, the solution to the generalized Poisson equation is singular at the locations of the fixed point charges. The standard method for regularizing the solution is to subtract the Coulomb potential, Φ_c, which analytically removes the singularities. The resulting potential is known as the "reaction potential",

$$u(\vec{r}) = \Phi(\vec{r}) - \Phi_c(\vec{r}).$$

The reaction potential is the solution to the regularized generalized Poisson equation:

$$-\nabla \cdot (\varepsilon(\vec{r}) \nabla u(\vec{r})) = \nabla \cdot ((\varepsilon(\vec{r}) - \varepsilon_m) \nabla \Phi_c(\vec{r})),$$

with zero-at-infinity boundary conditions. Note that, if we assume that all the fixed charges are located within Ω_m, then the reaction potential is non-singular. If $\varepsilon(\vec{r})$ is smooth, then the reaction potential is smooth, since the source is the derivative of smooth functions. Once one computes the reaction potential, the electrostatic potential can be readily recovered as the sum of the reaction potential and the Coulomb potential.

The Poisson-Boltzmann equation is a second-order nonlinear partial differential equation, which can be interpreted as an extension of the generalized Poisson equation to account for mobile ions within the solvent. For a system with N_s species of mobile ions, each with charge q_i, the Poisson-Boltzmann equation can be written as

$$-\nabla \cdot (\varepsilon(\vec{r})\nabla\Phi(\vec{r})) = \rho_c(\vec{r}) + \sum_{i=1}^{N_s} \mathrm{sign}(q_i)\kappa_i^2(\vec{r}) \exp\left[-\frac{q_i\Phi(\vec{r})}{k_\mathrm{B}T}\right].$$

Here, k_B is Boltzmann's constant, T is temperature, and $\kappa_i^2(\vec{r})$ is a spatially-dependent function which describes the accessible volume for each ionic species. This function, κ_i^2, is zero within the molecular region, and equal to a positive constant in the solvent region.

This page intentionally left blank.

Chapter 3

Approximation methods

Overview

There are two common approaches for approximating long-range pairwise particle interactions. The first approach is to use a hierarchical clustering method (HCM), which subdivides the particles into a multilevel hierarchy of clusters. The interactions between particles are computed exactly for neighboring particles, and approximated for more distant particles. Examples of HCMs include tree methods [19], the fast multiple method (FMM) [29], and accelerated Cartesian expansions (ACE) [66, 3]. The second approach is to use a kernel splitting method (KSM), which splits the kernel into short-range and long-range parts. The short-range interactions are computed exactly, and the long-range interactions are approximated. Examples of KSMs include FFT-based methods, such as particle–mesh Ewald (PME) [22] and particle–particle particle–mesh (P^3M) [35, 7], and multiple grid methods such as the multilevel summation method (MSM) [11, 12, 63, 70, 47].

In the prototypical hierarchical clustering method, one starts by enclosing the N particles in a bounding box, which is then subdivided into eight rectangular cells. Each cell is then recursively subdivided, which results in an "octtree" structure. Each node in the tree corresponds to a cluster of particles in the hierarchy. Interactions between well-separated clusters are approximated using a series (e.g., multipole) expansion. The fast multipole method is an $\mathcal{O}(N)$ algorithm, which is particularly well-suited for systems without periodic boundaries, and with a non-uniform distribution of particles. One drawback of HCMs is that the resulting potential is not smooth, which results

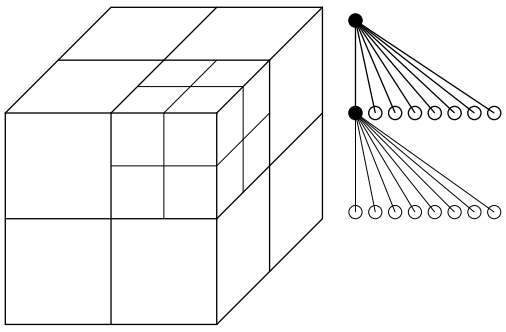
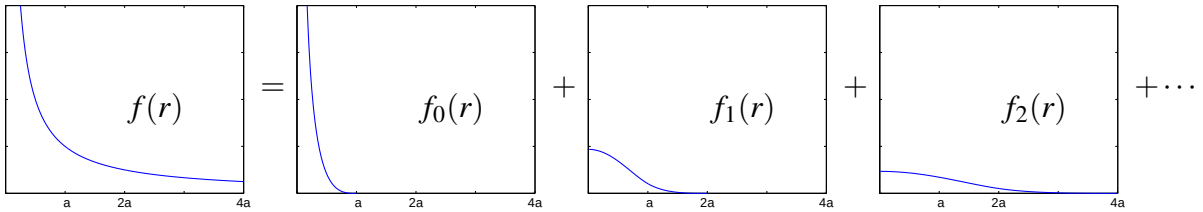


Figure 3.1. An unbalanced octtree with two-levels



in a discontinuous force between particles. This discontinuity can be attributed to the discontinuous change in the clustering as particles move across cell boundaries. However, the size of the discontinuity can be reduced by increasing the accuracy used in the series expansions.

The second approach is to use a kernel splitting method. Like HCMs, kernel splitting methods are based on the separation of length scales. The key difference is that KSMs do not split the particles, but instead split the kernel into a sum of short-range and long-range parts. For the multilevel summation method, the splitting is applied recursively, resulting a series of kernels. For FFT-based Ewald methods, a two-level splitting is used, and the long-range interactions are approximated in Fourier space. The particle mesh Ewald method specifies the Ewald splitting parameter, β_e so that the short-range calculations are $\mathcal{O}(N)$. However, the overall algorithm is $\mathcal{O}(N \log N)$ due to the use of the FFT.

Multilevel Summation

In this report, we focus our attention on the multilevel summation method (MSM) which was first introduced for integral transforms in 1990 [11] and later applied to compute pairwise interactions for electrostatics [63, 70, 47]. In this section, we provide a brief overview of the MSM, generally following the notation and progression presented in David Hardy's PhD thesis [34].

In the MSM, the kernel rewritten as a sum of kernels of increasing range and reduced variation. For example, a four-term splitting of the Coulomb potential can be written as

$$\frac{1}{|\vec{r}-\vec{r}'|} = f_0(|\vec{r}-\vec{r}'|) + f_1(|\vec{r}-\vec{r}'|) + f_2(|\vec{r}-\vec{r}'|) + f_3(|\vec{r}-\vec{r}'|).$$

The short-range part, $f_0(r)$, is calculated directly. The remaining parts are approximated by interpolation on a nested hierarchy of three-dimensional grids of increasing coarseness. This reduces the pairwise interaction problem to a calculation of short-range interactions between points on grids. A splitting parameter, a, determines the range of each kernel. The first three terms of the split kernel have a range of a, $2a$, and $4a$, respectively.

To construct the kernels, one starts with a sequence of smoothing functions, $g_a(\vec{r},\vec{r}')$, which have the property that

$$g_a(\vec{r},\vec{r}') = \frac{1}{|\vec{r}-\vec{r}'|}, \quad \text{for} \quad |\vec{r}-\vec{r}'| > a,$$

but with less variation than original Coulomb potential when $|\vec{r}-\vec{r}'| < a$. Using these smoothing

functions, the Coulomb potential, $|\vec{r}-\vec{r}'|^{-1}$, can be written as a telescoping series, e.g.,

$$f_0(|\vec{r}-\vec{r}'|) = |\vec{r}-\vec{r}'|^{-1} - g_a(\vec{r},\vec{r}'), \qquad f_1(|\vec{r}-\vec{r}'|) = g_a(\vec{r},\vec{r}') - g_{2a}(\vec{r},\vec{r}'),$$
$$f_2(|\vec{r}-\vec{r}'|) = g_{2a}(\vec{r},\vec{r}') - g_{4a}(\vec{r},\vec{r}'), \qquad f_3(|\vec{r}-\vec{r}'|) = g_{4a}(\vec{r},\vec{r}').$$

By construction, f_0, f_1, and f_2 have support $[0,a]$, $[0,2a]$, and $[0,4a]$ respectively. The smoothness of the kernels depends on the smoothness of the smoothing functions. The key is to select each g_a with minimal variation so that it can be well-represented by its interpolant on a grid. The entire series of smoothing functions can be defined using a single master function, γ, with

$$g_a(\vec{r},\vec{r}') := \frac{1}{a}\gamma\left(\frac{1}{a}|\vec{r}-\vec{r}'|\right).$$

The master function has the property that $\gamma(r) = 1/r$ for $r > 1$, but is more smoothly varying than $1/r$ for $r < 1$. Note that at this point we have not introduced any approximations, but simply rewritten the original Coulomb kernel as a sum of kernels of increasing range.

The core approximation used by the multilevel summation method is interpolation of the slowly varying kernels on grids. The shortest-range kernel, f_0, is computed exactly. The slowly varying kernels are approximated using compactly supported nodal basis functions on grids. For example, the function $g_a(\vec{r},\vec{r}')$ can be approximated by the double sum

$$g_a(\vec{r},\vec{r}') \approx \sum_n \sum_m \phi_n^h(\vec{r}) g_a(\vec{r}_n^h, \vec{r}_m^h) \phi_m^h(\vec{r}').$$

Here, $\vec{r}_n^h$ and $\vec{r}_m^h$ are grid points on a regular grid with spacing h. The functions ϕ_n^h and ϕ_m^h are compactly supported nodal basis functions, which vanish outside of a neighborhood of $\vec{r}_n^h$ and $\vec{r}_m^h$ respectively. Applying this approximation to the pairwise interaction problem, we have

$$\frac{1}{2}C\sum_{i=1}^{N}\sum_{j=1}^{N} q_i q_j g_a(\vec{r}_i,\vec{r}_j) \approx \frac{1}{2}C\sum_{i=1}^{N}\sum_{j=1}^{N} q_i q_j \sum_n \sum_m \phi_n^h(\vec{r}_i) g_a(\vec{r}_n^h,\vec{r}_m^h)\phi_m^h(\vec{r}_j{}'),$$

or

$$\frac{1}{2}C\sum_{i=1}^{N}\sum_{j=1}^{N} q_i q_j g_a(\vec{r}_i,\vec{r}_j) \approx \frac{1}{2}C\sum_n \sum_m q_n^h q_m^h g_a(\vec{r}_n^h,\vec{r}_m^h),$$

where

$$q_n^h := \sum_{i=1}^{N} q_i \phi_n^h(\vec{r}_i).$$

Hence, through interpolation, the original pairwise kernel interactions have been approximated by pairwise interactions on a grid. The number of nonzero terms in the gridded sum depends on the ratio of a and h. Reducing the grid spacing, h, increases the accuracy and the number of nonzero terms in the series.

For g_{2a}, we repeat the interpolation process, but using a grid with twice the mesh spacing. The resulting gridded sum,

$$\frac{1}{2}C\sum_{i=1}^{N}\sum_{j=1}^{N} q_i q_j g_{2a}(\vec{r}_i,\vec{r}_j) \approx \frac{1}{2}C\sum_n \sum_m q_n^{2h} q_m^{2h} g_{2a}(\vec{r}_n^{2h},\vec{r}_m^{2h}),$$

will have the same number of nonzero terms, since the ratio of $2a$ to $2h$ is the same as the ratio of a to h.

As we mentioned in the introduction, the pairwise summation problem can be written using a matrix formulation,

$$E(x) = \frac{1}{2} q^{\mathsf{T}} K(x) q,$$

where the entries of the vector q are the charges, and entries of the matrix K are

$$K_{ij}(x) = C \begin{cases} 0, & i, j \text{ excluded,} \\ \|\vec{r}_j - \vec{r}_i\|^{-1}, & \text{otherwise.} \end{cases}$$

The matrix K is dense, but using the interpolation approximation can be written as a sum of products of sparse matrices. Let $I_h(x)$ be a rectangular interpolation matrix, which returns values at particle positions interpolated from gridded values on the grid with spacing h. The transpose of $I_h(x)$ returns gridded values computed from values at particle positions x. Hence, the ith row of $I_h(x)$ is the evaluation at $\vec{r}_i$ of the nodal basis functions associated with the gridpoints of the h grid. For the 4-level non-nested summation method, the matrix $K(x)$ is approximated by

$$K(x) \approx \hat{K}(x) + I_h(x) \hat{K}_h I_h(x)^{\mathsf{T}} + I_{2h}(x) \hat{K}_{2h} I_{2h}(x)^{\mathsf{T}} + I_{4h}(x) K_{4h} I_{4h}(x)^{\mathsf{T}}. \tag{3.1}$$

Here, $\hat{K}(x)$ is constructed using shortest-range kernel, f_0, and does not use any interpolation. The matrices $K_a(x)$, K_{2a}, and K_{4a} are constructed using f_1, f_2, and f_3 respectively, on grids with spacing h, $2h$, and $4h$ respectively. The matrices $I_h(x)$, $I_{2h}(x)$, and I_{4h} transfer the charge from values at atomic locations to grid locations on grids with spacing h, $2h$, and $4h$ respectively. For a particular choice of a and h, the short-range interaction matrices $\hat{K}(x)$, $\hat{K}_h$, $\hat{K}_{2h}$ have 140, 410, 410 nonzeros per row, respectively, and interpolation matrices $I_h(x)$, $I_{2h}(x)$, $I_{4h}(x)$ have 216 nonzeros per row. Note with $\mathcal{O}(\log N)$ levels, the total number of nonzeros in the expansion is $\mathcal{O}(N \log N)$. It has been shown that the optimal grid spacing h is nearly constant [70] with the accuracy controlled by the ratio of a to h.

For some types of interpolation [70, 34], the complexity can be reduced further. If we replace the coarser interpolation matrices with a product of interpolation matrices,

$$I_{2h}(x) = I_h(x) I_{2h}^h, \quad \text{and} \quad I_{4h}(x) = I_h(x) I_{2h}^h I_{4h}^{2h},$$

we can rewrite the approximation as a nested sum,

$$K(x) \approx \hat{K}(x) + I_h(x) \left(\hat{K}_h + I_{2h}^h \left(\hat{K}_{2h} + I_{4h}^{2h} K_{4h} (I_{4h}^{2h})^{\mathsf{T}} \right) (I_{2h}^h)^{\mathsf{T}} \right) I_h(x)^{\mathsf{T}}. \tag{3.2}$$

The interpolation matrices I_{2h}^h and I_{4h}^{2h} have the same number of nonzeros per row as before, but the dimensions are reduced. From h to $2h$, dimensions reduce by 1/8, and from $2h$ to $4h$ by another 1/8. For an sum with $\mathcal{O}(\log N)$ levels the number of nonzeros in the expansion is only $\mathcal{O}(N)$.

To approximate $E(x)$, we approximate $K(x)q$ then compute $q^{\mathsf{T}} K(x) q$. Applying the approximation to $K(x)$ to q we obtain

$$K(x)q \approx \hat{K}(x)q + I_h(x) \left(\hat{K}_h + I_{2h}^h \left(\hat{K}_{2h} + I_{4h}^{2h} K_{4h} (I_{4h}^{2h})^{\mathsf{T}} \right) (I_{2h}^h)^{\mathsf{T}} \right) I_h(x)^{\mathsf{T}} q.$$

To compute forces, we differentiate the approximation, which results in

$$-\frac{1}{2}q^{\mathsf{T}}\frac{\partial K(x)}{\partial x_i}q \approx -\frac{1}{2}q^{\mathsf{T}}\frac{\partial \hat{K}(x)}{\partial x_i}q - q^{\mathsf{T}}\frac{\partial I_h(x)}{\partial x}\left(\hat{K}_h + I_{2h}^h\left(\hat{K}_{2h} + I_{4h}^{2h}K_{4h}(I_{4h}^{2h})^{\mathsf{T}}\right)(I_{2h}^h)^{\mathsf{T}}\right)I_h(x)^{\mathsf{T}}q.$$

Note that only the finest level interpolation operator, I_h, and kernel, $\hat{K}$, are functions of the particle positions. The remaining interpolation operators and kernels are functions of the grid points, and do not change when particles move. This dramatically simplifies the expression for the force, and means that the forces are only explicitly computed on the finest level.

For the four level example, the computation proceeds as follows. On the finest level, compute

$$e_{\text{short}} = \hat{K}(x)q, \quad F_i^{\text{short}} = -\frac{1}{2}q^{\mathsf{T}}\frac{\partial \hat{K}(x)}{\partial x_i}q$$

Progressing forward through the grid hierarchy, compute

$$\begin{aligned} q^h &= (I_h)^{\mathsf{T}}q, \quad & e_{\text{short}}^h &= K_h q^h \\ q^{2h} &= (I_{2h}^h)^{\mathsf{T}}q^h, \quad & e_{\text{short}}^{2h} &= K_{2h}q^{2h} \\ q^{4h} &= (I_{4h}^{2h})^{\mathsf{T}}q^{2h}, \quad & e_{\text{short}}^{4h} &= K_{4h}q^{4h}. \end{aligned}$$

Progressing backward through the grid hierarchy, compute

$$\begin{aligned} e^{4h} &= e_{\text{short}}^{4h} \\ e^{2h} &= e_{\text{short}}^{2h} + I_{4h}^{2h}e^{4h}, \\ e^h &= e_{\text{short}}^h + I_{2h}^h e^{2h}, \end{aligned}$$

Finally, on the finest level, compute

$$E(x) \approx \frac{1}{2}q^{\mathsf{T}}e_{\text{short}} + \frac{1}{2}q^{\mathsf{T}}I_h e^h, \quad F_i \approx F_i^{\text{short}} - q^{\mathsf{T}}\frac{\partial I_h(x)}{\partial x_i}e^h,$$

where i is each of the $3N$ components of the particle positions.

Laplacian-centered Generalized Poisson Equation

We now return our attention to the generalized Poisson equation (GPE) introduced in Chapter 2. The discussion presented in this section is intentionally brief, with a full report included in the appendix.

Recall the regularized GPE with zero-at-infinity boundary conditions,

$$-\nabla\cdot(\varepsilon(\vec{r})\nabla u(\vec{r})) = \nabla\cdot((\varepsilon(\vec{r})-\varepsilon_m)\nabla\Phi_c(\vec{r})).$$

As discussed in Chapter 2, ε is the dielectric function, Φ_c is the Coulomb potential for a fixed point charge distribution ρ_c, and u is the unknown reaction potential. The dielectric function is constant

in the molecular interior, with value ε_m, and constant in the solvent region, with value ε_s. In a thin boundary layer, the function smoothly switches between ε_m and ε_s. The fixed charges are restricted to the molecular interior.

To reformulate the regularized GPE as an integral equation, let $\rho(\vec{r})$ be the effective charge distribution, which is defined by

$$\rho(\vec{r}) = -\varepsilon_m \nabla \cdot \nabla u(\vec{r}),$$

where $u(\vec{r})$ is the unknown reaction potential. Given an effective charge distribution, the reaction potential can be reconstructed using Green's second identity,

$$u(\vec{r}) = \frac{1}{\varepsilon_m} \int \frac{\rho(\vec{r}')}{4\pi|\vec{r}-\vec{r}'|} d\vec{r}'.$$

Inserting this expression into the regularized GPE, we obtain

$$\frac{\varepsilon(\vec{r})}{\varepsilon_m}\rho(\vec{r}) - \left(\nabla \frac{\varepsilon(\vec{r})}{\varepsilon_m}\right) \cdot \left(\nabla \int \frac{\rho(\vec{r}')}{4\pi|\vec{r}-\vec{r}'|} d\vec{r}'\right) = \varepsilon_m \nabla \cdot \left(\left(\frac{\varepsilon(\vec{r})}{\varepsilon_m} - 1\right) \nabla \Phi_c(\vec{r})\right),$$

or

$$\rho(\vec{r}) - \left(\nabla \ln\left(\frac{\varepsilon(\vec{r})}{\varepsilon_m}\right)\right) \cdot \left(\nabla \int \frac{\rho(\vec{r}')}{4\pi|\vec{r}-\vec{r}'|} d\vec{r}'\right) = \varepsilon_m \left(\nabla \ln\left(\frac{\varepsilon(\vec{r})}{\varepsilon_m}\right)\right) \cdot \nabla \Phi_c(\vec{r}), \quad \vec{r} \in \Omega_b,$$

$$\rho(\vec{r}) = 0, \quad \vec{r} \notin \Omega_b$$

where we have used the fact that $\nabla \cdot \nabla \Phi_c = 0$ when $\varepsilon \neq \varepsilon_m$ and that ε is constant outside of the boundary region, Ω_b. Hence, the effective charge distribution, ρ, is zero everywhere outside of a thin boundary region, Ω_b, between the molecular and solvent regions (see Figure 2.4).

To discretize ρ, we use discontinuous constant elements,

$$\rho(\vec{r}) \approx \rho_h(\vec{r}) = \sum_j q_j^h \psi_j^h(\vec{r}),$$

on regular mesh with cubic elements of length h in each direction. The unknown coefficients are denoted by q_j^h. Inserting this expression into the integral equation, we obtain

$$\rho_h(\vec{r}) - \left(\nabla \ln\left(\frac{\varepsilon(\vec{r})}{\varepsilon_m}\right)\right) \cdot \left(\nabla \int \frac{\rho_h(\vec{r}')}{4\pi|\vec{r}-\vec{r}'|} d\vec{r}'\right) \approx \varepsilon_m \left(\nabla \ln\left(\frac{\varepsilon(\vec{r})}{\varepsilon_m}\right)\right) \cdot \nabla \Phi_c(\vec{r}), \quad \vec{r} \in \Omega_b.$$

Multiplying both sides by ψ_i^h and integrating results in a system of discrete equations for the unknown coefficients,

$$q_i^h h^3 - \sum_j q_j^h \int_{E_i^h} \left(\nabla \ln\left(\frac{\varepsilon(\vec{r})}{\varepsilon_m}\right)\right) \cdot \left(\nabla \int_{E_j^h} \frac{1}{4\pi|\vec{r}-\vec{r}'|} d\vec{r}'\right) d\vec{r} = \varepsilon_m \int_{E_i^h} \left(\nabla \ln\left(\frac{\varepsilon(\vec{r})}{\varepsilon_m}\right)\right) \cdot \nabla \Phi_c(\vec{r}) d\vec{r},$$

where E_i^h is the $h \times h \times h$ cubic support of the discontinuous constant basis function, ψ_i^h. In matrix form, the problem can be written as

$$h^3(\mathbf{I} - \mathbf{K})\mathbf{q} = \mathbf{b}, \tag{3.3}$$

where $\mathbf{I}$ is the identity matrix,

$$K_{ij} = \frac{1}{h^3} \int_{E_i^h} \left(\nabla \ln \left(\frac{\varepsilon(\vec{r})}{\varepsilon_m} \right) \right) \cdot \left(\nabla \int_{E_j^h} \frac{1}{4\pi |\vec{r} - \vec{r}'|} d\vec{r}' \right) d\vec{r},$$

and

$$b_i = \varepsilon_m \int_{E_i^h} \left(\nabla \ln \left(\frac{\varepsilon(\vec{r})}{\varepsilon_m} \right) \right) \cdot \nabla \Phi_c(\vec{r}) d\vec{r}.$$

The matrix in equation 3.3 is well-conditioned, but dense. To solve this linear system of equations, we apply the biconjugate gradient method, which requires the matrix-vector product, but does not require explicit construction of the matrix. The matrix-vector product is approximated by first replacing $\ln(\varepsilon(\vec{r})/\varepsilon_m)$ with a trilinear interpolant, then using the multilevel summation method to compute the convolution. The same technique is used for computing the entries in $\mathbf{b}$. To reconstruct the reaction potential, u, from the effective potential, we again use the multilevel summation method.

The computational cost of the resulting algorithm scales linearly with the number of grid points in the boundary region. Since the boundary region has a fixed width, the number of grid points in the boundary region scales linearly as a function the surface area of the molecule and sublinearly as a function of the volume of the molecule. Detailed numerical results are presented in the appendix.

This page intentionally left blank.

Chapter 4

Numerical results

In this section, we test the efficiency and scalability of the multilevel summation method applied to the problem of computing the long-range electrostatic forces for a periodic box of water molecules. This is a prototypical problem for explicit solvent simulations with molecular dynamics, where 90 percent of simulation box is occupied by water. Periodic boundary conditions are used to reduce surface effects, and a large simulation box is needed to reduce spurious finite-size effects.

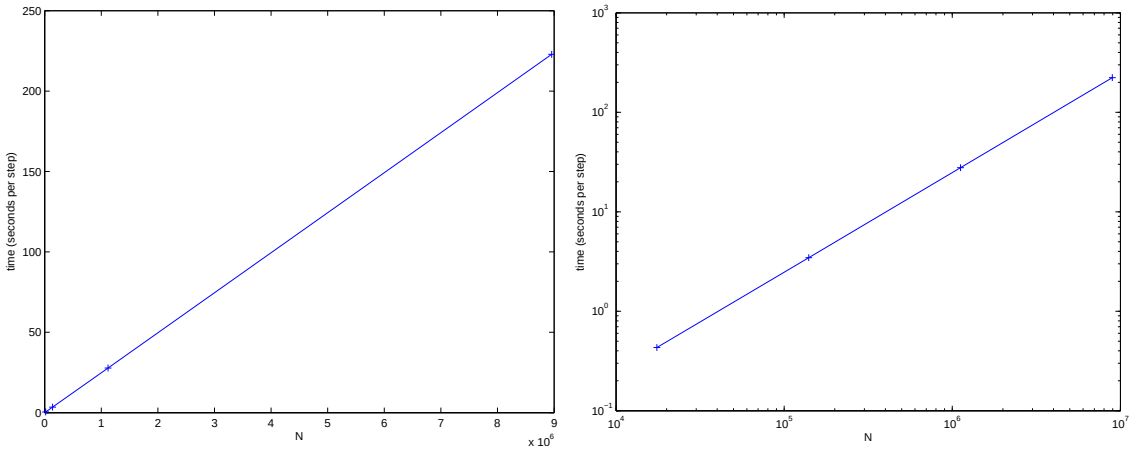


Figure 4.1. Per-step simulation time for 10 steps of molecular dynamics using the multilevel summation method applied to a box of water with N = 17496, 139968, 1119744, and 8957952 atoms. The plot on the right is the same as the left, but with a logarithmic scale.

All of the numerical experiments were conducted using the LAMMPS molecular simulation software package. The multilevel summation method was initially released in LAMMPS in September 2012 by Stan Moore, Stephen Bond, and Paul Crozier. Over the past year, the efficiency and capabilities of the initial implementation has been subsequently improved by Stan Moore, Paul Crozier, and others. We should stress that the implementation is still relatively new, compared to more well-tested and optimized algorithms in LAMMPS. As a result, we are most concerned with showing that the method scales, and expect that the relative speed will improve over time as the implementation is further developed.

To test the scalability of the algorithm, we performed 10 steps of NVE molecular dynamics with a box of 5832, 46656, 373248, and 2985984 water molecules (or N = 17496, 139968, 1119744, and 8957952 atoms). An initial box of 5832 water molecules was used to generate the configurations for all four simulations, successively doubling the box in all three coordinate directions. In all four cases, the multilevel summation method was applied with eighth order interpolation and a relative tolerance of 1 percent. The finest grid used by each simulation was proportional to the system size, with the smallest simulation using a $32 \times 32 \times 32$ grid and the largest using a $256 \times 256 \times 256$ grid. Roughly the same value of a was used in all four simulations. To further improve the efficiency, the shortest range MSM forces were computed at the same time as the other non-bonded interactions. In Figure 4.1, the per-step time to compute the long-range forces is shown as a function of the number of atoms. As expected, the computational cost scales linearly with the number of atoms.

	Time in seconds						
	MSM			PME			Ewald
N	Loop	Pair	Grid	Loop	Pair	Grid	Loop
17496	4.3201	1.469	2.8294	2.0355	1.814	0.199	2.19
139968	34.554	11.83	22.526	16.319	14.51	1.611	45.3
1119744	277.76	95.88	180.21	131.67	117.2	12.76	2239
8957952	2228.7	773.4	1442.2	1052.9	930.5	107.7	-

Table 4.1. Performance comparison for 10 MD steps

In Table 4.1 we present a performance comparison of three different long-range force algorithms in LAMMPS: Multilevel Summation (MSM), Particle Mesh Ewald (PME), and Ewald. Just as in the last test, we performed 10 steps of NVE molecular dynamics with a box of 5832, 46656, 373248, and 2985984 water molecules (or N = 17496, 139968, 1119744, and 8957952 atoms). The relative tolerance was set to one percent for all three methods. In the table, "Loop" is the total simulation loop time, "Pair" is the time spent computing the pairwise short-range forces, and "Grid" is the time spent computing the long range forces. The PME method is about twice as fast as MSM in all of the tests, which we partially attribute to the fact that relatively little effort has been devoted to optimizing the MSM implementation. In the MSM simulations, approximately 65 percent of the computation time is spent computing the gridded long-range forces. In contrast, in the PME simulations, only about 10 percent of the computation time is spent on the grid calculations. In the PME simulations, the smallest simulation used a $9 \times 9 \times 9$ grid, and the largest simulation used a $72 \times 72 \times 72$ grid. This is significantly smaller than the finest grids used by MSM which ranged from $32 \times 32 \times 32$ up to $256 \times 256 \times 256$. For the same short-range splitting parameter, the computational cost of the PME method and MSM should be nearly the same, if the limiting step is the memory access time. Although PME is asymptotically an $O(N \log N)$ method, so little time was spent in the grid calculations that we only observed the transient $O(N)$ complexity. As expected, both PME and MSM are significantly faster than Ewald for large problems, which scales as $O(N^{3/2})$.

Chapter 5

Conclusion

In this report, we provided an overview of the long-range pairwise interaction problem, with a focus on the Coulomb problem, and the Generalized Poisson Equation. We discussed popular methods for approximating long-range pairwise interactions, with an emphasis on the multilevel summation method (MSM). As an application, we derived an integral equation formulation of the regularized Generalized Poisson Equation, using the effective charge as the unknown field variable. After discretization, the resulting system of equations required computation of a convolution with a long-range kernel, which we approximated using MSM. In a sequence of numerical experiments, we tested the implementation of MSM in the LAMMPS software package. Our preliminary testing demonstrated that the computational cost of MSM scales linearly as expected, but the method is still less efficient than PME for moderately sized problems. We anticipate that efficiency of MSM will improve over time, as more effort is spent optimizing the code.

This page intentionally left blank.

References

[1] M. P. Allen and D. J. Tildesley. *Computer Simulation of Liquids*. Oxford Science, Oxford, 1987.

[2] János G. Àngyán, Christophe Chipot, François Dehez, Christof Hättig, Georg Jansen, and Claude Millot. OPEP: A tool for the optimal partitioning of electric properties. *J. Comput. Chem.*, 24:997–1008, 2003.

[3] A. D. Baczewski and B. Shanker. An O(N) method for rapidly computing periodic potentials using accelerated Cartesian expansions. *J. Comput. Phys.*, page submitted, 2011. arXiv preprint:1107.3069.

[4] N. A. Baker, M. J. Holst, and F. Wang. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation II: refinement at solvent accessible surfaces in biomolecular systems. *J. Comput. Chem.*, 21:1343–1352, 2000.

[5] Nathan A. Baker, Donald Bashford, and David A. Case. Implicit solvent electrostatics in biomolecular simulation. In B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schütte, and R. Skeel, editors, *New Algorithms for Macromolecular Simulation*, volume 49 of *Lecture Notes in Computational Science and Engineering*, pages 263–295. Springer-Verlag, 2005.

[6] Nathan A. Baker, David Sept, Simpson Joseph, Michael J. Holst, and J. Andrew McCammon. Electrostatics of nanosystems: Application to microtubules and the ribosome. *Proc. Nat. Acad. Sci.*, 98:10037–10041, 2001.

[7] V. Ballenegger, J. J. Cerda, O. Lenz, and C. Holm. The optimal P3M algorithm for computing electrostatic energies in periodic systems. *J. Chem. Phys.*, 128:34109 (31 pages), 2008.

[8] Danny Barash, Linjing Yang, Xiaoliang Qian, and Tamar Schlick. Inherent speedup limitations in multiple time step/particle mesh Ewald algorithms. *J. Comput. Chem.*, 24:77–88, 2003.

[9] Dmitrii Beglov and Benoît Roux. Solvation of complex molecules in a polar liquid: An integral equation theory. *J. Chem. Phys.*, 104(21):8678–8689, 1996.

[10] J. J. Biesiadecki and R. D. Skeel. Dangers of multiple-time-step methods. *J. Comput. Phys.*, 109:318–328, 1993.

[11] A. Brandt and A. A. Lubrecht. Multilevel matrix multiplication and fast solution of integral equations. *J. Comput. Phys.*, 90:348–370, 1990.

[12] A. Brandt and C. H. Venner. Multilevel evaluation of integral transforms with asymptotically smooth kernels. *SIAM J. Sci. Comput.*, 19:468–492, 1998.

[13] James M. Briggs and J. Andrew McCammon. Computation unravels mysteries of molecular biophysics. *Computers in Physics*, 6(3):238–243, 1992.

[14] Robert K. Brunner and Laxmikant V. Kale. Handling application-induced load imbalance using parallel objects. In Takayasu Ito and Taiichi Yuasa, editors, *Parallel and Distributed Computing for Symbolic and Irregular Applications*, pages 167–181. World Scientific Publishing, 2000.

[15] Trevor Cickovski, Chris Sweet, and Jesús Izaguirre. MDL overview. `https://simtk.org/home/mdl`.

[16] C. M. Cortis and R. A. Friesner. Numerical solution of the Poisson-Boltzmann equation using tetrahedral finite-element meshes. *J. Comput. Chem.*, 18:1591–1608, 2000.

[17] M. E. Davis and J. A. McCammon. Solving the finite difference linearized Poisson-Boltzmann equation: A comparison of relaxation and conjugate gradient methods. *J. Comput. Chem.*, 10:386–391, 1989.

[18] S. W. de Leeuw, J. W. Perram, and E. R. Smith. Simulation of electrostatic systems in periodic boundary conditions. I. Lattice sums and dielectric constants. *Proc. R. Soc. Lond. A*, 373:27–56, 1980.

[19] Zhong-Hui Duan and Robert Krasny. An adaptive treecode for computing nonbonded potential energy in classical molecular systems. *J. Comput. Chem.*, 22:184–195, 2001.

[20] R. Elber. Long-timescale simulation methods. *Curr. Opin. Struc. Biol.*, 15:151–156, 2005.

[21] Robert D. Engle, Robert D. Skeel, and Matthew Drees. Monitoring energy drift with shadow Hamiltonians. *J. Comput. Phys.*, 206(2):432–452, July 1, 2005.

[22] U. Essmann, L. Perera, M. L. Berkowitz, T. Darden, H. Lee, and L. Pederson. A smooth particle mesh Ewald method. *J. Chem. Phys.*, 103:8577–8593, 1995.

[23] Michael Feig and Charles L. Brooks III. Recent advances in the development and application of implicit solvent models in biomolecule simulations. *Curr. Op. Struct. Bio.*, 14(2):217–224, 2004.

[24] Michael Feig, Alexey Onufriev, Michael S. Lee, Wonpil Im, David A. Case, and Charles L. Brooks III. Performance comparison of generalized Born and Poisson methods in the calculation of electrostatic solvation energies for protein structures. *J. Comput. Chem.*, 25(2):265–284, 2004.

[25] Daan Frenkel and Berend Smit. *Understanding Molecular Simulation*. Academic Press, New York, 2nd edition, 2002.

[26] Razif R. Gabdoulline and Rebecca C. Wade. Biomolecular diffusional association. *Curr. Opin. Struc. Biol.*, 12:204–213, 2002.

[27] W. H. Geng, S. N. Yu, and G. W. Wei. Treatment of charge singularities in implicit solvent models. *J. Chem. Phys.*, 127:114106 (20 pages), 2007.

[28] M. K. Gilson, K. A. Sharp, and B. H. Honig. Calculating the electrostatic potential of molecules in solution: Method and error assessment. *J. Comput. Chem.*, 9:327–335, 1987.

[29] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *J. Comput. Phys.*, 73:325–348, 1987.

[30] L. Greengard and V. Rokhlin. A new version of the fast multipole method for the Laplace equation in three dimensions. *Acta Numerica*, 6:229–269, 1997.

[31] H. Grubmüller, H. Heller, A. Windemuth, and K. Schulten. Generalized Verlet algorithm for efficient molecular dynamics simulations with long-range interactions. *Molecular Simulation*, 6:121–142, 1991.

[32] T. A. Halgren and W. Damm. Polarizable force fields. *Curr. Opin. Struct. Biol.*, 11:236–242, 2001.

[33] David J. Hardy. NAMD-Lite. `http://www.ks.uiuc.edu/Development/MDTools/namdlite/`.

[34] David Joseph Hardy. *Multilevel Summation for the Fast Evaluation of Forces for the Simulation of Biomolecules*. PhD thesis, Univ. of Illinois at Urbana-Champaign, 2006.

[35] R. W. Hockney and J. W. Eastwood. *Computer Simulation Using Particles*. McGraw-Hill, New York, 1981.

[36] M. J. Holst, N. A. Baker, and F. Wang. Adaptive multilevel finite element solution of the Poisson-Boltzmann equation I: Algorithms and examples. *J. Comput. Chem.*, 21:1319–1342, 2000.

[37] M. J. Holst and F. Saied. Numerical solution of the nonlinear Poisson-Boltzmann equation: Developing more robust and efficient methods. *J. Comput. Chem.*, 16:337–364, 1995.

[38] Michael Holst. Adaptive numerical treatment of elliptic systems on manifolds. *Adv. Comput. Math.*, 15:139–191, 2001.

[39] William Humphrey, Andrew Dalke, and Klaus Schulten. VMD – Visual Molecular Dynamics. *J. Molec. Graphics*, 14:33–38, 1996.

[40] Wonpil Im, Dmitrii Beglov, and Benoît Roux. Continuum solvation model: computation of electrostatic forces from numerical solutions to the Poisson-Boltzmann equation. *Comput. Phys. Comm.*, 111:59–75, 1998.

[41] Jesús A. Izaguirre, Scott S. Hampton, and Thierry Matthey. Parallel multigrid summation for the N-body problem. *J. Parallel Distributed Computing*, 65:949–962, 2005.

[42] J. D. Jackson. *Classical Electrodynamics*. John Wiley & Sons, Inc., New York, NY, 2nd edition, 1975.

[43] Hans Johansen and Phillip Colella. A cartesian grid embedded boundary method for Poissonś equation on irregular domains. *J. Comput. Phys.*, 147(1):60–85, 1998.

[44] W. L. Jorgensen, D. S. Maxwell, and J. Tirado-Rives. Development and testing of the OPLS all-atom force field on conformational energetics and properties of organic liquids. *J. Am. Chem. Soc.*, 118:11225–11236, 1996.

[45] Patrice Koehl. Electrostatics calculations: Latest methodological advances. *Curr. Opin. Struct. Biol.*, 16:142–151, 2006.

[46] Shihhsien S. Kuo, Michael D. Altman, Jaydeep P. Bardhan, Bruce Tidor, and Jacob K. White. Fast methods for simulation of biomolecule electrostatics. In *ICCAD '02: Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design*, pages 466–473, New York, NY, USA, 2002. ACM Press.

[47] M. S. Lee, Jr. F. R. Salsbury, and M. A. Olson. An efficient hybrid implicit/explicit solvent method for biomolecular simulations that uses a generalized Born reaction field and multigrid enhancements. *J. Comput. Chem.*, 25:1967–1978, 2004.

[48] Oren E. Livne and Achi Brandt. N roots of the secular equation in $O(N)$ operations. *SIAM J. Matrix Anal. and Appl.*, 24:439–453, 2002.

[49] Benzhuo Lu, Xiaolin Cheng, Jingfang Huang, and J. Andrew McCammon. Order *n* algorithm for computation of electrostatic interactions in biomolecular systems. *Proc. Natl. Acad. Sci USA*, 103(51):19314–19319, 2006.

[50] Qun Ma, Jesús Izaguirre, and Robert D. Skeel. Nonlinear instability in multiple time-stepping molecular dynamics. In *Proceedings of the 18th ACM Symposium on Applied Computing (SAC'03)*, pages 167–171, 2003.

[51] Qun Ma, Jesús Izaguirre, and Robert D. Skeel. Verlet-I/r-RESPA is limited by nonlinear instability. *SIAM J. Sci. Comput.*, 24(6):1951–1973, May 6, 2003.

[52] Thierry Matthey, Trevor Cickovski, Scott Hampton, Alice Ko, Qun Ma, Matthew Nyerges, Troy Raeder, Thomas Slabach, and Jesús A. Izaguirre. ProtoMol, an object-oriented framework for prototyping novel algorithms for molecular dynamics. *ACM Trans. Math. Softw.*, 20:237–265, 2004.

[53] J. A. McCammon. Greater computing power and collaboration help advance molecular dynamics simulations. *NPACI & SDSC Envision*, 13(1):2–3, 1998.

[54] J. A. McCammon and S. C. Harvey. *Dynamics of proteins and nucleic acids*. Cambridge University Press, Cambridge, 1987.

[55] G. N. Milstein. *The Numerical Integration of Stochastic Differential Equations*. Urals Univ. Press, Sverdlovsk, 1988. English ed. by Kluwer Academic Publishers, Dordrecht, 1995.

[56] Aiichiro Nakano. Parallel multilevel preconditioned conjugate-gradient approach to variable-charge molecular dynamics. *Computer Physics Comm.*, 104:59–69, 1997.

[57] Mark Nelson, William Humphrey, Attila Gursoy, Andrew Dalke, Laxmikant Kalé, Robert D. Skeel, and Klaus Schulten. NAMD—a parallel, object-oriented molecular dynamics program. *Intl. J. Supercomput. Applics. High Performance Computing*, 10:251–268, Winter 1996.

[58] Mafalda Nina, Dmitri Beglov, and Benoît Roux. Atomic radii for continuum electrostatics calculations based on molecular dynamics free energy simulations. *J. Phys. Chem. B*, 101:5239–5248, 1997.

[59] Sandeep Patel, Jr. Alexander D. MacKerell, and Charles L. Brooks III. CHARMM fluctuating charge force field for proteins, ii: Protein/solvent properties from molecular dynamics simulations using a nonadditive electrostatic model. *J. Comput. Chem.*, 25:1504–1514, 2004.

[60] James C. Phillips, Rosemary Braun, Wei Wang, James Gumbart, Emad Tajkhorshid, Elizbeth Villa, Christophe Chipot, Robert D. Skeel, Laxmikant Kalé, and Klaus Schulten. Scalable molecular dynamics with NAMD. *J. Comput. Chem.*, 26:1781–1802, 2005.

[61] E. L. Pollock and J. Glosli. Comments on P^3M, FMM, and the Ewald method for large periodic coulombic systems. *Computer Phys. Commun.*, 95:93–110, 1996.

[62] W. Rankin and J. Board. A portable distributed implementation of the parallel multipole tree algorithm. *The 4th International Symposium on High Performance Distributed Computing, Proceedings*, pages 17–22, 1995. [Duke University Technical Report 95-002].

[63] Bilha Sandak. Multiscale fast summation of long range charge and dipolar interactions. *J. Comput. Chem*, 22:717–731, 2001.

[64] Michel Sanner. PMV overview. `http://mgltools.scripps.edu/packages/pmv`.

[65] Tamar Schlick. *Molecular Modeling and Simulation: An Interdisciplinary Guide*. Springer-Verlag, New York, 2002.

[66] B. Shanker and H. Huang. Accelerated cartesian expansions – a fast method for computing of potentials of the form $R^{-\nu}$ for all real ν. *J. Comput. Phys.*, 226:732–753, 2007.

[67] K. A. Sharp and B. Honig. Electrostatic interactions in macromolecules: Theory and applications. *Ann. Rev. Biophys. Biophys. Chem.*, 19:301–332, 1990.

[68] Robert D. Skeel, David J. Hardy, and James C. Phillips. Correcting mesh-based force calculations to conserve both energy and momentum in molecular dynamics simulations. *J. Comput. Phys.*, 225(1):1–5, July 1, 2007.

[69] Robert D. Skeel and K. Srinivas. Nonlinear stability analysis of area-preserving integrators. *SIAM J. Numer. Anal.*, 38:129–148, 2000.

[70] Robert D. Skeel, Ismail Tezcan, and David J. Hardy. Multiple grid methods for classical molecular dynamics. *J. Comput. Chem.*, 23:673–684, 2002.

[71] W. Clark Still, Anna Tempczyk, Ronald C. Hawley, and Thomas Hendrickson. Semianalytical treatment of solvation for molecular mechanics and dynamics. *J. Am. Chem. Soc.*, 112(16):6127–6129, 1990.

[72] John E. Stone, James C. Phillips, Peter L. Freddolino, David J. Hardy, Leonardo G. Trabuco, and Klaus Schulten. Accelerating molecular modeling applications with graphics processors. *J. Comput. Chem.*, 28:2618–2640, 2007.

[73] Y. Sun and R. A. Latour. Comparison of implicit solvent models for the simulation of protein-surface interactions. *J. Comput. Chem.*, 27(16):1908–1922, 2006.

[74] Ilario Tironi, Rene Sperb, Paul E. Smith, and Wilfred F. van Gunsteren. A generalized reaction field method for molecular dynamics simulations. *J. Chem. Phys.*, 102:5451–5459, 1995.

[75] Abdulnour Toukmaji, Celeste Sagui, John Board, and Tom Darden. Efficient particle-mesh Ewald based approach to fixed and induced dipolar interactions. *J. Chem. Phys.*, 113:10913–10927, 2000.

[76] M. Tuckerman, B. J. Berne, and G. J. Martyna. Reversible multiple time scale molecular dynamics. *J. Chem. Phys.*, 97:1990–2001, 1992.

[77] C. Verlinde and W. Hol. Structure-based drug design: Progress, results, and challenges. *Structure*, 2:577–587, July 15, 1994.

[78] Andreas Vitalis, Nathan A. Baker, and J. Andrew McCammon. ISIM: a program for grand canonical Monte Carlo simulations of the ionic environment of biomolecules. *Mol. Sim.*, 30:45–61, 2004.

[79] Y. N. Vorobjev and H. A. Scheraga. A fast adaptive multigrid boundary element method for macromolecular electrostatic computations in a solvent. *J. Comput. Chem.*, 18:569–583, 1997.

[80] Wei Wang. *Fast Polarizable Force Field Computation in Biomolecular Simulations*. PhD thesis, Univ. of Illinois at Urbana-Champaign, 2005. Also Department of Computer Science Report No. UIUCDCS-R-2005-2522, 2005. Available online at `ftp://ftp.cs.uiuc.edu/pub/dept/tech_reports/2005/UIUCDCS-R-2005-2522.pdf.gz`.

[81] Wei Wang and Robert D. Skeel. Fast evaluation of polarizable forces. *J. Chem. Phys.*, 123:164107 (12 pages), 2005.

[82] P. J. Whittle and T. L Blundell. Protein structure-based drug design. *Annu. Rev. Biophys. Biomol. Struct.*, 23:349–375, 1994.

[83] X. Wu and B. R. Brooks. Isotropic periodic sum: a method for the calculation of long-range interactions. *J. Chem. Phys.*, 122:44107 (18 pages), 2005.

[84] R. J. Zahaur and R. S. Morgan. The rigorous computation of the molecular electric-potential. *J. Comput. Chem.*, 9:171–187, 1988.

[85] H.-X. Zhou. Boundary element solution of macromolecular electrostatics: interaction energy between two proteins. *Biophys. J.*, 65:955–963, 1993.

[86] R. H. Zhou, E. Harder, H. F. Xu, and B. J. Berne. Efficient multiple time step method for use with Ewald and particle mesh Ewald for large biomolecular systems. *J. Chem. Phys.*, 115, 2001.

[87] Zhongxiang Zhou, Philip Payne, Max Vasquez, Nat Kuhn, and Michael Levitt. Finite-difference solution of the Poisson-Boltzmann equation: Complete elimination of self-energy. *J. Comput. Chem.*, 17(11):1344–1351, 1996.

[88] Gang Zou and Robert D. Skeel. Robust biased Brownian dynamics for rate constant calculation. *Biophys. J.*, 85(4):2147–2157, October 2003.

[89] Gang Zou and Robert D. Skeel. Robust variance reduction for random walk methods. *SIAM J. Sci. Comput.*, 25(6):1964–1981, May 25, 2004.

[90] Gang Zou, Robert D. Skeel, and Shankar Subramaniam. Biased Brownian dynamics for rate constant calculation. *Biophys. J.*, 79:638–645, 2000.

This page intentionally left blank.

Appendix A

Application to the Generalized Poisson Equation

In this section, we include a copy of a report by Dmytro S. Yershov, Stephen D. Bond, and Robert D. Skeel on a "fast Laplacian-centered method for the generalized Poisson equation". The method applies the multilevel summation method to an integral formulation of the generalized Poisson equation. In the Laplacian-centered approach, one solves for an unknown effective charge distribution, which is the Laplacian of the unknown reaction potential. The reaction potential can be reconstructed by solving the related Poisson equation using a fast Poisson solver, like the multilevel summation method. One benefit of using a Laplacian-centered approach is that effective charge distribution has lower regularity, and is supported on a thin boundary region at the solute–solvant interface. The integral equation for the effective charge distribution is inherently well-conditioned, and can be solved with a few iterations using a standard Krylov method. A summary of the results can be found in Chapter 3, and the complete report on this application of the multilevel summation method is included in the remaining sections of the appendix.

A Fast Laplacian-Centered Method for the Generalized Poisson Equation

Dmytro S. Yershov[a,1], Stephen D. Bond[b,2], Robert D. Skeel[c,3]

[a]*University of Illinois at Urbana-Champaign, Department of Computer Science, 201 N. Goodwin Ave., Urbana, IL 61801, USA*
[b]*Sandia National Laboratories, P.O. Box 5800, MS 1318, Albuquerque, NM 87185, USA*
[c]*Purdue University, Department of Computer Science, 305 N. University St., West Lafayette, IN 47907, USA*

1. Motivation

Solvation effects play a critical role in determining the structure and function of biomolecular systems [1, 2, 3, 4]. Accurate calculation of solvation energetics is an essential component of molecular dynamics (MD) [5, 6], Brownian dynamics (BD) [7], and Monte-Carlo methods [8]. Properly modeling solvent interactions with explicit water molecules is computationally expensive due to the need to evaluate many more terms in the energy function and to sample the many configurations of water molecules. The addition of explicit water molecules around a biomolecule typically increases the total number of atoms in the system by a factor of eight to ten.

For large systems, one method for reducing this expense is through an implicit solvent model, which replaces the explicit atoms with a dielectric continuum. The most popular model for implicit solvent is generalized Born, which represents the biomolecule by a union of spheres and approximates the solvation energy using an effective radial potential [9, 10]. A more accurate implicit solvent description can be obtained from the generalized Poisson equation (GPE) or the Poisson-Boltzmann equation (PBE), which are elliptic partial differential equations (PDEs) for the electrostatic potential. In fact, generalized Poisson and Poisson-Boltzmann are often used for fitting the parameters in the generalized Born model [11, 12, 13].

Historically, the use of PDE methods for electrostatics has been considered too expensive for use in dynamics or sampling, even in comparison to explicit solvent [14]. The most popular numerical methods for solving the PBE (and linearized PBE) include finite difference/volume [15, 16, 17, 18, 19, 20], boundary element [21, 22, 23, 7, 24, 25], boundary integral [26, 27], and finite

[1]D. S. Yershov, yershov2@illinois.edu, was supported in part by the National Science Foundation (CCF 08-30578).

[2]S. D. Bond, sdbond@sandia.gov, was supported in part by the National Science Foundation (CCF 08-30578), Sandia National Laboratories LDRD program, and by the Applied Mathematics Program within the Department of Energy (DOE) Office of Advanced Scientific Computing Research (ASCR) as part of the Collaboratory on Mathematics for Mesoscopic Modeling of Materials (CM4). Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

[3]R. D. Skeel, skeel@cs.purdue.edu, is supported in part by the National Science Foundation (CCF 08-30582).

element methods [28, 29, 30, 31]. For highly charged macromolecules, the mean-field approximation in the PBE is no longer valid and grand canonical Monte Carlo has been used to generate ion distributions to sample varying solutions to the GPE [32]. A fast GPE solver is essential for this method to be computationally feasible.

In this paper, we propose a fast method for calculating the electrostatic potential for the GPE with a soft solvent-solute interface. The proposed method represents the solution in terms of its Laplacian, which is proportional to an "effective" charge distribution, ρ, which is compactly supported in a boundary layer along the solvent-solute interface. The equation for ρ is a linear integral equation with a simple kernel, which when discretized, results in a well-conditioned linear system that can be solved iteratively. Once this effective charge distribution is found, the corresponding electrostatic potential can be computed by convolution with the Green's function of the Laplacian.

In the proposed approach, a new challenge arises from discretization of an integral convolution, which generically leads to dense matrices. Standard methods for computing the matrix-vector product require $O(N^2)$ operations, where N is the number of degrees of freedom of a given discretization. To overcome this difficulty we use the *Multilevel Summation Method* (MSM) [33, 34]. MSM is an N-body solver which uses a series of sparse linear matrices defined on a set of nested grids to approximate the corresponding integral convolution. The cost of computing each matrix-vector product is proportional to the number degrees of freedom of the corresponding grid. As a result, for a fixed tolerance, the total cost of the procedure scales as $O(N)$.

The remainder of the paper is organized as follows. In Section 2, we discuss the generalized Poisson equation, the regularized Poisson equation for the reaction potential, and the integral equation for the effective charge distribution. In Section 3, we discuss the projection method [35] to solve the resulting integral equation. The iterative linear solver and running time analysis are presented in Section 4. In Section 5, we apply MSM to reduce the running time of the iterative solver. In Section 6, the error and the running time of the algorithm are investigated through a series of numerical experiments. Finally, we present our conclusions in Section 7.

2. Problem formulation

The generalized Poisson equation (GPE) is a second-order elliptic partial differential equation, which can solved to obtain the electrostatic potential of a solute molecule immersed in an implicit solvent. The equation for the unknown electrostatic potential, Φ, can be written as

$$-\nabla \cdot (\epsilon(\boldsymbol{r})\nabla\Phi(\boldsymbol{r})) = \rho_{\mathrm{c}}(\boldsymbol{r}), \quad \boldsymbol{r} \in \mathbb{R}^3\,, \tag{1}$$

in which Φ is an unknown electrostatic potential field, ϵ is a space-dependent dielectric permittivity of a medium, and ρ_{c} is an explicit charge distribution. Boundary conditions for (1) are posed at infinity, requiring $\Phi(\boldsymbol{r}) \to 0$ as $\|\boldsymbol{r}\| \to \infty$.

We assume that ϵ takes a constant value ϵ_{m} in the molecular region, Ω_{m}, and a constant value ϵ_{s}, in the solvent region, Ω_{s}; see Figure 1. Let ϵ vary continuously in the boundary region, Ω_{b}, so that a soft solvent-solute interface is defined. Finally, assume the explicit charge distribution is described as a superposition of point charges, i.e.

$$\rho_{\mathrm{c}}(\boldsymbol{r}) = \sum_{p=1}^{N} q_{\mathrm{c},p}\delta^3(\boldsymbol{r} - \boldsymbol{r}_{\mathrm{c},p})\,, \tag{2}$$

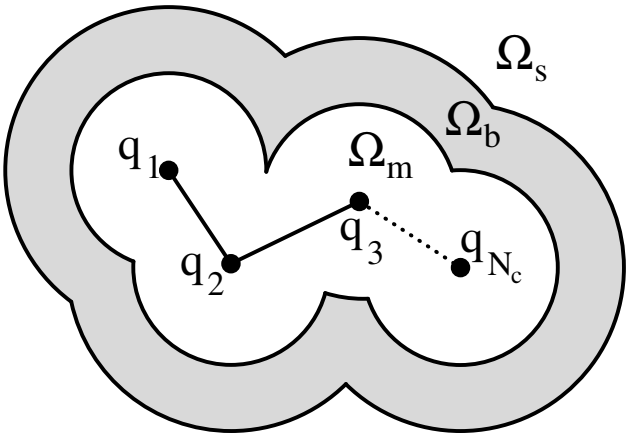


Figure 1: Space decomposition near a molecule

in which $q_{c,p}$ is the magnitude of pth point charge, $\boldsymbol{r}_{c,p}$ is the position vector of pth point charge, and $\delta^3(\boldsymbol{r}) = \delta(x)\delta(y)\delta(z)$ is a three-dimensional delta distribution centered at $\boldsymbol{r} = [x, y, z]$.

The solution of (1) with the right-hand side (2) is singular, with $|\Phi(\boldsymbol{r})| \to \infty$ as $\boldsymbol{r} \to \boldsymbol{r}_{c,p}$ for any p, which causes difficulties for most numerical methods. To address this issue, the problem is regularized by subtracting the singular Coulomb potential from the solution. This approach leads to the *regularized* Poisson equation for the remainder potential.

To derive the regularized GPE, consider the Coulomb potential for N point charges, $q_{c,p}$, centered at positions $\boldsymbol{r}_{c,p}$,

$$\Phi_c(\boldsymbol{r}) = \frac{1}{\epsilon_m} \int_{\mathbb{R}^3} \frac{\rho_c(\boldsymbol{r}')}{4\pi|\boldsymbol{r} - \boldsymbol{r}'|} \, \mathrm{d}^3\boldsymbol{r}\,' = \frac{1}{\epsilon_m} \sum_{p=1}^{N} \frac{q_{c,p}}{4\pi|\boldsymbol{r} - \boldsymbol{r}_{c,p}|} \,. \tag{3}$$

This potential satisfies the Poisson equation

$$-\nabla \cdot (\epsilon_m \nabla \Phi_c(\boldsymbol{r})) = \rho_c(\boldsymbol{r}) \,. \tag{4}$$

Subtract (4) from (1) to obtain the regularized Poisson equation:

$$-\nabla \cdot (\epsilon(\boldsymbol{r}) \nabla u(\boldsymbol{r})) = \nabla \cdot ((\epsilon(\boldsymbol{r}) - \epsilon_m) \nabla \Phi_c(\boldsymbol{r})) \,, \tag{5}$$

in which $u(\boldsymbol{r}) = \Phi(\boldsymbol{r}) - \Phi_c(\boldsymbol{r})$ is called the *reaction* potential. Boundary conditions for (5) are posed at infinity, with $u(\boldsymbol{r}) \to 0$ as $\|\boldsymbol{r}\| \to \infty$. Note that if ϵ is smooth, the right-hand side in (5) is smooth; hence, the reaction potential is also smooth. The solution of (1) is given as a superposition of the reaction potential and the Coulomb potential, i.e. $\Phi(\boldsymbol{r}) = u(\boldsymbol{r}) + \Phi_c(\boldsymbol{r})$. A standard approach to find an approximate solution for (1) is to find an approximate solution for (5) and to superimpose it with the Coulomb potential given in (3).

The reaction potential is still difficult to approximate numerically, because it is nonzero everywhere in $\mathbb{R}^3$, and $\mathbb{R}^3$ is unbounded. We circumvent this difficulty by reformulating (5) in terms of the Fredholm integral equation of the second kind for the *effective* charge distribution. We show that the effective charge is trivial at every point outside of Ω_b.

Consider an expansion of (5) multiplied by $\epsilon_{\mathrm{m}}/\epsilon(\boldsymbol{r})$,

$$-\epsilon_{\mathrm{m}}\nabla^2 u(\boldsymbol{r}) - \epsilon_{\mathrm{m}}\nabla \ln(\epsilon(\boldsymbol{r})/\epsilon_{\mathrm{m}}) \cdot \nabla u(\boldsymbol{r}) = \epsilon_{\mathrm{m}}\nabla \ln(\epsilon(\boldsymbol{r})/\epsilon_{\mathrm{m}}) \cdot \nabla \Phi_{\mathrm{c}}(\boldsymbol{r}) \,, \quad \boldsymbol{r} \in \Omega_{\mathrm{b}} \,. \tag{6}$$

Note that $-\epsilon_{\mathrm{m}}\nabla^2 u(\boldsymbol{r})$ is zero for all $\boldsymbol{r}$ such that $\nabla\epsilon(\boldsymbol{r}) = 0$, or, simply, $\boldsymbol{r} \in \mathbb{R}^3 \setminus \Omega_{\mathrm{b}}$. Define the effective charge distribution to be $\rho(\boldsymbol{r}) = -\epsilon_{\mathrm{m}}\nabla^2 u(\boldsymbol{r})$. In this case the effective charge is concentrated at the soft interface and it is a smooth function, generating a smooth reaction potential. This is in contrast to the case of a discontinuous dielectric permittivity, in which the effective charge is concentrated on the surface of the molecule and the reaction potential is a single layer potential.

The reaction potential is derived from the effective charge distribution using the second Green's identity

$$u(\boldsymbol{r}) = \frac{1}{\epsilon_{\mathrm{m}}} \int \frac{\rho(\boldsymbol{r}')}{4\pi|\boldsymbol{r} - \boldsymbol{r}'|} \, \mathrm{d}\boldsymbol{r}' \,. \tag{7}$$

Substituting (7) into (6), obtain the Fredholm integral equation of the second kind for the effective charge distribution

$$\rho(\boldsymbol{r}) - \nabla \ln(\epsilon(\boldsymbol{r})/\epsilon_{\mathrm{m}}) \cdot \nabla \int \frac{\rho(\boldsymbol{r}')}{4\pi|\boldsymbol{r} - \boldsymbol{r}'|} \, \mathrm{d}\boldsymbol{r}' \; = \epsilon_{\mathrm{m}}\nabla \ln(\epsilon(\boldsymbol{r})/\epsilon_{\mathrm{m}}) \cdot \nabla \Phi_{\mathrm{c}}(\boldsymbol{r}) \,. \tag{8}$$

In (8), the function $\nabla \ln(\epsilon(\boldsymbol{r})/\epsilon_{\mathrm{m}}) \cdot \nabla(4\pi|\boldsymbol{r} - \boldsymbol{r}'|)^{-1}$ is called the kernel of the integral operator.

3. Numerical Discretization

In this section, we build a numerical method to approximate the solution of (8). We begin with the discretization of Ω_{b} using a regular Cartesian mesh. Let h be the mesh step size, and N be the total number of cells in the discretization. The cells are indexed using a multi-index vector notation, in which $\boldsymbol{i} = [i_x, i_y, i_z]$ corresponds to a grid cell with integer coordinates i_x, i_y, and i_z. Finally, assume I is the set of all vector indices of a given discretization.

For simplicity, we use a piecewise constant approximation of the effective charge distribution. Let $\hat{\rho}(\boldsymbol{r}) = \sum_{\boldsymbol{j}\in I} q_{\boldsymbol{j}}\phi_{\boldsymbol{j}}(\boldsymbol{r})$ be the approximate effective charge distribution, where $\phi_{\boldsymbol{j}}(\boldsymbol{r})$ are piecewise constant cell basis functions defined as follows:

$$\phi_{\boldsymbol{j}}(\boldsymbol{r}) = \begin{cases} 1 & \text{if } \boldsymbol{r} \in \square_{\boldsymbol{j}}^h \\ 0 & \text{if otherwise} \end{cases} \,. \tag{9}$$

In the above $\square_{\boldsymbol{j}}^h = \{\boldsymbol{r} \in \mathbb{R}^3 \mid |\boldsymbol{r} - h\boldsymbol{j}|_\infty \le h/2\}$ and $|\cdot|_\infty$ is the infinity-norm in $\mathbb{R}^3$.

Define the residual by substituting $\hat{\rho}$ into (8),

$$\hat{\rho}(\boldsymbol{r}) - \nabla \ln(\epsilon(\boldsymbol{r})/\epsilon_{\mathrm{m}}) \cdot \nabla \int \frac{\hat{\rho}(\boldsymbol{r}')}{4\pi|\boldsymbol{r} - \boldsymbol{r}'|} \, \mathrm{d}\boldsymbol{r}' \; - \epsilon_{\mathrm{m}}\nabla \ln(\epsilon(\boldsymbol{r})/\epsilon_{\mathrm{m}}) \cdot \nabla \Phi_{\mathrm{c}}(\boldsymbol{r}) \,. \tag{10}$$

The approximate solution is found by minimizing this residual. We implement the projection method [35], in which a system of linear equations is derived by minimizing the projection of the residual on the linear subspace spanned by test functions.

To derive the resulting system of linear equation, we first need to define an inner product function space in which projection takes place. Consider $L^2(\mathbb{R}^3)$, the space of all real-valued

square-integrable functions in $\mathbb{R}^3$. Further, for any two functions $f,\, g \in L^2(\mathbb{R}^3)$, define the inner product as $\langle f,\, g\rangle = \int_{\Omega_b} f(\boldsymbol{r})g(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}$. The function space $L^2(\mathbb{R}^3)$ equipped with an inner product $\langle\cdot,\,\cdot\rangle$ is an inner product linear space.

We choose test functions to be cell basis functions, ϕ_i. Hence, to minimize the projection of the residual, it is sufficient to require the inner product of all basis functions with the residual to vanish. Using the inner product definition and (10), we find that the vector of coefficients $\boldsymbol{q} = [q_1, \ldots q_N]$ satisfies the system of linear equations

$$\boldsymbol{A}\boldsymbol{q} = \boldsymbol{b}\,, \tag{11}$$

where $\boldsymbol{A} = h^3(\boldsymbol{I} - \boldsymbol{K})$, $\boldsymbol{I}$ is the identity matrix, elements of $\boldsymbol{K}$ are

$$K_{ij} = h^{-3}\int_{\square_i^h} \nabla\ln(\epsilon(\boldsymbol{r})/\epsilon_\mathrm{m})\cdot\nabla\left[\int_{\square_j^h}\frac{1}{4\pi|\boldsymbol{r}-\boldsymbol{r}\,'|}\,\mathrm{d}\boldsymbol{r}'\right]\mathrm{d}\boldsymbol{r}\,, \tag{12}$$

and components of vector $\boldsymbol{b}$ are

$$b_i = \epsilon_\mathrm{m}\int_{\square_i^h}\nabla\ln(\epsilon(\boldsymbol{r})/\epsilon_\mathrm{m})\cdot\nabla\Phi_\mathrm{c}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\,. \tag{13}$$

To derive fully discrete quadrature formulas for the elements of $\boldsymbol{K}$ and $\boldsymbol{b}$, the approximation of $\nabla\ln(\epsilon(\boldsymbol{r})/\epsilon_\mathrm{m})$ is necessary. We use piecewise tri-linear nodal basis functions to approximate $\nabla\ln(\epsilon(\boldsymbol{r})/\epsilon_\mathrm{m})$ as follows

$$\ln(\epsilon(\boldsymbol{r})/\epsilon_\mathrm{m}) \cong \sum_{\boldsymbol{k}=-1,1}\ln(\epsilon(\boldsymbol{r}_i + h\boldsymbol{k}/2)/\epsilon_\mathrm{m})\,\psi_{\boldsymbol{k}}((\boldsymbol{r}-\boldsymbol{r}_i)/h)\,, \tag{14}$$

in which $\psi_{\boldsymbol{k}}(\boldsymbol{r}) = (k_x x + 1/2)(k_y y + 1/2)(k_z z + 1/2)$, and $\boldsymbol{k}$ is a vector index taking values ±1 for each of its component k_x, k_y, and k_z independently. Substituting (14) into the integral in (12), we obtain

$$K_{ij} = \sum_{\boldsymbol{k}=-1,1}\ln(\epsilon(\boldsymbol{r}_i + \frac{h}{2}\boldsymbol{k})/\epsilon_\mathrm{m})\,K_{ij\,\boldsymbol{k}}\,, \tag{15}$$

in which

$$K_{ij\,\boldsymbol{k}} = \int_{\square_i^h}\nabla\psi_{\boldsymbol{k}}\Big(\frac{\boldsymbol{r}-\boldsymbol{r}_i}{h}\Big)\cdot\nabla\left[\int_{\square_j^h}\frac{1}{4\pi|\boldsymbol{r}-\boldsymbol{r}\,'|}\,\mathrm{d}\boldsymbol{r}'\right]\mathrm{d}\boldsymbol{r}\,. \tag{16}$$

Changing variables $(\boldsymbol{r}-\boldsymbol{r}_i)/h \to \boldsymbol{r}$ and $(\boldsymbol{r}'-\boldsymbol{r}'_j)/h \to \boldsymbol{r}'$ in the integral above results in the final form,

$$K_{ij\,\boldsymbol{k}} = \int_{\square_0^1}\nabla\psi_{\boldsymbol{k}}(\boldsymbol{r})\cdot\nabla\left[\int_{\square_0^1}\frac{1}{4\pi|\boldsymbol{r}-\boldsymbol{r}'+\boldsymbol{i}-\boldsymbol{j}|}\,\mathrm{d}\boldsymbol{r}'\right]\mathrm{d}\boldsymbol{r}\,. \tag{17}$$

Similarly, the integral in (13) becomes

$$b_i = \epsilon_\mathrm{m}\sum_{\boldsymbol{k}=-1,1}\ln(\epsilon(\boldsymbol{r}_i + \frac{h}{2}\boldsymbol{k})/\epsilon_\mathrm{m})\int_{\square_i^h}\nabla\psi_{\boldsymbol{k}}\Big(\frac{\boldsymbol{r}-\boldsymbol{r}_i}{h}\Big)\cdot\nabla\Phi_\mathrm{c}(\boldsymbol{r})\,\mathrm{d}\boldsymbol{r}\,.$$

Note that elements $K_{ij\,\boldsymbol{k}}$ are problem independent, and therefore can be precomputed in advance. The method we used to precompute $K_{ij\,\boldsymbol{k}}$ lies beyond the scope of this paper. Nevertheless,

we highlight the main ideas on how to evaluate these terms: The integral in (15) is singular if $|\boldsymbol{i}-\boldsymbol{j}|_\infty \leq 1$, and special treatment is necessary. The singularity is removed by changing variables $x \to x^\alpha$, $y \to y^\beta$, and $z \to z^\gamma$, in which α, β, and γ are sufficiently large integers. After the singularity is removed, a standard adaptive quadrature is applied to evaluate the integrals. We precomputed $K_{\boldsymbol{ij\,k}}$ with relative error less than 5×10^{-6} for the numerical experiments discussed in Section 6.

4. Iterative Solver

Solving (11) using direct methods, such as Gaussian elimination, would require order N^3 arithmetic operations, since the matrix is dense. To reduce the computational cost, we will use an iterative method, and use a fast summation method to reduce the cost of each matrix-vector product (discussed in section 5).

Since matrix $\boldsymbol{A}$ is not symmetric, and, generally, not positive definite, we apply BiCG-Stab iterative method to solve (11). At each iteration of the algorithm a matrix-vector product involving $\boldsymbol{A}$ is computed twice. Provided the condition number of matrix $\boldsymbol{A}$ is bounded, the number of BiCG-Stab iterations required to achieve a given level of accuracy is bounded. Therefore, for a well-conditioned system, the cost of applying the iterative linear solver is proportional to the cost of the matrix-vector product. In the next section, we show how the fast Multilevel Summation Method (MSM) can be used to reduce the complexity of the matrix-vector product to $O(N)$, and, hence, to reduce the overall complexity of the algorithm to the same order.

5. Fast matrix-vector product computations using Multilevel Summation Method

The fast Multilevel Summation Method was originally proposed to evaluate the electrostatic potential and force fields of a set of point charges distributed arbitrarily in $\mathbb{R}^3$ [33, 34]. For a specified tolerance level, the method computes electrostatic potential and forces in linear time with respect to the number of charges. This is achieved by splitting the Green's function into a series of smooth functions and interpolating on a set of nested grids. Note that the integral kernel in (15) contains the electrostatic Green's function, and it exhibits similar behavior. Hence, we adopt the ideas of MSM to approximate the matrix-vector product for a dense matrix $\boldsymbol{K}$. We show that (for a fixed tolerance) the matrix-vector product can be computed in linear time with respect to the vector length, i.e. it is of order N. Moreover, the accuracy of the approximation can be adjusted with a single tuning parameter.

The approximation of $\boldsymbol{K}$ is done in two steps. The first step is to split the integral kernel $1/|\boldsymbol{r}-\boldsymbol{r}'|$ in (17) into rapidly varying and slowly varying components. Let $G_0 = 1/r$ for $r \in \mathbb{R}$, and consider a splitting $G_0 = (G_0 - G_1) + G_1$, such that G_1 is a smooth function of argument r. Assume that values of G_1 are equal to those of G_0 for $r > a$; see Figure 2. Parameter a is called the *cutoff radius*. If we define $G_{0,1} = G_0 - G_1$, the functions $G_{0,1}$ and G_1 correspond to the rapidly and slowly varying components of G_0. Using this splitting for G_0, the corresponding integral kernel splitting is as follows

$$\frac{1}{|\boldsymbol{r}-\boldsymbol{r}'+\boldsymbol{i}-\boldsymbol{j}|} = G_0(|\boldsymbol{r}-\boldsymbol{r}'+\boldsymbol{i}-\boldsymbol{j}|) = G_{0,1}(|\boldsymbol{r}-\boldsymbol{r}'+\boldsymbol{i}-\boldsymbol{j}|) + G_1(|\boldsymbol{r}-\boldsymbol{r}'+\boldsymbol{i}-\boldsymbol{j}|)\,. \tag{18}$$

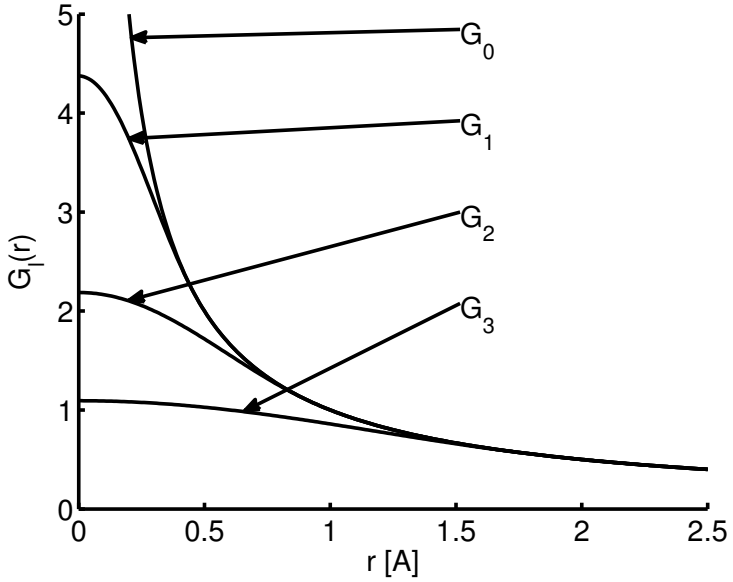


Figure 2: Radial components of the original Green's function $G_0(r)$ and smoothed Green's functions $G_1(r)$, $G_2(r)$ and $G_3(r)$ for cutoff radius $a = 0.5$.

Applying the integral kernel splitting to (17) induces a splitting of $\boldsymbol{K}$. Moreover, the matrix splitting inherits the properties of the kernel splitting, i.e. $\boldsymbol{K}$ splits into rapidly,

$$(K_{0,1})_{\boldsymbol{ij}\,\boldsymbol{k}} = \int_{\square_0^1} \nabla\psi_{\boldsymbol{k}}(\boldsymbol{r}) \cdot \nabla \left[\int_{\square_0^1} \frac{1}{4\pi} G_{0,1}(|\boldsymbol{r} - \boldsymbol{r}' + \boldsymbol{i} - \boldsymbol{j}|)\, \mathrm{d}\boldsymbol{r}' \right] \mathrm{d}\boldsymbol{r} \;, \tag{19}$$

and slowly varying components,

$$(K_1)_{\boldsymbol{ij}\,\boldsymbol{k}} = \int_{\square_0^1} \nabla\psi_{\boldsymbol{k}}(\boldsymbol{r}) \cdot \nabla \left[\int_{\square_0^1} \frac{1}{4\pi} G_1(|\boldsymbol{r} + \boldsymbol{i} - \boldsymbol{r}' - \boldsymbol{j}|)\, \mathrm{d}\boldsymbol{r}' \right] \mathrm{d}\boldsymbol{r} \;. \tag{20}$$

Since $G_{0,1}$ vanishes for all $r \geq a$, component $(\boldsymbol{K}_{0,1})_{\boldsymbol{ij}} = 0$ if $|\boldsymbol{i} - \boldsymbol{j}|_2 > a + 1$, in which $|\cdot|_2$ is the Euclidean distance in $\mathbb{R}^3$. Hence, the matrix $\boldsymbol{K}_{0,1}$ is sparse with no more than $8(a+1)^3$ nonzero entries per row. On the other hand, the matrix $\boldsymbol{K}_1$ is dense, because G_1 is nontrivial for all argument values.

The second step is to interpolate G_1 on a coarser grid, using nodal basis functions $\xi_{\boldsymbol{i}'}$ and $\xi_{\boldsymbol{j}'}$,

$$G_1(|\boldsymbol{r} + \boldsymbol{i} - \boldsymbol{r}' - \boldsymbol{j}|) \approx \sum_{\boldsymbol{i}',\boldsymbol{j}'} G_1(|\boldsymbol{i}' - \boldsymbol{j}'|)\xi_{\boldsymbol{i}'}(\boldsymbol{r} + \boldsymbol{i})\xi_{\boldsymbol{j}'}(\boldsymbol{r}' + \boldsymbol{j}) \,, \tag{21}$$

in which $\boldsymbol{i}'$ and $\boldsymbol{j}'$ are vector indices of coarse grid points. Substitute (21) into (20) to obtain

$$(K_1)_{\boldsymbol{ij}\,\boldsymbol{k}} \approx \sum_{\boldsymbol{i}'\,\boldsymbol{j}'} \frac{1}{4\pi} G_1(|\boldsymbol{i}' - \boldsymbol{j}'|) \int_{\square_0^1} \nabla\psi_{\boldsymbol{k}}(\boldsymbol{r} - \boldsymbol{i}) \cdot \nabla\xi_{\boldsymbol{i}'}(\boldsymbol{r} + \boldsymbol{i})\, \mathrm{d}\boldsymbol{r} \int_{\square_0^1} \xi_{\boldsymbol{j}'}(\boldsymbol{r}' + \boldsymbol{j})\, \mathrm{d}\boldsymbol{r}' \;. \tag{22}$$

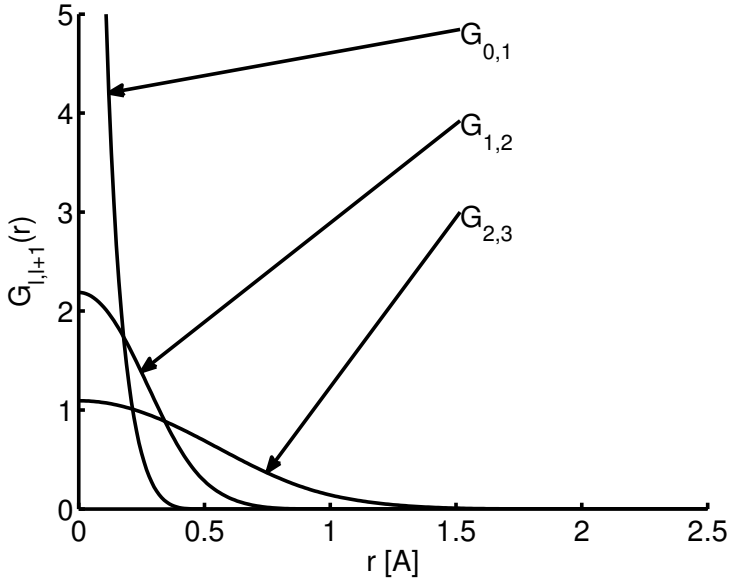


Figure 3: Radial components of Green's functions $G_{0,1}$, $G_{1,2}$, and $G_{2,3}$ for cutoff radius $a = 0.5$.

Hence, the matrix $\boldsymbol{K}_1$ is approximated with the product $\boldsymbol{P}_1\boldsymbol{K}'_1\boldsymbol{R}_1$, in which

$$(P_1)_{\boldsymbol{i}\boldsymbol{i}'} = \sum_{\boldsymbol{k}=-1,1} \ln(\epsilon(\boldsymbol{r}_i + \frac{h}{2}\boldsymbol{k})/\epsilon_{\mathrm{m}}) \int_{\square_0^1} \nabla\psi_{\boldsymbol{k}}(\boldsymbol{r}-\boldsymbol{i}) \cdot \nabla\xi_{\boldsymbol{i}'}(\boldsymbol{r}+\boldsymbol{i})\,\mathrm{d}\boldsymbol{r}\ , \tag{23}$$

$$(K'_1)_{\boldsymbol{i}'\boldsymbol{j}'} = \frac{1}{4\pi} G_1(|\boldsymbol{i}'-\boldsymbol{j}'|)\,, \tag{24}$$

$$(R_1)_{\boldsymbol{j}'\boldsymbol{j}} = \int_{\square_0^1} \xi_{\boldsymbol{j}'}(\boldsymbol{r}+\boldsymbol{j})\,\mathrm{d}\boldsymbol{r}\ . \tag{25}$$

If $|\boldsymbol{i}-\boldsymbol{i}'|_2$ is greater than the ξ support set diameter, then the integral in (23) vanishes. Similarly, if $|\boldsymbol{j}-\boldsymbol{j}'|_2$ is greater than the diameter of the support set, then the integral in (23) vanishes. Hence, matrices $\boldsymbol{P}_1$ and $\boldsymbol{R}_1$ are sparse if the support of ξ is compact. The matrix $\boldsymbol{K}'$, on the other hand, remains dense due to nonlocality of G_1. However, the dimensionality of $\boldsymbol{K}'_1$ is reduced compared with that of $\boldsymbol{K}_1$, because indices $\boldsymbol{i}'$ and $\boldsymbol{j}'$ correspond to a coarser grid.

Repeating the two steps outlined above, and we can build the two level approximation of $\boldsymbol{K}$,

$$\boldsymbol{K} = \boldsymbol{K}_{0,1} + \boldsymbol{P}_1\Big(\boldsymbol{K}_{1,2} + \boldsymbol{P}_2\boldsymbol{K}_2\boldsymbol{R}_2\Big)\boldsymbol{R}_1\,, \tag{26}$$

in which $(\boldsymbol{K}_2)_{\boldsymbol{i}''\boldsymbol{j}''} = G_2(|\boldsymbol{i}''-\boldsymbol{j}''|)/4\pi$ and the smooth function G_2 is equal to G_1 for $r > 2a$ (see Figure 2); $(\boldsymbol{K}_{1,2})_{\boldsymbol{i}'\boldsymbol{j}'} = G_{1,2}(|\boldsymbol{i}'-\boldsymbol{j}'|)/4\pi$ and $G_{1,2} = G_1 - G_2$ (see Figure 3); $(P_2)_{\boldsymbol{i}'\boldsymbol{i}''} = \xi_{\boldsymbol{i}''}(\boldsymbol{i}')$ and $(R_2)_{\boldsymbol{j}''\boldsymbol{j}'} = \xi_{\boldsymbol{j}''}(\boldsymbol{j}')$; and vector indices $\boldsymbol{i}''$ and $\boldsymbol{j}''$ correspond to the second level coarse grid. Applying these two steps recursively, we derive a multilevel approximation for matrix $\boldsymbol{K}$ in the form

$$\boldsymbol{K} \approx \boldsymbol{K}_{0,1} + \boldsymbol{P}_1\Big(\boldsymbol{K}_{1,2} + \boldsymbol{P}_2(\boldsymbol{K}_{2,3} + \ldots \boldsymbol{P}_M(\boldsymbol{K}_M)\boldsymbol{R}_M \ldots)\boldsymbol{R}_2\Big)\boldsymbol{R}_1\,, \tag{27}$$

in which M is the level number, the matrix $\boldsymbol{K}_M$ is of a fixed but small dimension, matrices $\boldsymbol{K}_{l,l+1}$, $\boldsymbol{P}_l$, and $\boldsymbol{R}_l$ are sparse matrices defined on lth coarse grid, and the index l ($1 \leq l \leq M$) indicates the corresponding level.

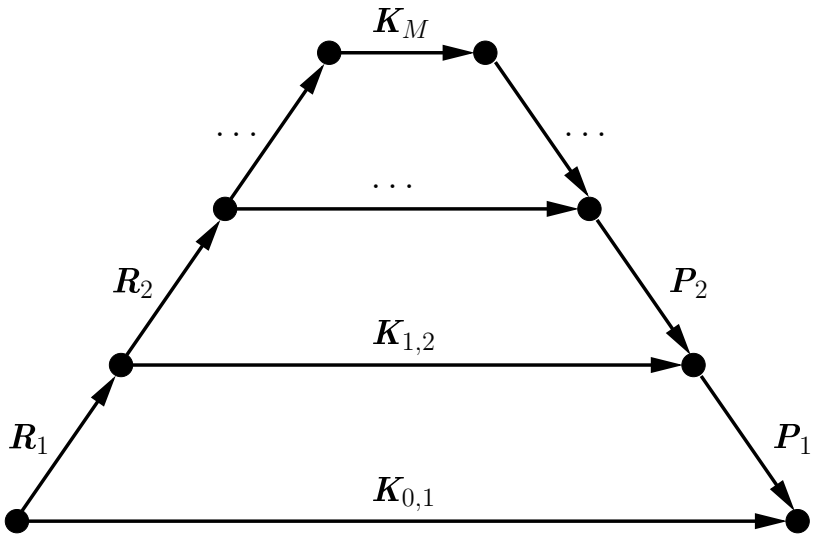


Figure 4: Multilevel summation method diagram.

Figure 4 illustrates the computation of the approximate matrix-vector product using MSM. The computation of a matrix-vector product is carried out as follows: 1) for any given input vector, successively compute the matrix-vector product of the restriction matrices $\boldsymbol{R}_l$ in ascending order of l and store the resulting vectors; 2) compute matrix-vector products for matrices $\boldsymbol{K}_{l-1,l}$ and $\boldsymbol{K}_M$ with corresponding restricted vectors; 3) in descending order of l compute matrix-vector products for the prolongation matrices $\boldsymbol{P}_l$ and add the prolongated vector with the current level vector at each step.

The cost of computing MSM approximation is the cumulative cost of all matrix-vector products in the diagram outlined above. First, we consider the vector size at each level. For 3D grids it is typical for the coarse grid to have $1/8$ of the previous level's grid points. Hence, if the size of the finest grid (zeroth level) is N, then the size of the grid at the first level is $N/8$, at the second level $N/64$, and so on. Second, we consider the cost of computing the restricted and prolongated vectors. As we have shown before, matrices $\boldsymbol{R}_l$ and $\boldsymbol{P}_l$ are sparse with at most D^3 nonzeroes per row, in which D is the diameter of the support set of ξ. Hence, the cost of computing the matrix-vector product for either $\boldsymbol{R}_l$ or $\boldsymbol{P}_l$ is at most $D^3N/8^l$ operations. Third, we consider the cost of computing the matrix-vector product for all $\boldsymbol{K}_{l,l+1}$. The number of nonzeroes per row for each $\boldsymbol{K}_{l,l+1}$ does not exceed $8(a+1)^3$, and that the number of grid points at level M does not exceed $8(a+1)^3$. Hence, computing the matrix-vector product for $\boldsymbol{K}_{l,l+1}$ takes at most $8(a+1)^3N/8^l$ operations for all levels. Finally, adding all costs we derive the total cost of MSM,

$$2\sum_{l=1}^{M}\frac{D^3N}{8^l}+\sum_{l=0}^{M}\frac{8(a+1)^3N}{8^l}\leq\frac{1}{7}(2D^3+16(a+1)^3)N\,. \tag{28}$$

Hence, the computational complexity of MSM is of order N.

Since interpolation of the kernel is introduced at each level, MSM is only an approximation of the matrix-vector product involving $\boldsymbol{K}$. To analyze the approximation error, we calculate the interpolation error at each level. Assuming the worst case scenario, the approximation error is the accumulation of the interpolation errors. Using pth degree polynomial basis functions ξ, the interpolation error is proportional to the pth derivative of the smoothed kernel G_l multiplied by the level l grid step size. Assume G_l is such that its pth derivative exists, and it is bounded by $p!(2^l a)^{-p-1}$ (see [34] for more details). The lth level grid step size is given by 2^l. Therefore, the kernel interpolation error at level l is proportional to $2^{-pl}a^{-(p+1)}$, and the total error of MSM is proportional to

$$\sum_{l=1}^{M} 2^{-pl}a^{-(p+1)} \leq \frac{1}{1-2^{-p}}\frac{1}{a^{p+1}} \leq \frac{1}{a^{p+1}}\,. \tag{29}$$

Note that the error is reduced if a is increased. Therefore, the cutoff radius plays the role of the error tuning parameter for MSM.

6. Results and Discussion

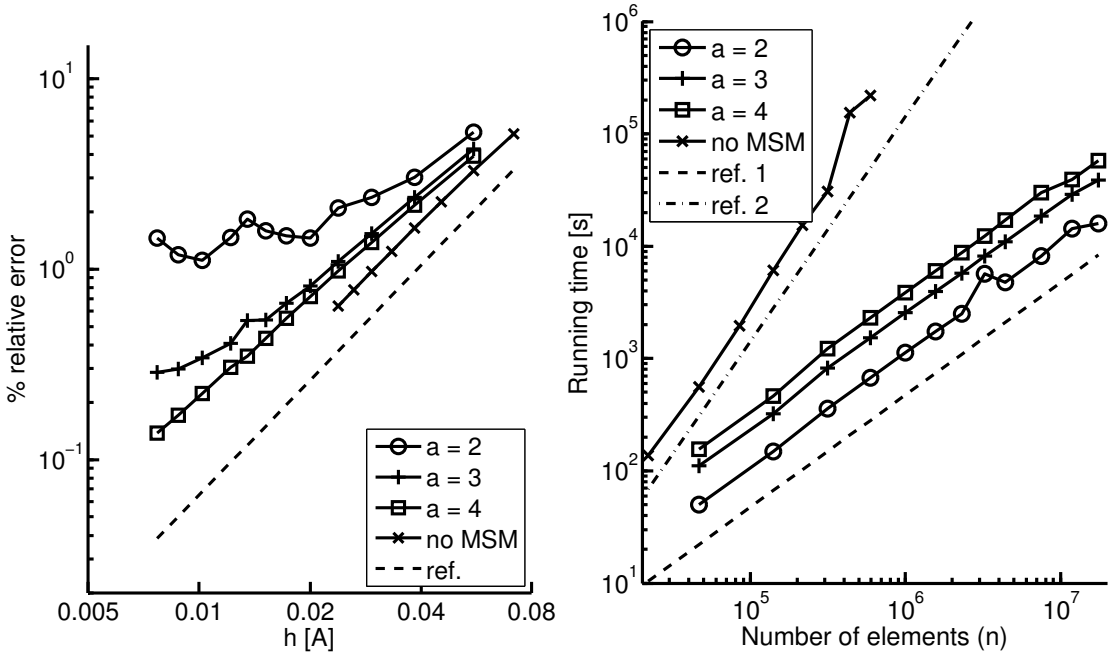
In this section, we analyze the convergence rate and computational complexity of the proposed algorithm through a series of numerical experiments. The first system we consider is the Born ion, which consists of a single point charge located at the origin of an infinite dielectric medium with spherically symmetric permittivity coefficient, i.e. $\epsilon(\boldsymbol{r}) = \epsilon(|\boldsymbol{r}|)$. Due to the spherical symmetry, both the effective charge distribution and the electric potential are spherically symmetric as well. Moreover, the Born ion problem admits an analytic solution for the effective charge distribution,

$$\rho_{\text{Born}}(\boldsymbol{r}) = -q\epsilon_{\text{m}}\frac{|\nabla\epsilon(\boldsymbol{r})|}{|\boldsymbol{r}|^2\epsilon^2(\boldsymbol{r})}\,. \tag{30}$$

To investigate the convergence of the proposed algorithm without using MSM, we use the Born ion problem with the following parameters: the dielectric permittivity $\epsilon(r) = \epsilon_{\text{m}} = 1$ for $r < 0.5$Å and $\epsilon(r) = \epsilon_{\text{s}} = 20$ for $r > 1$Å; the discretization mesh is uniform with the step size varying between 0.008Å and 0.07Å. Figure 5a shows the relative error in the 2-norm of the effective charge distribution related to the discretization mesh step size; see "no MSM" line. Additionally, a dashed reference line with slope 2 is shown. Note that convergence of the proposed algorithm is of second order for the piecewise constant basis functions. The convergence result is in agreement with the theoretical developments presented in Section 3.

Figure 5b depicts the relation between the running time of BiCG-Stab linear solver and the number of discretization points used; see "no MSM" line. A dot-dashed line with slope 2 is shown in Figure 5b for the reference. As we can see, computing the matrix-vector product directly takes order N^2 seconds, as expected.

Next, we study the error and running time of the proposed method for the Born ion problem when MSM is applied. We use a 3rd degree polynomial to interpolate smoothed kernels on coarse grids. From Figure 5b we observe that the running time for MSM is proportional to the number of grid cells in the discretization (unit slope dashed reference line is shown). The proportionality constant depends on the cutoff radius a, however. The observed behavior is in agreement with the theoretical findings of Section 5.



(a) Error (b) Cost

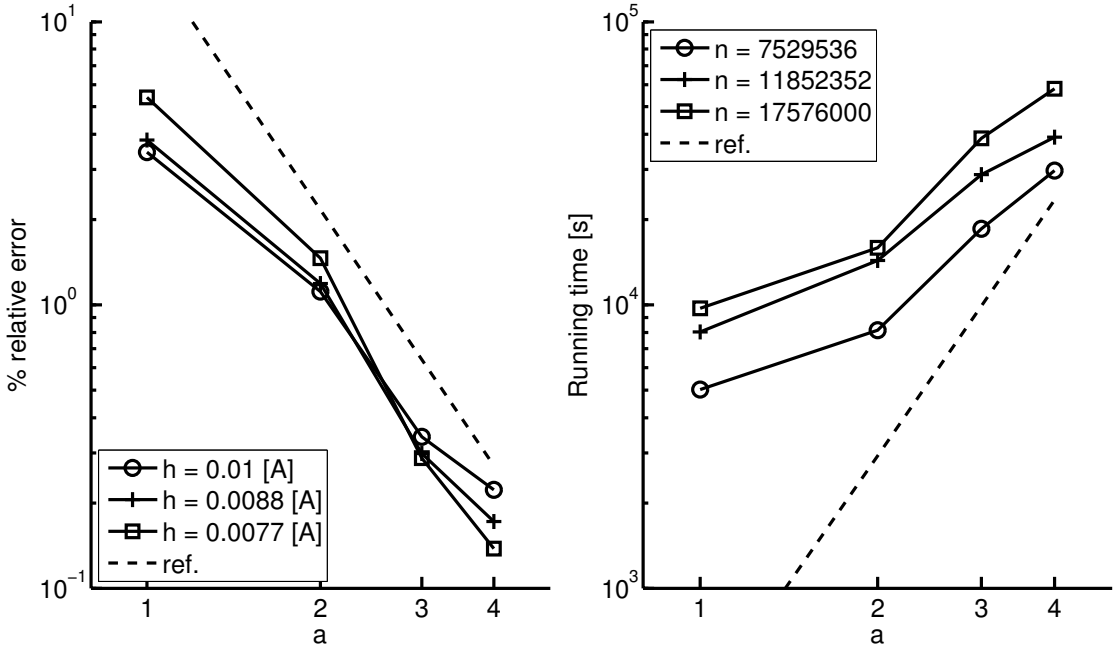
Figure 5: MSM with 3rd degree interpolation

The 2-norm error of the numerical solution with respect to the mesh step size h is plotted on Figure 5a, along with a dashed reference line of slope 2. Observe that the convergence rate is of second order if measured on coarse meshes. However, stagnation occurs on fine meshes. This is due to MSM approximation error, which is not affected by mesh refinement, but it can be reduced by increasing the cutoff radius. Thus, as the cutoff radius is increased, stagnation occurs at finer meshes, and it is virtually unobservable for $a = 4$ in our experiments; see Figure 5a.

Finally, we analyze the effect of varying the cutoff radius on the error and running time of the algorithm. Figure 6a demonstrates the relation between the relative error of the approximate solution and the cutoff radius. Note that for fine meshes ($h \leq 0.027$Å) the error decreases at the rate a^{-3}. Finally, Figure 6b shows that the computation time increase roughly at the rate a^3 as the cutoff radius increases. These results are in a good agreement with the theory as well.

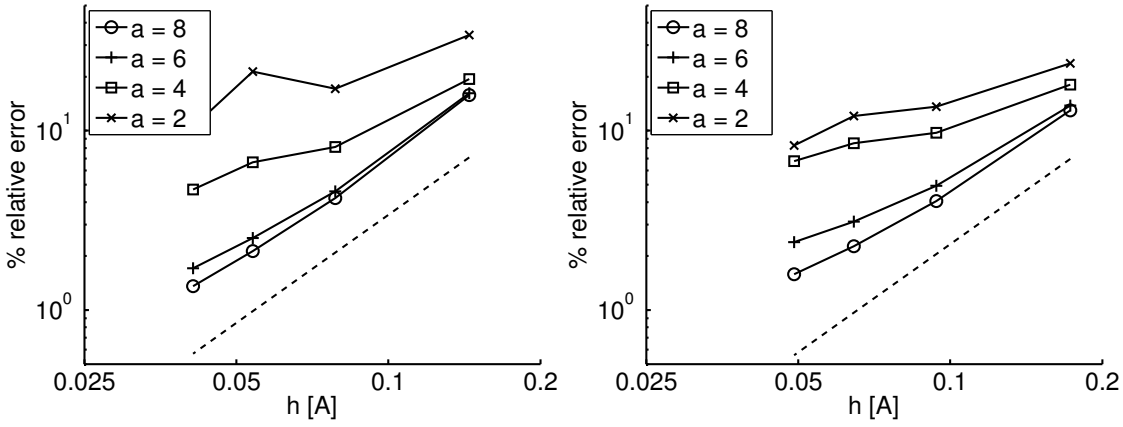
To illustrate the applicability of the proposed method for real systems, we perform computations of the polar solvation free energy, which is calculated as $\frac{1}{2}\sum_p u(\boldsymbol{r}_{\mathrm{c},p})q_{\mathrm{c},p}$, for small molecules (methanol and acetate). The solver is used to calculate the reaction field $u(\boldsymbol{r})$ at charge location $\boldsymbol{r}_{\mathrm{c},i}$. The dielectric permittivity is taken to be $\epsilon_{\mathrm{m}} = 1$ inside the molecule region and $\epsilon_{\mathrm{s}} = 80$ in the solvent region. We assume a linear relation between the dielectric permittivity $\epsilon(\boldsymbol{r})$ and the solvent concentration. The solvent concentration is approximated as a product of spherical bump functions centered at the explicit charge locations [20], with inner and outer radii equal to 100% and 120% of the charge radius, correspondingly.

To estimate the true value of the solvation free energy we use the least squares method. We assume the solution error to be proportional to the discretization error (h^2) plus the error of MSM



(a) Error (b) Cost

Figure 6: MSM with 5th degree interpolation



(a) Methanol (b) Acetate

Figure 7: Relative error in solvation free energy

(a^{-3}). Moreover, we assume that these two errors are independent. With just two computations for different grid step sizes, we are able to reconstruct the free energy value to a greater accuracy using this method. The resulting solvation free energy is -10.04 *kcal/mol* for methanol and -91.622 *kcal/mol* for acetate.

In Figure 7a and Figure 7b the relative errors for different step sizes and cutoff radii are depicted. A dashed reference line of slope 2 is shown. From the figures, we observe that the estimated error in the free energy is of the same order as the computed relative error in the effective charge distribution for a given step size h. Moreover, we conclude that a step size of ≈ 0.5Å yields relative error of less than 2%. This result can be used to compare our method with existing numerical models.

7. Conclusion

In conclusion, we investigated a numerical procedure for the Generalized Poisson Equation with continuously varying dielectric permittivity of the medium. Under the assumption that the permittivity varies in a bounded region Ω_b, we reformulated the GPE as a Fredholm integral equation of the second kind for the effective charge distribution. We showed, that the superposition of uniform medium electric potential fields of the effective charge distribution and implicit molecule charges gives the original GPE solution. Hence, by solving the integral equation, the solution to GPE is recovered via Coulomb's law.

The projection method, when applied to the integral equation, results in a system of linear equations. The solution to this system is a vector of coefficients for the approximate charge distribution in the basis of the selected finite element subspace. Since the integral kernel has infinite support, the matrix of the system is dense. Hence, solving the system with a direct solver takes order N^3 operations. Application of an iterative solver, however, reduces the cost to order N^2 if the matrix is well conditioned. Further reduction in cost using the fast Multilevel Summation Methods was investigated.

The MSM, originally developed as a fast N-body solver, is modified to compute the matrix-vector product of the discrete integral transform in linear time. The modification relies on the fact that the integral kernel resembles the electrostatic Green's function. The kernel is successively split into rapidly and slowly varying components, with further interpolation of the slowly varying component on a coarser grid. This way, it is possible to approximate the matrix-vector product with a sequence of sparse matrix-vector products with ever decreasing dimensionality, and to control the error of this approximation with the splitting cutoff radius. For a fixed accuracy, the MSM method computes the approximate matrix-vector product in linear time.

Finally, the Born ion problem and the solvation free energy of two small molecules were studied numerically. The experiments are in complete agreement with theoretical findings. Hence we conclude the following about the proposed method:

- For a desired error tolerance, the method achieves optimal complexity of order N (here N is the finite element space dimension);
- The numerical discretization error is controlled by the mesh step size (h) and is proportional to h^2 for piecewise constant basis functions;

- MSM error is controlled by the cutoff radius (a) and is proportional to a^{-p};
- The algorithm can be extended to larger systems to compute physical properties of various macromolecules.

References

[1] D. Levitt, Electrostatic calculations for an ion channel. I. energy and potential profiles and interactions between ions, Biophys. J. 22 (2) (1978) 209–219.

[2] M. Davis, J. McCammon, Calculating electrostatic forces from grid-calculated potentials, J. Comput. Chem. 11 (3) (1990) 401–409.

[3] B. Honig, A. Nicholls, Classical electrostatics in biology and chemistry, Science 268 (5214) (1995) 1144–1149.

[4] B. Roux, T. Simonson, Implicit solvent models, Biophys. Chem. 78 (1-2) (1999) 1–20.

[5] W. Im, D. Beglov, B. Roux, Continuum solvation model: computation of electrostatic forces from numerical solutions to the Poisson-Boltzmann equation, Comput. Phys. Commun. 111 (1998) 59–75.

[6] T. Schlick, Molecular Modeling and Simulation: An Interdisciplinary Guide, Springer-Verlag, New York, 2002.

[7] H.-X. Zhou, Boundary element solution of macromolecular electrostatics: interaction energy between two proteins, Biophys. J. 65 (1993) 955–963.

[8] D. Frenkel, B. Smit, Understanding Molecular Simulation, 2nd Edition, Academic Press, New York, 2002.

[9] W. C. Still, A. Tempczyk, R. C. Hawley, T. Hendrickson, Semianalytical treatment of solvation for molecular mechanics and dynamics, J. Am. Chem. Soc. 112 (16) (1990) 6127–6129.

[10] M. Feig, C. L. Brooks III, Recent advances in the development and application of implicit solvent models in biomolecule simulations, Curr. Opin. Struct. Biol. 14 (2) (2004) 217–224.

[11] M. Feig, A. Onufriev, M. S. Lee, W. Im, D. A. Case, C. L. Brooks III, Performance comparison of generalized Born and Poisson methods in the calculation of electrostatic solvation energies for protein structures, J. Comput. Chem. 25 (2) (2004) 265–284.

[12] N. A. Baker, D. Bashford, D. A. Case, Implicit solvent electrostatics in biomolecular simulation, in: B. Leimkuhler, C. Chipot, R. Elber, A. Laaksonen, A. Mark, T. Schlick, C. Schütte, R. Skeel (Eds.), New Algorithms for Macromolecular Simulation, Vol. 49 of Lecture Notes in Computational Science and Engineering, Springer-Verlag, 2005, pp. 263–295.

[13] Y. Sun, R. A. Latour, Comparison of implicit solvent models for the simulation of protein-surface interactions, J. Comput. Chem. 27 (16) (2006) 1908–1922.

[14] P. Koehl, Electrostatics calculations: Latest methodological advances, Curr. Opin. Struct. Biol. 16 (2006) 142–151.

[15] J. Warwicker, H. C. Watson, Calculation of the electric potential in the active site cleft due to alpha-helix dipoles, J. Molec. Biol. 157 (4) (1982) 671–679.

[16] M. K. Gilson, K. A. Sharp, B. H. Honig, Calculating the electrostatic potential of molecules in solution: Method and error assessment, J. Comput. Chem. 9 (1987) 327–335.

[17] M. E. Davis, J. A. McCammon, Solving the finite difference linearized Poisson-Boltzmann equation: A comparison of relaxation and conjugate gradient methods, J. Comput. Chem. 10 (1989) 386–391.

[18] M. J. Holst, F. Saied, Numerical solution of the nonlinear Poisson-Boltzmann equation: Developing more robust and efficient methods, J. Comput. Chem. 16 (1995) 337–364.

[19] N. A. Baker, D. Sept, S. Joseph, M. J. Holst, J. A. McCammon, Electrostatics of nanosystems: Application to microtubules and the ribosome, Proc. Nat. Acad. Sci. 98 (2001) 10037–10041.

[20] A. J. Grant, B. T. Pickup, A. Nicholls, A smooth permittivity function for Poisson-Boltzmann solvation methods, J. Comput. Chem. 22 (6) (2001) 608–640.

[21] R. Zauhar, R. Morgan, A new method for computing the macromolecular electric potential, J. Molec. Biol. 186 (4) (1985) 815–820.

[22] R. J. Zahaur, R. S. Morgan, The rigorous computation of the molecular electric-potential, J. Comput. Chem. 9 (1988) 171–187.

[23] Y. N. Vorobjev, J. A. Grant, H. A. Scheraga, A combined iterative and boundary-element approach for solution of the nonlinear Poisson-Boltzmann equation, J. Am. Chem. Soc. 114 (9) (1992) 3189–3196.

[24] R. Bharadwaj, A. Windemuth, S. Sridharan, B. Honig, A. Nicholls, The fast multipole boundary element method for molecular electrostatics: An optimal approach for large systems, J. Comput. Chem. 16 (7) (1995) 898–913.

[25] Y. N. Vorobjev, H. A. Scheraga, A fast adaptive multigrid boundary element method for macromolecular electrostatic computations in a solvent, J. Comput. Chem. 18 (4) (1997) 569–583.

[26] S. S. Kuo, M. D. Altman, J. P. Bardhan, B. Tidor, J. K. White, Fast methods for simulation of biomolecule electrostatics, in: ICCAD '02: Proceedings of the 2002 IEEE/ACM international conference on Computer-aided design, ACM Press, New York, NY, USA, 2002, pp. 466–473.

[27] B. Lu, X. Cheng, J. Huang, J. A. McCammon, Order n algorithm for computation of electrostatic interactions in biomolecular systems, Proc. Natl. Acad. Sci USA 103 (51) (2006) 19314–19319.

[28] W. H. Orttung, Direct solution of the Poisson equation for biomolecules of arbitrary shape, polarizability density, and charge distribution, Ann. N.Y. Acad. Sci. 303 (1977) 22–37.

[29] C. M. Cortis, R. A. Friesner, Numerical solution of the Poisson-Boltzmann equation using tetrahedral finite-element meshes, J. Comput. Chem. 18 (2000) 1591–1608.

[30] M. J. Holst, N. A. Baker, F. Wang, Adaptive multilevel finite element solution of the Poisson-Boltzmann equation I: Algorithms and examples, J. Comput. Chem. 21 (2000) 1319–1342.

[31] N. A. Baker, M. J. Holst, F. Wang, Adaptive multilevel finite element solution of the Poisson-Boltzmann equation II: refinement at solvent accessible surfaces in biomolecular systems, J. Comput. Chem. 21 (2000) 1343–1352.

[32] A. Vitalis, N. A. Baker, J. A. McCammon, ISIM: a program for grand canonical Monte Carlo simulations of the ionic environment of biomolecules, Molec. Sim. 30 (2004) 45–61.

[33] R. D. Skeel, I. Tezcan, D. J. Hardy, Multiple grid methods for classical molecular dynamics, J. Comput. Chem. 23 (2002) 673–684.

[34] D. J. Hardy, Multilevel summation for the fast evaluation of forces for the simulation of biomolecules, Ph.D. thesis, Univ. of Illinois at Urbana-Champaign (2006).

[35] K. Atkinson, W. Han, Theoretical Numerical Analysis: A Functional Analysis Framework, 2nd Edition, Springer, New York, 2005.

DISTRIBUTION:

1	MS 0899	Technical Library, 9536 (electronic copy)
1	MS 0359	D. Chavez, LDRD Office, 1911

Sandia National Laboratories