Efficient Numerical Algorithms for Surface Formulations of Mathematical Models for Biomolecule Analysis and Design

by

Jaydeep Porter Bardhan

S. B., E. E., Massachusetts Institute of Technology (2000)
M. Eng., E. E. C. S., Massachusetts Institute of Technology (2001)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

June 2006

Author . . . . . .
Department of Electrical Engineering and Computer Science
May 10, 2006

Certified by . . . . . .
Jacob K. White
Cecil H. Green Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Certified by . . . . . .
Bruce Tidor
Professor of Biological Engineering and Computer Science
Thesis Supervisor

Accepted by . . . . . .
Arthur C. Smith
Chairman, Department Committee on Graduate Students

Efficient Numerical Algorithms for Surface Formulations of Mathematical Models for Biomolecule Analysis and Design

by

Jaydeep Porter Bardhan

Submitted to the Department of Electrical Engineering and Computer Science
on May 10, 2006, in partial fulfillment of the
requirements for the degree of
Doctor of Philosophy in Electrical Engineering and Computer Science

Abstract

This thesis presents a set of numerical techniques that extend and improve computational modeling approaches for biomolecule analysis and design. The presented research focuses on surface formulations of modeling problems related to the estimation of the energetic cost to transfer a biomolecule from the gas phase to aqueous solution. The thesis discusses four contributions to modeling biomolecular interactions. First, the thesis presents an approach to allow accurate discretization of the most prevalent mathematical definitions of the biomolecule–solvent interface; also presented are a number of accurate techniques for numerically integrating possibly singular functions over the discretized surfaces. Such techniques are essential for solving surface formulations numerically. The second part of the thesis presents a fast multiscale numerical algorithm, FFTSVD, that efficiently solves large boundary-element method problems in biomolecule electrostatics. The algorithm synthesizes elements of other popular fast algorithms to achieve excellent efficiency and flexibility. The third thesis component describes an integral-equation formulation and boundary-element method implementation for biomolecule electrostatic analysis. The formulation and implementation allow the solution of complicated molecular topologies and physical models. Furthermore, by applying the methods developed in the first half of the thesis, the implementation can deliver superior accuracy for competitive performance. Finally, the thesis describes a highly efficient numerical method for calculating a biomolecular charge distribution that minimizes the free energy change of binding to another molecule. The approach, which represents a novel PDE-constrained methodology, builds on well-developed physical theory. Computational results illustrate not only the method's improved performance but also its application to realistic biomolecule problems.

Thesis Supervisor: Jacob K. White
Title: Cecil H. Green Professor of Electrical Engineering and Computer Science

Thesis Supervisor: Bruce Tidor
Title: Professor of Biological Engineering and Computer Science

Acknowledgments

It has been much easier to write the main text of this thesis than to try to properly acknowledge the many people who helped make the last ten years of my life so incredible.

First, I am very blessed to have participated in an extensive five-year collaboration with two fantastic thesis advisors, Professors Jacob White and Bruce Tidor, and with an enormously talented fellow student, Michael Altman. I can't thank Jacob and Bruce enough for their guidance, encouragement, and support, both financially and academically. Jacob took me on as a student despite our mutual ignorance of numerical algorithms in biology, and I am thrilled to have had the opportunity to learn how an applied numerical analyst enters a new field. Bruce has been incredibly patient with me as I've learned about computational methods in biology. I am also grateful to Professor Steve Senturia, who has been a wonderful mentor and friend since supervising my M. Eng. thesis. Jacob, Bruce, and Steve have been and always will be some of my greatest heroes.

I am deeply indebted to Professor Mike Gilson at the University of Maryland at College Park, who generously shared his valuable time to serve as a member of my thesis committee. I would like to also thank several other excellent teachers who have helped me along, each in their own way: Irv Siegler, Dave Sonday, Dr. Ron Loomis, Judy Cross, and Carol Case. The only commensurate expression of gratitude that I may offer my teachers is to model my career of service on theirs.

I have been privileged to collaborate with many talented and dedicated researchers. Working with Malty has been an exhilarating experience, and I will miss him very much. Shihhsien Kuo led the initial biomolecule electrostatics modeling research in our collaboration. Jung Hoon Lee and I spent many weeeks discussing possible approaches for fast electrostatic optimization, and after the development of the co-optimization method we spent many more weeks discussing variations and the method's strengths and weaknesses. Dave Willis, with whom I've shared an office since joining Jacob's group, has played an essential role in helping clarify issues regarding curved-element methods in BEM. During the summer of 2003, I thoroughly enjoyed working with Dr. Steve Benson and Dr. Sven Leyffer at the Mathematics and Computer Science (MCS) Division at Argonne National Laboratory. Kevin Chu, Bree Aldridge, and Sumita Pennathur have been terrific collaborators on a number of fun research projects.

Many past and current students and post-doctoral researchers have been subjected to a steady flow of my ill-thought-out and generally worthless ideas. Their patience and comments have been invaluable. I would like to acknowledge the following members of the Computational Prototyping Group: Prof. Luca Daniel, Xin Wang, Annie Vithayathil, Xin Hu, Ben Song, Zhenhai Zhu, John Rockway, Michal Rewienski, Joe Kanapka, Tom Klemas, Carlos Pinto-Coelho, Dmitri Vasilyev, Kin Sou, Brad Bond, Bo Kim, and Homer Reid. I would also like to thank the following members from the Tidor lab: Woody Sherman, Mark Bathe, Prof. David Green, David Huggins, Brian Joughin, Mala Radhakrishnan, Bracken King, Shaun Lippow, Aurore Zyto, Kathryn Armstrong, A. Katharina Wilkins, Bambang Adiwijaya, Caitlin Bever, Josh Apgar, Patricio Ramirez, and Jared Toettcher.

Worthy of far more credit than I can offer here are the people with whom I've lived over the past six years: Jim Kim, Burak and Oya and Baris Temelkuran, Michael Ep-

stein, Sudeb Dalai, Andrew Wallace, Andrew Hogue, Ben Roberts, Lars Blackmore, Jeff Leblanc, Andrew Martinez, Eric Cohen, Isaac Dinner, Ricardo Garcia, Brian Savery, and Richard Koche. For their patience, support, and ability to tune out my ten million awful renditions of "Don't Think Twice," I am truly grateful. In addition, I must thank the numerous friends who have let me stay with them on my travels. I am especially grateful to Steve and JaNette Lefkowitz for opening their home to me during the summer of 2003 while I worked at Argonne. I owe Ken Conley an enormous debt for being such a good friend and gracious host during my regular trips to California (and immediately upon arrival, In-N-Out Burger). Tiffany and Andrew (and now Sydney and Alex also!) Hogue have several times offered a most welcome respite from MIT. I am also profoundly grateful for the hospitality of Desiree Syn, Paul Chen, Mike Boucher, and Neil Bardhan.

If you've read this far, you are probably either looking for your name or merely trying to look like you're reading the thesis. I would like to thank many friends: Jerry Hughes, Bryan Morrissey, Alan Krasnick, Randy Graebner, Robert Tunick, Steve Lefkowitz, Ken Conley, Andrew Hogue, Eric Cohen, Jeff Leblanc, Russell Spieler, Dylan Hirsch-Shell, Mark Brigham, Chetak Reshamwala, Stephanie Lin, Sumita Pennathur, Holland Dieringer, Jaimie Knowles, and Jim Carton. I have really appreciated the perspective and advice of Luke Theogarajan and John Rockway as I have made my way through graduate school. Erik Deutsch, Raj Sood, Annie Vithayathil, and Xin Hu have been great listeners and even better friends. The staff at Bukowski's are owed a special shout out for their patience and good humor. Some very specific thanks are also due: to Erik Deutsch, for offering a safe, quiet place to live and work as I prepared my thesis and defense; to Belle Wang, for the *shakabuku*; to Neil Basu, for the gum; to Dr. Sheila Combs, for listening; to Speed Levitch, for being himself; and to Kyla Mitsunaga, for being so wonderfully supportive. Words are totally insufficient to adequately thank Andrew Martinez; happily, they are also unnecessary. Through e-mail, phone, and too-infrequent real-life gatherings, my spirits have been buoyed unflaggingly by the ECS, which has grown from a group comprised of close childhood friends—Sam and Beth Spillman, Mike Moore, Todd and Jason Brown, Jeremy Cortese, and my brother Neil—to include Jessica Sullivan-Brown, Andi Keller-Moore, Aidan Moore, and Dr. Stephanie Cortese.

Finally, through all of this, my family have been my most stalwart supporters. My Boro Pishimoshoi and Pishimoni, in particular, have encouraged me for as long as I can remember to pursue this dream, as have my extended family from grandparents on down. My parents have played the most important role in all of this. Since dropping me off at the Student Center in August 1996, they have listened patiently to all my triumphs, frustrations, and failures, small and large. Their advice was always helpful, whether offering a parent's perspective or relating stories from graduate school. Discussions with dad were illuminating and I always welcomed his analysis of difficult situations. Mom's home-baked goodies always brightened my week, reminding me that yes, somewhere out there beyond these walls, someone was thinking of me. Without their support, I would have turned away from this path long ago. This thesis is dedicated to them.

Contents

[1] To be submitted [38] with Appendices A, B, and C.

[2]This chapter was published previously in a special issue of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems [83] with J. P. Bardhan and M. D. Altman as joint first authors.

[3]To be submitted [50].

[4]To be submitted [164]; some portions of this chapter have been published previously [165, 166].

[5]To be submitted as an appendix with Chapter 2 [38].

[6]To be submitted as an appendix with Chapter 2 [38].

[7]To be submitted as an appendix with Chapter 2 [38].

[8]This appendix appeared in the proceedings of the 2005 IEEE Conference on Engineering in Medicine and Biology [36].

List of Figures

List of Tables

Chapter 1

Introduction

It has long been recognized that computer simulations of interactions between biological molecules hold tremendous value not only for designing molecules, but also for analyzing the mechanisms of interaction between molecules. Such simulations can help guide a course of experimental studies and, in addition, complement experiment by enabling the comparison of the energetics of different interactions in ways not accessible to experiment. Accordingly, with the rapid increase in computer processing capabilities there has been a correspondingly large growth in the study of numerical techniques for biomolecule simulation. The wide range of available methods reflects the numerous types of problems studied. Investigations of processes such as catalysis can require extremely accurate quantum mechanical modeling [1]. At the other end of the computational spectrum, many problems in molecular design have intractably large search spaces, and therefore solution methods include not only highly approximate methods for evaluating interactions but also careful search algorithms to strongly limit computational complexity [2].

In many problems in molecular analysis and design, the concept of a molecule's *solvation free energy* is a valuable tool for analyzing biomolecular structure–function relationships and interactions [3]. This free energy, denoted by ΔG^0_{solv}, is defined to be the difference between the free energy of the molecule in solution and its free energy in a gas-phase reference state. Such a quantity is useful because it allows the decomposition of complex processes such as binding, whose energetics may not be readily estimated, into a set of simpler thermodynamic steps whose energies are perhaps easier to estimate. Figure 1-1

illustrates a thermodynamic cycle that can be used to estimate the free energy of binding between two molecules. The unbound state is assumed to have the binding partners, labeled L for ligand and R for receptor, infinitely separated in solution. Each binding partner is transferred from solvent to a low-dielectric gas phase, and then the partners are bound in vacuum. Transferring the complex into solvent then completes the cycle that determines the binding free energy in solution. The binding free energy estimate is therefore obtained using three solvation free energies and a gas-phase binding free energy; the important point is that all of these quantities are more easily evaluated. Thermodynamic cycles such as this one are thus helpful not only as computational tools to decompose difficult calculations, but also as theoretical tools that allow more fine-grained energetic analysis.

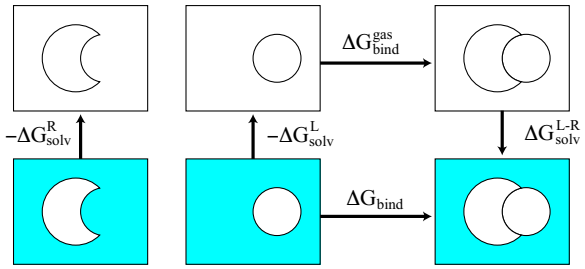


Figure 1-1: A thermodynamic cycle illustrating the utility of solvation free energy calculations for estimating binding free energies. The shaded region on the lower set of panels represent aqueous solvent. The upper panels represent a uniform low dielectric with zero ionic strength throughout. One can determine the binding free energy by adding the free energies associated with de-solvating the two unbound partners, complexing them in the gas phase, and re-solvating the complex.

Unfortunately, the calculation of solvation free energies represents one of the most difficult challenges in molecular modeling. The many solvent molecules and possibly salt ions that surround biomolecules present a basically intractable many-body problem. Their treatment is essential, however, because virtually all biological reactions occur in aqueous solution. The most accurate mathematical models of these physical systems — high-level quantum mechanics — are far too computationally demanding to be of practical use for most problems. Even molecular dynamics (MD) simulations, which integrate Newton's laws of motion, can require prohibitive computational resources to calculate quantities of interest. Modeling the energetics of water–solvent interactions using MD requires the sampling of the enormous phase space associated with the solvent molecules and ions. Com-

mon techniques for calculating these energetics, such as free energy perturbation methods [4], cannot always be converged given reasonable amounts of computational work. It should be noted that these methods for calculating free energies, which do not require the intermediate solvation calculations shown in Figure 1-1, may be applied to calculate binding free energies directly [5], although the more common approach is to compute binding free energy changes resulting from chemical change. However, even the direct calculations face the limitations imposed by the sampling problem.

In contrast to expensive explicit-solvent methods, there exist much faster techniques to estimate the solute–solvent interactions using an implicit representation of the solvent in which zero or only a small number of solvent molecules are treated explicitly. For a review of implicit-solvent models, see [6]. These models, which are often based on continuum theory, offer an attractive tradeoff between computational efficiency and accuracy. Continuum models have been shown to offer good agreement with their much more computationally intensive counterparts [7,8], and for many problems involving small-molecule design or the modification or analysis of large molecules such as proteins, the loss of accuracy relative to explicit-solvent simulations is acceptable. In continuum models of solvation, the solvation free energy of a molecule is commonly considered to be the sum of two components [3]:

$$\Delta G^{0}_{\text{solv}} = \Delta G^{0,\text{np}}_{\text{solv}} + \Delta G^{0,\text{es}}_{\text{solv}}. \tag{1.1}$$

The first free energy is called the nonpolar contribution to solvation; this term accounts for the van der Waals interactions between solute and solvent as well as for the entropic cost associated with excluding solvent molecules from the solute volume. This term is commonly estimated to grow in proportion with the surface area of the solute [8].

The electrostatic solvation free energy $\Delta G^{0,\text{es}}_{\text{solv}}$ accounts for the electrostatic enthalpy as well as the solvent entropy associated with the solute charge distribution. Continuum electrostatic theory is commonly used to calculate a molecule's electrostatic solvation free energy [3, 9]. These models generally treat the electrostatic potential in the molecule and in aqueous solvent with a symmetric, monovalent salt as obeying the Poisson–Boltzmann

equation

$$\nabla \cdot (\varepsilon(r)\nabla\varphi(r)) = -\rho(r) + \kappa^2(r)\sinh(\varphi(r)), \tag{1.2}$$

where $\varphi(r)$ is the potential at a point r, $\varepsilon(r)$ is the permittivity, $\rho(r)$ is a distribution of fixed charge, and the modified inverse Debye screening length κ describes the screening effect by mobile salt ions. Many biological systems with relatively low charge density can be modeled using the much simpler linearized equation

$$\nabla \cdot (\varepsilon(r)\nabla\varphi(r)) = -\rho(r) + \kappa^2(r)\varphi(r), \tag{1.3}$$

and this thesis focuses exclusively on this form. The solute interior and solvent exterior regions are generally treated as homogeneous dielectric continua with possible salt treatment, with the boundary between interior and exterior defined in relation to a set of sphere centers and their radii, where each sphere represents an atom or group of atoms. The dielectric constant in the molecule is typically taken to be between 2 and 4 [9], although some recent work has used dielectric constants up to 20 for surface groups [10]; a dielectric constant of 2 represents electronic polarization only, and slightly higher dielectric constants are often used to account for minor fluctuations in molecular structure. The dielectric constant in the solvent is usually modeled with that of bulk water, which is approximately 80. The solute charge distribution is taken to be a set of discrete point charges located at the atom (or group) centers. The point charge values and sphere radii are commonly assigned using either molecular mechanics force fields such as CHARMM22 [11], parameter sets specifically fit for electrostatic calculations [12], or quantum mechanical calculations for charges (for a recent article reviewing such methods, see [13]) in conjunction with force-field radii.

It should be noted that continuum-model solvation free energies are often computed for a single, static molecular structure [3]. In reality, of course, the molecule is not static in shape but fluctuates, and the free energy is an ensemble average. The structure lowest in energy will contribute the most to the average, and therefore most single-structure calculations rely on either time-averaged structures generated from a molecular dynamics (MD) trajectory [9], energy-minimized structures, or atomic coordinates obtained from X-ray crystallography or NMR experiments. Recently, there has been a movement towards

the use of multiple structures in calculating solvation free energies and binding free energies [14].

Warwicker and Watson presented the first numerical simulations of a continuum model for realistic biomolecule geometries [15], and since then a vast number of other numerical approaches based on finite-difference, finite-element, and boundary-element methods have been presented (for a small but representative selection, see [16–33]). These approaches have enabled a wide range of computational studies over the previous two decades. However, as we demonstrate in this thesis, there exist important calculations for which standard finite-difference methods are unable to achieve a desirable level of accuracy. Non-rigid binding free energy calculations, for instance, can entail calculating the difference between comparably large solvation free energies. Not infrequently, the approximate error in the solvation energies is of comparable magnitude to their difference, and therefore significant skepticism is in order when interpreting the results of such calculations. The bulk of this thesis therefore focuses on the development of numerical methods that can find highly accurate solutions to the models used without inordinately high requirements for computational resources such as memory or time.

The need for accurate solution may also be motivated philosophically. All numerical methods necessarily return approximate answers to the unobtainable exact PDE solution, and the models for the nonpolar and electrostatic solvation free energies are themselves somewhat approximate. The compounding of approximations strains the credibility of predictions so obtained: where should fault be assigned if the predictions are proven incorrect? Which component of the predictive process warrants attention for improvement? One of the most important guiding principles for this thesis research is that rigor demands that uncertain models should be solved as exactingly as possible when used for design or for studying mechanisms. Resolving numerical uncertainty strengthens not only the trustworthiness of the predictions, but also critically enables experimental results to feed directly back into clarifying the models. It should be noted that this modeling philosophy argues for the use of continuum models, whose mathematical properties are relatively well understood, over explicit-solvent simulations for which convergence properties are not as clear.

The thesis is organized into six chapters. Chapter 2 describes an approach for accu-

rately discretizing three of the solute–solvent definitions most commonly used in molecular modeling; these surfaces define a boundary between the interior of a biomolecule and an exterior region—either the gas phase or solvent. We define two classes of compact curved surfaces, or panels, into which these boundaries can be discretized essentially exactly. Importantly, we present numerical integration techniques specialized for the curved integration domains and for the integrands of interest, which may be singular. In Chapter 3 we present a specialized algorithm to rapidly solve boundary-element-method problems in biomolecule electrostatics. Our algorithm, which we call FFTSVD, can also be applied to modeling problems in other domains of potential theory, including fluidic simulation and electromagnetics. Chapter 4 discusses a large-scale, boundary-element-method implementation for biomolecule electrostatics. The implementation uses the FFTSVD fast algorithm and curved boundary elements to achieve high accuracy without sacrificing computational efficiency. Furthermore, the boundary-integral-equation formulation is much more general than those presented in the literature, and this generality allows a previously unavailable unified treatment of complex molecular topologies such as solvent-filled cavities as well as ion-exclusion (Stern) layers. Chapter 5 describes a novel, highly efficient numerical approach for calculating a biomolecule charge distribution that optimizes the free energy of binding to another molecule. This approach builds on physical theory developed by Kangas, Lee, and Tidor [34, 35] and represents an unusual approach to PDE-constrained optimization. Application to a realistic test case validates the approach and application to simple test problems illustrate the method's improved performance. Chapter 6 summarizes the main contributions of the thesis and suggests ideas for future work in this area.

Several appendices have been included. The first three offer details regarding curved-panel discretization and integration. Appendix D addresses a popular integral formulation for electrostatics in non-ionic solution [36]. The formulation accuracy is highly sensitive to the process by which the integral equations are discretized, and we show that a process known as qualocation offers accuracy superior to the commonly used centroid-collocation methods [37].

Chapter 2

Numerical Integration Techniques for Curved-Element Discretizations of Molecule–Solvent Interfaces[1]

We explore the use of exact representations of solute–solvent interfaces in surface formulations of biophysical modeling problems. Following and refining Zauhar's work [39], we define two classes of curved elements that can exactly discretize the van der Waals, solvent-accessible, and solvent-excluded surfaces. This work presents numerical integration techniques specialized for the curvature of these surfaces and for the singular integrals required to solve boundary-integral formulations of continuum electrostatics problems using boundary-element methods (BEM). The integration methods are applied to surface-Generalized-Born (sGB), surface-continuum van der Waals (scvdW), and boundary-element electrostatics problems. Results demonstrate that electrostatics BEM using curved elements with piecewise-constant basis functions and centroid collocation is nearly ten times more accurate than planar-element BEM. Furthermore, the sGB and scvdW calculations give exceptional accuracy even for coarsely discretized surfaces. The extra accuracy is attributed to the correct representation of the solute–solvent interface.

[1] To be submitted [38] with Appendices A, B, and C.

2.1 INTRODUCTION

Several important problems in molecular physics can be modeled using boundary integral equations or surface integrals over the molecular surfaces. Continuum electrostatics models based on Tanford–Kirkwood theory [40] give rise to variable-coefficient Poisson or Poisson–Boltzmann partial differential equations that can be converted to boundary integral equations. The generalized-Born model [41], commonly used to estimate electrostatic interactions, can also be transformed to a surface formulation [42]. Recently, Levy *et al.* presented a continuum model for estimating the van der Waals interaction energy between a molecular solute and surrounding aqueous solvent [8]; this model can also be solved using surface methods [43].

Surface formulations offer several advantages for numerical computation. Boundary-integral-equation problems require the solution of two-dimensional rather than three-dimensional problems, requiring correspondingly fewer unknowns and therefore less computer memory resources. In addition, exterior problems — those requiring discretization of an infinite or semi-infinite volume domain — are reduced to problems over compact domains. For most problems of interest, these domains are complicated surfaces for which there exist no closed-form expressions for the associated integrals. To facilitate numerical solution, a complicated surface is usually approximated as the union of a set of simpler subdomains for which integration techniques are known. Commonly, these subdomains, which are called boundary elements, or panels, are planar triangles or quadrilaterals. There exists a large body of literature devoted to the evaluation of integrals over these domains (see, for examples, references [44–46]).

In many physical modeling problems at the molecular scale, the surfaces of interest are curved, representing an atom or a collection of atoms. Even when surface discretizations can be readily obtained, integrating singular or near-singular functions over curved surfaces poses a challenge. Numerical quadrature techniques have been developed for quadratically curved surfaces (defined by curves along the element edges) [47] and B-splines [48], but relatively few numerical integration techniques specialized to molecular shapes have been presented [29, 39]. For boundary-element methods, improved accuracy is often achieved

by using higher-order basis functions on planar, quadratic, or cubic boundary elements. Unfortunately, basis functions of infinitely high order would fail to give correct answers for these problems, because the surface discretizations only approximate the true geometry. The failure of such methods even in ideal thought-experiments highlights a fundamental limitation imposed by inexact surface representation: increasing the number of surface elements improves both the basis set and the geometrical approximation, and it can be difficult to assess the relative importance of these effects in order to determine where effort should be made to achieve an optimal trade-off between accuracy computational expense.

In this work we explore the impact of using curved-element rather than planar-element discretizations of the solute–solvent interface for several types of molecular modeling problems. First, we define two classes of curved boundary elements that can exactly represent three of the most common molecular boundary definitions. Second, we develop efficient numerical techniques to evaluate singular and near-singular integrals over the curved elements. Using these methods, we calculate Generalized Born radii, solute–solvent van der Waals interaction energies, and electrostatic components of solvation energies. Our work on curved boundary elements most closely resembles the work of Zauhar [39] and that of Liang and Subramaniam [29]. We present nearly exact discretizations of solvent-excluded surfaces [49], in contrast to the approximate solvent-accessible surfaces of Liang and Subramaniam and the smoothed solvent-excluded surfaces presented by Zauhar. In addition, we describe numerical integration techniques designed to treat the curved-element singular and near-singular integrals required for numerical solution of the boundary-integral equations. One of our most significant findings is that if the accurate surface geometry is used, then only a relatively small number of discretization degrees of freedom are needed to achieve high accuracy. The very large number of degrees of freedom required for convergence of other methods contribute mainly to improving the accuracy of the geometric representation.

In Section 2.2 we introduce several physical problems that can be addressed by solving boundary integral equations or by integrating functions over solute–solvent interfaces, and also briefly describe popular interface definitions and discretization approaches. Curved elements that can exactly represent the relevant boundaries are defined in Section 2.3, and

in Section 2.4 we present accurate and efficient numerical integration methods for these curved boundaries. Validation of the surface discretizations and the integration techniques, as well as demonstration of the advantages of curved-element surface methods, are given in Section 2.5. Conclusions are in Section 2.6.

2.2 BACKGROUND

2.2.1 Surface Formulations of Biophysical Problems

Molecular Electrostatics

Figure 2-1 illustrates the mixed discrete–continuum electrostatics model [3, 9]. The molecular interior is defined to be a homogeneous region with low permittivity, denoted ε_I, and the molecule's charge distribution is taken to be a set of n_c discrete point charges, which are often located at the atomic nuclei. In this low-permittivity region the electrostatic potential satisfies a Poisson equation. The solvent region exterior to the boundary Ω is assumed to be a homogeneous medium with much higher permittivity than the interior, which is denoted by ε_{II}, and a Debye screening parameter κ. In this exterior region, the potential satisfies the linearized Poisson–Boltzmann equation. The Richards molecular surface [49] is commonly used to define the boundary Ω.

The Poisson problem in the interior and the linearized Poisson–Boltzmann problem in the exterior are coupled by continuity conditions at the boundary [51]. These coupled partial differential equations can be converted to integral equations in several ways. Problems in non-ionic solutions (those with $\kappa = 0$ in the solvent region) can be solved using the induced surface-charge method [17, 19]. When the ionic strength is non-zero, Green's theorem can be applied to derive either a mixed first-second-kind integral formulation [21] or a purely second-kind formulation [23]. Chipman [52] has described and compared these and other formulations. We present the mixed formulation originally presented by Yoon and Lenhoff [21].

Applying Green's theorem in both regions and applying the continuity conditions gives

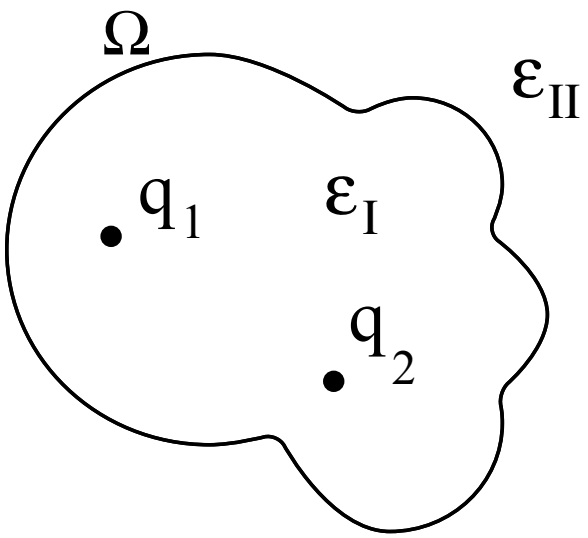


Figure 2-1: A mixed discrete-continuum model for biomolecule electrostatics. The surface Ω represents the dielectric boundary between regions with dielectric constants ε_I and ε_{II}. Partial atomic charges are located in region I, with illustrative charges q_1 at r_1 and q_2 at r_2. The Debye screening parameter κ is zero within region I and may be non-zero in region II. In work not described here, an ion-exclusion layer may also be treated [50].

the coupled integral equations

$$\frac{1}{2}\varphi(r_\Omega)+\oint_\Omega \varphi(r')\frac{\partial G_I}{\partial n(r')}(r_\Omega;r')dA'-\oint_\Omega \frac{\partial \varphi}{\partial n(r')}(r')G_I(r_\Omega;r')dA' = \sum_{i=1}^{n_c}\frac{q_i}{\varepsilon_I}G_I(r_\Omega;r_i); \tag{2.1}$$

$$\frac{1}{2}\varphi(r_\Omega)-\oint_\Omega \varphi(r')\frac{\partial G_{II}}{\partial n}r'(r_\Omega;r')dA'+\frac{\varepsilon_I}{\varepsilon_{II}}\oint_\Omega \frac{\partial \varphi}{\partial n}r'(r')G_{II}(r_\Omega;r')dA' = 0. \tag{2.2}$$

Here, r_Ω is a point on the surface; r' is the integration variable on the surface; $n(r')$ is the normal at r' pointing into solvent; $\oint$ denotes the principal value integral taken in the limit as a field point approaches r' from the inside; $\varphi(r)$ and $\frac{\partial\varphi}{\partial n}(r)$ denote the potential and its normal derivative at the surface; and $G_I(r;r')$ and $G_{II}(r;r')$ are the free-space Green's functions for the governing equations in the two regions. Typically, $G_I(r;r')=\frac{1}{4\pi||r-r'||}$ and $G_{II}(r;r')=\frac{\exp(-\kappa||r-r'||)}{4\pi||r-r'||}$.

To solve Equations 2.1 and 2.2 using a boundary-element method, the solute–solvent boundary is discretized and the surface variables are approximated as weighted sums of compactly supported basis functions, where the weights are selected so that the discretized

integrals match a set of constraints (see, for example, [53,54]). In collocation methods, the residual is forced to be exactly zero at a set of points on the surface; in Galerkin methods, the residual is required to be orthogonal to the basis functions. Using collocation and piecewise constant basis functions such that the i^{th} basis function is unity on the i^{th} boundary element and zero elsewhere, we form a dense block matrix whose entries take the form

$$\int_{\text{element } j} K(r_i; r')dA', \tag{2.3}$$

where r_i denotes the collocation point associated with the i^{th} boundary element and $K(r;r')$ is either a Green's function or a Green's function derivative with respect to the surface normal at r'.

Surface Generalized Born

The Generalized Born (GB) model of solute–solvent electrostatic interactions yields a more easily computed approximation to energies derived by solving the Poisson–Boltzmann equation [41]. The GB pairwise energy $U_{i,j}$ between charges i and j is given by the equation

$$U_{i,j} = -\frac{1}{2}(1/\varepsilon_{II} - 1/\varepsilon_I)\frac{q_i q_j}{\sqrt{r_{ij}^2 + R_i R_j \exp(-r_{ij}^2/4R_iR_j)}}, \tag{2.4}$$

where q_i and q_j are the charge values and R_i and R_j are the Born radii. The Born radius R_i for an atom or group of the solute is defined such that a sphere with radius R_i and centrally-located unit charge has solvation energy equal to that of the entire molecule if $q_i = 1$ and $q_j = 0 \ \forall j \neq i$.

Still *et al.* proposed to calculate the Born radius R_i by relating the volume integral

$$\int_{V_{\text{int}}} \frac{1}{||r' - r_i||^4} dV' \tag{2.5}$$

to the analytical expression for the solvation energy of a centrally located charge in a spherical dielectric cavity [41]. In this equation, V_{int} is the volume of the solute interior and r' denotes the integration variable. Similar expressions to calculate Born radii have also

been presented [42, 55, 56]. Ghosh *et al.* introduced the surface-Generalized Born (S-GB) method [42], in which an application of the divergence theorem converts Equation 2.5 to the surface integral

$$\int_S \frac{(r'-r_i)^T \hat{n}(r')}{||r'-r_i||^4} dA', \tag{2.6}$$

where S denotes the dielectric boundary, which we again assume to be the Richards molecular surface.

Continuum van der Waals

Levy *et al.* described a continuum method to model the van der Waals interactions between solute and solvent [8], based on assuming a spherical model for a water molecule. In this model, the interaction energy is then expressed as an integral over the solvent volume,

$$U_{\mathrm{vdW}}^{(i)} = \sum_{i=1}^{n} \left(\int_{\mathrm{solvent}} \rho_{\mathrm{w}} u_{\mathrm{vdW}}^{(i)}(r) dV' \right), \tag{2.7}$$

where n denotes the number of atoms in the solute, ρ_w the bulk water number density, and $u_{\mathrm{vdW}}^{(i)}(r)$ the van der Waals potential between atom i and a water molecule located at a distance $r = ||r' - r_i||$ from the atom center r_i.

The van der Waals potential is defined by the distance from a water molecule center to an atom center, so the solvent-accessible surface [57] is the natural solute–solvent boundary definition for the integral in 2.7. If the van der Waals potential is modeled by the Lennard-Jones 6/12 function,

$$u_{\mathrm{vdW}}^{(i)}(r) = \frac{A^{(i)}}{r^{12}} - \frac{B^{(i)}}{r^6}, \tag{2.8}$$

then the divergence theorem applied to 2.7 yields

$$\int_V \left(\frac{A}{r^{12}} - \frac{B}{r^6} \right) dV = \int_S \frac{\partial}{\partial n} \left(\frac{A}{90r^{10}} - \frac{B}{12r^4} \right) dS \ . \tag{2.9}$$

2.2.2 Defining Molecule–Solvent Interfaces

Figure 2-2 illustrates the three most prevalent definitions for the solute–solvent boundary. A molecule's van der Waals surface, as shown in Figure 2-2(a), is defined to be the bound-

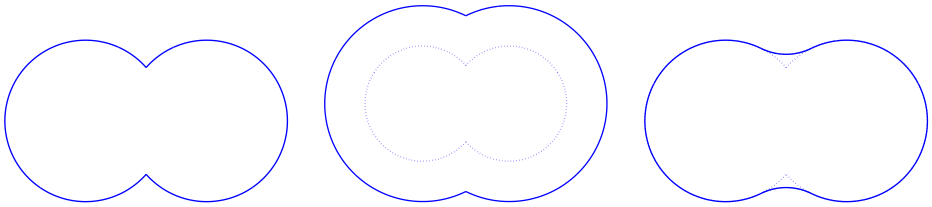


Figure 2-2: Three definitions of solute–solvent boundaries: (a) van der Waals surface. (b) Solvent-accessible surface. (c) Solvent-excluded (molecular) surface. The dotted lines in (b) and (c) denote the van der Waals surface.

ary of a union of spheres. Each sphere represents an atom centered at a particular location in space and the sphere radius is set to the atom's van der Waals radius; for reduced-atom models such as the polar-hydrogen CHARMM19 model [58], some spheres represent groups of atoms. The Lee and Richards solvent-accessible surface [57], depicted in Figure 2-2(b), is also a union of spheres; in this definition, each sphere's radius is equal to the atom or group's van der Waals radius plus the radius of a spherical probe molecule that is rolled over the union of atoms. The portion of each atom's surface that is exposed to solvent can be described as the intersection of the sphere's surface with a set of half-spaces [59], and each such piece of surface is called a patch.

Richards defined the molecular surface, or solvent-excluded surface [49], and Connolly [59] presented an algorithm for its analytical determination. As illustrated in Figure 2-2(c), the molecular surface is defined by rolling a probe sphere over the union of spheres with van der Waals radii; the surface consists of the set of points of the probe sphere's closest approach to the boundary of the union. In this definition, the regions of the molecular surface that correspond to probe positions at which the probe contacts the sphere union at only one position are said to belong to the *contact surface*; such convex, spherical surface patches are called caps [59]. In contrast, the *reentrant surface* comprises regions that correspond to probe positions at which the probe touches the sphere union at multiple points. Where the probe touches two spheres of the union, its movement is restricted by one degree of freedom; a toroidal, or belt, piece of surface is then produced as the probe rotates about the axis defined by the two sphere centers. Where the probe touches the union at three or more points, a concave spherical surface patch is defined; this type of face is

termed a pit. All three types of surface patches, or faces, are bounded by circular arcs, and molecular surfaces can be represented exactly as a finite union of different instances of these surfaces [59].

Many researchers have presented algorithms to discretize solvent-excluded and solvent-accessible surfaces [39, 60–68]. The algorithms take as input the atom centers and their radii, as well as the probe sphere radius, and return a set of boundary elements that approximate the molecular or accessible surface. Generally, to improve the surface approximation one uses a larger number of smaller elements. Most work has focused on generating planar-triangle-based surface discretizations, but several groups have developed more sophisticated approaches. Zauhar and Morgan have reported cubically-curved elements [19, 69], Juffer *et al.* used cubic interpolation [23], Bajaj *et al.* used B-spline patches [60], Bordner and Huber used quadratically-curved elements [70]. Zauhar has presented an approach to exactly discretize a smooth approximation to the molecular surface such that the surface has a continuous normal [39]. Liang *et al.* find an exact solvent-accessible surface derived from alpha shapes [29, 63, 64], but solve problems on an exactly-curved approximation to this surface. Our approach exactly discretizes the Richards molecular surface using Connolly's method and we solve problems on this exact representation using numerical integration techniques specialized for these surfaces.

2.3 SURFACE DISCRETIZATION

As discussed in Section 2.2.2, three common solute–solvent boundary definitions can be represented as the union of portions of toruses and spheres, where the surface construction ensures that the boundaries between different surface patches are formed by arcs of circles. In this section we define two classes of curved surface elements that permit the exact discretization of the solute–solvent boundaries.

2.3.1 Toroidal Element Definition

A torus is defined by revolving a circle about an axis that lies in the same plane as the circle. The circle center, normal, and revolution axis together define a local coordinate system, and

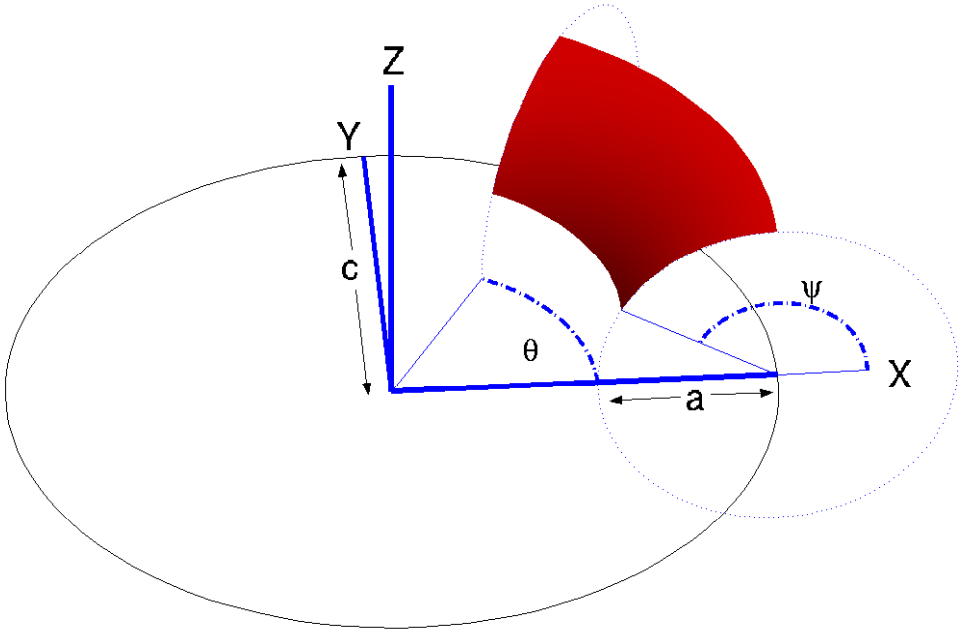


Figure 2-3: Specification of a torus and a torus element with $0 \leq \theta \leq \pi/3$ and $\pi/2 \leq \psi \leq 5\pi/6$.

it is useful to describe the torus as having an outer radius *c*, which is the shortest distance between the circle center and the revolution axis, and inner radius *a*, which is the radius of the circle. We define *z* to be the axis of revolution, *y* to coincide with the normal to the original circle, and the origin such that the circle origin lies in the $x-y$ plane. Two angular coordinates θ and ψ, both in the interval $[0, 2\pi]$, suffice to specify any point on the torus. The angle θ describes the azimuthal angle of the point relative to the *x* axis in the $x-y$ plane. The angle ψ determines the point's position on the circle at θ, and is defined such that $\psi = 0$ points radially outward from the origin and $\psi = \pi$ points radially inward. We define a *torus element* as the portion of a torus with angular coordinates $\theta_1 \leq \theta \leq \theta_2$ and $\psi_1 \leq \psi \leq \psi_2$. An arbitrary toroidal element is shown in Figure 2-3. The circle center, as it revolves around the axis of revolution *z*, traces a circle, which is shown in black in the Figure. We number and define the edges of the torus in a right-handed manner (*i.e.*, the interior of the element is to the left as one traverses the edges). Because the toroidal surface patches form part of the reentrant surface, the torus element normal points into the finite volume enclosed by the torus.

2.3.2 Spherical Element Definition

We define a *generalized spherical triangle* (GST) to be a three-sided region of a sphere's surface whose edges are formed by three circular arcs [43]. The arcs are not permitted to intersect except at their endpoints, which are the vertices of the generalized spherical triangle. Furthermore, at the vertices formed by adjacent arcs, the local interior angles must be less than π radians. This definition contrasts with a regular spherical triangle, whose arcs are portions of great circles on the sphere. Figure 2-4 illustrates a GST in which one arc is a portion of a small circle and the others belong to great circles. The arcs are oriented and numbered in a right-handed fashion, following standard mathematical convention. Convex spherical faces have a normal pointing away from the sphere center; concave faces have a normal pointing towards the sphere center, because the concave faces must point out into the solvent region. A surface-representation error results if only great-circle arcs are allowed to form the element boundaries, because small-circle arcs are needed to resolve the boundaries between surface patches [39]. Liang and Subramaniam generated curved-element discretizations by projecting the edges of a triangulated surface out to the sphere [29]; the surface elements so generated have exact curvature but their edges are all arcs of great circles.

2.4 CURVED-ELEMENT INTEGRATION METHODS

In this section, numerical techniques are presented to evaluate integrals of the form

$$\Phi(r) = \int_{\Omega} K(r;r')dA', \tag{2.10}$$

where Ω is either a toroidal or generalized spherical triangle element, as defined in Section 2.3. For the problems discussed in this work, the function $K(r;r')$ is singular at $r = r'$ and decays monotonically to zero as $||r - r'|| \to \infty$. For smooth integrands such as far-field integrals in which r is far from Ω, the integration may be performed using numerical quadrature. We present specialized methods for smooth integrands in Section 2.4.1. Integrals for which $r \in \Omega$, or is sufficiently close that the integrand varies extremely rapidly,

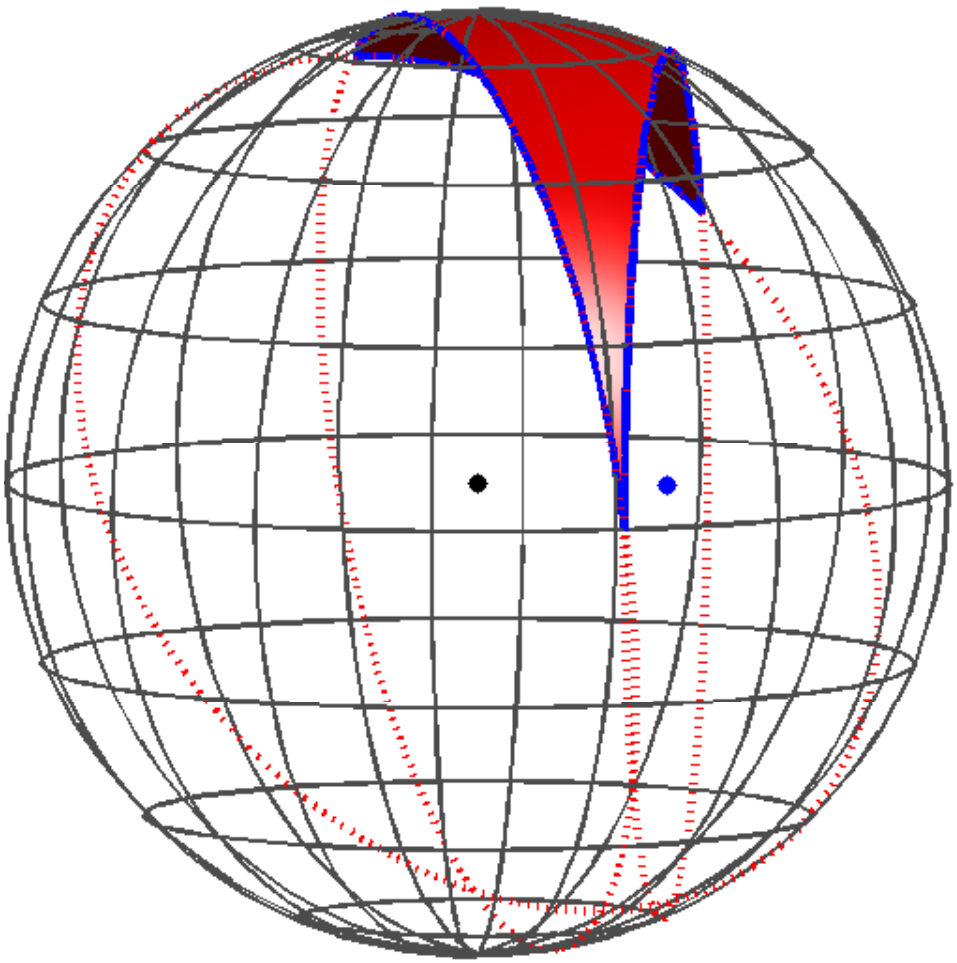


Figure 2-4: A generalized spherical triangle (GST) with one bounding edge belonging to the circle centered at the blue dot. The remaining edges belong to great circles on the sphere.

are called near-field integrals, and require special techniques, which we present in Section 2.4.2.

2.4.1 Far-Field Quadrature

When the evaluation point r in Equation 2.10 is sufficiently far from the domain of integration Ω, $K(r;r')$ varies smoothly over Ω and therefore relatively low-order numerical quadrature suffices to provide accurate results. A q^{th}-order quadrature rule estimates the integral of a function f over a domain β as a weighted sum of function evaluations at q specified points in β:

$$\int_{\beta} f(x)dx \approx \sum_{i=1}^{q} w_i f(x_i), \tag{2.11}$$

The values w_i are called quadrature weights and the points x_i are called quadrature points. Many types of quadrature rules are designed such that they give exact or nearly exact results if the domain is simple and the integrand is a sufficiently low-order polynomial. For simple integration domains like planar triangles, well-established rules such as those presented by Stroud [44] offer excellent accuracy.

To integrate a function over a more complex domain Ω, one typically determines a smooth coordinate transformation M from a simple domain Γ, which has a known quadrature rule, to the domain of integration Ω. Applying the chain rule transforms the integral of Equation 2.10 to the form

$$\Phi(r) = \int_{\Gamma} K(r;M(\hat{r}'))\ \left|J(\hat{r}')\right|\ d\hat{A}', \tag{2.12}$$

where $\hat{r}'$ denotes the integration variable in Γ and $|J(\hat{r}')|$ is the determinant of the Jacobian of M at $\hat{r}'$. A q^{th}-order quadrature rule for the domain Γ allows Equation 2.10 to be approximated as:

$$\Phi(r) \approx \sum_{i=1}^{q} w_i\, K(r;M(\hat{r}_i'))\ \left|J(\hat{r}_i')\right|. \tag{2.13}$$

Because the original integrand, over Ω, is multiplied in the new integral by the Jacobian determinant $|J|$, it is essential that the product of the original integrand and the coordinate transformation be smooth; that is, $K|J|$ should vary smoothly over Γ.

We now describe such coordinate transformations for the curved elements presented in the preceding section.

Generalized Spherical Triangle Coordinate Transformation

Zauhar has presented one coordinate transformation between a planar triangle and what we have defined as the generalized spherical triangle [39]. We present an independently-derived alternative. Figure 2-5 illustrates the coordinate transformation from the standard planar triangle of Figure 2-5(a) with vertices $\{(0,0)^T;(1,0)^T;(0,1)^T\}$, to a GST, shown in top and side views in Figures 2-5(b) and 2-5(c). The GST has been oriented such that the longest arc, labeled a_1, lies in a plane perpendicular to the x axis and the arc midpoint lies in the $x-z$ plane. This mapping is guaranteed to exist if the vertex v_3 is further from the plane of arc a_1 than any other point on the arcs a_2 and a_3 and if in addition v_3 lies above the $x-y$ plane; such restrictions are easily imposed during surface discretization. The standard triangle parametric coordinates $(\xi,\eta)^T$ are first mapped to a spherical coordinate system $(\theta,\psi)^T$ as shown in Figures 2-5(b) and (c), and then trivially transformed to Cartesian coordinates. The angle ψ measures the angle from the positive x axis and the angle θ measures rotation about the x axis such that a point with $\theta = 0$ lies in the $x-z$ plane.

The reference triangle edge from $\hat{v}_3 = (0,1)^T$ to $\hat{v}_1 = (0,0)^T$ is mapped to the GST edge from v_3 to v_1. Letting $(\theta_i,\psi_i)^T$ denote the spherical coordinates of GST vertex v_i, it is clear that $\psi_1 = \psi_2$, and that for all points in the GST, $\psi \leq \psi_3$. As shown in Figure 2-5, every line of constant η in the standard triangle is mapped to an arc of the circle defined by $\psi = \psi_1 + \eta(\psi_3 - \psi_1)$. The arc endpoints are defined by the intersection of the circle at elevation angle ψ with the arcs a_2 and a_3. A point $(\xi,\eta)^T$ in the reference triangle is mapped to this arc by mapping the point's parametric distance $s = \frac{\xi}{1-\eta}$ to a parameterized form of the arc at ψ between a_2 and a_3. Appendix B contains the full derivation of the coordinate transformation and its Jacobian.

Toroidal Element Coordinate Transformation

A torus element is isomorphic to a rectangle. A simple mapping suffices to transform the unit rectangle, with vertices $\{(0,0)^T;(0,1)^T;(1,1)^T;(1,0)^T\}$, to an arbitrary torus element.

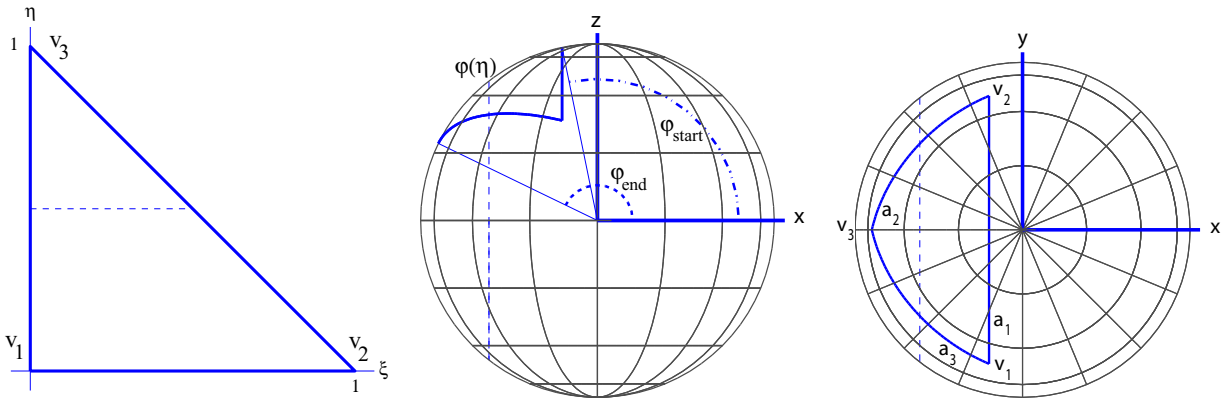


Figure 2-5: (a) The standard unit triangle in parametric coordinate space. (b) A GST viewed from the negative *y*-axis. The angle ψ is measured relative to the positive *x*-axis. Each ψ is mapped to one plane with normal along the *x*-axis; the plane intersects the sphere and defines a circle. (c) A GST viewed from the positive *z*-axis. Dashed lines indicate the circle of intersection between the sphere surface and the plane specified by φ. The image of the standard-triangle vertices under the coordinate transformation are labeled.

For the torus in Figure 2-3, with outer radius c, inner radius a, centered at the origin and with axis of revolution along the Z axis, the Cartesian coordinates of a point at $(\xi,\eta)^T$ in parametric coordinates are

$$r = \begin{bmatrix} c + a\cos(\psi)\cos(\theta) \\ c + a\cos(\psi)\sin(\theta) \\ a\sin(\psi) \end{bmatrix}, \tag{2.14}$$

where

$$\theta = \theta_1 + \xi(\theta_2 - \theta_1); \tag{2.15}$$

$$\psi = \psi_1 + \eta(\psi_2 - \psi_1). \tag{2.16}$$

The determinant of the Jacobian is

$$|J| = a(c + a\cos(\psi))\,|\psi_2 - \psi_1|\,|\theta_2 - \theta_1|\,. \tag{2.17}$$

The coordinates of a point on an arbitrarily positioned torus can be computed by applying a simple affine transform.

2.4.2 Near-field Integration Techniques

The integrands of interest have singularities as the evaluation point approaches the domain of integration. As a result, even high-order Gaussian quadrature rules fail to accurately approximate the singular and near-singular integrals; more sophisticated techniques are required. In this section we present techniques for integrating the Laplace kernel $K(r;r') = 1/(4\pi||r-r'||)$ and its normal derivative $\frac{\partial K}{\partial n(r')}(r;r')$. Appendix C describes how these methods may be adapted for the linearized Poisson–Boltzmann, surface-Generalized Born, and continuum-van der Waals kernels.

Single-Layer Potential

The integral

$$\Phi(r) = \int_{\Omega} \frac{1}{4\pi||r-r'||} dA' \tag{2.18}$$

is referred to as the single-layer potential because it represents the potential induced by a unit-density monopole charge layer on the integration domain.

Spherical Element Single-Layer. When Ω is a generalized spherical triangle, the method of Wang *et al.* can be applied to evaluate the integral in Equation 2.18 [43, 71]. Figure 2-6 illustrates the approach. For a given generalized spherical triangle, we define a flat reference element Γ that lies in the plane tangent to the spherical element at the GST centroid. After selecting a reference element, one finds a polynomial distribution of monopole charge on the reference element such that the reference-element induced potential accurately reproduces the potential induced by the original distribution on the curved element. For uniform distributions, the relation

$$\int_{\Omega} G(r;r')dA' = \int_{\Gamma} G(r;\hat{r}') \left[\frac{G(r;M(\hat{r}'))}{G(r;\hat{r}')} \left| J(\hat{r}') \right| \right] d\hat{A}', \tag{2.19}$$

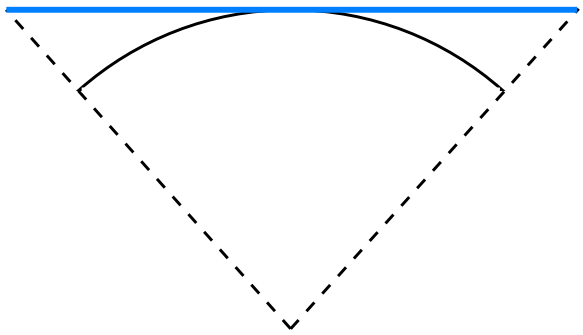


Figure 2-6: Schematic of the approach for evaluating the potential induced by a distribution of monopole charge on a generalized spherical triangle.

defines the reference-element monopole charge distribution that exactly reproduces the curved-element induced potential. In Equation 2.19, $\hat{r}'$ is a point in the flat element, $M(\hat{r}')$ is its image under the coordinate transformation from Γ to Ω, and $J(\hat{r}')$ is the Jacobian of the mapping. Because the flat element is tangent at the centroid and the sphere has constant curvature, the term in parentheses uniformly approaches $|J|$ at the centroid; for centroid-collocation with piecewise-constant basis functions, this is the only type of singular integral that must be evaluated. The term is actually smooth over the entire domain, and therefore it can be accurately approximated using a low-order polynomial [71]. Then the single-layer potential is calculated as

$$\int_{\Gamma} G(r;\hat{r}') \left[\frac{G(r;M(\hat{r}'))}{G(r;\hat{r}')} \left| J(\hat{r}') \right| \right] d\hat{A}' \approx \sum_{i,j} \alpha_{i,j} \int_{\Gamma} \xi^i \eta^j G(r;\hat{r}') d\hat{A}', \tag{2.20}$$

where the set $\{\alpha_{i,j}\}$ denote the polynomial coefficients and ξ and η denote a local coordinate system on the flat reference element such that one vertex is at $(0,0)^T$. Newman has presented techniques to analytically evaluate the monomial integrals $\int \xi^i \eta^j G(r;\hat{r}') d\hat{A}'$ for planar polygons, and Wang has described an alternative [46, 72].

The coefficients $\{\alpha_{i,j}\}$ are found by least-squares solution of the Vandermonde matrix equation

$$\begin{bmatrix} 1 & \xi_1 & \eta_1 & \dots & \xi_1^k \eta_1^l \\ 1 & \xi_2 & \eta_2 & \dots & \xi_2^k \eta_2^l \\ \vdots & & & \ddots & \vdots \\ 1 & \xi_n & \eta_n & \dots & \xi_n^k \eta_n^l \end{bmatrix} \begin{bmatrix} \alpha_{0,0} \\ \alpha_{1,0} \\ \vdots \\ \alpha_{k,l} \end{bmatrix} = \begin{bmatrix} \frac{G(r;M(\hat{r}_1))}{G(r;\hat{r}_1)} |J(\hat{r}_1)| \\ \frac{G(r;M(\hat{r}_2))}{G(r;\hat{r}_2)} |J(\hat{r}_2)| \\ \vdots \\ \frac{G(r;M(\hat{r}_n))}{G(r;\hat{r}_n)} |J(\hat{r}_n)| \end{bmatrix}, \tag{2.21}$$

where $\hat{r}_i = (\xi_i, \eta_i)^T$ denotes the i^{th} of n sample points, where n must be greater than the number of coefficients to be fit.

The flat reference element can be defined in one of two ways. In the first, the flat element edges are defined by casting rays from the sphere center through the GST boundary arcs to the tangent plane. Boundary arcs that are segments of great circles map to straight lines in this projective transformation, and any arc belonging to a small circle becomes a portion of a conic curve (either a hyperbola or an ellipse). The monomial integrals can then be evaluated by analytical integration over a triangular domain, followed by addition or subtraction, as necessary, of the result of numerical quadrature over the conic region [43]. An alternative method is to project the GST vertices to the tangent plane, which defines a triangle. The mapping between this reference triangle and the GST is then a composition of two mappings: the first transforms the reference triangle to the standard triangle, and the second transforms the standard triangle to the GST. The first mapping is straightforward, and methods for the second mapping have been presented in Section 2.4.1.

We emphasize that our selection of a flat reference element that lies tangent at the GST centroid suffices for the kernels specified in this work and for BEM approaches based on piecewise-constant basis functions and centroid-collocation; other problems may require that a reference element be defined in relation to the evaluation point [71].

Toroidal Element Single-Layer. When Ω is a toroidal element, the previously-described polynomial-fitting method is difficult to apply because the torus surface has unequal radii of curvature at most points. As a result, the ratio $G(r; M(\hat{r}'))/G(r; \hat{r}')$ takes different limits depending on the direction from which r' approaches r, and this phenomenon necessitates the development of more complicated coordinate transformations. Instead, recursive subdivision is applied to evaluate near-field integrals.

The element integral is evaluated in one of two ways. We denote the element centroid by r_c and its area by A. If the evaluation point r satisfies $||r - r_c|| < 5\sqrt{A}$, the element is split into four sub-elements defined by equally dividing the angular ranges. The sub-element integrations are then evaluated independently. Further subdivision may be required, depending on the position of the evaluation point relative to the four new centroids and the

new element areas. The second near-field integration method is applied when the evaluation point lies at the element centroid (*i.e.*, $r = r_c$). This case arises during the solution of boundary-element-method problems solved using centroid-collocation schemes. Symmetry in the θ direction allows these integrals to be evaluated for half the computational expense of a full subdivision. Both subdivision integration methods halt the subdivision when the divided elements have no edges longer than 10^{-5} Å.

Double-Layer Potential

The double-layer integral

$$\Phi(r) = \int_{\Omega} \frac{\partial}{\partial n(r')} \left(\frac{1}{4\pi||r - r'||} \right) dA' \tag{2.22}$$

represents the potential due to a unit-density dipole charge layer on the domain. The Wang *et al.* approach for double-layer integrals cannot be used for singular integrals. The ratio $\frac{K(r;M(\hat{r}'))}{K(r;\hat{r}')}$ is not defined on the reference element because $K(r;\hat{r}') = \frac{\partial}{\partial n(r')} \left(\frac{1}{4\pi||r-r'||} \right)$ vanishes for all $r' \neq r$ in the plane of the dipole layer.

We instead use the double-layer calculation presented by Willis *et al.* [73], which extends the work of Newman [46]. Recall that the potential induced by a normally-oriented dipole charge layer of uniform density equals the solid angle subtended by the integration domain at the evaluation point r [51]. Exploiting this characteristic, Newman derived an analytic expression for the double-layer potential induced by a uniform dipole distribution on a boundary element bounded by straight line segments [46]. Willis *et al.* extended Newman's work to uniform distributions on curved elements, noting that the subtended solid angle can be found easily using quadrature [73].

Figure 2-7 illustrates this approach for evaluating the double-layer potential in Equation 2.22. The evaluation point r is translated to the origin and the coordinate system is rotated so that the element centroid lies on the z axis. We define a sphere of unit radius centered at the origin and cast rays from the origin through the element edges to the sphere surface. The projected edges define the subtended angle, which we can compute by inte-

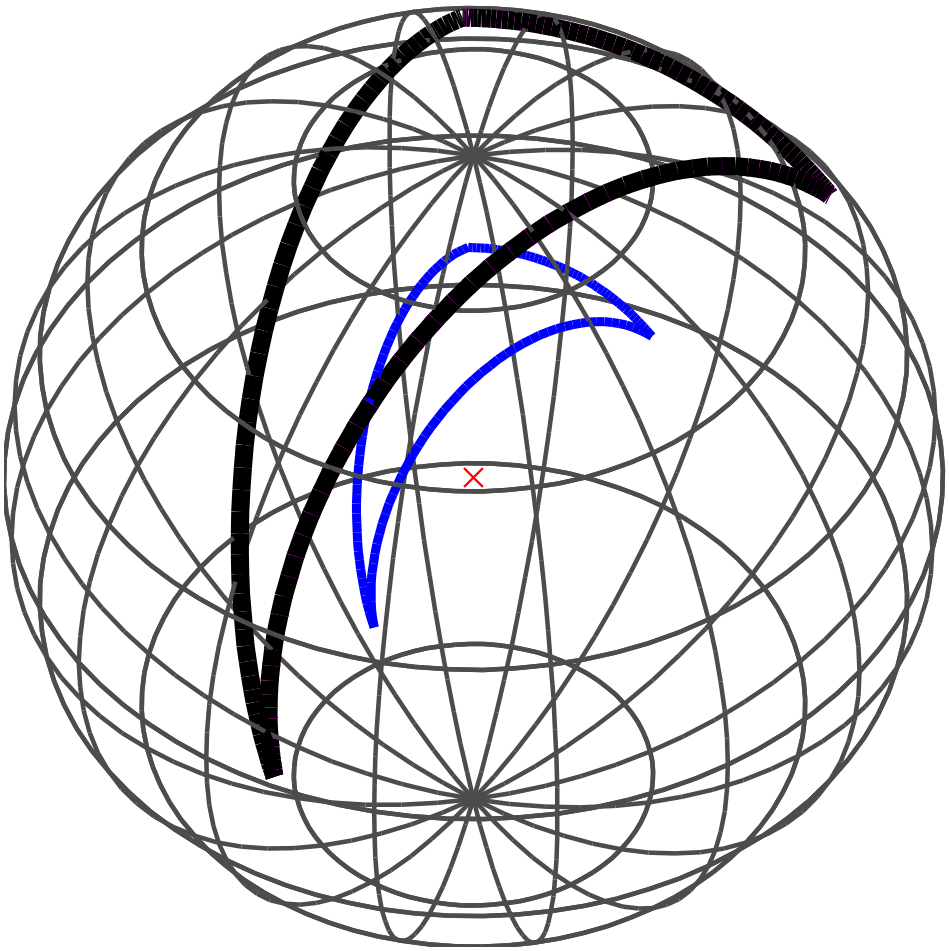


Figure 2-7: The Newman approach to calculating the potential induced by a uniform distribution of a normally-oriented dipole charge layer [46]. The circle at the center of the sphere denotes the point at which the potential is to be determined; the thin arcs form the edges of a GST; the thick lines are the projection of the GST bounding arcs to the sphere. The double-layer potential is the solid angle bounded by the thick lines.

grating the bounded area:

$$A = \int_0^{2\pi} \int_0^{\phi(\theta)} 1 \cdot \sin(\phi) d\phi d\theta = \int_0^{2\pi} (1 - \cos(\phi(\theta))\, d\theta \tag{2.23}$$

Separating the integration into a sum of integrals over each of the n_e circular arcs that form the element boundary, and changing variables from θ to a parametric t along the arcs, we have

$$A = \sum_{i=1}^{n_e} \int_0^1 (1 - \cos(\phi(\theta_i(t)))) \frac{d\theta_i}{dt} dt, \tag{2.24}$$

where t is the parametric coordinate along the i^{th} edge, and $\theta_i(t)$ is the azimuthal angle of the point at position t along the i^{th} arc.

The directional character of the double-layer potential deserves comment. The integral of Equation 2.22 is discontinuous as the evaluation point r approaches and passes through the surface. The value of the integral is defined to be the limit as r approaches the surface; when $r \in \Omega$, therefore, the side from which r approaches the surface will determine the value of the integral. The two limits sum to 4π [51]. By convention, we assume that the integral has been taken as the evaluation point approaches from the side opposite the normal direction.

An alternate approach, applicable only to uniform distributions, can also be taken. According to the Gauss–Bonnet theorem [74], the area bounded by the projected arcs can be determined following integration of the geodesic curvature of the projected edges. Finally, we note that the Willis *et al.* approach is applicable not only to spherical and toroidal surface elements, but also to many other types of curved surface elements [73].

2.5 RESULTS

We have generated several curved-element discretizations using the process outlined in Appendix A, and implemented the numerical integration methods in both C and Matlab [75]. Flat-triangular surface discretizations have been produced using Connolly's Molecular Surface Package [76]. We first present results that validate the surface discretizations and the integration techniques; we then demonstrate the advantages of curved-element surface

methods with several representative calculations on small molecules.

2.5.1 Validating the Surface Discretization

The surface area of both GST and toroidal elements can be calculated analytically. The Gauss–Bonnet theorem [74], when applied to a compact manifold, relates the integral of the curvature over the surface to the integral of the geodesic curvature of the boundary and the corner angles. A generalized spherical triangle has constant curvature over its surface, and its bounding arcs have constant geodesic curvature, and accordingly its area may be calculated analytically without difficulty. This approach to surface area calculation was taken by Connolly [59] and we use it here to demonstrate the correctness of the surface discretization. The area of a toroidal element defined in Section 2.3.1 is seen to be

$$A = \left| (\theta_2 - \theta_1) \left(ac\,(\psi_2 - \psi_1) + a^2\,(\sin(\psi_2) - \sin(\psi_1)) \right) \right|. \tag{2.25}$$

We generated both flat-element and curved-element surface discretizations of several molecules at varying levels of refinement, using the Richards molecular surface definition [49] and the solvent-accessible surface. PARSE radii [12] were used for molecular surface generation and CHARMM22 radii [11] were used for solvent-accessible surfaces. Analytical areas of the discretizations were then computed and compared to the analytical areas calculated by MSP [76]. Tables 2.1 and 2.2 present the molecular-surface and solvent-accessible-surface results; the calculations illustrate that even coarse curved-element discretizations accurately capture the molecular surface geometry. Similar results (not shown) have been obtained for van der Waals surfaces, which have spherical but not toroidal elements. It is especially noteworthy that planar-element discretizations with significantly more elements than their curved-element counterparts have not converged to the correct surface area. The more correct geometric description inherent to curved-element methods could lead to significantly more accurate numerical calculations than those based on planar-element discretizations.

	METHOD				
GEOMETRY	ANALYTICAL AREA (Å^2)	AREA OF DISCRETIZED SURFACE			
		FLAT		CURVED	
		# ELEMENTS	AREA (Å^2)	# ELEMENTS	AREA (Å^2)
Atom	12.566				
COARSE[a]		74	11.516	40	12.566
MEDIUM[b]		270	12.249	70	12.566
FINE[c]		448	12.390	124	12.566
Parallel β dialanine[d]	241.642				
COARSE[a]		684	230.965	1326	241.642
MEDIUM[b]		1944	238.450	1781	241.642
FINE[c]		2904	239.617	2923	241.642
Barnase–barstar complex[e]	8269.077				
COARSE[a]		29728	7979.774	63915	8269.077
MEDIUM[b]		79104	8188.538	88860	8269.077
FINE[c]		149160	8407.962	133676	8269.077

Table 2.1: Comparison of discretized surface areas with analytical molecular (solvent-excluded) surface area. Probe radius is taken to be 1.4Å. [a]: MSP angle = 1.0, NETGEN level = VERY COARSE; [b]: MSP angle = 0.5, NETGEN level = COARSE; [c]: MSP angle = 0.4, NETGEN level = MEDIUM; [d]: structure preparation is described in Section 2.5.5; [e]: from [77], entry 1BRS in the Protein Databank [78]. All quantities are in Å^2 and have been rounded to the nearest 0.001 Å^2.

	METHOD				
PROBLEM	ANALYTICAL AREA (Å^2)	AREA OF DISCRETIZED SURFACE			
		FLAT		CURVED	
		# ELEMENTS	AREA (Å^2)	# ELEMENTS	AREA (Å^2)
Atom	72.382				
COARSE[a]		74	66.334	40	72.382
MEDIUM[b]		270	70.554	68	72.382
FINE[c]		448	71.368	124	72.382
Parallel β dialanine[d]	467.815				
COARSE[a]		396	437.304	564	467.815
MEDIUM[b]		1268	459.406	714	467.815
FINE[c]		1846	462.617	1064	467.815
Barnase–barstar complex[e]	9152.150				
COARSE[a]		10643	8785.722	20053	9152.150
MEDIUM[b]		31800	9094.782	25835	9152.150
FINE[c]		87178	9571.220	38767	9152.150

Table 2.2: Comparison of discretized surface areas with analytical solvent-accessible surface area. Probe radius is taken to be 1.4Å. [a]: MSP angle = 1.0, NETGEN level = VERY COARSE; [b]: MSP angle = 0.5, NETGEN level = COARSE; [c]: MSP angle = 0.4, NETGEN level = MEDIUM; [d]: structure preparation is described in Section 2.5.5 [e]: from [77], entry 1BRS in the Protein Databank [78]. All quantities are in Å^2 and have been rounded to the nearest 0.001 Å^2.

2.5.2 Validating Curved Boundary-Element Integration

To illustrate the correctness of the coordinate transformations and the polynomial-fitting method, we used our numerical quadrature techniques to compute the surface areas of molecular surfaces. The areas are not expected to exactly match the analytical results because the Jacobian determinants are not polynomial. Table 2.3 lists the pit, belt, and cap areas calculated by analytical and direct quadrature methods, and also by the polynomial-fitting method for the pit and cap surfaces.

2.5.3 Surface Generalized-Born Calculations

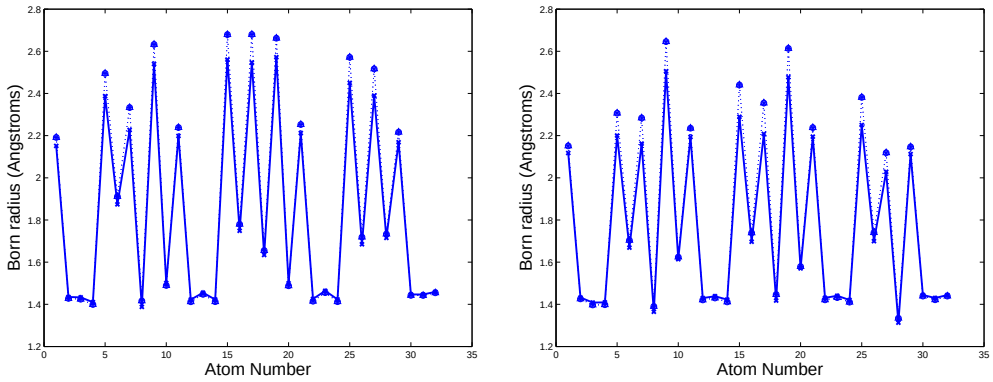
The surface discretization and integration techniques presented in this work have been used to calculate Born radii using the surface-Generalized-Born method introduced by Ghosh *et al.* [42] and surface formulations of the Grycuk [55] and Wojciechowski and Lesyng [56] Generalized Born models. The surface integrals associated with these calculations are never singular because every evaluation point is the center of a sphere. Figure 2-8 is a plot of the Born radii computed for the α-helical dialanine and parallel-β dialanine molecules using a surface formulation of the Grycuk method; results are shown for several levels of surface discretization. Also shown are the Born radii calculated by BEM solution of the Yoon and Lenhoff formulation of the Poisson–Boltzmann equation [21]. These calculations take $\varepsilon_I = 4$, $\varepsilon_{II} = 80$, and $\kappa = 0$. Note that the surface-Generalized Born radii do not appreciably change as the discretization is refined. Similar results are obtained using the method of Ghosh *et al.* , or that presented by Wojciechowski and Lesyng (data not shown).

2.5.4 Continuum van der Waals Calculations

The surface-continuum van der Waals formulation has been implemented [43] and tested for four of the alanine dipeptide conformations presented by Scarsdale *et al.* [79]. Curved-element discretizations of the solvent-accessible surface were generated using OPLS all-atom radii [80] and a probe radius of 0.85Å, in accordance with the Levy *et al.* parameterization [8] for the TIP4P water model [81]. The Lennard-Jones coefficients for each

	METHOD						
		COARSE		MEDIUM		FINE	
PROBLEM	ANALYTICAL	DIRECT	FIT	DIRECT	FIT	DIRECT	FIT
Atom							
CAP	12.566	12.566	12.567	12.566	12.566	12.566	12.566
Parallel β dialanine							
PIT	18.719	18.719	18.720	18.719	18.719	18.719	18.719
BELT	77.565	77.565	–	77.565	–	77.565	–
CAP	145.358	145.358	145.340	145.358	145.354	145.358	145.358
Barnase–barstar complex							
PIT	2453.293	2453.240	2453.390	2453.292	2453.300	2453.293	2453.291
BELT	3195.626	3195.626	–	3195.626	–	3195.626	–
CAP	2620.158	2620.130	2619.698	2620.154	2620.056	2620.157	2620.137

Table 2.3: Comparison of pit, belt, and cap areas computed by analytical, direct quadrature, and polynomial-fitting methods, using the molecular surface discretizations of Section 2.5.1. All quantities are in $Å^2$ and have been rounded to the nearest 0.001 $Å^2$.



(a) (b)

Figure 2-8: Generalized Born radii calculated by boundary-element method solution of the linearized Poisson–Boltzmann equation (LPBE) and by evaluating surface integrals based on the GB model proposed by Grycuk [55]. The LPBE radii are plotted in solid lines with x and the GB radii are plotted with dashed lines and triangles, circles, dots, and stars. (a) Alpha-helix dialanine. (b) Beta-sheet dialanine.

surface integral of the form in Equation 2.9 are determined by appropriately mixing the well depths ε and the diameters σ for each OPLS atom type and the TIP4P water model. Table 2.4 shows the calculated energies at each discretization.

2.5.5 Electrostatics Problems

The electrostatic component of the solvation energy for several small boundary-element systems has been computed using the Yoon and Lenhoff integral formulation (Equations 2.1 and 2.2) and dense preconditioned GMRES [82]. Larger systems must be solved using fast, kernel-independent BEM algorithms such as the fast multipole method or FFTSVD [28,32, 83]. As described in Section 2.2.1, we have used piecewise-constant basis functions and centroid collocation. For all calculations, we assume that the solute region has $\varepsilon_I = 4$ and the solvent region has $\varepsilon_{II} = 80$.

Spherical Geometry

The solvation energy of a centrally-located charge in a spherical low-dielectric cavity can be computed analytically if the Laplace equation holds in the solvent region, or numerically using spherical harmonics if the linearized Poisson–Boltzmann equation holds in

	c5		αR		c7eq		c7ax	
Volume	-10.1365		-9.8917		-10.0190		-9.9199	
	# Elements	Energy	# Elements	Energy	# Elements	Energy	# Elements	Energy
Surface	429	-10.1369	486	-9.8918	357	-10.0193	421	-9.9201
	558	-10.1366	611	-9.8918	479	-10.0192	541	-9.9200
	901	-10.1365	1033	-9.8917	793	-10.0191	863	-9.9199
	1912	-10.1365	2069	-9.8917	1746	-10.0190	1782	-9.9199
	4877	-10.1365	5247	-9.8917	4245	-10.0190	4585	-9.9199
	10035	-10.1365	10829	-9.8917	10418	-10.0190	10755	-9.9199

Table 2.4: Solute–solvent van der Waals interaction energies estimated using a surface formulation of the Levy *et al.* continuum van der Waals model and curved surface elements. All energies are in kcal/mol and have been rounded to the nearest 10^{-4} kcal/mol.

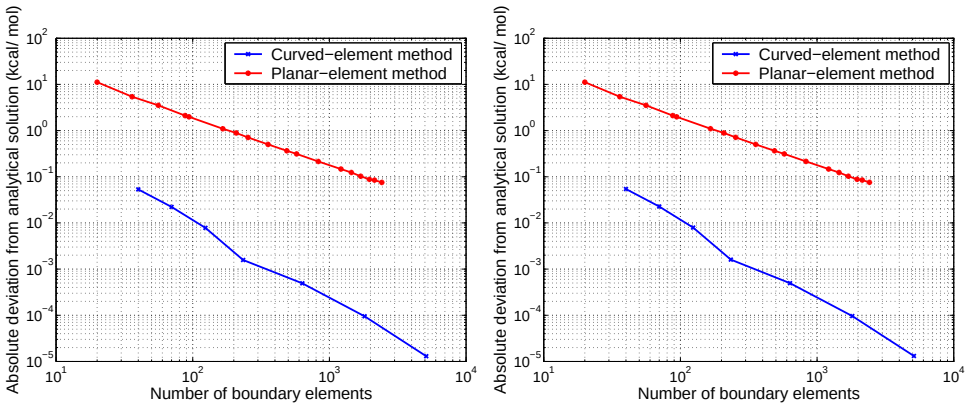


Figure 2-9: Convergence of solvation free energies for a centrally located charge in a 1 Å sphere, calculated by BEM numerical solution of the Yoon and Lenhoff integral equations. For both cases $\varepsilon_I = 4$ and $\varepsilon_{II} = 80$. (a) $\kappa = 0$ Å $^{-1}$. (b)$\kappa = 0.124$ Å $^{-1}$.

the solvent region. Figure 2-9 illustrates the improved accuracy of curved-element BEM relative to planar-element methods; Figure 2-9(a) plots convergence for non-ionic solutions (*i.e.*, $\kappa = 0$ Å $^{-1}$) and Figure 2-9(b) plots convergence to the analytical result when $\kappa = 0.124$ Å $^{-1}$.

Dialanine

CHARMM [58] with the CHARMM22 parameter set [11] was used to generate two conformations of dialanine (two alanine residues with an acetylated N-terminus and N-methylamide at the C-terminus). One conformation takes average ϕ and ψ angles for a parallel β-sheet ($\phi = -119^\circ$, $\psi = +113^\circ$); the other conformation takes the average angles for a right-handed α-helix ($\phi = -57^\circ$, $\psi = -47^\circ$) [84].

Alanine Dipeptide

Scarsdale *et al.* has presented energy-minimized atomic coordinates for several conformations of the alanine dipeptide [79]. A set of curved-element surface discretizations at varying refinement were generated using these coordinates, PARSE radii and partial charges, and a probe radius of 1.4 Å. Comparing the calculated planar-element and curved-element energies to their values at the finest discretizations, and plotting the absolute deviation as a function of the number of elements, one obtains Figure 2-10.

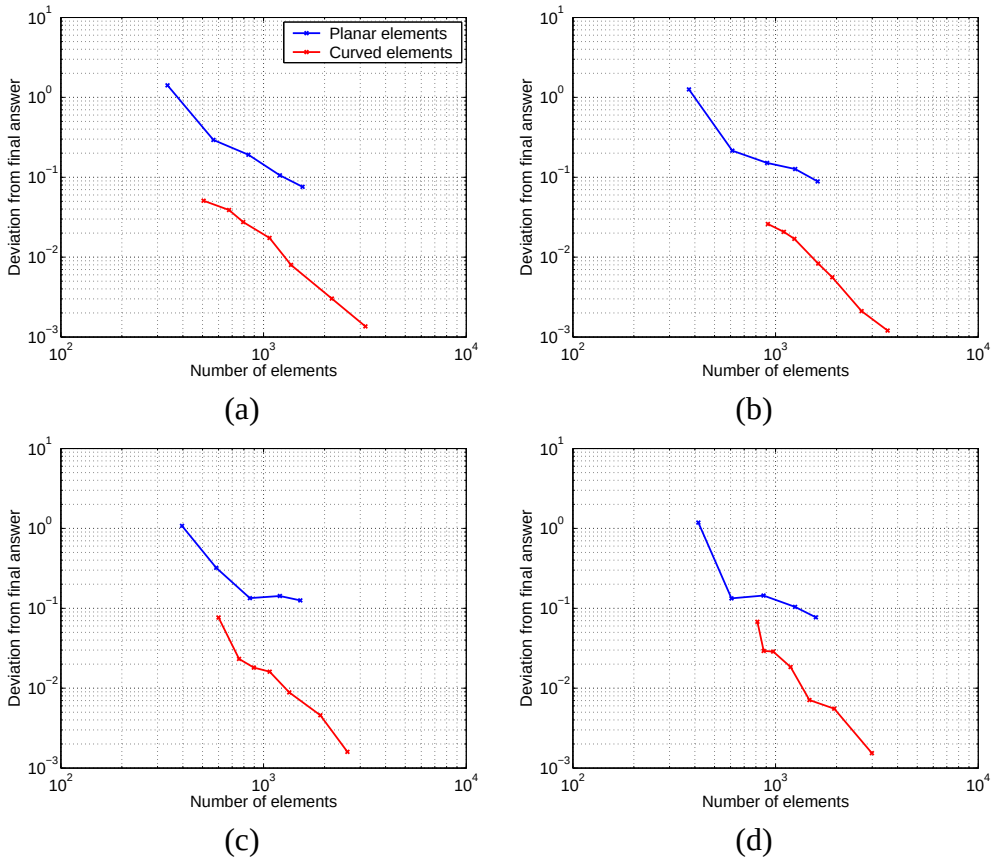


Figure 2-10: Solvation free energies for four conformers of the alanine dipeptide; atom centers are those presented in [79] and PARSE atomic radii and partial charges have been used [12]. (a) c5 geometry. (b) αR geometry. (c) c7ax geometry. (d) c7eq geometry.

2.6 DISCUSSION

We have defined two classes of compact, curved, two-dimensional surface elements that can be used to exactly describe arbitrary solute–solvent boundaries according to the most commonly used boundary definitions. These curved-element surface discretizations can be used in a number of surface formulations of biophysical modeling problems. To numerically evaluate the desired surface integrals over these domains, we have described a set of accurate, efficient techniques specialized for these domains. Computational results illustrate the advantages of curved-element surface discretizations relative to those based on planar triangles.

One significant advantage of the curved-element representations is that the geometry of the discretized surface does not change as the discretization is refined. In contrast, flat-element discretizations describe different boundaries at differing refinements, as do curved-element discretizations based on quadratic or cubic shapes. Curved-element methods based on our discretizations, however, are limited only by the accuracy of the integration method used, and, for boundary-element method problems, also by the order of the basis functions. The curved-element method presented here therefore offers an attractive approach for calculating Born radii via the SGB method and for computing solute–solvent van der Waals interactions using a continuum model. Furthermore, curved-element quadrature in the far-field is as efficient as far-field flat-element quadrature, because one can use the same order quadrature rules for both. As a result, problems that require the evaluation of many more far-field than near-field integrals can benefit significantly from curved-element methods without undue increase in computational expense. Finally, as a practical matter, the integration techniques presented in this work are straightforward to implement, requiring only a few hundred lines of MATLAB code, for example [75].

Although the near-field integration techniques for curved elements are significantly slower than those required for flat elements [45, 46], the extra accuracy afforded may be invaluable for problems that require highly accurate solutions. Because curved elements allow a significant reduction in the number of unknowns, such discretizations provide a promising approach to reach a target level of accuracy given constraints on computer mem-

ory. In the future we may extend these techniques to allow the evaluation of more complicated integrals, such as the potential induced by a polynomially-varying charge distribution on a curved element. Also, the curved-element discretization procedure may be modified to allow the production of coarser meshes.

Chapter 3

FFTSVD: A Fast Multiscale Boundary-Element Method Solver Suitable for Bio-MEMS and Biomolecule Simulation[1]

We present a fast boundary element method (BEM) algorithm that is well-suited for solving electrostatics problems that arise in traditional and Bio-MEMS design. The algorithm, FFTSVD, is Green's function independent for low-frequency kernels and efficient for inhomogeneous problems. FFTSVD is a multiscale algorithm that decomposes the problem domain using an octree and uses sampling to calculate low-rank approximations to dominant source distributions and responses. Long-range interactions at each length scale are computed using the FFT. Computational results illustrate that the FFTSVD algorithm performs better than precorrected-FFT style algorithms or the multipole style algorithms in FastCap.

[1]This chapter was published previously in a special issue of IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems [83] with J. P. Bardhan and M. D. Altman as joint first authors.

3.1 INTRODUCTION

Microelectromechanical systems (MEMS) have recently become a popular platform for biological experiments because they offer new avenues for investigating the structure and function of biological systems. Their chief advantages over traditional *in vitro* methods are reduced sample requirements, potentially improved detection sensitivity, and structures of approximately the same dimensions as the systems under investigation [85]. Devices have been presented for sorting cells [86], separating and sequencing DNA [87], and biomolecule detection [88]. Furthermore, because arrays of sensors can be batch fabricated on a single device, parallel experiments and high-throughput analysis are readily performed. However, since microfabrication is relatively slow and expensive, numerical simulation of MEMS devices is an essential component of the design process [89, 90]. Design tools for integrated circuits cannot address multiphysics problems, and this has motivated the development of several computer-aided MEMS design software packages, most of which are based on the finite-element method (FEM) and the boundary-element method (BEM) [91].

BioMEMS, when applied to such problems as biomolecule detection, are often functionalized with receptor molecules that bind targets of interest [92]. Molecular labels can also be used to aid in the detection process [93]. However, the interactions between these molecules, the MEMS device, and the solvent environment are often neglected during computational prototyping. In other fields, such as computational chemistry and chemical engineering, continuum models of solvation are often used to study the electrostatic component of these interactions [94]. These mean-field models permit the efficient calculation of many useful properties, including solvation energies and electrostatic fields [18, 20], and have been shown to correlate well with more expensive calculations that include explicit solvent [7]. However, continuum models are unable to resolve specific molecular interactions between solvent molecules and the solute. A variety of numerical techniques can be used to simulate the continuum models, including the finite-difference method (FDM), the finite-element method (FEM), and the boundary-element method (BEM) [21, 30, 95].

The boundary-element method has a number of advantages relative to FDM and FEM,

such as requiring only surface discretizations and exactly treating boundary conditions at infinity. However, the discretization of boundary integral equations produces dense linear systems whose memory costs scale as $O(n^2)$ and solution costs scale with $O(n^3)$, where n is the number of discretization unknowns. This rapid rise in cost with increasing problem complexity has motivated the development of accelerated BEM solvers. Preconditioned Krylov subspace techniques, combined with fast algorithms for computing matrix–vector (MV) products, can require as little as $O(n)$ memory and time to solve BEM problems [96]. Many such algorithms have been presented, including the fast multipole method (FMM) [97, 98], $\mathcal{H}$-matrices [99–101], the precorrected-FFT method [102], wavelet techniques [103, 104], FFT on multipoles [105, 106], kernel-independent multipole methods [107, 108], the hierarchical SVD method [109, 110], plane-wave expansion based approaches [111], and the PILOT algorithm [112]. Some algorithms, such as the original FMM, exploit the decay of the integral equation kernel; the precorrected-FFT method makes use of kernel shift-invariance. This work introduces an algorithm that combines the benefits of both of these approaches, leading to a method that has excellent memory and time efficiency even on highly inhomogeneous problems.

Fast BEM algorithms whose structures depend on kernel decay suffer from a common, well-known problem: computing medium- and long-range interactions is still expensive, even when their numerical low rank is exploited. For instance, in the fast multipole method, computing the M2L (multipole to local) products dominates the matrix–vector product time, since each cube can have as many as 124 or 189 interacting cubes, depending on the interaction list definition, and the work per M2L multiplication scales as $O(p^4)$, where p is the expansion order and is related to accuracy [97, 98, 113]. Much work has focused on reducing this cost; for the FMM, plane-wave expansions [111] diagonalize the M2L translation, but are typically only efficient for large p. The precorrected-FFT (pFFT) algorithm [102] relies on not the kernel's decay but rather its translation invariance to achieve high efficiency. The pFFT method is Green's function independent, even for highly oscillatory kernels. Consequently, the method has been applied in a number of different fields, including wide-band impedance extraction [114], microfluidics [72, 115, 116] and biomolecule electrostatics [117]. One weakness of the precorrected-FFT method is that its

efficiency decreases as the problem domain becomes increasingly inhomogeneous [102].

In this work, we introduce a fast BEM algorithm called FFTSVD. The method is well-suited to MEMS device simulation because it is Green's function independent and maintains high efficiency when solving inhomogeneous problems. The FFTSVD algorithm is similar to the PILOT algorithm introduced by Gope and Jandhyala [112], in that our algorithm is multiscale and based on an octree decomposition of the problem domain. Similar to PILOT and IES^3, our algorithm uses sampling and QR decomposition to calculate reduced representations for long-range interactions. The FFT is used to efficiently compute the interactions, as in the kernel-independent multipole method [108]. Numerical results from capacitance extraction problems demonstrate that FFTSVD is more memory efficient than FastCap or pFFT and that the algorithm does not have the homogeneity problem. In addition, we illustrate electrostatic force analysis by simulating a MEMS comb drive [72]. Finally, we demonstrate the method's kernel-independence by calculating the electrostatic free energy of transferring a small fluorescent molecule from the gas phase to aqueous solution, using an integral formulation of a popular continuum electrostatics model [21, 117].

The following section briefly describes a representative MEMS electrostatics problem, a boundary element method used to solve the problem, and a more complicated surface formulation for calculating the electrostatic component of the solvation energy of a biomolecule. Section 3.3 presents the FFTSVD algorithm. Computational results and performance comparisons appear in Section 3.4. Section 3.5 describes several algorithm variants and summarizes the chapter.

3.2 BACKGROUND EXAMPLES

In this section we describe two electrostatics problems that arise in BioMEMS design and describe how they can be addressed using BEM.

3.2.1 MEMS Electrostatic Force Calculation

Consider the electrostatically actuated MEMS comb drive illustrated in Figure 3-1. Two interdigitated polysilicon combs form the drive; one comb is fixed to the substrate and

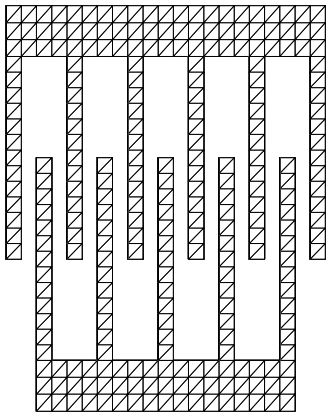


Figure 3-1: An electrostatically actuated MEMS comb drive.

the other is attached to a flexible tether. Applying a voltage difference to the two combs results in an electrostatic force between the two structures, and the tethered comb moves in response [72]. The electrostatic response of the system to an applied voltage difference can be calculated by solving the first-kind integral equation

$$\int_S \sigma(r')G(r;r')\,dA' = V(r), \tag{3.1}$$

where S is the union of the comb surfaces, $V(r)$ is the applied potential on the comb surfaces, $G(r;r') = 1/||r-r'||$ is the free-space Green's function, and $\sigma(r)$ is the charge density on the comb surfaces. Note that this is a standard capacitance extraction problem.

We can compute the axial electrostatic force between the combs by the relation

$$F(s) \;=\; -\frac{d}{ds}E = -\frac{d}{ds}\frac{1}{2}V^T C(s)V, \tag{3.2}$$

where $F(s)$ is the force in the axial direction, s is the separation between the combs, E is the electrostatic energy of the system, V is the vector of conductor potentials, and $C(s)$ is the capacitance matrix, written as a function of the comb separation.

To solve Equation 3.1 numerically,we discretize the surfaces into n_p panels and represent $\sigma(r)$, the charge density on the surface as a weighted combination of compactly

supported basis functions defined on the panels:

$$\sigma(r) = \sum_{i=1}^{n_p} x_i f_i(r). \tag{3.3}$$

Here, $f_i(r)$ is the i^{th} basis function and x_i the corresponding weight. Forcing the integral over the discretized surface to match the known potential at a set of collocation points, we form the dense linear system

$$Gx = b. \tag{3.4}$$

The Green's function matrix G is defined by

$$G_{ij} = \int f_j(r')G(r_i, r')\,da', \tag{3.5}$$

where r_i is the i^{th} collocation point and $b_i = V(r_i)$. Alternatively, one can use a Galerkin method, in which case

$$G_{ij} = \int\int f_i(r)\,f_j(r')G(r;r')\,dr\,dr' \tag{3.6}$$

and

$$b_i = \int f_i(r)\psi(r)\,dr. \tag{3.7}$$

The linear system of Equation 3.4 is solved using preconditioned GMRES [82].

3.2.2 BEM Simulation of Biomolecule Electrostatics

Electrostatic solvation energy, the cost of transferring a molecule from a nonpolar low dielectric medium to an aqueous solution with mobile ions, plays an important role in understanding molecular interactions and properties. To calculate solvation energy, continuum electrostatic models are commonly employed. Figure 3-2 illustrates one such model. The Richards molecular surface [49] is taken to define the boundary a that separates the biomolecule interior and the solvent exterior. The interior is modeled as a homogeneous region of low permittivity ε_I, where the potential $\varphi(r)$ is governed by the Poisson equation, and partial atomic charges on the biomolecule atoms are modeled as discrete point charges

at the atom centers:

$$\nabla^2\varphi(r) = -\sum_{i=1}^{n_c} \frac{q_i}{\varepsilon_I}\delta(r-r_i), \tag{3.8}$$

where n_c is the number of discrete point charges and q_i and r_i are the i^{th} charge's magnitude and location, respectively. In the solvent region, the linearized Poisson–Boltzmann equation

$$\nabla^2\varphi(r) = \kappa^2\varphi(r) \tag{3.9}$$

governs the potential, where κ, the inverse Debye screening length, depends on the concentration of ions in the solution and a higher permittivity ε_{II}. We write Green's theorem in the interior and exterior regions and then enforce continuity conditions at the boundary to produce a pair of coupled integral equations,

$$\begin{aligned} \frac{1}{2}\varphi(r_a) + \int\!\!\!\!\!\!-_a \varphi(r')\frac{\partial G_1}{\partial n}(r_a;r')dA' - \int\!\!\!\!\!\!-_a \frac{\partial \varphi}{\partial n}(r')G_1(r_a;r')dA' \\ = \sum_{i=1}^{n_c} \frac{q_i}{\varepsilon_I} G_1(r_a;r_i) \end{aligned} \tag{3.10}$$

$$\begin{aligned} \frac{1}{2}\varphi(r_a) - \int\!\!\!\!\!\!-_a \varphi(r')\frac{\partial G_2}{\partial n}(r_a;r')dA' + \frac{\varepsilon_I}{\varepsilon_{II}}\int\!\!\!\!\!\!-_a \frac{\partial \varphi}{\partial n}(r')G_2(r_a;r')dA' \\ = 0, \end{aligned} \tag{3.11}$$

where r_a is a point on the surface, $\int\!\!\!\!\!\!-$ denotes the Cauchy principal value integral, G_1 is the Laplace Green's function, G_2 is the real Helmholtz Green's function, $\frac{\partial G_i}{\partial n}$ denotes the appropriate double layer Green's function, $\varphi(r)$ is the potential on the surface, and $\frac{\partial \varphi}{\partial n}(r)$ is the normal derivative of the potential on the surface. Readers are referred to [21, 117] for detailed derivations of the formulation. To solve Equations 3.10 and 3.11 numerically we define a set of basis functions on the discretized surface and represent the surface potential and its normal derivative as weighted combinations of these basis functions:

$$\varphi(r) \approx \sum_i x_i f_i(r) \tag{3.12}$$

$$\frac{\partial \varphi}{\partial n}(r) \approx \sum_i y_i f_i(r). \tag{3.13}$$

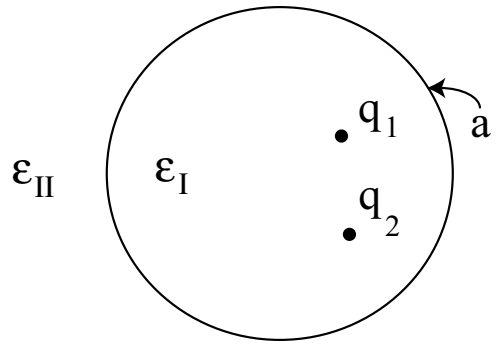


Figure 3-2: Continuum model for calculating biomolecule solvation.

We force the discretized integrals to exactly match the known surface conditions at the panel centroids; this produces the dense linear system

$$\begin{bmatrix} \frac{1}{2}I + \frac{\partial G_1}{\partial n} & -G_1 \\ \frac{1}{2}I - \frac{\partial G_2}{\partial n} & +\frac{\varepsilon_I}{\varepsilon_{II}}G_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \sum_k \frac{q_k}{\varepsilon_I} G_1(r;r_k) \\ 0 \end{bmatrix}, \tag{3.14}$$

where, denoting the i^{th} panel centroid as r_i, the block matrix entries are

$$G_{1,ij} = \oint f_j(r')G_1(r_i;r')dA' \tag{3.15}$$

$$\left(\frac{\partial G_1}{\partial n}\right)_{ij} = \oint f_j(r')\frac{\partial G_1}{\partial n(r')}(r_i;r')dA' \tag{3.16}$$

and the block matrices G_2 and $\frac{\partial G_2}{\partial n}$ are similarly defined. Note that boundary element method solution of this problem requires a Green's function independent fast algorithm.

3.3 FFTSVD ALGORITHM DETAILS

The FFTSVD is a multiscale algorithm like most fast algorithms for low frequency applications: to compute the total action of the integral operator on a vector, we separate its actions at different length scales and compute them separately, combining them only at the end. In describing the FFTSVD algorithm, it is helpful to think of the basis functions as sources, $\int f_i(r')G(r;r')dr'$ as the potential produced by source i, and the collocation points r_i as destinations. Multiplying x by G in Equation 3.4 is then computing potentials at all

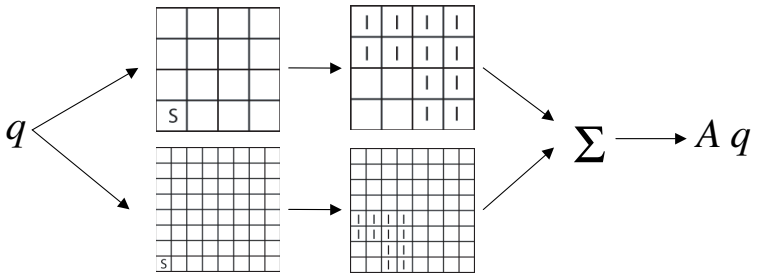


Figure 3-3: The multiscale approach to fast matrix multiplication.

the destinations due to all sources. Figure 3-3 illustrates the multiscale approach to fast matrix multiplication: the square *S* denotes a source, and the squares denoted *I* represent destinations.

3.3.1 Notation

Let *d* and *s* denote two sets of panels: then $G_{d,s}$ is the submatrix of *G* that maps sources in *s* to responses in *d*. The number of panels in set *i* is denoted by n_i.

3.3.2 Octree Decomposition

We first define the problem domain to be the union of all the sets of panels that comprise the discretized surfaces. We then place a bounding cube around the domain and recursively decompose the cube using octrees. Given a cube *s* at level *i*, the *nearest neighbors* N_s are those cubes at level *i* that share a face, edge, or vertex with *s*. The *interaction list* for *s* is denoted as I_s and defined to be the set of cubes at level *i* that are not nearest neighbors to *s* and not descended from any cube in an interaction list of an ancestor of *s* [118]. Figure 3-4 illustrates the exclusion process for a 2-D domain. At every level, each panel is assigned to the cube that contains its centroid. Where ambiguity will not result, *s* denotes either the cube itself or the set of panels assigned to it. This assignment rule ensures that each panel–panel interaction is treated exactly once.

The coarsest decomposition is termed level 0 and has 4^3 cubes; coarser decompositions have null interaction lists. We continue decomposing the domain until we reach a level *l* at which no cube is assigned more than $n_{p,max}$ destinations. At each level *i*, every cube *s* has

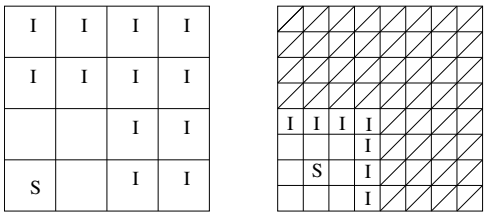


Figure 3-4: Interacting squares at two levels of decomposition.

a set of interacting cubes I_s that are well-separated from s with respect to the current cube size. Note that the definition of an interaction list is symmetric: $d \in I_s \rightarrow s \in I_d$.

3.3.3 Sampling Dominant Sources and Responses

One can compute the potential response φ_{I_s} in I_s due to a source q_s in s by the dense matrix-vector product

$$\begin{aligned} \varphi_{I_s} &= G_{I_s,s} q_s \\ G_{I_s,s} &\in \Re^{n_{I_s} \times n_s}. \end{aligned} \tag{3.17}$$

However, the separation between s and I_s motivates the approximation

$$\begin{aligned} G_{I_s,s} &\approx U_{I_s} V_{s,\mathrm{src}}^T \\ U_{I_s} &\in \Re^{n_{I_s} \times k} \\ V_{s,\mathrm{src}}^T &\in \Re^{k \times n_s} \\ k &\ll n_{I_s} \end{aligned} \tag{3.18}$$

where $V_{s,\mathrm{src}}$ has orthogonal columns [109]. The matrix $V_{s,\mathrm{src}}$ is small and represents the k source distributions in s that produce dominant effects in I_s. It is a reduced row basis for $G_{I_s,s}$. The projection of q_s onto $V_{s,\mathrm{src}}$ loosely parallels the fast multipole method's calculation of multipoles from sources, in the sense that both the multipole expansion and the product $V_{s,\mathrm{src}}^T q_s$ capture the important pieces of q_s when calculating far-field interactions. We call $V_{s,\mathrm{src}}$ the source compression matrix.

A similar low-rank approximation can be made to find the response in a cube d given a source distribution in I_d:

$$\begin{aligned} \varphi_d &= G_{d,I_d} q_{I_d} \\ &\approx U_{d,\text{dest}} V_{I_d}^T q_{I_d} \\ U_{d,\text{dest}} &\in \Re^{n_d \times k} \\ V_{I_d}^T &\in \Re^{k \times n_{I_d}} \\ k &\ll n_{I_d}. \end{aligned} \tag{3.19}$$

Here, $U_{d,\text{dest}}$ is small and represents the k dominant potential responses in d, the destination cube, due to source distributions in I_d. We call $U_{d,\text{dest}}$ the destination compression matrix; $U_{d,\text{dest}}$ is a reduced column basis for G_{d,I_d}.

Since it is impractical to compute $G_{I_s,s}$ and G_{s,I_s} for each cube s, we use a sampling procedure inspired by the Kapur and Long hierarchical SVD method [109]. Figures 3-5 and 3-6 illustrate the process of finding a reduced row basis $V_{s,\text{src}}$. To determine the row basis, we begin by selecting one destination per interacting cube, computing the corresponding rows of $G_{I_s,s}$, and performing rank-revealing QR factorization with reorthogonalization on the transpose of the submatrix. If the submatrix rank is less than half the number of sampled destinations, the QR-determined row basis is considered to be adequate. Otherwise, an additional destination is sampled for each interacting cube; the extra destination is chosen to be well-separated from the originally chosen destination. The transpose of the new submatrix is factorized and again required to have rank less than half the total number of samples. The process of resampling is continued until the required rank threshold is met.

To compute the reduced column basis $U_{d,\text{dest}}$ for the matrix G_{d,I_d}, we select a set of well separated panels in I_d, compute the corresponding columns of G_{d,I_d}, and QR factorize the submatrix.

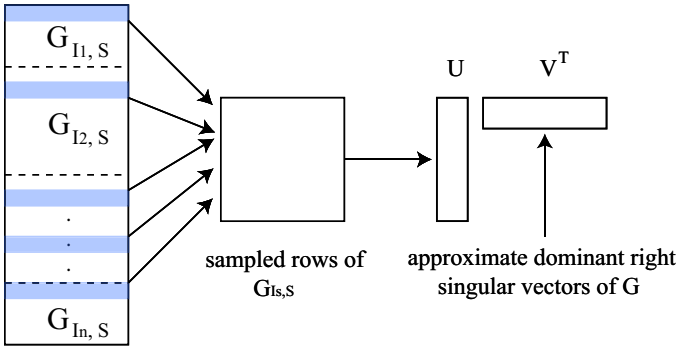


Figure 3-5: Computing dominant row basis for $G_{I_s,S}$ using sampling.

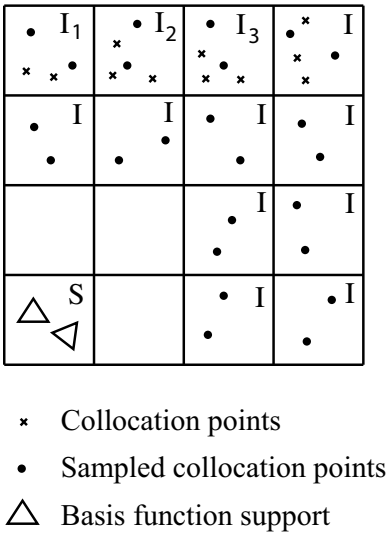


Figure 3-6: Sampling a small set of long-range interactions.

3.3.4 Computing Long-range Interactions

Consider two well separated cubes s and d. Because the cubes are well separated, we could find a low-rank approximation to $G_{d,s}$ by truncating its SVD:

$$\begin{aligned} \varphi_d &= G_{d,s} q_s && (3.20) \\ &= U_{d,s} \Sigma_{d,s} V_{d,s}^T q_s && (3.21) \\ &\approx \hat{U}_{d,s} \hat{\Sigma}_{d,s} \hat{V}_{d,s}^T q_s && (3.22) \end{aligned}$$

where the hat denotes trunctation to k columns, $k < n_s$. Since the source compression matrix $V_{s,\mathrm{src}}$ finds an approximation to the dominant row space of $G_{I_s,s}$, we expect that it also approximates the dominant row space of $G_{d,s}$, which is a submatrix of $G_{I_s,s}$. Similarly, we expect that $U_{d,\mathrm{dest}}$ approximates the dominant column space of $G_{d,s}$. A small matrix $K_{d,s}$ maps source distributions in the reduced basis $V_{s,\mathrm{src}}$ to responses in the reduced basis $U_{d,\mathrm{dest}}$:

$$\varphi_d = U_{d,\mathrm{dest}} K_{d,s} V_{s,\mathrm{src}}^T q_s, \tag{3.23}$$

and it is easy to see that

$$K_{d,s} = U_{d,\mathrm{dest}}^T G_{d,s} V_{s,\mathrm{src}}. \tag{3.24}$$

Note that $K_{d,s}$ is not diagonal because $U_{d,\mathrm{dest}}$ and $V_{s,\mathrm{src}}$ only approximate the singular vectors of $G_{d,s}$. If $V_{s,\mathrm{src}} \in \Re^{n_s \times k_s}$ and $U_{d,\mathrm{dest}} \in \Re^{n_d \times k_d}$, then $K_{d,s} \in \Re^{k_d \times k_s}$.

The action of the K matrices can be computed in a number of different ways: they can be computed explicitly, via multipoles, or via an FFT. Explicit storage is memory intensive, and multipole representations are Green's function dependent. We have therefore chosen to implement the memory-efficient, Green's function independent FFT translation method presented by Ying *et al.* [108].

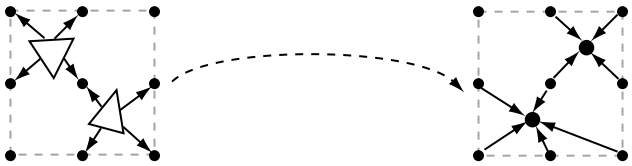


Figure 3-7: Schematic of the FFTSVD method for computing long-range interactions.

3.3.5 Diagonalizing Long-range Interactions with the FFT

Our method projects sources to a grid, uses an FFT convolution to accomplish translation between source and destination, and interpolates results back from the grid. Figure 3-7 illustrates the approach. We introduce two matrices: $P_{g,j}$ projects sources in cube j to the cube grid, and $I_{j,g}$ interpolates from the grid in cube j to the evaluation points in j. We use an equivalent density scheme similar to those used by Phillips and White [102] and Biros *et al.* [107] to determine the projection and interpolation matrices.

Projection Matrix Calculation

Given a cube s and the basis function weights q_s for panels in s, we wish to find a set of grid charges $q_{g,s}$ that reproduce the potential field far from s. We accomplish this by defining a sphere Γ bounding s and picking a set of quadrature points [119] on the sphere. Denoting quadrature point i on Γ by $r_{\Gamma,i}$, the mapping between q_s and the responses at the quadrature points can be written as $G_{\Gamma,s}$, where

$$G_{\Gamma,s,ij} = \int_{\text{panel } j} G(r_{\Gamma,i}; r')dr'. \tag{3.25}$$

The mapping between grid charges and responses at the quadrature points can be written as

$$G_{\Gamma,g,ij} = G(r_{\Gamma,i}, r_{g,j}) \tag{3.26}$$

where $r_{g,j}$ is the position of the j^{th} grid point. If more quadrature points than grid points are used for the matching, solving a least squares problem gives the desired projection $P_{g,s}$:

$$P_{g,s} = G_{\Gamma,g}^{-1} G_{\Gamma,s}. \tag{3.27}$$

In practice, one uses the singular value decomposition to solve for $P_{g,s}$.

Interpolation Matrix Calculation

Given grid potentials q_d in a cube d, we find the potentials φ_d at the panel centroids in d by interpolation. For problems in which centroid collocation is used to generate a linear system of equations, the interpolation matrix is calculated as

$$I_{d,g} = (G_{\Gamma,g}^{-1} G_{\Gamma,d})^T \tag{3.28}$$

where $G_{\Gamma,d}$ denotes the Green's function matrix from the quadrature points on Γ to the panel centroids in d. If Galerkin methods are used rather than centroid collocation, the interpolation matrix is the transpose of the projection matrix.

Diagonal Translation

Once the grid charges in s are known, a spatial convolution with the Green's function produces the potentials at the grid points in the destination cube d. This spatial convolution is diagonalized by the Fourier transform; we write the transform matrix as $\mathcal{F}$, its inverse by $\mathcal{F}^{-1}$, and the transform of the Green's function matrix by $\tilde{G}_{d,s}$. After calculating the grid potentials in d, interpolation produces the potentials at the desired evaluation points. The matrix $G_{d,s}$ is therefore written as

$$G_{d,s} = I_{d,g} \mathcal{F}^{-1} \tilde{G}_{d,s} \mathcal{F} P_{g,s}. \tag{3.29}$$

The products $I_{d,g}\mathcal{F}^{-1}$ and $\mathcal{F}P_{g,s}$ could be stored, but in our experience this precomputation only marginally improves the matrix–vector product time while increasing memory use since $\mathcal{F}$ and $\mathcal{F}^{-1}$ are padded and complex.

In addition to diagonalizing the translation operation between cubes, the FFT significantly decreases memory requirements. Using explicit K matrices requires storing a small dense matrix for each pair of cubes; using FFT translation eliminates the expensive per-pair matrix cost. Instead, each cube has its own P_g and I_g matrices, which are used for all

long-range interactions. In addition, because the Green's function is translationally invariant, we only need to store a small number of $\tilde{G}$ matrices for each octree level; each one represents a particular relative translation between source and destination cubes. Because these matrices are diagonal, storage requirements are minimal.

Since translation is the dominant cost in the FFTSVD matrix–vector product, efficient implementation of the translation procedure is essential to maximizing performance. The translation operation is simply an element-wise multiplication of two complex vectors, therefore, for g_p grid points per cube side, each translation vector is $(2g_p-1)^2[(2g_p-1)/2+1]$ complex numbers long when using the FFTW library [120]. This number takes into account padding and symmetry. For example, with $g_p = 3$, 75 complex numbers are required, resulting in 250 individual multiplies during the translation operation. This number has been reduced by taking advantage of vectorization. Many modern CPUs include instructions that can assist in multiplying complex numbers within a register, effectively halving the number of required multiplies. For comparison, standard fast multipole method translations require more multiplications since they are not diagonal, and cannot be vectorized as easily since they involve matrix–vector products. In addition, we have yet to exploit additional ways to accelerate the FFTSVD translation operation. These include using symmetries between related translation vectors ($\tilde{G}$), such as those that translate in opposite directions, and exploiting the fact that for axial translations, many $\tilde{G}$ elements are purely real.

3.3.6 Local Interactions

At the finest level of the decomposition, interactions between nearest neighbor cubes are computed directly by calculating the corresponding dense submatrices of G. These submatrices are denoted by $D_{i,j}$ where j is the source cube and i the destination. We bound the complexity of the local interaction computation by continuing the octree decomposition until each cube has fewer than $n_{p,\max}$ panels.

3.3.7 Algorithm Detail

The mapping from source cube s to destination cube d can thus be written as

$$\varphi_d = U_d \left(U_d^T I_{d,g}\right) \mathcal{F}^{-1} \tilde{G} \mathcal{F} \left(P_{g,s} V_s\right) V_s^T q_s \tag{3.30}$$

The computations are grouped to eliminate redundant multiplications; the matrix products $U_d^T I_{d,g}$ and $P_{g,s} V_s$ are stored for each cube rather than recomputed at every iteration. Below, we introduce the restriction operator $M_j^{(i)}$ that restricts a global vector to a local vector associated with cube j at level i; let the inverse operator map a local vector to the global by inserting appropriate zeros. Let L^i denote the set of cubes at level i. Given a charge vector q, the matrix–vector product is computed by the following procedure:

1. DOWNWARD PASS FOR LONG-RANGE INTERACTIONS: For levels $i = 0, 1, \ldots, l$:

   (a) PROJECT INTO DOMINANT SOURCE SPACE: For each cube $j \in L^i$, compute

   $$\zeta_j = \mathcal{F} \left(P_{g,j} V_{j,\mathrm{src}}\right) V_{j,\mathrm{src}}^T M_j^{(i)} q. \tag{3.31}$$

   (b) COMPUTE LONG-RANGE INTERACTIONS: For each cube $j \in L^i$, compute

   $$\nu_j = \sum_{s \in I_j} \tilde{G} \zeta_s. \tag{3.32}$$

   (c) DETERMINE TOTAL DOMINANT RESPONSE: For each cube $j \in L^i$, compute

   $$\varphi = \varphi + M_j^{(i),-1} U_{j,\mathrm{dest}} \left(U_{j,\mathrm{dest}}^T I_{j,g}\right) \mathcal{F}^{-1} \nu_j. \tag{3.33}$$

2. SUM DIRECT INTERACTIONS: For each cube d at level l, add the contributions from neighboring cubes N_d:

   $$\varphi = \varphi + M_d^{(l),-1} \sum_{s \in N_d} D_{d,s} M_s^{(l)} q. \tag{3.34}$$

3.4 COMPUTATIONAL RESULTS

To demonstrate the accuracy, speed, and memory efficiency of the FFTSVD algorithm, we have used FFTSVD to solve for self and mutual capacitances in various geometries. A MEMS comb drive example [72] illustrates electrostatic force calculation using FFTSVD. In addition, to show Green's function independence and use of double layer kernels, we have used FFTSVD to solve for the electrostatics of solvation for the highly charged dye molecule fluorescein. Fluorescein is often used as a fluorescent label in BioMEMS applications [121,122], and its electrostatic properties in aqueous solution modulate its interaction with other molecules and surfaces.

The FFTSVD algorithm has several adjustable parameters: ε_{QR} is the reduced basis tolerance; g_p is the number of FFT grid points on each side of a finest-level cube; $n_{p,\max}$ is the maximum number of panels in a finest-level cube; n_{quad} is the number of quadrature points used on the equivalent density sphere, tol_{GMRES} is the tolerance on the relative residual that the resulting linear equations are solved to. At the two finest levels, g_p FFT grid points per cube edge are used, and the number of grid points per edge increases by one for each successively coarser level; experience has shown that using different numbers of grid points per edge provides significant accuracy improvements for marginal memory and time costs. The parameters used for the following results are 10^{-4} for ε_{QR}, 3 for g_p, 32 for $n_{p,\max}$, 25 for n_{quad}, and 10^{-4} for tol_{GMRES} unless otherwise specified.

For capacitance calculations, we compare performance to FastCap, based on the fast multipole method [113], and fftcap++, based on the pFFT++ implementation of the pre-corrected-FFT method [123]. All programs were compiled with full optimizations using the Intel C++ compiler version 8.1 and benchmarked on an Intel Pentium 4 3.0-GHz desktop computer with 2 GB of RAM. All parameter settings in FastCap and fftcap++ were left at their defaults, except for the tolerance on solving the resulting linear equations, which was set to 10^{-4} unless otherwise specified.

3.4.1 Self-Capacitance of a Sphere

In order to test the accuracy of the FFTSVD method, we have applied it to solving for the self-capacitance of a unit 1-m radius sphere, a quantity known analytically. Figure 3-8 shows the improvement in accuracy with increasing sphere discretization for FFTSVD with values of 3 and 5 for g_p, 2nd and 4th order multipoles in FastCap, and default settings for fftcap++. A tolerance of 10^{-6} for the relative residual when solving the BEM equations was used in all programs. The analytical value for the self-capacitance of a 1-m radius sphere is 0.111265 nF as computed by Gauss' law. The results show that FFTSVD with a value of 3 for g_p tends to be more accurate than 2nd order multipoles in FastCap. In addition, FFTSVD with low values of g_p tends to overshoot the analytical solution while FastCap tends to undershoot with truncation of multipole order. These findings are consistent across many geometries when examining convergence behavior.

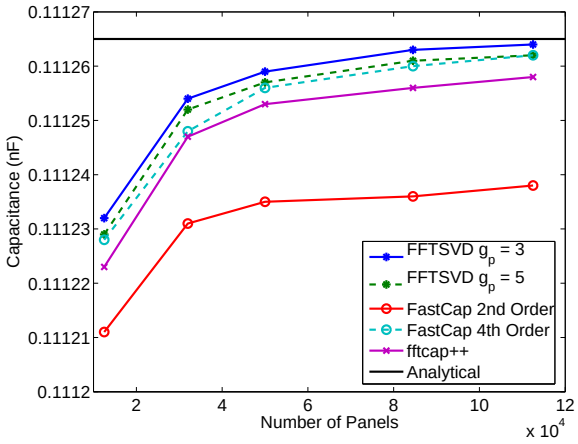


Figure 3-8: Accuracy versus number of panels for FFTSVD, FastCap and fftcap++ solving the unit sphere self-capacitance problem.

3.4.2 Woven Bus Example (Homogeneous Problem)

As stated previously, one of the advantages of the FFTSVD method is its use of diagonal translation operators. This advantage becomes apparent in cases of homogeneous geometry, since a large number of translation operations are required. To examine performance in a problem with homogeneous geometry, we have applied FFTSVD to solving for the mu-

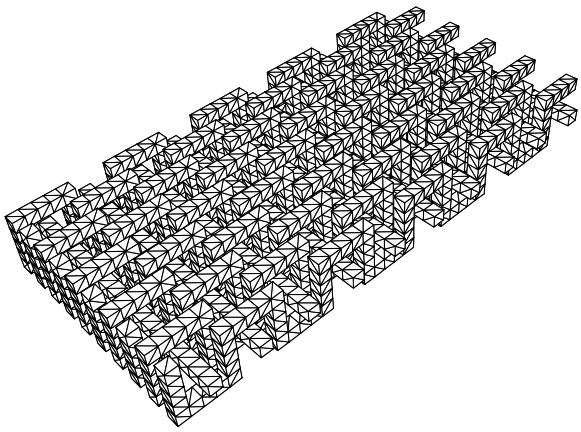


Figure 3-9: Homogeneous woven bus capacitance problem (woven10n01).

tual capacitances between woven bus conductors as in Figure 3-9. Table 3.1 summarizes the results for several woven bus capacitance problems. FFTSVD can achieve slightly better speed and memory performance than precorrected-FFT, which is expected to excel at problems with uniform distribution, and significantly better performance as compared to FastCap.

3.4.3 Inhomogeneous Capacitance Problem

One of the disadvantages of the precorrected-FFT method is that it lays down a uniform grid over the entire problem domain, and the simulation time grows roughly in proportion to the number of grid points. For simulations in which most of the domain is empty, therefore, the precorrected-FFT algorithm is inefficient. We have demonstrated this inefficiency, and FFTSVD's relative advantage, by configuring a set of conductors as shown in Figure 3-10. Almost all of the panels in this system are at the edges of a cube bounding the domain. Figure 3-11 plots the matrix–vector product times for the FFTSVD, FastCap and fftcap++ codes, and Figure 3-12 plots the memory requirements. As expected, the precorrected-FFT based fftcap++ code has poor performance, especially for fine discretizations of the inhomogeneous problem. FFTSVD performs consistently better than fftcap++ and generally better than FastCap. The sharp jumps in FFTSVD and fftcap++ matrix–vector product time with increasing panel count are due to a change in selection of the optimal octree

Table 3.1: Comparison of FastCap (FC), fftcap++ (FFT++) and FFTSVD (FS) performance in terms of matrix–vector product time (MV) and memory usage (MEM) on homogeneous woven bus capacitance problems with 2, 5 and 10 crossings (woven02n03, woven05n03, woven10n03) and 10 crossings with lower discretization (woven10n01).

Problem	Panels	FC MV (s)	FC MEM (MB)	FFT++ MV (s)	FFT++ MEM (MB)	FS MV (s)	FS MEM (MB)
woven02n03	3168	0.03	30	0.02	23	0.01	11
woven05n03	18720	0.17	205	0.22	411	0.09	110
woven10n01	8160	0.08	89	0.04	69	0.04	41
woven10n03	73440	0.73	901	0.51	818	0.41	466

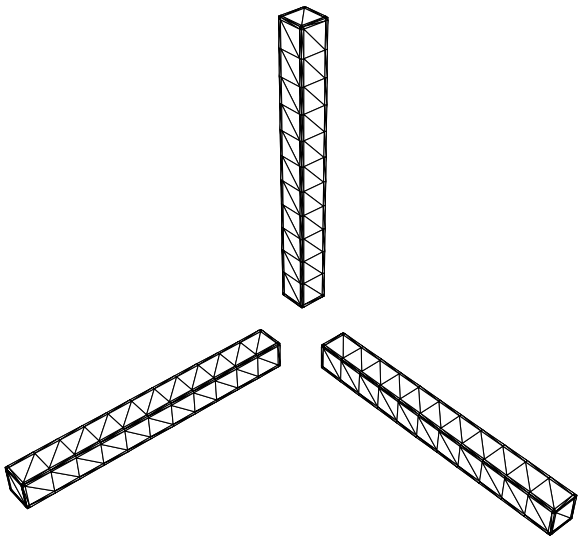


Figure 3-10: Inhomogeneous capacitance problem.

decomposition depth or FFT grid size, respectively.

3.4.4 MEMS Comb Drive

We have simulated the MEMS comb drive illustrated in Figure 3-1 [72]. We applied a voltage difference of 1 V to the two structures and used a fourth-order finite difference scheme to approximate the derivative in Equation 3.2. Because the finite-difference scheme for force calculation requires high accuracy in the capacitance calculations, more stringent parameters are required for these simulations. We have used $tol_{GMRES} = 10^{-6}$, $\varepsilon_{QR} = 10^{-6}$, $g_p = 5$, $n_{QUAD} = 64$, and for each discretization we have fixed $n_{p,max}$ such that the octree decomposition depth is equal for each of the four geometries.

The contribution of each panel to the axial force is plotted in Figure 3-13 and the total axial electrostatic force is plotted in Figure 3-14 as a function of the number of panels used to discretize the comb drive. We have used general triangles and note that the discretization scheme is poorly tuned for the calculation of electrostatic forces; nonuniform meshes achieve superior accuracy at reduced panel counts [124]. The force can also be calculated by integrating the squared charge density over the conductor surface, but this approach requires specialized treatment because the charge density becomes infinite at the edges and

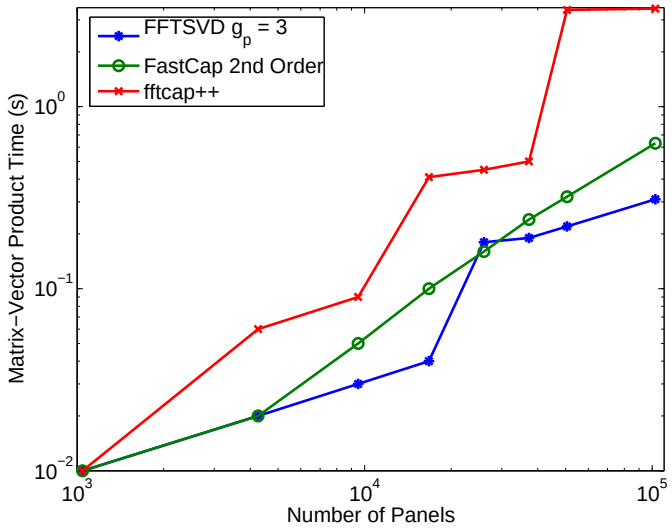


Figure 3-11: Matrix-vector product times for FFTSVD, FastCap and fftcap++ codes solving the inhomogeneous capacitance problem.

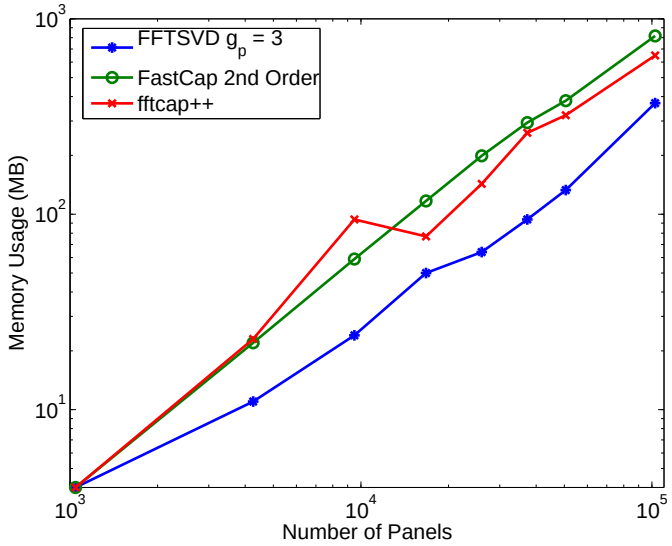


Figure 3-12: Memory requirements for FFTSVD, FastCap and fftcap++ codes solving the inhomogeneous capacitance problem.

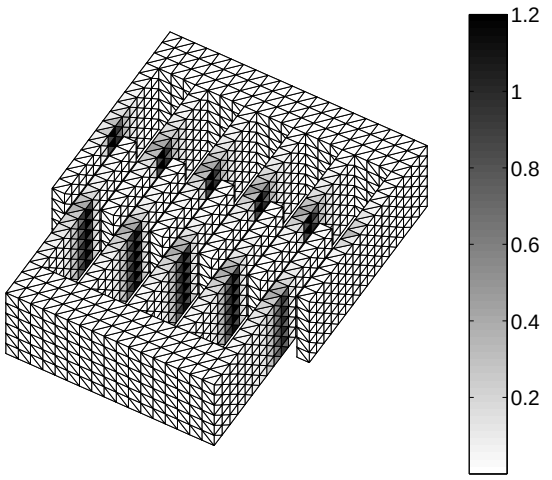


Figure 3-13: Magnitudes of panel contributions to the axial electrostatic force. Units are pN.

corners of the conductors [125, 126].

3.4.5 Solvation of Fluorescein

We have used the integral formulation in Equations 3.10 and 3.11 to calculate the solvation energy of fluorescein. To prepare a model for solvation calculations, its structure and partial atomic charges were determined from quantum mechanical calculations. Radii were assigned to each atom and used to generate a triangulation of the molecular surface. The interior of the fluorescein molecule was assigned a dielectric constant of 4, and the exterior was assigned a dielectric constant of 80 (for water) with an ionic strength of 0.145 M ($\kappa = 0.124\ \AA^{-1}$). FFTSVD was used to solve for both the electrostatic solvation energy (Figure 3-15), as well as the total electrostatic potential on the surface of the fluorescein molecule (Figure 3-16). We note that the long-range single- and double-layer integrals can be computed using only one set of of translation operations. Different projection operators are used to find the corresponding grid charges due to monopole and dipole distributions, and the grid charges can then be summed for translation.

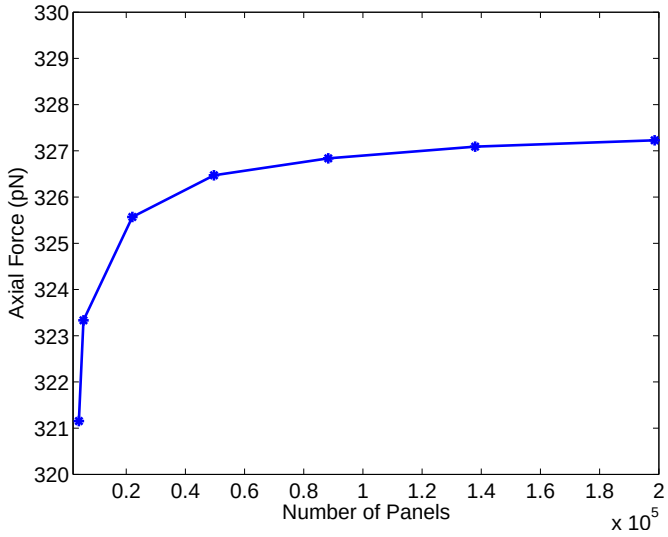


Figure 3-14: Calculated total axial electrostatic force on one comb.

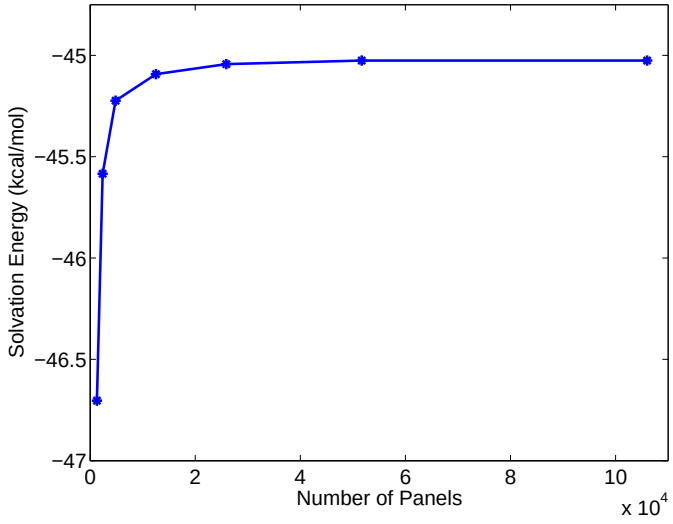


Figure 3-15: Computed electrostatic solvation energy of fluorescein with increasing problem discretization.

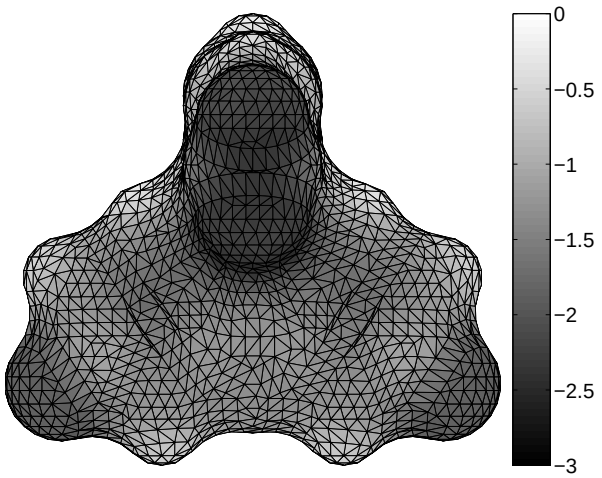


Figure 3-16: Electrostatic solvation potentials on the molecular surface of fluorescein. Units are kcal mol^{-1} e^{-1}.

3.5 DISCUSSION

3.5.1 Algorithm Variants

For problems with a small number of integral operators, memory constraints may not be a significant consideration. In these cases, the matrices $K_{d,s}$ can be stored explicitly. These $K_{d,s}$ matrices are computed using Equation 3.24, but instead of computing $G_{d,s}$ explicitly, we project, translate and interpolate an identity matrix using the methodology outlined in Section 3.3.5. Although setup time and memory use increase when explicit K-matrices are used, the matrix–vector product time is significantly reduced. We have also implemented a parameter that allows a tradeoff between speed and memory use through K-matrices. Pairs of interacting octree cubes that contain fewer panels than the parameter are handled with explicit K-matrices, while all other cubes use the FFT-based translation. In this manner, the balance between speed and memory can be fine-tuned for the given application.

It is also straightforward to create an FFTSVD variant that runs in linear time; the same method used to generate the projection and interpolation matrices can be used to create "upward pass" and "downward pass" operators such as those found in multipole algorithms. This variant algorithm is essentially equivalent to the kernel-independent method by Ying *et al.* [108], except that we allow all the grid charges to be nonzero. The Ying method, in

contrast, uses only grid charges on the surface of the cube.

The linear-time FFTSVD method requires a greater number of grid points per cube, due to the loss of degrees of freedom during each upward pass from child to parent cube. In addition, the SVD based compression of dominant sources and responses is no longer computed, since these bases are now taken directly from child cubes. This method is extremely memory efficient since dominant source and response bases are no longer stored, but it trades off performance to achieve it due to the larger required grid sizes.

Finally, the multilevel structure of FFTSVD allows easy parallelization. Each processor can be assigned responsibility for a set of cubes on coarse levels, and the computation can proceed independently until the final potential responses are summed. We have implemented parallel FFTSVD using both OpenMP and MPI libraries with good results.

3.5.2 Summary

We have developed a fast algorithm for computing the dense matrix–vector products required to solve boundary-element problems using Krylov subspace iterative methods. The FFTSVD method is a multiscale algorithm; an octree decomposes the matrix action into different length scales. For each length scale, we use sampling to calculate reduced bases for the interactions between well-separated groups of panels. The FFT is used to diagonalize the translation operation that computes the long-range interactions. The method described here relies on both kernel decay and translation invariance.

Numerical results illustrate that FFTSVD is much more memory-efficient than FastCap or precorrected-FFT, and that it is generally faster than either technique on a variety of problems. In addition, FFTSVD is Green's-function independent, unlike FastCap, and the method performs well even when the problem domain is sparsely populated, unlike precorrected-FFT. Our implementation is well-suited to solve problems with multiple dielectric regions. Finally, we note that the structure of the algorithm permits treatment of kernels that are not translation-invariant; for such problems,the K-matrix algorithm variant should be used rather than the FFT. Together, the algorithm's performance and flexibility make FFTSVD an excellent candidate for fast BEM solvers for microfluidic and micro-

electromechanical problems that appear in BioMEMS design.

Chapter 4

Accurate Solution of Multi-region Continuum Electrostatic Problems Using the Linearized Poisson–Boltzmann Equation and Curved Boundary Elements[1]

We present a boundary-element method (BEM) implementation for solving problems in biomolecular electrostatics using the linearized Poisson–Boltzmann equation. The motivating factor behind this implementation was the desire to create an efficient and accurate solver capable of precisely describing the molecular topologies prevalent in continuum models. Underlying this implementation are three key features that address many of the well-known practical challenges associated with the boundary-element method. First, we present a general boundary-integral approach capable of modeling an arbitrary number of embedded homogeneous dielectric regions with differing dielectric constants, possible salt treatment, and point charges. Second, molecular and accessible surfaces used to describe dielectric and ion-exclusion boundaries are discretized with curved boundary elements that

[1] To be submitted [50].

faithfully reproduce even complicated geometries. Robust numerical integration methods are employed to accurately evaluate singular and near-singular integrals over the curved boundary elements. Third, we avoid explicitly forming the dense BEM matrix, and instead solve the linear system with preconditioned GMRES, using the FFTSVD algorithm to accelerate matrix–vector multiplication. A comparison of the presented BEM implementation and standard finite-difference techniques demonstrates that for certain classes of electrostatic calculations, the improved convergence properties of the BEM approach can have a significant impact on computed energetics. These results suggest that solvers with improved accuracy may be important to ensure that predictions based on continuum models are limited by the models themselves rather than by errors in the models' numerical evaluation.

4.1 INTRODUCTION

Continuum theories of solvation have become common tools for molecular modeling, and have led to an improved understanding of electrostatic interactions in biomolecular systems [9, 94]. One of the most popular models of continuum solvation treats a molecule and its solvent environment as homogeneous regions of low and high dielectric constants respectively, with embedded point charges representing the molecular charge distribution and Debye–Hückel theory modeling the effect of salt. The linearized Poisson–Boltzmann equation governs this continuum model, and this equation has received much attention in recent years [6, 10, 127]. The linearized Poisson–Boltzmann equation (LPBE), an elliptic partial differential equation (PDE) [128], is well understood theoretically and can be solved numerically using a variety of techniques including finite-difference methods (FDM) [20, 22, 26, 31, 95, 129–131] finite-element methods (FEM) [25, 30, 132], and boundary-element methods (BEM) [17, 19, 23, 27, 28, 32, 52, 69, 70, 133–135].

Boundary-element methods offer several inherent advantages over volume-based methods for solving the LPBE with regions of homogeneous dielectric [54]. For example, the BEM only requires discretization of problem boundaries rather than the entire infinite domain, and inherently captures the correct zero-potential boundary condition at infinity. In

comparison to finite-difference methods, BEM has the ability to model point charges exactly rather than requiring grid projection.

Unfortunately, boundary-element methods require sophisticated numerical techniques in order to be competitive with the flexibility and performance of volume-based methods. Several challenges complicate the implementation of BEM techniques for biomolecule electrostatics. The first challenge arises from the surface-based analysis of the problem. Some Poisson–Boltzmann modeling problems require treatment of multiple embedded or disconnected regions with differing dielectric constants and screening parameters [136, 137]. These features allow the simulation of solvent-filled cavities within macromolecules, salt-filled regions in large cavities, and an ion-exclusion layer surrounding the molecule with solvent permittivity but no salt. Multiple regions are easily modeled in volume methods like FDM and FEM because the dielectric constant and the presence of salt can be assigned to each grid point or volume element independently. Implementing these features using BEM requires the discretization of every interface between dielectric regions and between those governed by differing PDEs. In contrast, volume-based methods need no additional degrees of freedom. Previous BEM approaches have addressed these limitations by developing specific formulations to treat multiple embedded dielectric regions without salt [136], multiple disconnected dielectric bodies with salt [137], and hybrid boundary-element/finite-difference methods to treat ion-exclusion layers [33].

A second important challenge for biomolecule BEM is the strong dependence of solution accuracy on the quality of the surface representation. In this work and in most others, the dielectric and ion-exclusion surfaces are described according to one of two definitions. Accessible surfaces [57] are defined as a union of spheres, where the atomic radii are expanded by a probe's radius. Molecular surfaces [49,59,138] represent the surface of closest approach of a probe sphere rolled over a union of spheres representing a molecule. These curved surfaces, which consist of portions of spheres and torii, are analytically defined but often difficult to discretize because the surfaces have cusps and singularities. Most boundary-element methods for solving the LPBE represent these surfaces approximately using large numbers of planar triangular elements, or panels, that can never truly capture the curved geometries. The importance of using curved panels has already been dis-

cussed [29, 38], but previous implementations have introduced other approximations. For example, other work has modified the molecular surface definition to avoid singularities and thin regions, used elements with low-order curvature that cannot accurately represent spheres or torii, or discretized surfaces using standard spherical triangles that cannot exactly represent the intersections between atoms.

A third challenge for BEM is that discretization of surface integral equations gives rise to dense linear systems of equations. As a result, memory costs scale quadratically in the number of unknowns. In contrast, the FDM and FEM generate sparse matrices that reflect the local nature of the differential operators. Solving the BEM linear system by matrix factorization requires $O(n^3)$ time, where n is the number of unknowns. Computational costs rapidly become prohibitive for systems with more than 10^4 unknowns, which is currently insufficient to accurately model large macromolecules such as proteins. The quadratic memory and cubic time costs can be reduced to linear or near-linear complexity by combining two approximation schemes. First, the linear systems are solved approximately, rather than exactly, using Krylov subspace iterative methods such as the conjugate gradient method (CG) or the generalized minimum residual algorithm (GMRES) [82]. Every iteration of a dense Krylov subspace method requires the multiplication of a vector by the BEM matrix, costing a prohibitive $O(n^2)$ memory and time. A second approximation reduces the matrix–vector product cost by interpreting the formation of the product as an n-body potential calculation [113]. This interpretation enables the use of techniques such as multipole methods [28, 32, 70, 97, 113, 134], or multiscale methods [135], to reduce the solution costs to $O(n)$ or $O(n\log n)$. Multipole methods require specialized expansions for every governing equation, and expansions for the LPBE have been developed in recent years [139]. One disadvantage of the fast multipole method (FMM) in particular is that the computational costs grow rapidly when improving accuracy [140] due to dense translation operations between multipole and local expansions, motivating the development of more efficient techniques [83, 108, 140].

Another challenge for the BEM is that the computation of elements in the dense systems of linear equations requires the integration of possibly singular functions over the panels used to discretize the boundary surfaces. These integrations can be interpreted as the cal-

culation of the potential, at an evaluation point due to a charge distribution defined on a boundary element. In contrast, the matrix elements for FDM and FEM problems are relatively easily computed. Although analytical expressions exist for the integral of the Laplace (Poisson) kernel over flat triangular panels [45, 46], integration of the the LPBE kernel, or integration over general curved domains, require numerical approximation. When the evaluation point is sufficiently far from the panel, quadrature rules can be used to perform numerical integration, even over curved panels [69, 133]. However, when the evaluation point is near or on the panel, even high-order quadrature rules do not suffice to capture the singularity. The evaluation of near-singular and singular integrals has been noted to be a limiting factor in the accuracy of BEM implementations for molecular electrostatics [27], and a variety of techniques have been developed to either avoid computing these integrals [27] or to approximate them with specialized quadrature rules [141].

In this work we present a boundary-element method implementation for solving the linearized Poisson–Boltzmann equation (LPBE) that addresses all of these challenges, with the ultimate goal of achieving high accuracy given reasonable computational resources. Three key features underlie the implementation. First, we have developed a general boundary-integral approach that can easily treat an arbitrary number of embedded regions of homogeneous dielectric with different dielectric constants and possibly salt. Second, the accessible and molecular surfaces are discretized using curved boundary elements that accurately capture the problem geometry, employing robust methods to compute self- and near-field integrals. Third, the dense linear systems are solved using preconditioned Krylov subspace methods and the FFTSVD algorithm [83].

Our Green's-theorem-based integral-equation formalism allows for ion-exclusion layers, solvent-filled cavities in the solute, and multiple homogeneous dielectric regions. Finite-difference and finite-element simulations have long been capable of modeling problems with these features, but this work presents the first detailed derivation for BEM treatment. The accessible and molecular surfaces are represented essentially exactly using curved boundary-element discretizations that accurately reproduce singularities, cusps, and thin regions. Accurate numerical integration techniques for the singular Laplace (Poisson) and LPBE Green's functions [38] allow the BEM to achieve exceptional accuracy. The

FFTSVD algorithm [83] efficiently sparsifies the dense BEM matrix, and memory and time requirements scale effectively linearly in the number of boundary elements. This fast BEM technique can be applied without modification to compress all of the integral operators in biomolecule electrostatics. Furthermore, the dense translation operations that dominate the FMM computational cost are replaced in the FFTSVD method with more efficient diagonal translations, allowing for a better trade-off between computational expense and accuracy.

After describing the boundary-element implementation, we present a set of computational experiments in order to assess the relative accuracy and computational cost of finite-difference and boundary-element method simulations for several categories of calculations. We calculate the electrostatic contributions to free energies of solvation for an analytically solvable sphere geometry, a short peptide derived from an HIV-1 substrate site [142], and the barnase–barstar protein complex [77]. We also compute rigid and non-rigid electrostatic binding free energies for the wild-type barnase–barstar complex as well as three single mutants. Solvation calculations demonstrate that the BEM presented here provides better convergence as a function of compute time. Rigid-binding results also suggest that the BEM converges more rapidly than FDM. However, when comparing differential rigid binding energies between wild-type and mutant protein complexes, where the structure remains the same except at the site of mutation, even low-resolution finite-difference simulations seem to accurately capture this difference. The curved BEM regains an accuracy advantage for differential non-rigid binding calculations, suggesting that the accuracy of finite-difference rigid binding may result from fortuitous cancellation of error. Finally, we demonstrate that the BEM implementation offers a clear advantage in accuracy and comparable simulation time for calculations that require repeated solution of the same problem geometry with different sets of atomic charges. Electrostatic component analysis [143–145] and charge optimization [34, 35] are types of calculations that fall into this category.

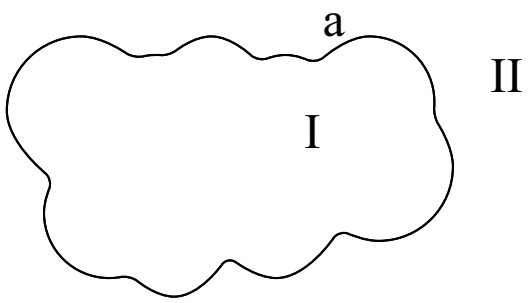


Figure 4-1: A one-surface problem in molecular electrostatics. The molecular interior (Region I) is surrounded by a salt solution with high dielectric constant and inverse Debye length κ (Region II).

4.2 THEORY

4.2.1 Green's theorem integral formulation

We begin our presentation of the multi-region integral formulation by deriving the one-surface Green's-theorem-based integral formulation described by Yoon and Lenhoff [21]. This method is also known as the non-derivative Green's theorem formulation [29, 52]. Figure 4-1 illustrates the problem and notation.

A single boundary a divides space into two regions. The molecular interior, labeled region I, has a uniform dielectric constant ε_I and contains n_c discrete point charges. The i^{th} point charge, located at r_i, is of value q_i. In region I, the electrostatic potential $\phi_I(r)$ is governed by a Poisson equation

$$\nabla^2\phi_I(r) = -\sum_{i=1}^{n_c} \frac{q_i}{\varepsilon_I}\delta(r-r_i), \tag{4.1}$$

where $\delta(r-r_i)$ is the Dirac delta function translated by r_i.

The solvent region II exterior to a represents solvent with mobile ions; we model the region as having a uniform dielectric constant ε_{II} and an inverse Debye length κ. In this region, the electrostatic potential $\phi_{II}(r)$ is assumed to obey the linearized Poisson–Boltzmann equation:

$$\nabla^2\phi_{II}(r) = \kappa^2\phi_{II}(r). \tag{4.2}$$

The free-space Green's functions for the Poisson and linearized Poisson–Boltzmann

equations are

$$G_I(r;r') = \frac{1}{4\pi||r-r'||} \text{ Region I} \tag{4.3}$$

$$G_{II}(r;r') = \frac{e^{-\kappa||r-r'||}}{4\pi||r-r'||} \text{ Region II} \tag{4.4}$$

respectively. Across the boundary surface a, the electrostatic potential and the normal displacement are continuous [51]. Using the relation $D = \varepsilon E$, where the electric field E satisfies $E = -\nabla\phi$, we can write the continuity conditions for a point r_a on the surface a as

$$\phi_I(r_a) = \phi_{II}(r_a) \tag{4.5}$$

$$\varepsilon_I \frac{\partial\phi_I}{\partial n}(r_a) = \varepsilon_{II} \frac{\partial\phi_{II}}{\partial n}(r_a). \tag{4.6}$$

In Equation 4.6, the normal direction is defined to point into the solvent region.

After specifying the problem domains and boundary conditions, one applies Green's theorem in both regions. Green's Theorem,

$$\int_V \left[\Psi\nabla^2\Phi - \Phi\nabla^2\Psi\right] dV = \int_\Omega \left[\Psi\frac{\partial\Phi}{\partial n} - \Phi\frac{\partial\Psi}{\partial n}\right] d\Omega, \tag{4.7}$$

where $\Psi(r)$ and $\Phi(r)$ are two scalar fields, allows the determination of the potential at a point in a volume V given the free-space Green's function for the governing equation in V as well as the potential and its normal derivative at the bounding surface Ω.

We first apply Green's theorem to find the potential at a point r_I in region I, which has the bounding surface $\Omega = a$. Using the Green's function (Equation 4.3) and substituting $\Psi(r') = G_I(r_I;r')$, $\Phi(r') = \phi_I(r')$, and Equation 4.1, we have

$$\int_V \left[G_I(r_I;r')\left(-\sum_{i=1}^{n_c}\frac{q_i}{\varepsilon_I}\delta(r-r_i)\right) - \phi_I(r')\nabla^2 G_I(r_I;r')\right] dV' = \int_a \left[G_I(r_I;r')\frac{\partial\phi_I}{\partial n}(r') - \phi_I(r')\frac{\partial G_I}{\partial n}(r_I;r')\right] dA'. \tag{4.8}$$

In Equation 4.8 and throughout this section, the normal derivative of G_I is taken with respect to the integration variable r': that is, $\frac{\partial G_I}{\partial n}(r_I;r')$ denotes the potential at r_I induced

by a normally-oriented dipole at r'. Simplifying the left-hand side using the definition of the Green's function,

$$\nabla^2 G_I(r_I; r') = -\delta(r_I - r'), \tag{4.9}$$

eliminates the volume integral in Equation 4.8, and by rearranging terms one obtains an expression for the potential at r_I as a function of the solute charge distribution and the boundary conditions:

$$\phi_I(r_I) = \sum_{i=1}^{n_c} \frac{q_i}{\varepsilon_I} G_I(r_I; r_i) + \int_a \left[G_I(r_I; r') \frac{\partial \phi_I}{\partial n}(r') - \phi_I(r') \frac{\partial G_I}{\partial n}(r_I; r') \right] dA'. \tag{4.10}$$

To apply Green's theorem in region II, one must first bound the region by introducing a hypothetical surface Γ at infinity, and using the substitutions $\Psi(r') = G_{II}(r_{II}; r')$, $\Phi(r') = \phi_{II}(r')$, Equation 4.4, and the LPBE Green's function definition. Assuming the potential obeys regularity conditions at infinity [23], the surface integrals over Γ vanish, and we can write the potential at a point r_{II} in region II as

$$\phi_{II}(r_{II}) = \int_a \left[G_{II}(r_{II}; r') \frac{\partial \phi_{II}}{\partial n}(r') - \phi_{II}(r') \frac{\partial G_{II}}{\partial n}(r_{II}; r') \right] dA', \tag{4.11}$$

and here, as in Equation 4.10, the normal direction is defined to point into region II.

We derive a pair of coupled integral equations by letting the points r_I and r_{II} approach a point r_a on the surface. Using Equation 4.10,

$$\begin{aligned} \phi_I(r_a) &= \lim_{r_I \to r_a} \phi_I(r_I) && (4.12) \\ &= \int_a G_I(r_a, r') \frac{\partial \phi_I}{\partial n}(r') dA' - \lim_{r_I \to r_a} \left[\int_a \phi(r') \frac{\partial G_I}{\partial n}(r_I; r') dA' \right] \\ &\quad + \sum_{i=1}^{n_c} \frac{q_i}{\varepsilon_I} G_I(r_a; r_i). && (4.13) \end{aligned}$$

The second term in Equation 4.13 can be interpreted as the potential induced by a dipole layer of charge on the surface. Such a potential is discontinuous as the evaluation point

crosses the surface and must be handled with care. We write

$$\phi_I(r_a) = \oint_a \left[G_I(r_a;r')] \frac{\partial \phi_I}{\partial n}(r') - \phi_I(r') \frac{\partial G_I}{\partial n}(r_a;r') \right] dA' + \frac{1}{2}\phi_I(r_a) + \sum_{i=1}^{n_c} \frac{q_i}{\varepsilon_I} G_I(r_a;r_i), \tag{4.14}$$

where $\oint$ represents a Cauchy principal value integral, and we assume that the limit as $r_I \to r_a$ has been taken from the direction opposite the normal. A similar limiting process applied to Equation 4.11, in which we let $r_{II} \to r_a$, yields

$$\phi_{II}(r_a) = \oint_a \left[-G_{II}(r_a;r')] \frac{\partial \phi_{II}}{\partial n}(r') + \phi_{II}(r') \frac{\partial G_{II}}{\partial n}(r_a;r') \right] dA' + \frac{1}{2}\phi_{II}(r_a). \tag{4.15}$$

Finally, we eliminate the unknowns $\phi_{II}(r_a)$ and $\frac{\partial \phi_{II}}{\partial n}(r_a)$ using the continuity conditions (Equations 4.5 and 4.6). Two coupled integral equations result:

$$\frac{1}{2}\phi_I(r_a) + \int_a \phi_I(r') \frac{\partial G_I}{\partial n}(r_a;r') dA' - \int_a \frac{\partial \phi_I}{\partial n}(r') G_I(r_a;r') dA' = \sum_{i=1}^{n_c} \frac{q_i}{\varepsilon_I} G_I(r_a;r_i) \tag{4.16}$$

$$\frac{1}{2}\phi_I(r_a) - \int_a \phi_I(r') \frac{\partial G_{II}}{\partial n}(r_a;r') dA' + \frac{\varepsilon_I}{\varepsilon_{II}} \int_a \frac{\partial \phi_I}{\partial n}(r') G_{II}(r_a;r') dA' = 0. \tag{4.17}$$

Introducing an abbreviated notation allows the equations to be written as

$$\begin{bmatrix} \frac{1}{2}I + D^a_{I,a} & -S^a_{I,a} \\ \frac{1}{2}I - D^a_{II,a} & \varepsilon_{I,II} S^a_{II,a} \end{bmatrix} \begin{bmatrix} \phi_a \\ \frac{\partial \phi_a}{\partial n} \end{bmatrix} = \begin{bmatrix} \sum_i \frac{q_i}{\varepsilon_I} G^a_{I,i} \\ 0 \end{bmatrix}, \tag{4.18}$$

where ϕ_a and $\frac{\partial \phi_a}{\partial n}$ denote the surface potential and normal displacement on a, I denotes the identity operator, $\varepsilon_{I,II}$ abbreviates $\frac{\varepsilon_I}{\varepsilon_{II}}$, and $S^u_{I,v}$ and $D^u_{I,v}$ denote the single- and double-layer operators that compute potential at the surface u due to a monopole or dipole charge density on surface v, given the Green's function $G_I(r;r')$. The operator $S^u_{I,v}$ is defined such that:

$$S^u_{I,v} \frac{\partial \phi_v}{\partial n} = \oint_v G_I(r_u;r') \frac{\partial \phi_v}{\partial n(r')}(r') dA'; \tag{4.19}$$

similarly,

$$D^u_{I,v}\phi_v = \oint_v \frac{\partial G_I}{\partial n(r')}(r_u; r')\phi_v(r')dA'. \tag{4.20}$$

In Equation 4.18, we have also defined $G^a_{I,i} = G_I(r_a; r_i)$.

4.2.2 Numerical solution using the boundary-element method

To simultaneously solve Equations 4.16 and 4.17 using the boundary-element method (BEM), we first approximate the surface variables $\phi_I(r_a)$ and $\frac{\partial \phi_I}{\partial n}(r_a)$ as weighted combinations of a set of n basis functions $\chi_1(r), \chi_2(r), \ldots, \chi_n(r)$ on the surface:

$$\phi_I(r_a) \approx \sum_{k=1}^{n} u_k \chi_k(r_a) \tag{4.21}$$

$$\frac{\partial \phi_I}{\partial n}(r_a) \approx \sum_{k=1}^{n} v_k \chi_k(r_a). \tag{4.22}$$

The unknown weights u_k and v_k are then found by forcing the integral equation to be satisfied as closely as possible in some choice of metric.

In this work, we discretize the surfaces into a discrete set of n_p non-overlapping curved boundary elements and use piecewise-constant basis functions that have a value of one on a single panel and are zero everywhere else:

$$\chi_k(r_a) = \begin{cases} 1 & \text{if } r_a \text{ is on panel } k \\ 0 & \text{otherwise.} \end{cases} \tag{4.23}$$

Defining the integral equation residual to be the difference between the known condition on the surface and the integral operator applied to the approximate solution, one can form a square linear system by forcing the residual to equal zero at the boundary-element centroids, a technique known as centroid collocation [53]. Using the piecewise-constant basis functions and denoting the centroid of panel i as r_{c_i}, the discretized (matrix) form of the operator $S^a_{I,a}$ from Equation 4.19 has entries

$$S_{i,j} = \int_{\text{panel } j} G_I(r_{c_i}; r')dA'_j, \tag{4.24}$$

and the double-layer discretized operator $D^a_{I,a}$ similarly has entries

$$D_{i,j} = \int_{\text{panel } j} \frac{\partial G_I}{\partial n(r')}(r_{c_i}; r')dA'_j. \tag{4.25}$$

The total matrix equation representing the discretized form of Equation 4.18 therefore has dimension $2n_p$. Once this equation is solved, the potential anywhere in space may be calculated using the discretized forms of Equations 4.10 and 4.11.

4.2.3 Extension to multiple dielectrics, solvent cavities, and ion-exclusion layers

Continuum electrostatics models of biomolecular systems can be defined by multiple embedded regions of differing homogeneous dielectric constant and salt treatment. Integral-equation formulations that can solve these problems often possess a complicated block structure because there exist numerous operators that couple variables on one surface to conditions on other surfaces. To illustrate this block structure, we next present Green's theorem formulations for two-surface and three-surface example problems. We then describe how a tree-based representation of the enclosed regions facilitates the determination of the appropriate Green's-theorem-based integral operator for arbitrary multi-region problems.

Two-surface formulation

Figure 4-2 is a schematic of a two-surface problem in molecular electrostatics; salt ions are not permitted to directly reach the molecular surface *a*, but instead are bounded by an accessible surface *b* a specified distance outside the molecule. The enclosed volume between the surfaces is termed the ion-exclusion layer. Region *I*, again representing the molecular interior, has dielectric constant ε_I and n_c point charges. The ion-exclusion layer, region *II*, has dielectric constant ε_{II}, and in this region the Laplace equation governs the electrostatic potential. Region *III* represents solvent with mobile ions and has dielectric constant ε_{III} (usually the same as ε_{II}) but contains a Debye–Hückel salt treatment; the potential in this region is governed by the linearized Poisson–Boltzmann equation. This

problem has continuity conditions at both a and b:

$$\begin{aligned}
\phi_I(r_a) &= \phi_{II}(r_a) && (4.26)\\
\varepsilon_I \frac{\partial \phi_I}{\partial n}(r_a) &= \varepsilon_{II} \frac{\partial \phi_{II}}{\partial n}(r_a) && (4.27)\\
\phi_{II}(r_b) &= \phi_{III}(r_b) && (4.28)\\
\varepsilon_{II} \frac{\partial \phi_{II}}{\partial n}(r_b) &= \varepsilon_{III} \frac{\partial \phi_{III}}{\partial n}(r_b). && (4.29)
\end{aligned}$$

The associated integral equations have four surface variables, which are the potential and normal derivative on both surfaces: ϕ_a, $\frac{\partial \phi_a}{\partial n}$, ϕ_b, $\frac{\partial \phi_b}{\partial n}$. The free-space Green's functions in each region are again denoted by G with the region label as subscript: $G_{II}(r;r')$, for instance, denotes the free-space Laplace Green's function. As in the one-surface derivation, we apply Green's theorem in each region using the appropriate substitutions, let the field points approach the bounding surfaces, and eliminate redundant variables using the continuity conditions. The resulting operator takes the form

$$\left[\begin{array}{cc|cc}
\frac{1}{2}I + D^a_{I,a} & -S^a_{I,a} & & \\
\frac{1}{2}I - D^a_{II,a} & +\varepsilon_{I,II} S^a_{II,a} & +D^a_{II,b} & -S^a_{II,b} \\ \hline
-D^b_{II,a} & +\varepsilon_{I,II} S^b_{II,a} & \frac{1}{2}I + D^b_{II,b} & -S^b_{II,b} \\
 & & \frac{1}{2}I - D^b_{III,b} & +\varepsilon_{II,III} S^b_{II,b}
\end{array}\right]
\left[\begin{array}{c} \phi_a \\ \frac{\partial \phi_a}{\partial n} \\ \phi_b \\ \frac{\partial \phi_b}{\partial n} \end{array}\right]
=
\left[\begin{array}{c} \sum_i \frac{q_i}{\varepsilon_I} G^a_{I,i} \\ 0 \\ 0 \\ 0 \end{array}\right], \quad (4.30)$$

which can be solved with the boundary-element method described above.

Note that the integral operator contains several zero blocks. These blocks arise from the application of Green's theorem in regions for which one or more surfaces do not form part of that region's bounding surface. For instance, surface b forms no portion of the bounding surface for region I, and consequently variables on surface b contribute nothing to the integral equation derived by applying Green's theorem in region I. Note also that two of the integral equations derive from the application of Green's theorem in region II.

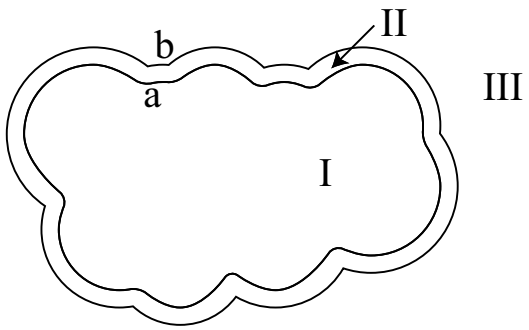


Figure 4-2: A two-surface problem in molecular electrostatics. The molecular interior (Region *I*) is surrounded by an ion-exclusion layer with solvent dielectric and no salt (Region *II*), which in turn is surrounded by solvent with a salt treatment (Region *III*).

Three surface formulation

To identify more general trends in the construction of multi-boundary integral operators, we extend the two surface formulation by adding a solvent-filled cavity inside the protein interior (Figure 4-3). In this problem and for the remainder of this section, we will follow the convention that region *I* is the outermost solvent region. The additional region *IV* has dielectric constant ε_{IV} (generally equal to ε_I and ε_{II}), and is not large enough to contain an ion-exclusion surface. Again, we apply Green's theorem in every region, take limits on the surface integrals as the field points approach the boundaries, and enforce continuity conditions. The resulting operator takes the form:

$$\left[\begin{array}{cc|cc|cc} \frac{1}{2}I+D^c_{IV,c} & -S^c_{IV,c} & & & & \\ \frac{1}{2}I-D^c_{III,c}+\varepsilon_{IV,III}S^c_{III,c} & +D^c_{III,b}-S^c_{III,b} & & & & \\ \hline -D^b_{III,c}+\varepsilon_{IV,III}S^b_{III,c} & \frac{1}{2}I+D^b_{III,b} & -S^b_{III,b} & & & \\ & \frac{1}{2}I-D^b_{II,b}+\varepsilon_{III,II}S^b_{II,b} & +D^b_{II,a}-S^b_{II,a} & & & \\ \hline & -D^a_{II,b}+\varepsilon_{III,II}S^a_{II,b} & \frac{1}{2}I+D^a_{II,a} & -S^a_{II,a} & & \\ & & \frac{1}{2}I-D^a_{I,a}+\varepsilon_{II,I}S^a_{I,a} & & & \end{array}\right] \left[\begin{array}{c} \phi_c \\ \frac{\partial\phi_c}{\partial n} \\ \hline \phi_b \\ \frac{\partial\phi_b}{\partial n} \\ \hline \phi_a \\ \frac{\partial\phi_a}{\partial n} \end{array}\right] = \left[\begin{array}{c} 0 \\ \frac{\sum_i \frac{q_i}{\varepsilon_{III}} G^c_{III,i}}{} \\ \sum_i \frac{q_i}{\varepsilon_{III}} G^b_{III,i} \\ 0 \\ \hline 0 \\ 0 \end{array}\right]. \tag{4.31}$$

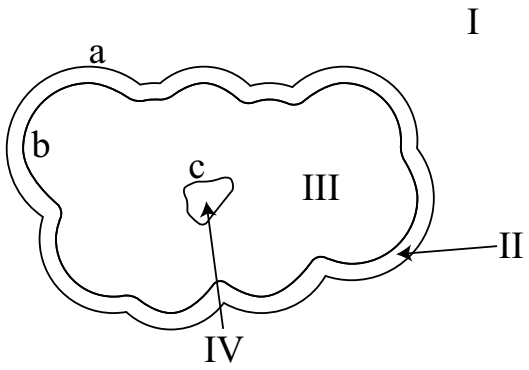


Figure 4-3: A three-surface problem in molecular electrostatics. This geometry is analogous to the two-surface problem (Figure 4-2) except that a solvent-filled cavity has been added within the molecular interior (Region *IV*). Note that in contrast to previous examples, the regions and surfaces have been labeled in reverse order.

In this expression, the charges in region *III* contribute to two of the integral equations, both of which derive from the application of Green's theorem in the region. The point-charge contributions are found by taking limits as the field point in the region approaches the exterior and interior bounding surfaces. Note that some of the off-diagonal 2×2 blocks contain nonzero 1×2 blocks. These operators represent the contribution of a region's interior (or exterior) bounding surface to the integral equation derived by letting the field point in the region approach an exterior (or interior) bounding surface.

Tree-based general formulation

To derive an integral operator for an arbitrary configuration of embedded boundaries with regions of differing homogeneous dielectric, point charges, and salt treatment, we represent the topology using a tree. Traversing the nodes of the tree, irrespective of the order in which they are visited, allows multiplication by the entire integral operator taking into account all necessary interactions. Each node of the tree represents one region, and is associated with a dielectric constant and possibly salt treatment or point charges. The tree is constructed such that the node for a given region X is assigned to be the child of the node corresponding to the region surrounding X. Region I, which is bounded only by a hypothetical surface at infinity, is defined to be the root node. Furthermore, we associate with each node the

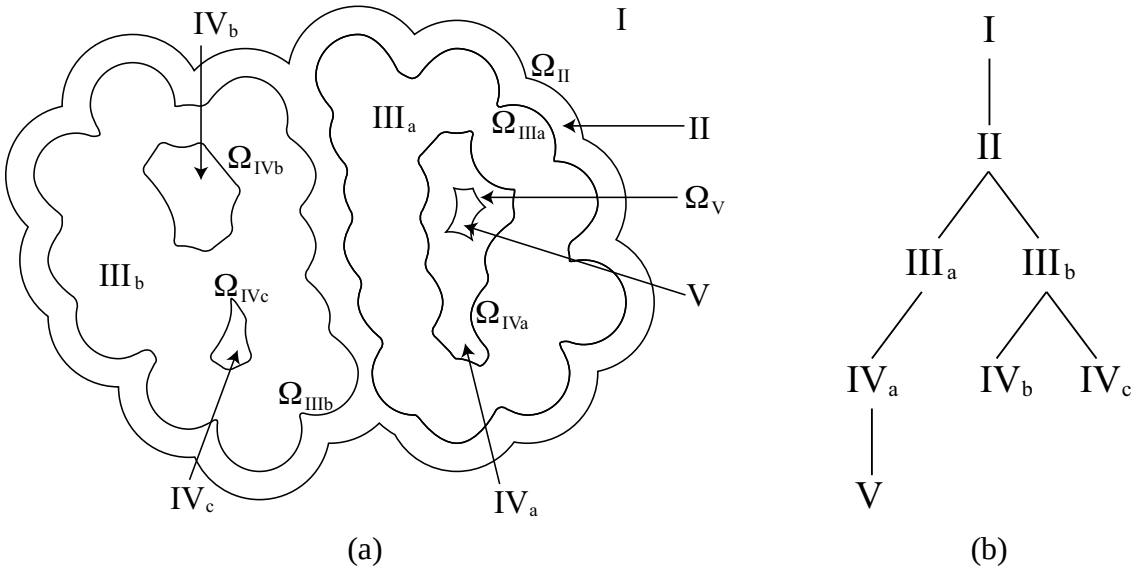


Figure 4-4: Tree representation of a general surface problem. The example molecular geometry shown in (a) might correspond to an encounter complex between two associating proteins (Regions III_a and III_b), surrounded by a single ion-exclusion layer (Region *II*), which in turn is surrounded by solvent with salt (Region *I*). The binding partners contain several solvent filled cavities (Regions IV_{a-c}), and one cavity is large enough to contain a small ion-exclusion layer (Region *V*). The tree representation for this example multi-surface geometry is shown in (b).

exterior bounding surface of the corresponding region. Figure 4-4(b) is a tree diagram constructed to describe the system shown in Figure 4-4(a).

The example geometry used here may be representative of an encounter complex in protein–protein binding, where two nearly associated binding partners (Regions III_a and III_b) are surrounded by a single ion-exclusion layer (Region *II*). There are also several solvent-filled cavities present in both binding partners (Regions IV_{a-c}), and one cavity is large enough to contain a small ion-exclusion layer (Region *V*).

Applying the multi-surface integral operator

A multi-region electrostatics problem with n surfaces generates a system of $2n$ coupled integral equations. For each surface, one writes Green's theorem for the regions exterior and interior to the surface and takes the appropriate limits as the evaluation points approach the surface. Accordingly, one may refer to the resulting integral equations as the exterior and interior equations corresponding to the surface.

An integral equation derived from an application of Green's theorem contains contributions from the surfaces bounding the region. As an example, consider the interior equation for surface Ω_{IIIb}. Applying Green's theorem in region III_b defines the potential at a point in this region as a function of the surface potential and its normal derivative on Ω_{IIIb}, Ω_{IVb}, and Ω_{IVc}. Taking the limit of the Green's theorem expression as the field point approaches Ω_{IIIb}, we obtain the interior equation. Clearly, a surface's interior equation contains contributions from the surface as well as its children. Similarly, a surface's exterior equation contains contributions from the surface, its parent, and its siblings. This can be seen by letting the field point approach any of the cavity surfaces.

Multi-surface problems demand that careful attention be paid to the definition of the surface normal. In this work we follow the mathematical convention that a normal always points outward from the finite volume enclosed by the surface. To apply the entire multi-surface operator for an arbitrary problem, we first define a tree such as shown in Figure 4-4(b). The tree is traversed depth first, and at each node we apply several integral operators, which in the discretized problem correspond to dense block matrix–vector multiplications. Because each block multiplication may be interpreted as the computation of the potential at a surface due to a distribution of monopole or dipole charge on another surface, we refer to the two surfaces as the *source surface* and the *destination surface*. The set of block multiplications is determined by the topology of the surfaces, and is defined such that every non-zero block in the integral operator is applied exactly once.

We define four types of block integral operators: the self-surface interior operator, the self-surface exterior operator, the non-self interior operators, and the non-self exterior operators. As previously discussed, each operator represents an interaction between two surfaces. The labels *interior* and *exterior* specify whether the integral operator arises from an application of Green's theorem to the region interior or exterior to the source surface. The self and non-self operators are distinguished because the discontinuity in the self operator double-layer calculation requires specific treatment.

For every node, the following block matrix-vector multiplications are performed. Let the current node correspond to the region X. Denote its parent region by W, sibling regions by S_i, and child regions by Y_i. Lowercase letters correspond to the outer bounding surfaces

for these regions. Every dense block is applied to the vector $\left(\phi_x, \frac{\partial \phi_x}{\partial n}\right)^{\mathrm{T}}$.

1. Apply the self-surface interior operator

$$\begin{bmatrix} \frac{1}{2}I + D^x_{X,x} & -S^x_{X,x} \end{bmatrix} \tag{4.32}$$

   and add the result to the node's interior equation.

2. Apply the self-surface exterior operator

$$\begin{bmatrix} \frac{1}{2}I - D^x_{W,x} & +\varepsilon_{X,W} S^x_{W,x} \end{bmatrix} \tag{4.33}$$

   and add the result to the node's exterior equation.

3. Apply the appropriate non-self exterior operator

$$\begin{bmatrix} -D^w_{W,x} & +\varepsilon_{X,W} S^w_{W,x} \end{bmatrix} \tag{4.34}$$

   and add the result to the interior equation of the *parent* node.

4. For each *sibling* node S_i, apply the appropriate non-self exterior operator

$$\begin{bmatrix} -D^{s_i}_{W,x} & +\varepsilon_{X,W} S^{s_i}_{W,x} \end{bmatrix} \tag{4.35}$$

   and add the result to the exterior equation of the sibling node.

5. For each *child* node Y_i, apply the appropriate non-self interior operator

$$\begin{bmatrix} +D^{y_i}_{X,x} & -S^{y_i}_{X,x} \end{bmatrix} \tag{4.36}$$

   and add the result to the exterior equation of the child node.

4.2.4 Matrix compression with the FFTSVD algorithm

As discussed in the Introduction, boundary-element methods give rise to dense matrix equations whose solution by LU factorization or Gaussian elimination requires $O(n^3)$ time and $O(n^2)$ memory for a system with n unknowns. Combining Krylov-subspace iterative methods with fast-solver algorithms reduces these costs to nearly $O(n)$. The Krylov method requires only a way to apply the matrix A to a vector; in contrast, LU factorization and Gauss elimination require explicit access to every entry of A. In this work, we use the FFTSVD algorithm [83] to rapidly apply the dense integral operators.

FFTSVD, like multipole methods, exploits the smooth decay of the Green's functions as the distance between source and evaluation point increases. Both types of methods use a spatial decomposition of the set of boundary elements to separate near-field interactions, which are computed exactly, from far-field or long-range interactions, which can be accurately approximated. The long-range interactions are approximated by projecting the dominant panel source distributions, computed using an approximate singular value decomposition (SVD), onto a grid. Grid–grid interactions are computed via the fast Fourier transform (FFT), and the dominant responses are interpolated back to the destination integral equation collocation points. An overview of the FFTSVD method is presented in Figure 4-5, and a fully detailed description of the algorithm can be found in reference [83].

For the general multi-boundary Green's theorem formulation, each node in the tree contains a FFTSVD-compressed operator that simultaneously stores both the single- and double-layer interactions between all panels that bound the region.

4.2.5 Preconditioning

It has been previously noted in the literature that the non-derivative Green's theorem formulation can lead to ill-conditioned systems of linear equations, especially with decreasing boundary-element size [32]. To address this issue, we have implemented preconditioning in order to efficiently solve these systems with iterative methods. By definition, a preconditioner is any matrix P such that the equation $PAx = Pb$ has better convergence properties than $Ax = b$ when the systems are solved iteratively. In general, Krylov iterative methods

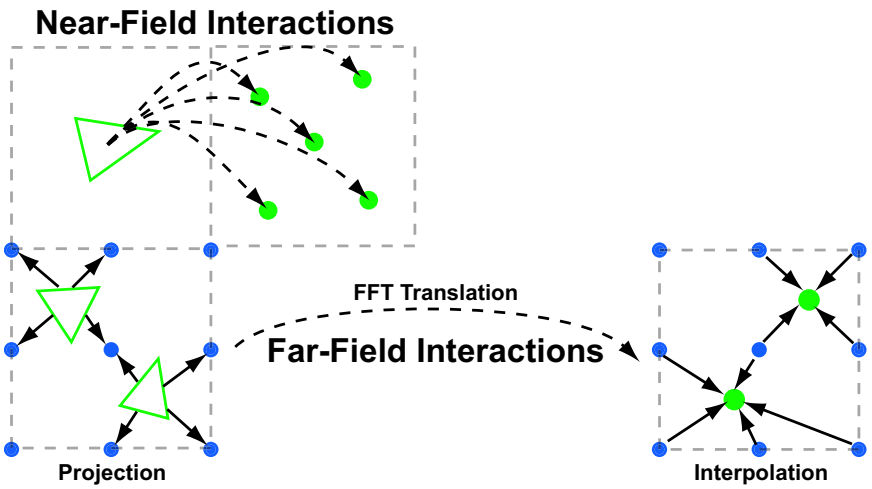


Figure 4-5: An overview of the FFTSVD matrix compression algorithm. FFTSVD uses a multi-level octree spatial decomposition to separate panel–evaluation point interactions into near- and far-field components at multiple length scales. When two cubes at the finest length scale are nearby, interactions are computed through direct integration. However, when two interacting cubes are well separated, dominant sources are projected onto a cubic grid and translated to a grid surrounding the recipient cube. The FFT is used to accelerate this translation operation. Finally, the grid potentials can be interpolated back onto the dominant responses of the panel centroids. This Figure has been adapted from [83].

are most efficient at solving linear systems with clustered eigenvalues [146]. Because the identity matrix I (or multiples) has an optimal clustering, P is generally selected such that $P \approx A^{-1}$ but is inexpensive to form and apply.

For the discretized integral operator matrices that arise from the Green's theorem formulation, the dominant entries tend to be the self-influence terms, for which the evaluation point is on the element over which the integral is performed. Consequently, a reasonable choice for P is the inverse of a sparse matrix that contains only these self-term entries. As an examination of Equations 4.18, 4.30, and 4.31 should make clear, the sparse matrix that includes just the self-influence terms is not diagonal, but no row has more than two non-zero off-diagonals.

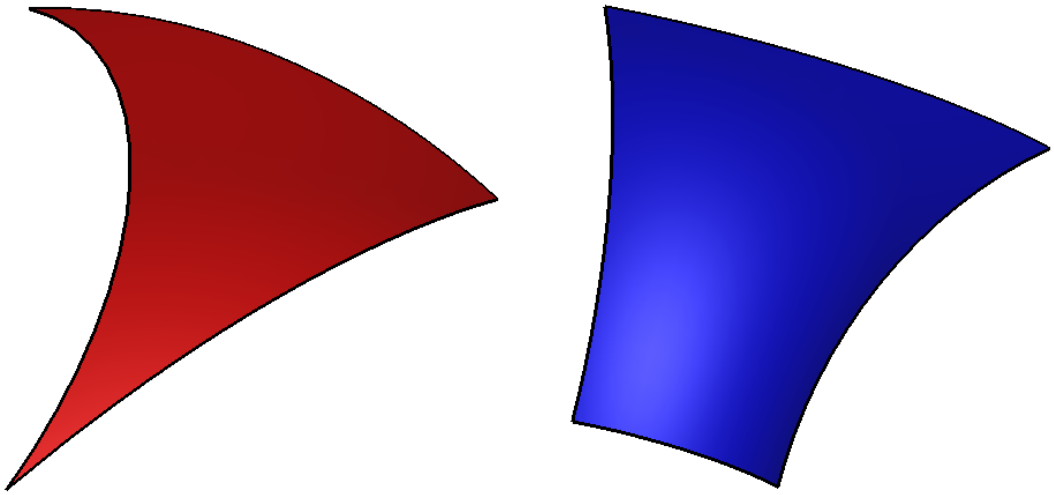
4.2.6 Curved panel discretization

In order to generate the basis functions used in the boundary-element method, we discretize the molecular and accessible surfaces that define the problem into curved elements that

can exactly represent the underlying geometry [38]. Accessible surfaces [57], also called expanded van der Waals surfaces, are generally used to model the ion-exclusion layer and can be completely described by convex spherical patches bounded by circular arcs. These circular arcs are not necessarily geodesic arcs, and thus we use the concept of a generalized spherical triangle (GST) (Figure 4-6(a)) [38, 39]. A GST is a three-sided curved element that lies on the surface of a sphere, where each edge is associated with a circular arc. If the arc center for all three edges happens to be the center of the sphere, a traditional spherical triangle is recovered. A spherical patch can be discretized into a set of GSTs by starting with a flat element triangulation, and then assigning the appropriate circular arc to each panel edge. Edges that lie along the interface between atoms are assigned non-geodesic arcs that follow the curve of intersection, while all other edges are assigned geodesic arcs.

Molecular surfaces [49, 59, 138], used here to model dielectric interfaces, are the surfaces of closest approach for the surface of a probe sphere that is rolled over a molecule. They can be described by three types of surface patches [59]. Convex spherical patches are defined where the probe sphere is in contact with only one atom, and can be described by portions of a sphere bounded by circular arcs and discretized with GSTs. Concave spherical re-entrant patches are formed when the probe touches three or more atoms simultaneously, and are also described by GSTs. When the probe simultaneously touches two atoms, a portion of a torus is generated. Toroidal regions are discretized into four-sided curved torus panels (Figure 4-6(b)) that are isomorphic to a rectangle. A fully meshed curved panel discretization for the barnase–barstar complex molecular surface is shown in Figure 4-7.

Techniques for integrating singular Green's functions over these curved GST and torus panels have been developed, and are discussed in detail in [38]. Briefly, when the evaluation point in the integrand is far away from the panel, low-order quadrature rules are used to perform numerical integration. These quadrature rules are generated by creating a smooth mapping between a reference flat triangle or rectangle (for GSTs and torus panels respectively) that relates a known quadrature rule on these simple domains [44] to those applicable on the curved panels. When the evaluation point is near or on the curved panel, even high-order quadrature rules do not suffice to capture the singularity. As a result, we adopt specialized methods for each panel type and Green's function. For the single-layer



(a) (b)

Figure 4-6: The two types of curved panels used to discretize accessible and molecular surfaces. A generalized spherical triangle (GST) (a), is a three-sided region on the surface of a sphere bounded by three circular arcs. These arcs are not necessarily geodesic arcs. Torus patches on molecular surfaces are discretized using toroidal panels (b), which are isomorphic to a rectangle.

Laplace (Poisson) kernel, we integrate over GSTs using a technique that reproduces the effect of panel curvature using a higher-order distribution on a reference flat triangle [71]. Single-layer Laplace integrals over torus panels are evaluated using a panel-splitting approach, which avoids integration near the singularity using recursive subdivision. When integrating the double-layer Laplace kernel in the near-field over both GST and torus panels, we exploit the fact that the double-layer potential is equal to the solid angle subtended by the curved panel when observed from the evaluation point [51, 73]. In order to integrate the linearized Poisson–Boltzmann kernel or its normal derivative in the near field, we adopt a previously presented desingularization technique [32]. This method divides the integral into a singular Laplace component that can be integrated as described above, and a smooth component that can be integrated using quadrature.

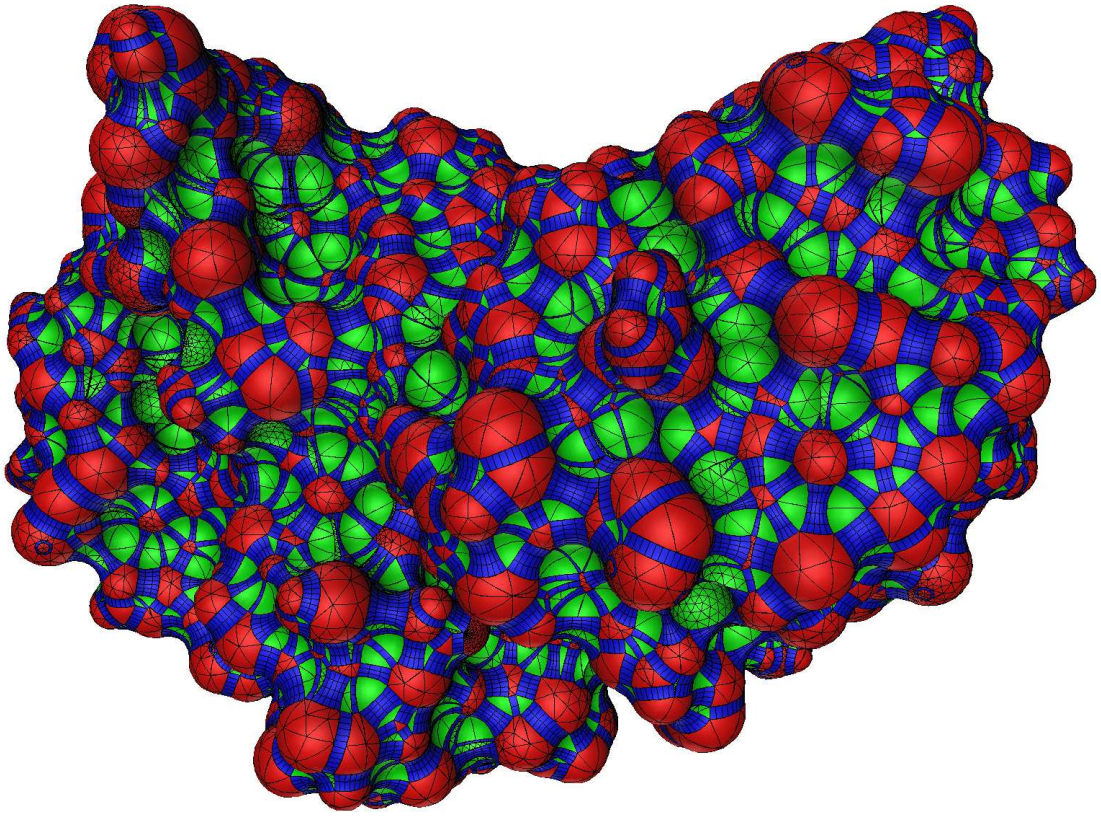


Figure 4-7: A rendering of a curved panel discretization for the molecular surface of the barnase–barstar protein complex. Red regions indicate convex spherical patches, green regions are re-entrant spherical patches, and blue regions are toroidal patches. Black lines indicate the boundaries between panels. The graphic depicts an approximation to the discretized geometry used for calculation. Every GST and torus panel has been approximated by a very large number of flat triangles for the purpose of visualization only, and the true surface normal in conjunction with Phong shading have been used to render the image.

4.3 COMPUTATIONAL DETAILS

4.3.1 Peptide and protein structure preparation

The structure of a peptide derived from an HIV-1 protease cleavage site was obtained from the Protein Data Bank (PDB) with accession code 1F7A [142]. This structure contains nine visible residues of a decameric peptide bound to an inactivated mutant of HIV-1 protease; only the peptide was considered in further calculations. An N-terminal acetyl blocking group and a C-terminal methylamide blocking group were added to the peptide. The wild-type structure for the barnase–barstar protein complex was also obtained from the PDB using accession code 1BRS [77]. To prepare this structure for calculation, we followed a previous protocol [147] where all but a set of 12 interfacial water molecules were removed. For both the peptide and barnase–barstar structures, hydrogen atoms were added using the HBUILD module [148] in the CHARMM computer program [58] using the PARAM22 parameter set [11] and a distance-dependent dielectric constant of 4. In addition, side-chain atoms that were missing from the crystal structures were rebuilt using CHARMM and the default PARAM22 geometry. All ionizable residues were left in their standard states at pH 7.

4.3.2 Modeling of barnase–barstar mutations

Three point mutations (E73Q in barnase, D39A in barstar, and T42A in barstar) were built into the barnase–barstar complex for subsequent analysis. The alanine mutations were created by cutting back the wild-type residue to the β-carbon. The E73Q mutation was built by sampling glutamine side-chain dihedral angles in 30-degree increments using CHARMM [58] and the PARAM22 parameter set [11]. For each sampled conformation, the side chain was energy minimized until convergence keeping all other atoms in the structure fixed. The lowest energy minimized geometry was taken to represent the E73Q mutation.

4.3.3 BEM and FFTSVD parameters

Parameters used in the FFTSVD algorithm included a drop tolerance of 10^{-5} for SVD compression, spatial decomposition until each cube contained no more than 32 panels, and a grid size of $4 \times 4 \times 4$ in each finest-level cube to represent dominant sources and responses during FFT translation. The boundary-element matrix equations were solved using the Krylov subspace method GMRES [82] to a relative residual of 10^{-6}. All curved BEM calculations were performed on a 2-way dual-core 2.0 GHz Opteron machine running a parallel version of the FFTSVD library. All presented timings are the sum of CPU usage across all four processors.

4.3.4 Finite-difference solver and parameters

In order to compare our curved-panel boundary-element solver to finite-difference methods (FDM), we have implemented a FDM solver using previously described techniques [22] and an analytical surface representation. This implementation uses successive over-relaxation (SOR) with an optimized acceleration factor to solve the finite-difference equations to a relative residual of 10^{-6}. In order to handle truncation of the boundary condition at infinity, a focusing scheme [95] was employed in all FDM calculations where the molecule of interest occupied first 23% and then 92% of the finite-difference grid. For the low-percent fill run, a Debye–Hückel screened potential in solvent dielectric was used to assign potentials to the boundary of the cubic grid. For the high-percent fill run, boundary potentials were taken by interpolation from the low-percent fill solution. Although it is common to average results from multiple translations of the molecule relative to the grid in order to reduce error due to the grid representation [95], only one placement was used here to make a fair comparison to the curved BEM, which is insensitive to translations or rotations of the geometry. Cubic grids used to discretize molecular geometries in the FDM spanned 129 to 481 grid points per Angstrom in increments of 32, which are all solvable within 4 GB of computer memory. These sizes correspond to grid resolutions of approximately 2.3 to 8.6 grid points per Angstrom for the barnase–barstar complex. All FDM calculations were performed in serial on a 2-way dual-core 2.0 GHz Opteron machine.

4.3.5 Electrostatic solvation and binding calculations

All continuum electrostatics calculations were performed using a molecular dielectric constant of 4, a solvent dielectric constant of 80, a molecular surface with probe radius 1.4 Å for dielectric interfaces, an accessible surface with probe radius 2.0 Å for ion-exclusion layers, and an ionic strength of 145 mM. In order to compute the solvation free energy of a molecule, we take the difference between the energy of the solvated state and a reference state where the solvent dielectric constant is equal to the molecular dielectric constant and no salt is present. The BEM calculates this energy difference directly, and an explicit reference state is not needed. In the FDM implementation, the energy of the reference state is explicitly computed to cancel grid energy.

For rigid-binding calculations, the electrostatic component of the free energy of binding was computed as the sum of Coulombic interactions in the bound state and the differential solvation energy between the bound complex and infinitely separated individual binding partners. For the FDM, proper grid placement was used to accelerate the calculation by cancelling the grid energy in the complex with grid energies for the individual binding partners. Because the BEM only computes the reaction potential rather than the total electrostatic potential, the Coulombic interactions between the binding parters must be explicitly added.

Non-rigid electrostatic binding energies were computed by first energy minimizing the geometry of the complex and each of the isolated binding partners separately. The minimization was performed using CHARMM and the PARAM22 parameter set, relaxing all atoms with 1,000 steps of adapted basis Newton–Raphson (ABNR) minimization using a distance-dependent dielectric constant of 4. The binding energy was then computed using a thermodynamic cycle where the two isolated binding partners were first desolvated to a vacuum with the molecular dielectric constant. In vacuum, the partners were deformed to their bound-state structures and then rigidly bound, computing all electrostatic changes with Coulomb's law in molecular dielectric. Finally, the complex was re-solvated. The sum of the energetic changes in these three steps was taken as the non-rigid electrostatic binding free energy. Due to the change in geometry between the bound and unbound states

in non-rigid binding, the FDM grid energy cancellation technique could not be used, and explicit reference states were employed for all FDM solvation calculations.

4.3.6 Generating curved panel discretizations

Molecular and accessible surfaces were discretized into curved panels starting with high-quality flat triangular meshes for spherical regions from the program NETGEN [149]. These panels were then converted, along with torus patches, to curved panels using previously described methods [38]. Curved-panel discretizations for molecular geometries were generated such that memory requirements did not exceed 4 GB. For the sphere test case, discretizations were obtained between roughly 80 and 58,000 curved panels including ion-exclusion and dielectric interface surfaces. For the peptide example, panel counts spanned approximately 5,200 to 128,000, and for the various barnase–barstar complexes, the span was roughly 92,000 to 310,000 total curved panels.

4.4 RESULTS AND DISCUSSION

For all calculations, we compared our boundary element results to those generated using finite-difference methods. Although geometric measures can be defined for such comparisons [28, 135], we chose to use compute time as our metric, to determine which method can achieve superior convergence properties given a certain amount of time. We could not guarantee that the geometry of the problem being solved is exactly the same in both methods because different algorithms were used to generate molecular boundaries. Therefore, for systems without closed-form solutions, the level of convergence for a particular method was assessed solely on how little the solution changed as the compute time increased.

4.4.1 Electrostatic solvation free energies

One of the simplest linearized Poisson-Boltzmann calculations is the computation of the electrostatic component of the free energy associated with the transfer of a molecule from low- to high-dielectric medium, where the high-dielectric region contains an ion-exclusion

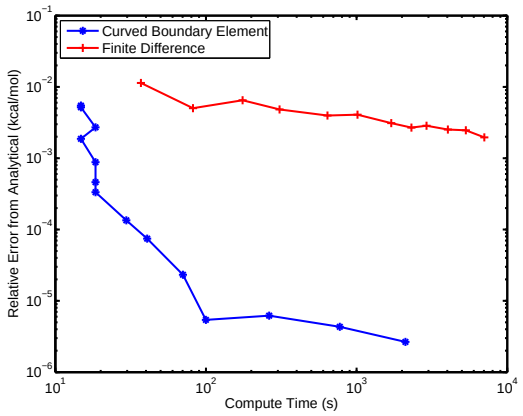


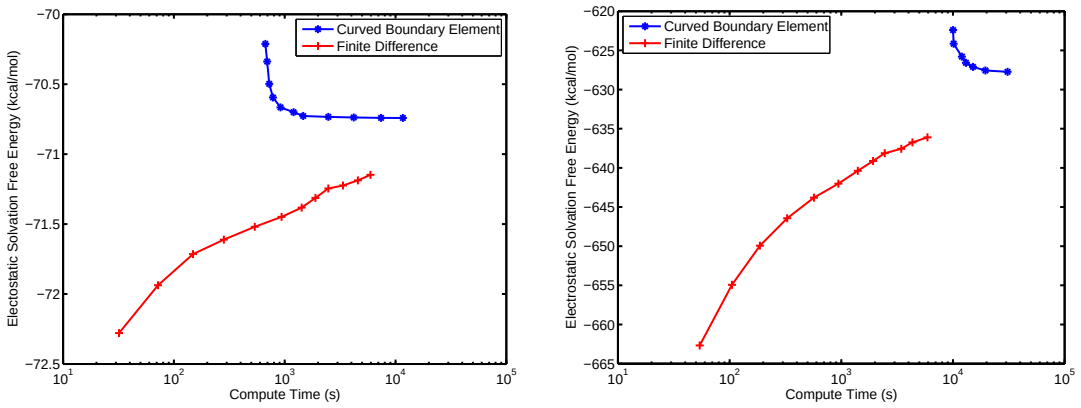
Figure 4-8: Convergence plot for the solvation free energy for a sphere with an eccentric charge and ion-exclusion layer. The relative error from the analytical solution is plotted as a function of compute time. Results are compared between the curved BEM and FDM implementations. The curved BEM accuracy is limited to 5–6 digits given the settings used in the FFTSVD matrix compression.

layer with salt outside. This quantity, known as the electrostatic solvation free energy, is useful in many calculations and forms the basis for computing more complex quantities such as electrostatic binding energies. We first validated the multi-surface formulation by computing the solvation free energy for a simple spherical test case, which has a closed-form solution. Then, we gauged the accuracy of the solver by examining more complicated geometries including a peptide derived from an HIV-1 substrate site and the barnase–barstar protein–protein complex.

Sphere with ion-exclusion layer

In order to test the correctness of the multiple surface formulation, the electrostatic solvation free energy for a sphere of radius 1 Å with a charge of $+1e$ placed 0.5 Å away from the center was computed. An ion-exclusion layer was added 2.0 Å outside the sphere surface, creating a two boundary problem. BEM and FDM solutions were compared to the analytical solvation energy for this geometry [35] to generate the convergence plot shown in Figure 4-8.

From the sphere convergence results, it is clear that the curved BEM method is able to achieve superior accuracy given the same amount of compute time as the finite difference method. For this problem, the FDM is limited to 2–3 digits of accuracy, even when using



(a) (b)

Figure 4-9: Computed solvation free energies, using curved BEM and FDM, for an HIV-1 substrate peptide (a) and the barnase–barstar complex (b). The absolute electrostatic solvation free energy is plotted as a function of compute time, and the selected discretizations used up to 4 GB of computer memory.

resolutions greater than 50 grid points per Angstrom. The limited ability of finite-difference methods to achieve high accuracy has been noted previously in the literature [150], although we obtain better than 1% accuracy on this sphere example. The accuracy of the curved BEM is limited to 5–6 digits given the settings selected in the FFTSVD matrix compression procedure. Additional accuracy can be achieved by increasing the size of the grids used to represent long-range interactions, at the expense of additional computational cost.

HIV-1 protease substrate peptide

To evaluate the method on a more complex example, the electrostatic solvation energy for a peptide derived from an HIV-1 substrate site was computed using BEM and FDM including salt and an ion-exclusion layer. The computed solvation energy was plotted as a function of compute time (Figure 4-9(a)). It is clear from examining Figure 4-9(a) that the solutions provided by the curved BEM implementation seem more converged than those obtained from the FDM. Although it is unclear whether the two methods will converge to the same answer for this complex geometry, the solution at the highest discretization levels for the curved BEM are changing by as little as 10^{-3} kcal/mol, while those from FDM are still changing on the order of tenths of kcal/mol.

Barnase–barstar complex

In order to be competitive with finite-difference methods, the curved boundary-element method presented here must be able to achieve high accuracy per unit compute time on large macromolecules, where the number of curved panels required to discretize the geometry can be large. To test the solver on a moderately sized protein system, we computed the solvation free energy of the barnase–barstar protein complex [77,151,152], a model binding system for which electrostatic interactions have been shown to be important [147,152–155]. In addition to an ion-exclusion layer, the problem geometry included four solvent-filled cavities inside the main dielectric boundary. A comparison between the BEM and FDM for computing the absolute solvation energy of this complex is shown in Figure 4-9(b). Even the finest BEM and FDM discretizations that can be solved on a computer with 4 GB of memory give answers that differ by 8–9 kcal/mol. Furthermore, it is difficult to establish whether the two methods will converge to the same answer. However, the curved BEM profile does appear to be relatively flat, even though the solution changed by approximately 0.2 kcal/mol between the two highest-resolution calculations.

As can be seen in Figure 4-9(b), even the lowest BEM discretization obtained for the barnase–barstar complex requires more compute time than the highest discretization used for the FDM. The timings for the FDM remain relatively constant across the presented problems because they depend primarily on the grid size. In contrast, the BEM requires more curved panels to discretize a larger molecular surface, resulting in significantly increased simulation cost.

The accuracy of the BEM scales with the panel density; accordingly, the larger barnase–barstar complex cannot be discretized at the same level as was feasible for the peptide example. The BEM-calculation solvation energies in Figures 4-9(a) and 4-9(b) exhibit similar curvature, and the "knees" of the two curves are separated by approximately a factor of ten in compute time. This difference is as expected considering the ratio of the surface areas for the peptide and barnase–barstar complex (952 $Å^2$ and 8019 $Å^2$ respectively). The level of FDM convergence might also be expected to suffer for larger problems due to decreasing grid resolution given the same number of grid points. Surprisingly, the FDM

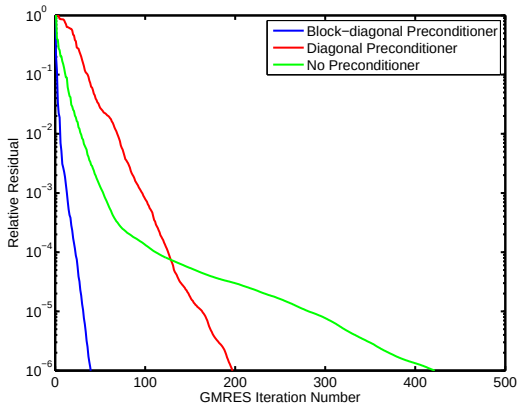


Figure 4-10: Comparison of preconditioning strategies when solving for the electrostatic solvation free energy of an HIV-1 protease substrate peptide discretized with 18,657 and 7,089 panels on the dielectric and ion-exclusion surfaces, respectively. The block-diagonal preconditioner significantly reduces the number of GMRES iterations required to solve the linear system of BEM equations to a relative residual of 10^{-6}.

appeared to lose less relative accuracy with increasing problem size as compared to the BEM. For the peptide and barnase–barstar solvation energies, the highest resolution FDM calculations were still changing by approximately 0.05 and 0.5 kcal/mol respectively. In the curved BEM results, they were changing by 0.001 and 0.2 kcal/mol, indicating a larger fold loss in convergence.

4.4.2 Importance of preconditioning

To demonstrate how effectively the block-diagonal preconditioner accelerates convergence of the iterative solution of the BEM equations, we repeated the solvation energy calculation for one discretization of the peptide example using several preconditioners. Specifically, we performed the calculation without preconditioning, with a purely diagonal preconditioner, and with the presented block-diagonal preconditioner. As shown in Figure 4-10, the number of GMRES iterations required to achieve a relative residual of 10^{-6} without preconditioning was 422. The purely diagonal preconditioner required 198 iterations, and the full block-diagonal preconditioner reduced this even further to 40 iterations. The block-diagonal preconditioner generally allows even complex geometries such as proteins to be solved to a relative residual of 10^{-6} in approximately 100 GMRES iterations or less.

4.4.3 Rigid electrostatic binding free energies

Another useful quantity often calculated using the LPBE model is the rigid electrostatic binding free energy between a pair of interacting molecules. One component of this quantity is the difference in solvation energy between the bound state and two unbound states where the binding partners are rigidly separated to infinity. This differential electrostatic solvation is added to the direct Coulombic interactions made between the partners in the bound state. To measure the role that LPBE solver accuracy plays in this class of calculations, as well as compare the curved BEM to FDM, we computed the rigid electrostatic binding free energies for the wild-type barnase–barstar complex and three experimentally characterized single mutants (E73Q in barnase, T42A and D39A in barstar) [152, 156, 157] that have been previously shown to have a significant effect on electrostatic binding calculations [153, 158–160]. These mutations were built into the wild-type barnase–barstar complex with minimal perturbation, where all atoms remained in the same position except at the site of mutation.

The results of these rigid electrostatic binding calculations are shown in Figure 4-11. For the wild-type barnase–barstar structure as well as the mutant complexes, the BEM calculations showed smaller changes in the computed energies with increasing problem discretization.

4.4.4 Differential rigid electrostatic binding free energies between mutants and wild type

Often, when comparing a set of protein mutations to identify those with improved electrostatic properties, one is more interested in the relative electrostatic rigid binding free energies as compared to wild type than the absolute binding energies themselves. To gauge the effect of solver accuracy on relative binding free energies, we calculated the difference in rigid electrostatic binding free energy between each mutant and the wild type at every level of problem discretization (Figure 4-12).

For all mutants studied, both methods appear to be converged to tenths of kcal/mol or better, and give very similar relative binding energies. Low discretizations of the FDM

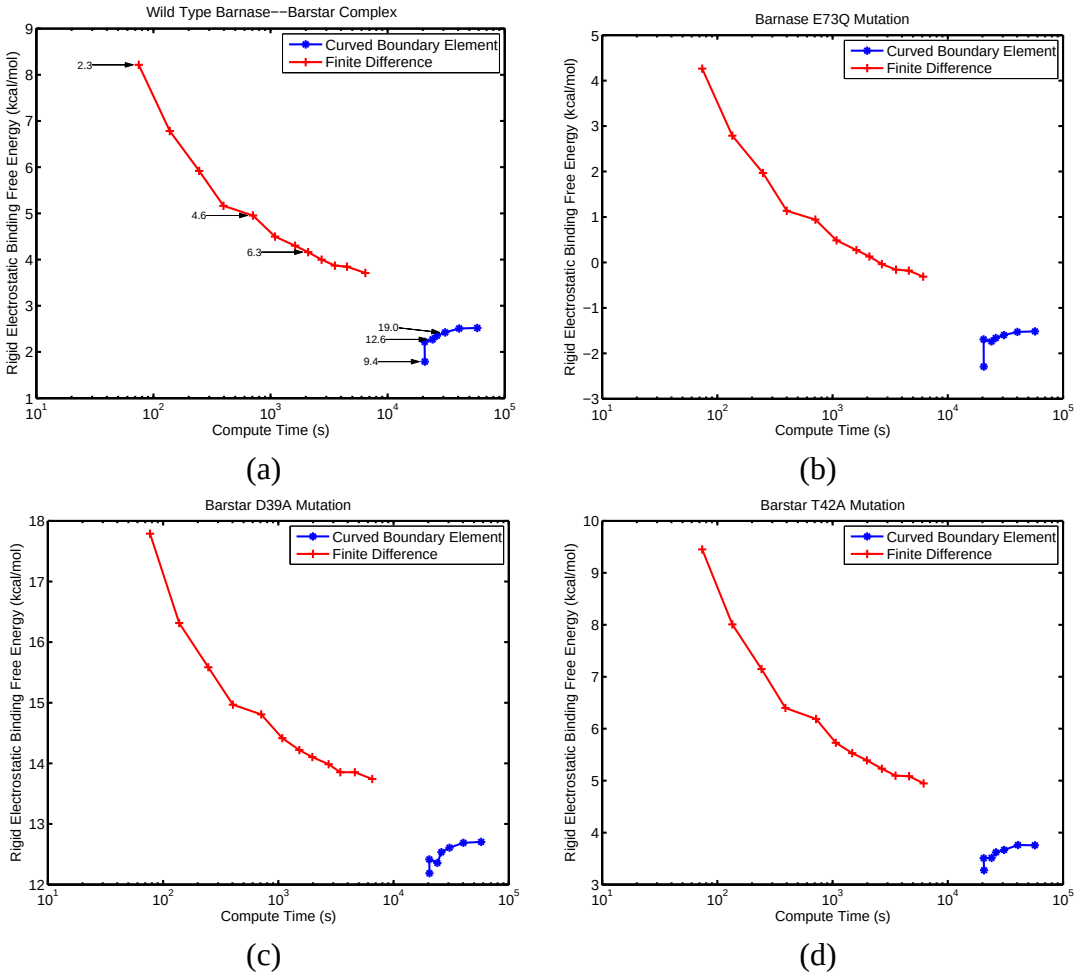


Figure 4-11: Comparison between curved BEM and FDM for computing the electrostatic component of the rigid binding free energy between the wild-type barnase–barstar complex (a), and three mutant complexes, E73Q in barstar (b), D39A in barnase (c), and T42A in barnase (d). The binding energy obtained is plotted as a function of the compute time required. In (a), several FDM and BEM results are labeled with their discretization level (grid points per Angstrom or panels per Angstrom^2, respectively).

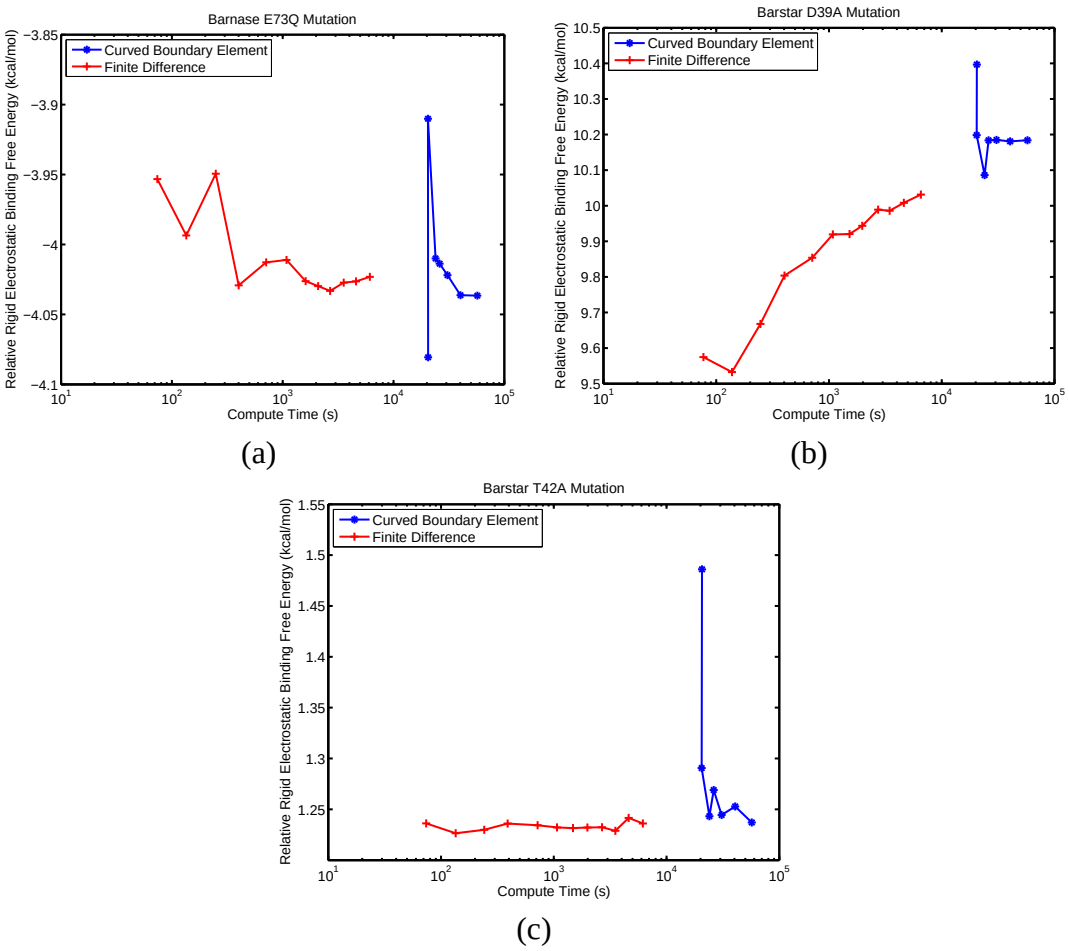


Figure 4-12: Comparison between curved BEM and FDM for computing relative rigid electrostatic binding energies between mutant and wild-type barnase–barstar complexes. Results are shown for the mutations E73Q in barstar (a), D39A in barnase (b), and T42A in barnase (c). The relative binding energy is plotted as a function of the compute time for the mutant complex rigid binding energy.

provide solutions very close to the final answer in a very short amount of time. This may be due to error cancellation because the mutant structures differ little from the wild type. For problems in which electrostatic energies are being compared between structures for which most atoms are located at identical positions, finite-difference methods may be a better choice than the boundary-element method presented here. Minimal-perturbation relative-binding calculations are often used when making predictions to improve protein binding or stability, especially in the field of protein design [2, 161, 162].

4.4.5 Non-rigid electrostatic binding free energies

The rigid binding model, although a useful approximation, is deficient in that it does not allow structural relaxation in the bound and unbound states. Consequently, a variety of methods have been presented in the literature for treating non-rigid effects in protein–protein binding using continuum electrostatics [14, 163]. One feature most techniques share is that there is no longer a direct correspondence between the majority of atomic coordinates in the bound and unbound states. As a result, we hypothesized that the FDM would no longer be able to take advantage of cancellation of error when computing non-rigid binding effects, and that the accuracy of the overall calculation would depend strongly on the ability to independently converge the solvation energy for each state. To test this idea, we implemented a crude non-rigid binding scheme involving independent minimization of the complex and unbound binding partners and a thermodynamic cycle to compute electrostatic energies. The non-rigid electrostatic binding energies for mutants were subtracted from those for the wild-type barnase–barstar complex to generate non-rigid relative binding energies.

As shown in Figure 4-13, the curved BEM method regains an accuracy advantage in non-rigid binding calculations. The curves in this plot resemble those from absolute binding energy calculations (Figure 4-11). The finite-difference solution does not appear to be well converged at low resolution, and seems to gradually approach the boundary-element solution.

Because grid cancellation could not be exploited in non-rigid binding to avoid reference state calculations in the FDM, we computed the solvation of each state independently allowing the protein complex or binding partners to fill the entire finite-difference grid. Therefore, when subtracting the solvation energies of binding partners from the bound complex, we were subtracting calculations solved at very different grid resolutions. To determine if this was responsible for the inability of FDM to converge relative non-rigid electrostatic binding energies, we repeated the calculation using fixed grid placement to ensure that the solvation energy of each state was computed at roughly the same number of grid points per Angstrom. However, this modification did not improve the ability of FDM to converge relative non-rigid binding energies (data not shown).

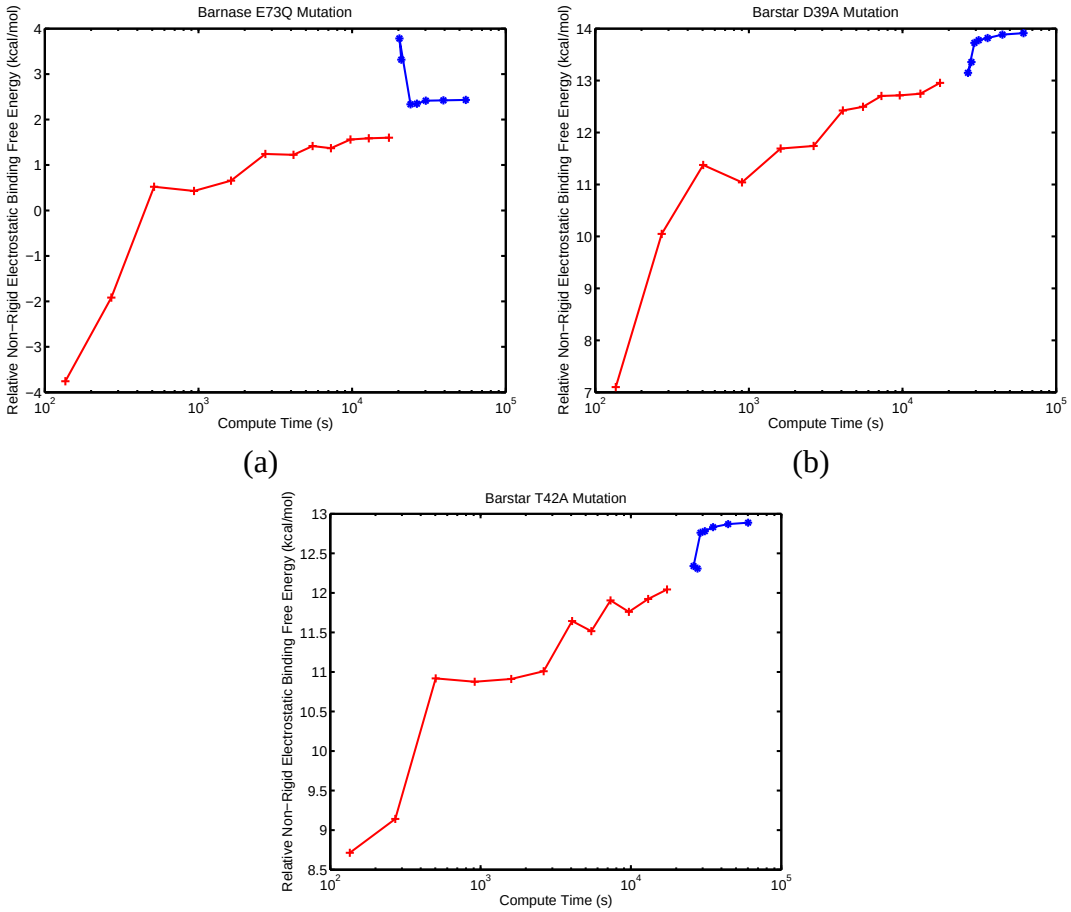


Figure 4-13: Comparison between curved BEM and FDM for computing relative non-rigid electrostatic binding energies between mutant and wild-type barnase–barstar complexes. Results are shown for the mutations E73Q on barstar (a), D39A on barnase (b), and T42A on barnase (c) The relative binding energy is plotted as a function of the compute time for the mutant complex non-rigid binding energy.

4.4.6 Multiple electrostatic solves for the same problem geometry

As shown in the previous Results sections, the curved BEM, although offering better convergence properties, is quite time consuming on large geometries such as proteins. The dominant computational cost in our implementation is compressing the integral operators using the FFTSVD algorithm, which primarily involves computing costly integrals over nearby curved panels. In contrast, the FDM requires very little time to initialize the system of linear equations and spends almost all compute time solving them. However, there exist several types of useful electrostatic calculations that involve multiple simulations of the same problem geometry; for these problems, the expensive BEM "setup" time can be amortized over all calculations.

One such example is charge optimization [34, 35, 147], which determines the optimal partial atomic charges for a ligand that minimize the electrostatic component of its binding free energy with a receptor molecule. In charge optimization, two geometries for the ligand are considered: the bound state, where it is complexed with the receptor molecule, and the unbound state, where it is isolated in solution. Each ligand charge is set to $+1e$ independently, leaving all others at zero, and one determines the difference in solvation potential at the ligand charge locations between the bound and unbound states by solving the LPBE twice. This produces the ligand desolvation matrix, an important component of the charge optimization equation [34, 35]. Overall, $2n$ solves of the LPBE are required, where n is the number of atoms in the ligand. When using the BEM, each state's integral operator only needs to be compressed once, and the compressed operator can be used to solve the n right-hand sides that only depend on the atomic charges.

To compare the performance of the curved BEM and FDM on a charge optimization problem, we computed the ligand desolvation matrix for barstar in the wild-type barnase–barstar complex. In total, 1403 simulations were performed in each of the bound and unbound states. In Table 4.1 we report the time required to compute the ligand desolvation matrix for three discretization levels of the finite-difference and curved boundary-element methods. The panel densities and grid spacings mentioned in Table 4.1 may be compared to the labeled points on the absolute binding free energy plot shown in Figure 4.4.3.

Method		
FDM	Grid Points Per Angstrom	Time (s)
	2.3	41,868
	4.6	637,930
	6.3	1,774,146
Curved BEM	Panels Per Angstrom^2	Time (s)
	9.4	755,343
	12.6	1,347,300
	19.0	2,024,024

Table 4.1: Compute time required to calculate the entries of the ligand desolvation matrix for barnase in the wild-type barnase–barstar complex. For both the curved BEM and FDM, the calculation was repeated at three discretization levels. For the curved BEM, the panel density reported is for all surfaces in the bound state geometry.

For the finer discretization BEM calculations, the compute time is comparable to that required for the finer FDM discretizations. Relating these discretization levels to the convergence plot suggests that for these multiple-solve problems, the BEM may offer superior accuracy for similar computational cost.

4.5 CONCLUSIONS

In conclusion, we have presented an implementation of the boundary-element method for linearized Poisson–Boltzmann continuum electrostatics that is capable of achieving high accuracy and solving the same topologies of dielectric boundaries, point charges, and salt regions that volume-based methods are capable of solving. Several techniques were employed to overcome several of the well-known practical limitations of the BEM. These included a general Green's-theorem integral formulation for multiple embedded regions, curved panel discretization with robust integration methods, and preconditioned Krylov subspace methods combined with matrix compression using the FFTSVD algorithm.

Comparing the performance of the curved BEM against a reference finite-difference solver identified types of calculations for which improved accuracy may be important. For example, when computing absolute electrostatic solvation free energies or the electrostatic component of rigid binding energies, the curved BEM method offers superior convergence properties. Even at the highest discretizations possible within 4 GB of computer mem-

ory, finite-difference methods did not appear to be converged, as the solutions continued to change significantly with increased expenditure of computing resources. However, when comparing differential rigid binding energies between mutant and wild-type protein complexes, even coarse finite-difference simulations sufficed to capture relative effects. This is not surprising considering that the local structural perturbations allow for cancellation of error. Relative rigid binding calculations with local geometry perturbations are prevalent in ranking the results of molecular design efforts [145], and finite-difference methods are an attractive tool for this class of computation. However, when non-rigid effects were introduced into the binding model, and the bound and unbound states were allowed to relax independently, finite-difference methods lost their convergence advantage. Therefore, as more sophisticated non-rigid models of binding are employed in ranking results of molecular design calculations, higher accuracy LPBE solvers such as the presented curved BEM may become necessary to make reliable predictions.

In the current implementation, the computational resources required to obtain solutions converged to tenths of kcal/mol on protein geometries are somewhat higher than what would be commonly available on a desktop workstation at this time. In order to compute a well converged protein solvation or binding energy in a few hours, a workstation with four processors and 4 GB of memory are currently required. Because the problem geometry is already represented essentially exactly, it is likely that the primary source of error in the method arises from the use of piecewise-constant representations of the surface variables. Higher-order basis functions may allow a significant reduction in the number of unknowns, and thus the required memory. However, two complications that may limit higher-order methods are that the numerical integrations are more time consuming, and that the compressibility of the discretized operator may decrease. It is not yet clear where the optimal trade-off lies between basis function complexity and these complications, and improvements in this area should be capable of reducing the time and memory usage of the curved BEM implementation to more accessible levels.

Chapter 5

A Co-optimization Approach for Optimizing Electrostatic Interactions between Biomolecules[1]

In this work we present a computationally efficient approach for calculating a molecular charge distribution that optimizes the electrostatic component of the free energy of binding to another molecule. The electrostatic optimization theory, developed over several years, can require substantial amounts of computer simulation prior to optimization, and our approach can in some circumstances greatly reduce the required computational expense. The approach uses an implicit matrix form for the objective function Hessian that directly couples the optimization to a numerical method used to simulate molecular electrostatics. The implicit-Hessian method can be applied to unconstrained as well as constrained optimization problems, and results illustrate not only that the method scales advantageously but also that realistic problems can be solved.

5.1 INTRODUCTION

A natural question in molecular design problems is whether a candidate design, or ligand, is optimal for binding the target, and if not, what modifications might be made to

[1]To be submitted [164]; some portions of this chapter have been published previously [165, 166].

improve binding affinity or specificity. If one accepts shape complementarity as a given requirement, close packing interactions are relatively fixed among candidates; however, the long-range electrostatic interactions possess highly variable energetics [34]. Furthermore, because the binding reaction sacrifices electrostatic interactions between solvent and ligand and between solvent and target for ones between target and ligand, the electrostatic contribution to binding free energies is not particularly intuitive. Therefore, the important but challenging goal is to design optimal electrostatic interactions to balance these competing terms. Questions in molecular biology regarding the evolution of biomolecules to serve specific functions or bind targets with high affinity and specificity, may also be interpreted as questions regarding optimization of a particular objective function [167].

The Tidor laboratory has developed over several years a theoretical framework for analyzing the optimality of the electrostatic interactions between molecules [34, 35, 147, 167–169]. The framework rests on a linear-response model of solute–solvent electrostatic interactions, and has generally been implemented using continuum electrostatic models (see, for example, references [3, 9]). The electrostatic binding free energy between spherical binding partners was studied, and it was shown that the multipole coefficients representing one partner's charge distribution could be optimized [34]. The convexity of the electrostatic component of a rigid binding free energy was demonstrated, and the approach was extended to allow not only multipole coefficients as a basis set for optimization, but also discrete point charges and an "inverse-image" basis [35]. Later work showed that under the continuum model and a small set of assumptions, the electrostatic binding free energy would be favorable for many realistic systems [168]. In addition, a measure of electrostatic specificity was defined and studied in the context of the affinity optimization framework [170].

The electrostatic optimization theory has been applied to study a number of molecular systems. Chong *et al.* [154] applied the original model [34] to barnase and found that small sets of biochemically reasonable charge distributions could closely reproduce the computed optimal charge distribution. Lee and Tidor studied the extremely tight-binding partners barnase and barstar [147, 167] and suggested that barstar is electrostatically optimized to bind to barnase. Other researchers have since reported contradictory results, but

the simulation methodologies have differed in significant details [155, 160]. In particular, the electrostatic simulations have been conducted using van der Waals surfaces rather than solvent-excluded surfaces to represent the solute–solvent dielectric interface; this choice appears to have been made so that the computed electrostatic binding free energies best match the total experimental binding free energies. In another application of the optimization theory, Kangas and Tidor studied the enzyme *B. subtilis* chorismate mutase [169], and their analysis indicated a particularly promising modification to improve the binding affinity of a transition-state analog inhibitor—the replacement of a carboxylate group by a nitro group. Mandal and Hilvert synthesized the proposed inhibitor and the resulting ligand bound the enzyme more tightly, in agreement with the computational analysis, to become the tightest-binding chorismate mutase inhibitor in the literature [171]. Sulea and Purisima have studied charge optimization to study problems ranging from cation–protein binding to the optimization of protein–protein interfaces [172]; in addition, they have explored the use of the charge optimization framework as a means to identify "hot spots" for binding [173]. Sims *et al.* studied two protein kinases, protein kinase A (PKA) and cyclin-dependent kinase 2 (CDK2), and several inhibitors [174]. Green and Tidor have applied charge optimization theory to two systems [145,175]. In one, they demonstrated that glutaminyl-tRNA synthetase is optimized for its substrates [175]. More recently, they proposed optimization-theory-based mutations to 5-Helix, which inhibits HIV-1 membrane fusion by gp41 [145]. Armstrong *et al.* have studied several inhibitors of neuraminidase and simulated a lead-optimization approach [176]. Very recently, Gilson explored a statistical-mechanical approach to extend the theory to allow the optimization of flexible ligands [177].

The application of electrostatic optimization theory is frustrated, however, by a computational difficulty that arises while optimizing large numbers of charges, or while exploring the optimality of multiple sets of binding partners. Implementations of the charge optimization process have required multiple solutions of the Poisson or linearized Poisson–Boltzmann equation to obtain complete information about the quadratic objective function prior to optimization [154]. A constant number of simulations is typically executed for each charge to be optimized [35, 154]; although this precomputation cost scales linearly, the proportionality constant can be quite large. In this chapter, we present a novel PDE-constrained

optimization approach that avoids the initial computation, and by doing so reduces the computational expense to be effectively independent of the number of charges [165, 166]. The approach achieves this high efficiency by breaking the black-box abstraction of the electrostatics solver and coupling the electrostatics simulation method directly to the optimization equations. Our work specifically focuses on boundary-element method (BEM) simulation coupled to optimization, but an analogous approach can be implemented using popular finite-difference methods [18, 20, 22, 26, 31].

The remainder of this chapter is outlined as follows. The following section briefly introduces the theory of electrostatic optimization as presented by Lee, Kangas, and Tidor, numerical methods for calculating electrostatics in a continuum model, and the coupled optimization-simulation, or co-optimization, approach that is the focus of this work. Section 5.3 describes several important facets of the implementation for the co-optimization method. Section 5.4 illustrates the method's application to several model problems and to a realistic problem in biomolecule electrostatic optimization. Section 5.5 concludes the chapter with a summary and brief discussion of promising future applications for the co-optimization technique.

5.2 THEORY

5.2.1 The Continuum Electrostatic Model and Numerical Simulation with Boundary-Element Methods

The electrostatic contribution to a binding free energy is commonly estimated using a thermodynamic cycle such as that shown in Figure 1. Three of the steps involve the transfer of a molecule or complex from a low-dielectric environment to the solvent. The difference in free energy between the two states is the solvation free energy, and the electrostatic component of this free energy is commonly estimated using a continuum electrostatic model [3,9]. In this section, we present a continuum model and a numerical method to compute the electrostatic solvation free energy of a solute. The presentation is directed towards expressing this change in free energy as a simple operator expression applied to the distribution of

charge in the solute. Figure 5-1 illustrates one continuum model. The molecule–solvent

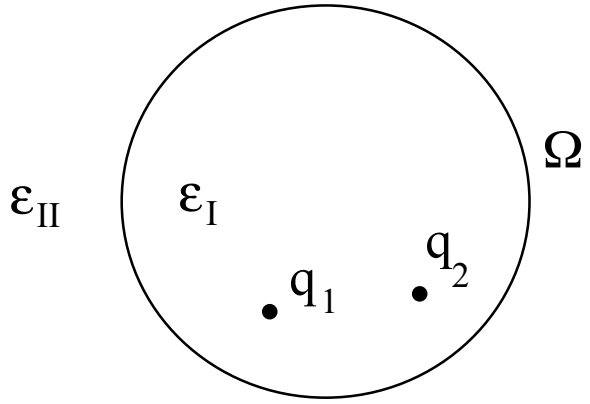


Figure 5-1: A continuum model for estimating the electrostatic component of a solute's solvation free energy. In this Figure, ε_I and ε_{II} represent the low-dielectric protein region and the high-dielectric solvent region. Ω is the boundary between the dielectric regions. q_1 and q_2 are two representative point charges in the protein.

boundary Ω, commonly taken to be the Richards molecular surface [49], separates the molecular interior, region I, from the solvent exterior, region II. The interior is modeled as a homogeneous dielectric with low dielectric constant ε_I and a charge distribution; in this work, we assume that the distribution consists of n_c discrete point charges, the i^{th} of which is located at r_i and has value q_i. The electrostatic potential in region I satisfies a Poisson equation:

$$\nabla^2\varphi_I(r) = -\sum_{i=1}^{n_c}\frac{q_i}{\varepsilon_I}\delta(r-r_i). \tag{5.1}$$

The solvent region is modeled as a homogeneous dielectric with high dielectric constant ε_{II} in which the electrostatic potential satisfies the Laplace equation

$$\nabla^2\varphi_{II}(r) = 0, \tag{5.2}$$

for non-ionic solutions, or the linearized Poisson–Boltzmann equation

$$\nabla^2\varphi_{II}(r) = \kappa^2\varphi_{II}(r), \tag{5.3}$$

where κ is the inverse Debye length, for dilute ionic solutions. The continuity of the potential and normal displacement furnish boundary conditions for both regions.

This set of coupled partial differential equations (PDE) can be transformed into boundary-integral equations [17, 21, 23, 28, 83, 178] and solved numerically using the boundary-element method (*e.g.*, reference [54]). Here, simply to obtain the general operator expression for the electrostatic solvation free energy, we present one widely used integral formulation, the apparent surface charge (ASC) method, that models electrostatics in non-ionic solutions [16, 17, 19, 36, 178] and a simple boundary-element method to solve it. More complex integral formulations allow the treatment of dilute ionic solutions [21, 23] and geometries with multiple dielectric regions [83]. We describe the ASC formulation to introduce the operators inherent to boundary-element simulations in electrostatics; the remainder of the theory section will apply these operators.

In the apparent surface charge method, one solves an equivalent problem with uniform dielectric constant ε_I throughout space and finds a fictitious distribution of charge on the surface that reproduces the continuity conditions of the original problem. This fictitious surface charge, which we denote by $\sigma_p(r)$, for a point r on the surface, satisfies the integral equation

$$\frac{\varepsilon_I+\varepsilon_{II}}{2\varepsilon_I(\varepsilon_I-\varepsilon_{II})}\sigma_p(r)+\oint_\Omega \frac{\partial}{\partial n(r)}\frac{\sigma_p(r')dA'}{4\pi\varepsilon_I||r-r'||}=-\frac{\partial}{\partial n(r)}\sum_{i=1}^{n_c}\frac{q_i}{4\pi\varepsilon_I||r-r_i||}. \tag{5.4}$$

Once the apparent surface charge is found, the reaction potential induced at a point r_I in the solute by polarization of the solute in response to the solute charges may be computed according to

$$\varphi_R(r_I)=\int_\Omega \frac{\sigma_p(r')}{4\pi\varepsilon_I||r_I-r'||}d\Omega. \tag{5.5}$$

The vector of reaction potentials at the charge locations can therefore be seen as the image of the charge distribution under three linear mappings:

$$\varphi_R=M_3M_2^{-1}M_1q. \tag{5.6}$$

The first, M_1, maps the charge distribution to the induced field at the dielectric boundary; that is, applied to q it gives the RHS in Equation 5.4. The operator M_2^{-1} maps the induced field to the induced surface charge, calculating $\sigma_p(r)$ given the RHS in Equation 5.4. Fi-

nally, the integral operator M_3 maps the induced surface charge to the reaction potentials at the charge locations.

To solve Equation 5.4 for general geometries and charge distributions, one introduces a set of basis functions defined on the surface, represents the unknown function $\sigma(r)$ approximately as a weighted combination of the basis functions, and chooses the weights to make the approximate solution satisfy the discretized integral equation as closely as possible in some metric. Usually, it is convenient to discretize the surface into a set of surface, or boundary, elements, before defining the basis functions. Commonly, these elements are planar triangles [21, 178], although curved-element discretizations have been described by several groups [19, 23, 38]. Using a set of n_p piecewise-constant basis functions defined such that

$$\chi_i(r) = \begin{cases} 1 & \text{if } r \text{ is on panel } i \\ 0 & \text{otherwise,} \end{cases} \tag{5.7}$$

and using a Galerkin method [54] in which the inner integral is evaluated via one-point quadrature, one obtains the dense linear system $M_2 x = M_1 q$, in which x_i represents the unknown weight on the i^{th} basis function, and the system entries are

$$M_{2,ii} = \frac{\hat{\varepsilon}}{2\varepsilon_I}\alpha_i + \oint_{\text{panel } i} \frac{\partial}{\partial n(r)} \frac{\alpha_i \, dA}{4\pi\varepsilon_I ||r - r_{c_i}||} \tag{5.8}$$

$$M_{2,ij} = \int_{\text{panel } i} \frac{\partial}{\partial n(r)} \frac{\alpha_j \, dA}{4\pi\varepsilon_I ||r - r_{c_j}||} \quad (i \neq j) \tag{5.9}$$

$$M_{1,ij} = -\int_{\text{panel } i} \frac{\partial}{\partial n(r)} \frac{q_j \, dA}{4\pi\varepsilon_I ||r - r_j||}, \tag{5.10}$$

where α_i denotes the area of panel i, $\hat{\varepsilon} = \frac{\varepsilon_I + \varepsilon_{II}}{\varepsilon_I - \varepsilon_{II}}$, $n(r)$ denotes the outward normal at r, r_{c_i} denotes the centroid of panel i, and $\oint$ denotes a Cauchy principal value integral. The approach presented here differs slightly from, and offers improved accuracy relative to, the centroid-collocation BEM for the ASC, which essentially approximates the outer Galerkin integral using one-point quadrature [36, 37].

Because the charge distribution is assumed to be a set of discrete point charges, the difference in electrostatic free energy between the uniform ε_I domain and the mixed dielectric

problem is simply

$$\Delta G = \frac{1}{2}\varphi_R^T q, \tag{5.11}$$

where φ_R denotes the vector of reaction potentials computed at the n_c charge locations. The operators M_1, M_2, and M_3 are all dense matrices when discretized, and the electrostatic component of the solvation free energy can be written as

$$\Delta G_{solv}^{0,es} = q^T S q \tag{5.12}$$

where we have defined the solvation matrix $S = \frac{1}{2} M_3 M_2^{-1} M_1$.

Because solving dense matrix equations with n unknowns using LU factorization requires $O(n^2)$ memory and $O(n^3)$ time, more efficient methods have been developed to reduce these demands to linear or near-linear scaling [28, 83, 96, 178]. These methods combine Krylov-subspace iterative methods [146] such as GMRES [82] with fast, approximate algorithms to apply the discretized integral operator matrix to a vector. The fast multipole method [96, 97] is one such algorithm. This work reports results computed using the precorrected-FFT algorithm [102] and the FFTSVD algorithm [83].

5.2.2 Electrostatic Optimization

Writing down the electrostatic contribution to each of the steps in Figure 5-2, we have

$$\Delta G_{bind}^{0,es} = -\Delta G_{solv,L}^{0,es} - \Delta G_{solv,R}^{0,es} + \Delta G_{bind,low\,\varepsilon}^{0,es} + \Delta G_{solv,L-R}^{0,es}, \tag{5.13}$$

where the solvation free energies for the ligand, receptor, and complex are denoted by the subscripts L, R, and $L-R$, and the Coulomb interaction energy in ε_I between the partners is written $\Delta G_{bind,low\,\varepsilon}^{0,es}$. Using Equation 5.12 for the three solvation terms, one obtains

$$\Delta G_{bind}^{0,es} = -q_L^T L_{unbound} q_L - q_R^T R_{unbound} q_R + (G q_R)^T q_L + q_C^T C_{bound} q_C, \tag{5.14}$$

where q_L and q_R denote the ligand and receptor charge distributions, $q_C = (q_L, q_R)^T$ is the union of these distributions, L, R, and C denote the appropriate desolvation penalty

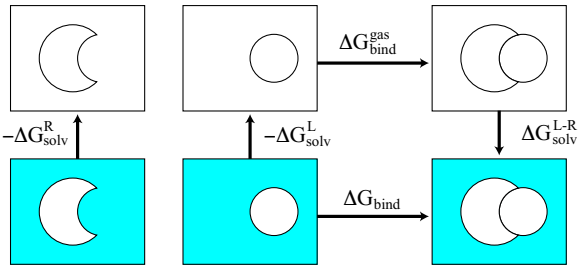


Figure 5-2: A thermodynamic cycle for estimating binding free energies. The shaded region on the lower set of panels represent aqueous solvent. The upper panels represent a uniform low dielectric with zero ionic strength throughout.

matrices, and the electrostatic component of the low-dielectric binding free energy, which is simply the Coulomb energy between the ligand and receptor charge distributions, has been written $(Gq_R)^T q_L$, where the matrix G maps receptor charge values to Coulomb potentials at the ligand-charge locations given the bound-state geometry.

We consider the variational electrostatic binding free energy $\Delta G^{0,var}_{bind}$ [35], which is the portion of $\Delta G^{0,es}_{bind}$ that is dependent on the ligand charges. We therefore drop the second term in Equation 5.14 and remove the contribution of the final term that depends only on the receptor charges. Then Equation 5.14 can be rewritten as

$$\Delta G^{0,var}_{bind} = -q_L^T L_{unbound} q_L + q_L^T L_{bound} q_L + q_R^T G^T q_L + q_R^T C^{L,R}_{bound} q_L. \tag{5.15}$$

The final two energy terms are both linear in the ligand charge values, and the expression

$$c = Gq_R + C^{L,R}_{bound} q_R, \tag{5.16}$$

which represents the total receptor-charge-induced field at the ligand charges in the bound state, may be used to further simplify Equation 5.15:

$$\Delta G^{0,var}_{bind} = q_L^T L_{bound} q_L - q_L^T L_{unbound} q_L + c^T q_L \tag{5.17}$$

$$= q_L^T \left(L_{bound} - L_{unbound}\right) q_L + c^T q_L. \tag{5.18}$$

Kangas and Tidor showed that the difference of the two symmetric positive definite operators $L_{unbound}$ and L_{bound} is nonnegative definite if one assumes that the ligand binds

rigidly [35]. The variational electrostatic binding free energy is therefore a convex function with respect to the ligand charge distribution, and there exists a unique minimal free energy.

5.2.3 Co-Optimization: Coupling Simulation and Optimization

In this section we introduce the essential idea behind the co-optimization approach to biomolecule electrostatic optimization. Suppose that we wished to find the minimum of the function

$$\text{minimize} \quad x^T Lx + c^T x, \tag{5.19}$$

where $L \in \Re^{n_c \times n_c}$ is symmetric, non-negative definite, and can be written as

$$L = M_3 M_2^{-1} M_1, \tag{5.20}$$

where $M_3 \in \Re^{n_c \times m}$, $M_2 \in \Re^{m \times m}$, and $M_1 \in \Re^{m \times n_c}$ are dense matrices, and $m \gg n_c$. The minimizer x^* is found where the gradient is zero:

$$2Lx^* = -c. \tag{5.21}$$

If the constituent matrices M_i are too large to be stored in memory, but have some properties enabling their actions on vectors to be computed approximately, then one could calculate L one column at a time by repeatedly solving

$$M_2 y_i = M_1 e_i, \tag{5.22}$$

using Krylov-subspace iterative methods [146]; here e_i represents the canonical i^{th} unit vector with 1 at position i and zero everywhere else. Krylov methods for solving linear systems generate a sequence of increasingly accurate approximate solutions $\{\hat{y}_i^1, \hat{y}_i^2, \ldots\}$, and at the k^{th} iteration the iterate lies in the span first k Krylov vectors

$$(M_1 e_i, M_2 M_1 e_i, \ldots, M_2^{k-1} M_1 e_i).$$

Once an acceptable approximation is found, one multiplies $\hat{y}_i$ by M_3 to obtain the i^{th} column of L. Such an approach obviously requires n_c solutions of Equation 5.22 before the minimum can be found. A simple alternative to finding x^* without column-by-column calculation of L would be to use Krylov-subspace methods to solve $Lx = b$. While forming the power series $\{b, Lb, L^2b, \ldots\}$, however, every application of L requires a solution of Equation 5.22. If L is difficult to precondition, or if multiple optimizations are to be performed, the total number of solves of Equation 5.22 can quickly approach or exceed n_c. The difficulty is that applying L requires an application of M_2^{-1}, which in turn requires an iterative solve.

Directly applying M_2 does not require an iterative solve, however, and the co-optimization has been designed to solve $2Lx = -c$ without ever actually applying M_2^{-1}. The co-optimization idea is to solve an *equivalent system* of equations that recovers the same optimizer x^* by introducing an auxiliary variable y^*. For x^* to solve Equation 5.21, the two variables must satisfy

$$M_1 x^* = M_2 y^* \tag{5.23}$$

$$2M_3 y^* = -c, \tag{5.24}$$

which can be written in matrix form as

$$\begin{bmatrix} 0 & 2M_3 \\ M_1 & -M_2 \end{bmatrix} \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} -c \\ 0 \end{bmatrix}. \tag{5.25}$$

The form $M_3 M_2^{-1} M_1$ resembles the Schur complement of a 2-by-2 block matrix reduced during block factorization; that is, to solve

$$\begin{bmatrix} A & B \\ C & D \end{bmatrix} \begin{bmatrix} x^* \\ y^* \end{bmatrix} = \begin{bmatrix} e \\ f \end{bmatrix}, \tag{5.26}$$

one can first solve the Schur system

$$(A - BD^{-1}C)x = e - BD^{-1}Cf \tag{5.27}$$

for x and then back-substitute to calculate y. Letting $A = 0$ and $f = 0$ makes the parallel clear. For this reason, we refer to the co-optimization approach as a reverse-Schur complement method.

5.2.4 Biomolecule Electrostatic Co-Optimization

In this section we present methods for solving biomolecule electrostatic optimization problems using the co-optimization approach. We formulate three of the most common types of electrostatic optimization problems: unconstrained, linear-equality constrained, and box-inequality constrained. Section 3.2 will describe preconditioning techniques for the resulting co-optimization systems.

Unconstrained Optimization

To solve the unconstrained program

$$\text{minimize} \quad q^T \left(L_{bound} - L_{unbound}\right) q + c^T q \tag{5.28}$$

it suffices to set the objective gradient to zero [35]:

$$2\left(L_{bound} - L_{unbound}\right) q = -c. \tag{5.29}$$

This system may be solved analogously to Equation 5.21, because both of the solvation matrices have the Schur complement operator form. Here, two reverse Schur complements are needed. The two sets of introduced auxiliary variables are the basis-function weights for the bound and unbound boundary-element problems; hence, using the apparent surface charge integral formulation to solve the electrostatic problems, the resulting system is of the form

$$\begin{bmatrix} 0 & 2M_3^b & -2M_3^u \\ M_1^b & -M_2^b & 0 \\ M_1^u & 0 & -M_2^u \end{bmatrix} \begin{bmatrix} q^* \\ x^{b,*} \\ x^{u,*} \end{bmatrix} = \begin{bmatrix} -c \\ 0 \\ 0 \end{bmatrix}. \tag{5.30}$$

Note that this system of equations solves three problems simultaneously: the optimization problem, the bound-state electrostatic problem, and the unbound-state electrostatic problem.

Optimizing Problems with Linear Equality Constraints

It is often desirable to impose linear equality constraints in the optimization. One may wish to force the total ligand charge, or a subset of charges, to have a particular sum (for instance, on an amino-acid side chain) [167, 169, 173, 174]. A solution to the resulting optimization problem

$$\begin{aligned} &\text{minimize} \quad q^T \left(L_{bound} - L_{unbound}\right) q + c^T q \\ &\text{subject to} \quad A_c q = b \end{aligned} \tag{5.31}$$

may be found using a single linear solve, because the optimality conditions are linear. Typically, the constraint matrix A_c has entries that are either zero or one, and the right-hand-side vector b has integer entries. The co-optimization system to be solved is

$$\begin{bmatrix} 0 & A_c^T & 2M_3^b & -2M_3^u \\ A_c & & & \\ M_1^b & & -M_2^b & \\ M_1^u & & & -M_2^u \end{bmatrix} \begin{bmatrix} q^* \\ \lambda^* \\ x^{b,*} \\ x^{u,*} \end{bmatrix} = \begin{bmatrix} -c \\ b \\ 0 \\ 0 \end{bmatrix}. \tag{5.32}$$

Optimizing Problems with Linear Inequality and Equality Constraints

Bound constraints are often imposed on each charge to ensure that the calculated charges are physically reasonable [167, 169, 173, 174]. The optimality conditions for inequality-constrained problems are nonlinear, and solution methods for such problems are correspondingly more complex. The linearly constrained quadratic program to be solved is

$$\begin{aligned} &\text{minimize} \quad q^T \left(L_{bound} - L_{unbound}\right) q + c^T q \\ &\text{subject to} \quad A_c q = b \\ &\qquad \text{and} \quad m_i \leq q_i \leq M_i, \forall i \in \{1, \ldots, n_c\}, \end{aligned} \tag{5.33}$$

where m_i and M_i represent the lower and upper bounds on the value for the i^{th} charge. Defining $L = L_{bound} - L_{unbound}$, we transform Equation 5.33 into the standard form for a quadratic program,

$$\begin{aligned} \text{minimize} \quad & y^T Q y + d^T y \\ \text{subject to} \quad & Ay = h \\ \text{and} \quad & y \geq 0 \end{aligned} \tag{5.34}$$

with the substitutions

$$m + t = \quad q,\ t \geq 0 \tag{5.35}$$

$$q + r = \quad M,\ r \geq 0 \tag{5.36}$$

$$y = \quad \begin{bmatrix} t \\ r \end{bmatrix} \tag{5.37}$$

$$d = \quad c + Lm \tag{5.38}$$

$$h = \quad \begin{bmatrix} b - A_c m \\ M - m \end{bmatrix} \tag{5.39}$$

$$Q = \quad \begin{bmatrix} L & 0 \\ 0 & 0 \end{bmatrix} \tag{5.40}$$

$$A = \quad \begin{bmatrix} A_c & \\ I & I \end{bmatrix} \tag{5.41}$$

$$\tag{5.42}$$

Because the objective is convex and the constraints are linear, the program of Equation 5.34 satisfies a constraint qualification [179], and consequently to find a global minimizer it suffices to find a primal vector y^*, a Lagrange multiplier vector λ^*, and a dual slack vector

s^* that satisfy the Karush–Kuhn–Tucker (KKT) conditions:

$$\begin{aligned} s^* &= 2Qy^* + d - A^T\lambda^* && (5.43)\\ Ay^* &= h && (5.44)\\ 0 &= y_i^* s_i^* \ \forall i \in \{1,\ldots,2n_c\} && (5.45)\\ (y^*, s^*) &\geq 0. && (5.46) \end{aligned}$$

To find such a set of vectors, we use a primal-dual interior-point method described in reference [180]. Such methods calculate (y^*, λ^*, s^*) using a modified Newton–Raphson method, finding the roots of the vector-valued function

$$F(y,\lambda,s) = \begin{bmatrix} 2Qy + d - A^T\lambda - s \\ Ay - b \\ Ys \end{bmatrix}, \quad (5.47)$$

where Y is a diagonal matrix with $Y_{i,i} = y_i$. The steps are scaled to ensure that Equation 5.46 holds for every iterate and biased to keep the pairwise products $y_i s_i$ approximately equal [180].

The Newton–Raphson step at iteration k is computed by linearizing F about the current iterate and solving $J\Delta x = -F + \xi$, where J is the Jacobian at the current iterate, F is the current function value, ξ biases the step, and Δx is the computed step. For F of the form of Equation 5.47, we solve the modified Newton–Raphson equation

$$\begin{bmatrix} 2Q & -A^T & -I \\ A & 0 & 0 \\ S^k & 0 & Y^k \end{bmatrix} \begin{bmatrix} \Delta y^{k+1} \\ \Delta\lambda^{k+1} \\ \Delta s^{k+1} \end{bmatrix} = \begin{bmatrix} -d + s^k - 2Qy^k + A^T\lambda^k \\ b - Ay^k \\ -Y^k S^k e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \hat{\sigma}\frac{(y^k)^T s^k}{2n_c} \end{bmatrix}, \quad (5.48)$$

where the second term on the right-hand side is the bias that keeps the products $y_i s_i$ approximately equal. An iterate (y^k, λ^k, s^k) that satisfies the equality constraints and satisfies $y_i s_i = \tau \ \forall i$ for some positive τ is said to be on the central path [180], and optimization is most rapid close to this path. Two reverse-Schur complements unfold Equation 5.48,

resulting in the system

$$\left[\begin{array}{cc|c|c|c|cc} 0 & 0 & -A_c^T & -I & -I & 2M_3^b & -2M_3^u \\ 0 & 0 & 0 & -I & & 0 & 0 \\ \hline A_c & 0 & 0 & 0 & 0 & 0 & 0 \\ I & I & 0 & 0 & 0 & 0 & 0 \\ \hline S^k & & 0 & 0 & Y^k & 0 & 0 \\ \hline M_1^b & 0 & 0 & 0 & 0 & -M_2^b & 0 \\ M_1^u & 0 & 0 & 0 & 0 & 0 & -M_2^u \end{array}\right] \left[\begin{array}{c} \Delta t^{k+1} \\ \Delta r^{k+1} \\ \hline \Delta\lambda_c^{k+1} \\ \Delta\lambda_t^{k+1} \\ \hline \Delta s^{k+1} \\ \hline \Delta x^{k+1,b} \\ \Delta x^{k+1,u} \end{array}\right] = \tag{5.49}$$

$$\left[\begin{array}{c} -d - 2L(m+t^k) + s_t^k + A_c^T\lambda_c^k + \lambda_t^k \\ s_r^k + \lambda_t^k \\ \hline b - A_c m - A_c t^k \\ M - m - t^k - r^k \\ \hline -Y^k S^k e + \hat{\sigma}\frac{(y^k)^T s^k}{2n_c} e \\ \hline -M_1^b(t^k+m) - M_2^b x^{k,b} \\ -M_1^u(t^k+m) - M_2^u x^{k,u} \end{array}\right],$$

where we have denoted the Lagrange multipliers of the two block rows of A as λ_c and λ_t, respectively.

5.2.5 Co-Optimization Method Analysis

Numerical calculation of the explicit Hessian via repeated solution of the linearized Poisson–Boltzmann problem produces a matrix L that contains minor, unphysical asymmetry, and can be highly ill-conditioned. Symmetrizing the Hessian with the simple rule $L \leftarrow (L + L^T)/2$ frequently produces a matrix with very small negative eigenvalues. The eigenspace corresponding to unphysical eigenvalues is commonly either removed from the optimization search space, or heavily penalized [167, 170]. Such regularization methods are not feasible for implicit-Hessian approaches because the matrix entries are not explicitly available, and every multiplication by L is expensive. Although the implicit-Hessian method has not been analyzed completely yet, one may argue that co-optimization linear systems

have spectral properties that tend to favor regularized approximate solutions when they are solved via Krylov subspace methods. The argument is based on nearly-ideal preconditioning of the unconstrained co-optimization system, and the analysis resembles the approach of De Sturler and Liesen [181], which drew in turn from work by Murphy *et al.* [182].

Consider solving the unconstrained optimization problem introduced in Section 5.2.3, and assume that we know the exact inverse M_2^{-1} of the dense matrix M_2. This matrix, which could be used to ideally precondition the BEM system, can be used to design a preconditioner for Equation 5.25 . The preconditioner

$$P = \begin{bmatrix} I & \\ & M_2^{-1} \end{bmatrix} \tag{5.50}$$

produces the preconditioned system matrix

$$PA = \begin{bmatrix} 0 & 2M_3 \\ M_2^{-1}M_1 & -I \end{bmatrix}. \tag{5.51}$$

We now show that if L is nonsingular, the preconditioned matrix PA has $2n$ or $2n+1$ eigenvalues. The eigenvalue equation for PA is

$$\begin{bmatrix} & 2M_3 \\ M_2^{-1}M_1 & -I \end{bmatrix} \begin{bmatrix} u \\ v \end{bmatrix} = \lambda \begin{bmatrix} u \\ v \end{bmatrix}. \tag{5.52}$$

The second equation gives the relation

$$M_2^{-1}M_1 u = (1+\lambda), \tag{5.53}$$

which can be substituted into the first equation to give an eigenvalue equation for $L = M_3 M_2^{-1} M_1$:

$$M_3 M_2^{-1} M_1 u = \frac{1}{2}\lambda(1+\lambda)u. \tag{5.54}$$

Therefore every eigenvalue μ of L is associated with two eigenvalues λ_+ and λ_- of the preconditioned matrix PA, and these eigenvalues can be obtained by solving the quadratic

equation:

$$\lambda = \frac{-1 \pm \sqrt{1+8\mu}}{2}. \tag{5.55}$$

If L is nonsingular, this relation gives $2n$ distinct eigenvalues for PA. To deduce the remainder of the spectrum, note that M_3 can have rank no greater than n; assuming it has full rank (which it must, for L to be nonsingular), its nullspace is of rank $m-n$. Picking an arbitrary normalized vector v from this subspace, it is clear that $(0, v^T)^T$ is an eigenvector of the preconditioned matrix, with unity as the corresponding eigenvalue. Note that the largest magnitude eigenvalues of L are mapped to the largest magnitude eigenvalues of the preconditioned system, subject to a square-root scale and shift. As a result, the dominant search directions will be explored during the early Krylov iterations.

In general, M_2^{-1} is not available; if instead the BEM preconditioner is written $M_2^{-1} + E$, where E is the perturbation from the ideal preconditioner, the perturbation from the preconditioned reverse-Schur system is

$$\begin{bmatrix} 0 & 0 \\ EM_1 & -EM_2 \end{bmatrix}. \tag{5.56}$$

If $||E||$ is small, the eigenvalues of the inexactly preconditioned system might be expected to lie close to those from the exactly preconditioned system, depending on the condition number of the eigenvector matrix [183]. However, such an analysis has not yet been performed.

5.3 IMPLEMENTATION

5.3.1 Preconditioning

The co-optimization approach requires the solution of one or more large linear systems with block structure in which several of the largest blocks are dense and cannot be stored explicitly. As discussed in Section 5.2.1, BEM problems are commonly solved using a combination of Krylov-subspace iterative methods and fast algorithms for approximately calculating the required dense matrix–vector (MV) products. For the co-optimization ap-

proach to offer competitive performance, we must be able to solve the co-optimization systems using relatively few MV products. Otherwise, it may be faster to calculate the full Hessian explicitly using effective BEM preconditioners [83].

We begin by defining an approximate Hessian $\hat{L} = \hat{L}_b - \hat{L}_u$, where we have defined two approximate desolvation penalty matrices. These matrices take the form

$$\hat{L}_{b/u} = M_{3,b/u} P_{2,b/u}^{-1} M_{1,b/u}, \tag{5.57}$$

where M_1 and M_3 correspond to the operators of the same name that were discussed in Section 5.2.1, and P_2 denotes the preconditioner for the corresponding BEM system; *i.e.*, when solving the bound-state boundary-element electrostatics problem one solves

$$P_{2,b} M_{2,b} x = P_{2,b} M_{1,b} q_L, \tag{5.58}$$

where q_L is the vector of ligand charges.

Using the approximate Hessian, we can write preconditioners for the co-optimization problems. The unconstrained problem (5.30) may be solved efficiently using the matrix

$$P_{unc} = \begin{bmatrix} \hat{L} & & \\ & P_b & \\ & & P_u \end{bmatrix} \tag{5.59}$$

as a preconditioner. For the equality constrained co-optimization system (5.32), we use

$$P_{eq} = \left[\begin{array}{cc|c} \hat{L} & A_c^T & \\ A_c & & \\ \hline & & \begin{matrix} P_b & \\ & P_u \end{matrix} \end{array}\right] \tag{5.60}$$

as a preconditioner. For box-constrained optimization problems solved using the Hessian-implicit primal-dual method, each modified Newton–Raphson step found using Equation 5.50 requires its own preconditioner because the system matrices depend on the current iterate.

Using the same notation as in Section 5.2.4, the system's preconditioner takes the form

$$P_{box} = \left[\begin{array}{ccc|cc} \hat{L} & -A^T & -I & & \\ A & & & & \\ S^k & & Y^k & & \\ \hline & & & P_b & \\ & & & & P_u \end{array}\right]. \tag{5.61}$$

5.3.2 Accelerating Primal-Dual Method Convergence

The centering parameter $\hat{\sigma}$ in Equation 5.48 dictates how strongly the algorithm attempts to keep the pairwise products $y_i^k s_i^k$ equal. If $\hat{\sigma}$ is set close to unity, iterates stay close to the central path and the algorithm is robust, but the algorithm makes slow progress towards an optimal solution. If instead $\hat{\sigma}$ is set very small, progress can be rapid but the optimization may stagnate. If an iterate approaches the boundary of the feasible region $(y,s) > 0$, the algorithm can make unacceptably slow progress. Wright suggested setting $\hat{\sigma} = 0.4$ for every iteration [180]. This balances robustness against convergence. For biomolecule optimization problems, we have studied a set of simple model problems of varying size and designed a new rule that picks $\hat{\sigma}^k$, the centering parameter at the k^{th} iteration, based on a rule dependent largely on the previous step multiplier α^{k-1}.

Algorithm 1 *Choosing centering parameter* $\hat{\sigma}^k$*:*

$\hat{\sigma}^k = 0.4$
if $\alpha^{k-1} > 0.7$
  $\hat{\sigma}^k = 0.1$
if $\alpha^{k-1} > 0.95$ *and* $k > 8$
  $\hat{\sigma}^k = 0.01$

This schedule was determined by practical experience with different model problems. The heuristic assumes that significant progress on the previous iteration has left the current iterate in a position to make good progress again. This assumption is generally safe after a few iterations, and the two cases in which $\hat{\sigma}^k < 0.4$ address its shortcomings.

5.4 COMPUTATIONAL RESULTS

5.4.1 Co-Optimization Method Scales Advantageously

We have examined the performance of the implicit-Hessian approach relative to explicit-Hessian optimization methods. Test optimization problems for these studies were generated using a fixed geometry of concentric spheres of radius 2 Å and 4 Å. The n_c ligand charges were randomly placed in the ligand and receptor spheres as appropriate, and n_e random equality constraints were imposed. Random box constraint vectors m and M were generated. The Yoon and Lenhoff Green's theorem formulation was used to calculate reaction potentials at the ligand charge locations assuming $\varepsilon = 4$ in the solute and $\varepsilon = 80$ in the solvent, with $\kappa = 0.124$ Å $^{-1}$.

The unconstrained and linear-equality constrained optimization problems can be solved completely using a single Krylov subspace solve, as discussed in Section 5.2.4. Accordingly, the computational advantage of using the Hessian-implicit method is evident even for very small optimization problems. Figure 5-3 is a plot of the cost, measured by the number of applied matrix–vector products, needed to solve equality-constrained problems using the implicit-Hessian method or by explicitly calculating the Hessian. These simulations were performed using a large-scale implementation based on the pFFT++ (precorrected-FFT++) fast BEM library [114] and the PETSc scientific library [184]; the linear system of Equation 5.32 was solved to 10^{-4} relative tolerance using GMRES.

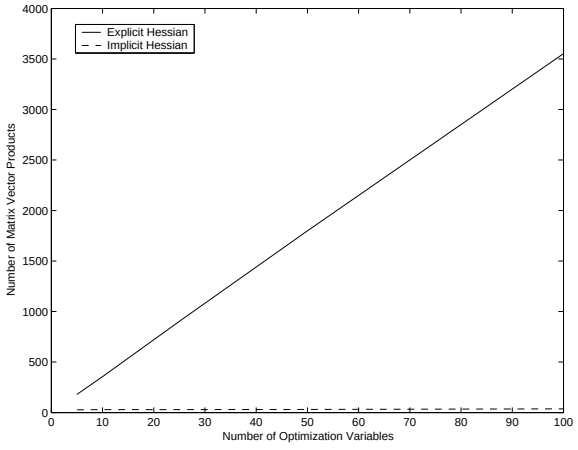


Figure 5-3: Performance of new algorithm on equality constrained problems.

We have also studied the computational scaling of the Hessian-implicit primal-dual method [165]. The Hessian-implicit solver was implemented in MATLAB [75], and the unbound and bound surfaces were discretized using 124 and 166 panels. The biased Newton–Raphson steps were calculated using GMRES [82] solved to a tolerance of 10^{-8}. The optimization was said to be converged when the slackness violation $y^T s$ was less than $\frac{10^{-6}}{n_c}$. Figure 5-4 is a plot of the number of matrix–vector products required to solve sample problems using the implicit- and explicit-Hessian methods.

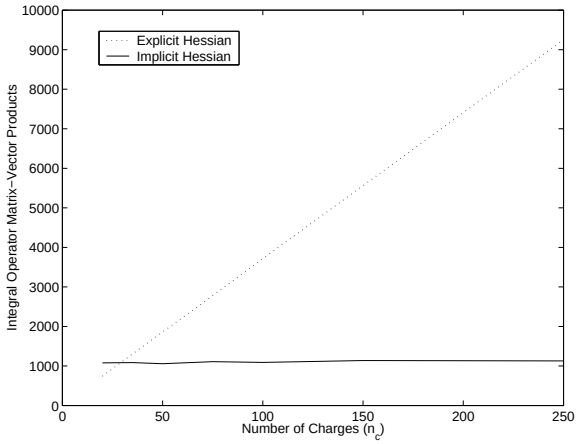


Figure 5-4: Computational scaling of the Hessian-implicit primal-dual method.

5.4.2 Comparison to Alternative Methods

We compared the Hessian-implicit primal-dual method to the simple implicit-Hessian alternative scheme mentioned in Section 2.3, and solved several inequality-constrained problems using both HIPD and the primal-dual interior-point optimization code KNITRO [185]. KNITRO implements a barrier method and solves each subproblem using sequential quadratic programming, each iteration of which is solved using conjugate gradients (CG). Each CG iteration requires one multiplication by the Hessian L, and therefore an iterative solve of the bound and unbound BEM problems. We solved each optimization problem in KNITRO problem by explicitly computing the Hessian and letting KNITRO use the Hessian. The cost was then estimated by multiplying the total number of KNITRO CG iterations by the average number of BEM matrix–vector multiplications required to find each column of

L. Figure 5-5 is a plot of the computational cost of each method for several problems of varying dimension. It is clear that HIPD offers superior performance; however, it should be noted that the KNITRO linear solves have not been preconditioned using the approximate Hessian $\hat{L}$.

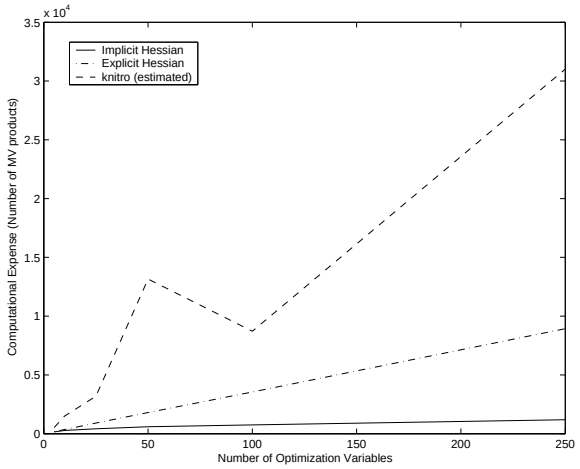


Figure 5-5: Performance of proposed Hessian-implicit method and an alternative approach for problems with linear equality and inequality constraints.

The original implementation of the HIPD method [165], which relied on the relatively conservative choice of centering parameter $\hat{\sigma} = 0.4$ as discussed in Section 5.3.2. We have compared the performance of the more aggressive schedule to the conservative algorithm; Figure 5-6 illustrates that the presented algorithm is approximately twice as fast.

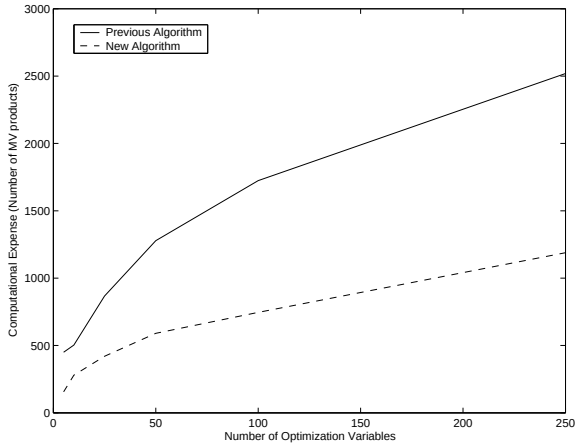


Figure 5-6: Performance of original and current implementations.

5.4.3 Realistic Biomolecule Optimization Problem: ECM/TSA

We first demonstrated in reference [166] that the co-optimization approach was viable for problems of biological significance. For this demonstration we studied *E. coli* chorismate mutase (ECM) and a transition state analog (TSA) inhibitor [169]. The TSA molecule has 26 charges to be optimized, and the Hessian-implicit system solved at every iteration has over 130,000 unknowns. In Figure 5-7 are plotted the optimal TSA charges computed using an explicit-Hessian optimizer and the PETSc-based, precorrected-FFT-accelerated co-optimization solver. The total charge has been constrained to sum to $-2e$, and each charge has been constrained to have magnitude less than $0.85e$. The primal-dual method was terminated when $y^T s < 10^{-4}/\sqrt{2n_c}$.

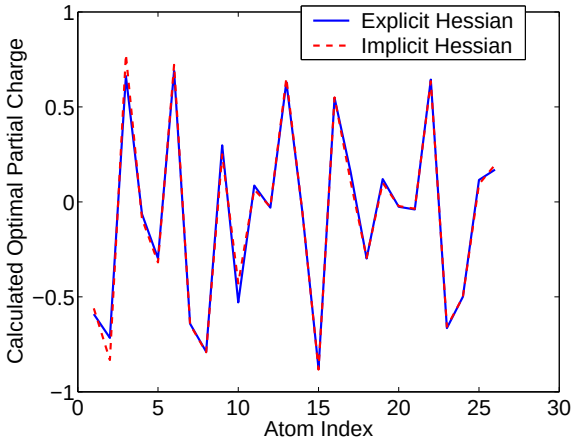


Figure 5-7: Optimal charges computed using the implicit- and explicit-Hessian optimization methods.

5.5 DISCUSSION

In this chapter, we have presented an alternative approach for solving problems in biomolecule electrostatic optimization. Our implicit-Hessian optimization technique combines Krylov-subspace iterative methods, fast boundary-element method solvers, and the optimization problems directly. By breaking the abstraction between simulation methods and the optimization, the method achieves exceptional performance, effectively reducing the computational expense for some problems from linear-time to constant-time. The implicit-Hessian

approach can be applied to constrained as well as unconstrained problems, and we have successfully applied the method to a realistic example in biomolecule design. Although a convergence analysis for these methods has so far not been possible, an analysis restricted to unconstrained problems suggests that these methods should be robust.

We note that there exist applications in which repeated optimizations will be executed for the same geometry but with varying constraints [174]. Such investigations are better suited to be studied the explicit-Hessian approach, because the cost to precompute the explicit Hessian is effectively amortized over all optimizations. That is, compared to the Hessian calculation cost, optimization is effectively free. In contrast, the implicit-Hessian cost remains a non-negligible constant for each solve. We are currently exploring possible ways to reuse computation between optimizations.

Future applications will focus on problems in which the implicit method may be used reliably. Buried, near-buried, and small ligands tend to have well-conditioned eigenspaces. These problems, which do not require explicit modification of the Hessian [169], are well-suited to the new method. Because these problems will likely not fully exploit the method's advantageous scaling, the new method is best used to investigate problems in which only a small number of optimizations are to be performed for each geometry, where each ligand is of small to moderate size, and multiple binding geometries are to be studied for each ligand.

For instance, one interesting application might be to extend the work of Sulea and Purisima [173]. This work generated a large number of possible ligand geometries, each of which was designed with high shape complementarity to a particular region of the protein surface. They optimized a single central charge for each ligand and used this charge value, and its effect on binding free energy, as tools to characterize surface reactivity and identify likely binding sites. The implicit-Hessian approach would allow many chemically reasonable geometries such as carboxylates or amino-acid side chains, to be substituted in these ligand geometries, rather than just single charges, at comparable cost to the original analysis presented in reference [173].

Another profitable use may be to explore optimization of flexible ligands. Although Kangas and Tidor proved convexity of the electrostatic optimization problem for rigid

binding, clearly many binding reactions involve ligand conformational change, and an optimization theory for these cases could significantly impact the computational ligand design community. Gilson [177] has shown that in general, conformational changes on binding give rise to non-convex objective functions, but there may exist restricted classes of problems in which flexible-ligand optimization may be performed. The implicit-Hessian method may permit an extensive exploration of non-rigid optimization problems in order to identify these classes.

Chapter 6

Conclusion

Computational modeling of interactions between biomolecules has become an essential tool for biological science and engineering. Despite their practical value, these computer simulations can implement only crude approximate models for the interactions — even the simplest of diatoms cannot be solved exactly with quantum mechanics! That these computational models are approximate should not be cause to brush them aside. Instead, the numerical methods and simulation protocols be developed carefully and in accordance with the models' inherent uncertainties.

The philosophy underlying much of this thesis work is that design problems and investigations of mechanism, when studied using an uncertain model, deserve the model's stringent solution. Excessively approximate numerical methods can render useless even well-conceived scientific studies, because computationally-based hypotheses — for instance, that a particular functional group is likely to be enriched in a set of tight-binding ligands — should be based on the mathematical model itself, or in other words on an explicitly stated set of assumptions. Careless or inappropriately applied numerical techniques can generate hypotheses that reflect computational artifacts rather than the model. As a result, the best-case scenario for a model simulated poorly is that it remains untested and therefore untrusted. In a worst-case scenario, experimental evidence supports a hypothesis that owes more to numerical error than to the model, and careless analysis leads one to conclude validity of the model.

In this thesis, we have presented a highly accurate boundary-element method solver for

biomolecule electrostatics problems. The implementation is based on three core techniques developed during the thesis research: a general Green's-theorem-based integral formulation for treating multiple embedded homogeneous regions, the fast BEM algorithm FFTSVD, and a set of techniques to discretize solute–solvent interfaces using curved boundary elements and to integrate singular functions over the elements. The challenge involved in developing an accurate solver that could remain competitive with existing finite-difference solvers was a rather surprising result, particularly for researchers from other domains; in electrical engineering, for instance, surface formulations offer a much clearer practical advantage as well as the numerous theoretical advantages discussed in Chapter 4. The remarkable performance of modern FDM for molecular electrostatics may be attributed in large part to the extensive and thorough numerical experiments performed over many years by several groups, most notably those led by Professors Barry Honig and J. Andrew McCammon.

The philosophy discussed in the preceding paragraph suggests, however, that new numerical approaches may be warranted to optimally exploit the continuing explosion in computational processing power. The thesis has presented a set of numerical techniques to improve the accuracy and efficiency with which one may calculate important components of a molecule's solvation free energy. The described techniques solve continuum-theory-based surface formulations of these molecular modeling problems. The thesis contributions may be grouped into four areas.

First, we have advanced two of the popular boundary-integral-equation formulations of continuum electrostatics models for biomolecule analysis. One formulation we have studied is the non-derivative Green's theorem formulation [21]. We have extended this mixed first-second kind equation to treat multiple boundaries separating regions of differing homogeneous dielectric constant with possible salt treatment [83]. In particular, we can model the ion-exclusion layer surrounding the molecular solute, as well as ion-exclusion layers that may exist inside water-filled cavities within the solute. In addition, we have examined discretization of the induced-surface charge or equivalent-charge formulation, which is a purely second-kind integral equation that can be used to study electrostatic interactions in non-ionic solution. We have demonstrated that the solution accuracy depends strongly on

the process of discretizing the integral equations [36]. Many other integral formulations of biomolecule electrostatics exist, however, and it seems likely that there remain theoretical and numerical improvements that may be made to these as well. For instance, the purely second-kind formulation described by Juffer *et al.* contains an integral operator that resembles that of the equivalent-charge formulation [23]. The qualocation method of Tausch, Wang, and White [37] may therefore find advantageous application there just as it did for the equivalent-charge formulation.

Second, we have designed, implemented, and optimized a fast, kernel-independent algorithm, called FFTSVD, to numerically solve boundary-integral equations on complex molecular geometries. The FFTSVD algorithm rapidly computes the matrix-vector products required to solve the BEM equations using preconditioned Krylov-subspace iterative methods [50]. The algorithm combines an octree decomposition of the problem domain with a sampling-based reduced-basis representation of the long-range interactions. The interactions between reduced-bases are calculated efficiently via the FFT. The structure of our algorithm is well-suited for solving problems in which the boundary-elements occupy a small fraction of a bounding cube surrounding the problem domain. In addition, the multi-level approach to multiplication suggests a natural and efficient parallelization. Developing a production-quality BEM solver based on this algorithm is an important goal for future work, because it will enable the solution of larger biomolecule problems as well as the modeling of problems in other domains such as micro- and nano-fluidics.

Curved-panel methods for surface formulations in biomolecular modeling comprise the third major contribution of this thesis [38]. First, we have defined two classes of curved panels that are general enough to allow the essentially exact discretization of van der Waals, solvent-accessible, and solvent-excluded surfaces. Second, we have described one method for obtaining such discretizations given a set of sphere centers, their radii, and the radius of the probe sphere to be rolled around the sphere union. Third, we have demonstrated a number of numerical integration techniques for evaluating far-field, near-singular, and singular integrals over these curved panels. The boundary-element electrostatics research discussed in this thesis has focused entirely on using piecewise-constant basis functions and centroid collocation. Higher-order basis functions may significantly reduce the amount of

computer memory resources required to reach a given level of accuracy, and for this reason we expect future work in this area to study ways by which such basis functions may be integrated efficiently.

The thesis has also developed a coupled simulation/optimization approach to efficiently solve for the charge distribution in a biomolecule that optimizes the free energy of binding to another molecule, given that the potential response is linear, that the molecules bind rigidly, and that no charge transfer occurs on binding [165]. This co-optimization technique, which in spirit resembles PDE-constrained optimization methods, relies on an implicit representation of the Hessian and solves the optimization problem simultaneously with two electrostatic simulations, using preconditioned Krylov subspace iterative methods. This *implicit-Hessian* method can be applied to unconstrained problems as well as those with linear equality and inequality constraints. We have applied the co-optimization method to realistic biomolecule optimization problems. The method was applied to a small validating test case, that of *E. coli* chorismate mutase and a transition-state analog inhibitor [166]. The results demonstrated that the accelerated solution method solves the optimization problem and that the computed optimal charges closely match the optimal charges calculated by the traditional charge-optimization approach in which the full explicit Hessian is calculated one column at a time. In a second, ongoing investigation, we are studying multiple ligands of the serine protease thrombin [186] and their relative electrostatic optimalities. The co-optimization technique may find profitable future applications in rational drug design processes as well as in studies of protein–ligand and protein–protein interactions.

It seems likely that some of the most exciting numerical work in the future will lie in the areas of accelerating the simulation of closely-related physical problems, and in coupling physical simulation with optimization. The simulation of multiple geometries is recognized to be a useful approach for several types of important calculations such as pKa shifts [187] and binding free energies [188]. The development of accelerated methods for biophysical simulation could possibly feed back to many other domains of engineering, including aerospace, electrical, and mechanical. Such a cross-over would not only have significant impact for design processes, but more importantly it would represent a step

forward for bridging the computational life sciences with more traditional computational research communities.

The coupling of simulation and optimization is a new and rapidly growing field of current interest. Molecular design represents one of the most challenging problems for which such approaches may be conceived. As already discussed, the extant mathematical models are relatively simplistic and highly approximate; furthermore, the high-dimensional search spaces are discrete. The development of efficient methods to prune the search space will certainly continue to be an important and active area of research. Many traditional mathematical programming approaches begin from a complete specification of the objective function and constraints. However, for some types of molecular design problems such as the electrostatic optimization problem, even obtaining such complete information can be prohibitive or infeasible. A exciting, inherently multi-disciplinary paradigm is emerging to address these and similar challenges in many domains: tremendous acceleration can be achieved by breaking the abstraction between the optimization method and the means used to obtain information about the objective. Branch-and-bound methods, for instance, offer in a sense a means to coarse-grain the search process by using approximate methods to bound the objective function at each branch. As a second example, the implicit-Hessian method for electrostatic optimization breaks the black-box abstraction of the linearized Poisson–Boltzmann solver. Rather than introducing approximations to the LPBE calculations, the discretized model is itself coupled directly to the optimization process and a self-consistent solution is obtained directly. This intimate coupling almost entirely eliminates the need to precompute information about the objective function.

Appendix A

Extracting Curved Panel Discretizations[1]

Accessible and van der Waals Surfaces

Accessible and van der Waals surfaces can be described by a set of spherical patches, where each patch represents a solvent-exposed portion of an atom. When an atom (or a probe-radius-expanded atom) intersects another, the two sphere surfaces form a circle of intersection, and all the atom's surface beyond the plane of this circle is buried inside the other atom. Consequently, each spherical patch can be described by an intersection of the sphere and a set of half-spaces, which are derived by analytically solving for the planes of intersection between the given sphere and all the intersecting spheres. To mesh a spherical patch, we first obtain a high-quality flat triangular discretization using the program NETGEN [149]. NETGEN meshes surfaces based on a constructive solid geometry (CSG) scheme in which geometries are defined using boolean operations on primitives such as spheres and half-spaces.

Once the discretization is obtained, each planar triangle is converted to a GST by assigning an arc center to each edge. If an edge lies on one of the half-space planes, its arc center is assigned to be the center of the circle of intersection that defines the half-space. Occasionally, coarse triangular discretizations contain triangles whose edges lie on more

[1] To be submitted as an appendix with Chapter 2 [38].

than one plane. These situations do not reflect the molecular geometry but instead are a consequence of the NETGEN discretization procedure; such geometries are therefore discretized more finely. If a planar-triangle edge does not lie in a half-space plane, the arc center is assigned to be the center of the sphere; as a result, the corresponding GST arc is part of a great circle. After forming the GST, it is checked to ensure that it conforms to the definition presented in Section 3.2. Specifically, it is ensured that the arcs only intersect at their end points and that the internal jump angles are less than π radians. If any GST fails these checks, the entire spherical patch is rediscretized at a finer level.

Molecular Surfaces

Molecular surfaces are discretized in two stages. In the first stage, we increase the atomic radii by the probe radius and use NETGEN to generate a solvent-accessible surface by meshing the union of the expanded spheres. During the discretization process, NETGEN determines every point on the accessible surface where three or more expanded atoms simultaneously intersect, as well as every circular arc generated by the intersection of two expanded sphere surfaces. The intersection of three or more arcs becomes a fixed probe position for the molecular surface. The probe position generates one or more concave-spherical patches of reentrant surface because this point is simultaneously a probe-radius distance away from three or more atoms. Each circular arc connects two fixed probe positions along the intersection of two expanded atoms. Because the arc is composed of points equidistant from exactly two atoms, this arc indicates the presence of a toroidal surface patch. The accurate determination of these features is valuable during the second stage of discretization, in which the specified spherical and toroidal patches are meshed directly.

Spherical Contact Patches

Spherical contact patches on molecular surfaces are generated for every solvent-exposed atom. The patches are meshed similarly to the spherical patches on van der Waals and accessible surfaces; however, contact patches on molecular surfaces are bounded by the half-space planes located at sphere–torus intersections rather than at sphere–sphere inter-

sections. The positions of these shifted planes are computed analytically by determining the point of tangency between the given sphere and the probe sphere when it simultaneously touches each neighboring atom.

Spherical Reentrant Patches

Spherical reentrant patches are meshed by placing a sphere of radius equal to the probe radius at each triple or higher intersection point determined during the discretization of the solvent-accessible surface. Recall that these intersection points are formed where multiple circular arcs meet, and that these arcs represent toroidal patches. The spherical reentrant patch is therefore intersected with three or more half-space planes, each of which represents a boundary between the probe sphere and the toroidal patch extracted from the corresponding circular arc.

Each plane is analytically defined by three points: the center of the probe sphere and the centers of the two atoms associated with the torus. When necessary, additional half-space planes are generated from probe-probe intersections in a manner similar to accessible surface meshing. Once the probe sphere and half-spaces have been identified, discretization proceeds identically to accessible spherical patch meshing.

Toroidal Patches

Each circular arc of the accessible surface is associated with one toroidal patch on the molecular surface. The arc traces out the path taken by the center of the sphere as it rolls tangent to its two associated atoms. Therefore, the toroidal patch is a portion of a torus centered at the analytical center of the circle of intersection between the two expanded atoms of the accessible surface. The torus's principal x and y axes lie in the circle plane and the z axis is parallel to the vector pointing between the atom centers. The torus's inner radius a is the probe radius, and the outer radius c is the radius of the intersection circle.

If two probe positions terminate the accessible-surface arc, the toroidal patch will be bounded in θ. The range in θ is determined by fixing one torus principal axis to point from the torus center to the first probe position and then by taking the dot product of this axis with the vector pointing from the torus center to the second probe position. If the accessible-

surface arc is not terminated by probe positions, the torus is complete, and spans $[0, 2\pi]$ in the θ direction.

The bounds on ψ are found by the following procedure: specify an arbitrary probe position on the accessible-surface circle of intersection. Then compute the vector pointing from the probe center to the center of the torus. Take the dot product of this vector with one pointing from the probe position to the center of each of the torus's associated atoms. Each dot product is the cosine of one of the bounding angles ψ.

If the torus has an outer radius less than its inner radius (*i.e.*, $c < a$), and if in addition the range in ψ overlaps the range $[\pi - \arccos(\frac{c}{a}), \pi + \arccos(\frac{c}{a})]$, then the toroidal patch consists of two disconnected pieces of surface. The two regions of such a self-intersecting torus are meshed separately.

Once the bounds on the toroidal patch are determined, the region is discretized into toroidal panels by dividing the ranges of θ and ψ into an integral number of pieces such that the arc lengths of the panel edges are similar to those generated for GST panels.

Appendix B

Coordinate Transformation from the Standard Triangle to the Generalized Spherical Triangle[1]

In this appendix we describe how the parametric coordinates (ξ,η) map to a point (x,y,z) on a GST, and how we compute $|J|$, the determinant of the transformation Jacobian. Figure 2-5 illustrates the spherical coordinate system; the coordinate $\psi \in [0,\pi]$ describes the angle from the positive x axis, and the coordinate $\theta \in [0,2\pi]$ describes the angle from the positive z axis. The angles ψ_{start} and ψ_{end} are defined as shown in the Figure. For any point (ξ,η) we define a circle $C(\eta)$ as shown; this circle is the set of points on the sphere at

$$\psi(\eta) = \psi_{start} + \eta(\psi_{end} - \psi_{start}). \tag{B.1}$$

Obviously $\frac{\partial\psi}{\partial\eta}(\eta) = \psi_{end} - \psi_{start}$. The intersection of $C(\eta)$ with the two arcs a_2 and a_3 produce two points r_2 and r_3, which are defined to be at $(\theta_{start}(\eta),\psi(\eta))$ and $(\theta_{end}(\eta),\psi(\eta))$. The θ coordinate of the mapped point is set to

$$\theta(\xi,\eta) = \theta_{start}(\eta) + \frac{\xi}{1-\eta}\left(\theta_{end}(\eta) - \theta_{start}(\eta)\right). \tag{B.2}$$

[1]To be submitted as an appendix with Chapter 2 [38].

We have also the first derivatives

$$\frac{\partial\theta}{\partial\xi}(\xi,\eta) = \frac{1}{1-\eta}(\theta_{end}(\eta)-\theta_{start}(\eta)); \tag{B.3}$$

$$\frac{\partial\theta}{\partial\eta}(\xi,\eta) = \frac{\partial\theta_{start}}{\partial\eta}(\eta)+\frac{\xi}{1-\eta}\left(\frac{\partial\theta_{end}}{\partial\eta}(\eta)-\frac{\partial\theta_{start}}{\partial\eta}(\eta)\right) + \frac{\xi}{(1-\eta)^2}\left(\theta_{end}(\eta)-\theta_{start}(\eta)\right). \tag{B.4}$$

Denoting the mapped point by $\vec{r}$, the Jacobian determinant is

$$|J| = \left|\frac{d\vec{r}}{d\xi}\times\frac{d\vec{r}}{d\eta}\right|, \tag{B.5}$$

where

$$\frac{d\vec{r}}{d\xi} = \frac{\partial\vec{r}}{\partial\theta}\frac{\partial\theta}{\partial\xi}+\frac{\partial\vec{r}}{\partial\psi}\frac{\partial\psi}{\partial\xi}. \tag{B.6}$$

$$\frac{d\vec{r}}{d\eta} = \frac{\partial\vec{r}}{\partial\theta}\frac{\partial\theta}{\partial\eta}+\frac{\partial\vec{r}}{\partial\psi}\frac{\partial\psi}{\partial\eta}. \tag{B.7}$$

Trivially, we have

$$\frac{\partial\theta}{\partial\xi} = \frac{\theta_{end}(\eta)-\theta_{start}(\eta)}{1-\eta} \tag{B.8}$$

$$\frac{\partial\psi}{\partial\eta} = \psi_{end}-\psi_{start} \tag{B.9}$$

$$\frac{\partial\psi}{\partial\xi} = 0. \tag{B.10}$$

The derivative $\frac{\partial\theta}{\partial\eta}$ is more challenging to calculate. The rotation angle θ_{start}, defined by the relation

$$\theta_{start}(\eta) = \tan^{-1}(\frac{y(\eta)}{z(\eta)}), \tag{B.11}$$

has the first derivative

$$\frac{d\theta_{start}}{d\eta} = \frac{1}{1+(\frac{y(\eta)}{z(\eta)})^2}\frac{z\frac{dy}{d\eta}-y\frac{dz}{d\eta}}{z(\eta)^2}, \tag{B.12}$$

where we have omitted adding the subscript *start* to the variables y and z, and the angle θ_{end} is defined analogously.

The derivatives $\frac{dy}{d\eta}(\eta)$ and $\frac{dz}{d\eta}(\eta)$ are defined by finding the angle α such that r_3 satisfies

$$\vec{r}_3 = \vec{r}_{center} + \vec{x}\cos(\alpha) + \vec{y}\sin(\alpha), \tag{B.13}$$

where r_{center} is the center of the circle defining the GST arc and x and y form an orthonormal basis for the plane in which the arc lies. We then find the needed derivatives by

$$\frac{d\vec{r}_3}{d\alpha} = -\vec{x}\sin(\alpha) + \vec{y}\cos(\alpha) \tag{B.14}$$

$$\frac{d\alpha}{d\eta} = \left(\frac{dx}{d\alpha}\left(\frac{dx}{d\eta}\right)^{-1}\right)^{-1} \tag{B.15}$$

$$\frac{d\vec{r}_3}{d\eta} = \frac{d\vec{r}_3}{d\alpha}\frac{d\alpha}{d\eta} \tag{B.16}$$

and taking the y and z components of $\frac{d\vec{r}_3}{d\eta}$.

Appendix C

Curved Panel Integration Techniques for Other Integrands[1]

Linearized Poisson–Boltzmann Kernel

The single-layer linearized Poisson–Boltzmann integrals

$$\Phi(r) = \int_\Omega \frac{e^{-\kappa||r-r'|}}{4\pi||r-r'||} dA' \tag{C.1}$$

can be evaluated by decomposing the integral into a sum of two more easily computed integrals [32]:

$$\Phi(r) = \int_\Omega \frac{1}{4\pi||r-r'||} dA' - \int_\Omega \frac{1-e^{-\kappa||r-r'||}}{4\pi||r-r'||}. \tag{C.2}$$

The first term is merely the single-layer Laplace integral, whose calculation we have already discussed. The second term is very smooth in the near-field, and can therefore be integrated using the quadrature schemes described in Section 5.2. In the far-field, the entire integral in Equation C.1 can be computed easily using direct quadrature.

Double-layer linearized Poisson–Boltzmann integrals can be computed in an exactly analoguous fashion.

[1] To be submitted as an appendix with Chapter 2 [38].

Surface-Generalized-Born Kernels

The surface-Generalized-Born integrals all take the form of Equation 2.6 but with different exponents depending on whether one begins from the volume formulations of Still *et al.* , Grycuk, or Wojciechowski and Lesyng [41, 55, 56]. The required curved-element integrals are all nonsingular because the evaluation points are always sphere centers. The integrands' rapid decay allows far-field quadrature to be used to compute all needed interactions.

Continuum van der Waals Kernels

The surface continuum van der Waals method requires evaluation of surface integrals of the form shown in Equation 2.9, where again the evaluation points are always sphere centers. The cvdW integrals over the solvent-accessible surface are therefore never singular, and again far-field quadrature techniques may be used.

Appendix D

Accurate Discretization of the Apparent-Surface Charge Formulation for Biomolecule Electrostatics in Non-ionic Solutions[1]

ABSTRACT

The electrostatic interactions between biomolecules and solvent are generally difficult to model because there exist an enormous number of solvent degrees of freedom. Continuum electrostatic models provide an approximate method to analyze these interactions; these models are typically solved numerically in either differential or integral form. In this paper we demonstrate the importance of using an appropriate numerical technique, called qualocation, for a popular integral formulation of the electrostatics problem. Numerical results illustrate that qualocation exhibits superior accuracy relative to naive implementations. We also show that the integral formulation is extremely well-conditioned and converges rapidly

[1]This appendix appeared in the proceedings of the 2005 IEEE Conference on Engineering in Medicine and Biology [36].

when iterative methods are used to solve the discretized integral equation.

D.1 INTRODUCTION

Electrostatic interactions within and between biomolecules are known to play important structural and functional roles [9, 189]. Analyzing these interactions computationally is challenging because solvent molecules surround the biomolecules of interest, so the physical problem has an enormous number of degrees of freedom. Monte Carlo and molecular dynamics methods [7, 9, 190–193] treat all or most of the solvent molecules explicitly, but for many problems the computational expense is prohibitive.

Continuum models offer an alternative approach to studying biomolecule electrostatics [9, 15, 26, 40, 94]. In these models, macroscopic laws of electrostatics are assumed to hold in the molecule interior and in the solvent, and the resulting systems of partial differential equations are solved numerically on a computer. Finite difference methods, finite element methods, and boundary element methods (BEM) have all been applied to the biomolecule electrostatics problem [18,21,23,31,130,194]. The boundary element method offers numerical advantages such as an improved representation of the biomolecule–solvent interface and exact treatment of discrete point charges. Here we study an integral formulation and boundary element technique for solving biomolecule electrostatics problems in which the solvent ionic strength is zero.

The integral formulation, called the equivalent charge formulation (ECF), has been previously discussed in the literature [16, 195]. In this work we demonstrate that a numerical technique called qualocation [37] substantially improves accuracy when compared to naive implementations of the integral formulation. The qualocation method can be applied to many types of BEM problems in addition to the biomolecule problem discussed here.

The following section introduces the electrostatics model and the boundary element method used to solve the model numerically. Section D.3 presents the ECF–qualocation method and Section D.4 illustrates the method's performance with computational results. Section D.5 summarizes the paper.

D.2 BACKGROUND

D.2.1 Mixed Discrete-Continuum Electrostatics Model

Figure D-1 illustrates the mixed discrete–continuum electrostatics model. The boundary Ω separates the molecular interior from the solvent exterior; Ω is taken to be the Richards molecular surface [49], which is formed by rolling a probe sphere around the union of van der Waals–radius spheres located at the atom centers. We treat the molecular interior as a homogeneous medium with permittivity ε_I, in which the electrostatic potential obeys the Poisson equation

$$\nabla^2\varphi_I(r) = -\sum_{i=1}^{n_c} \frac{q_i}{\varepsilon_I}\delta(r-r_i), \tag{D.1}$$

where n_c is the number of discrete point charges and r_i and q_i denote the location and value of the i^{th} charge. The solvent region is treated as a homogeneous medium with a much higher permittivity ε_{II}, and in this region the Laplace equation holds:

$$\nabla^2\varphi_{II}(r) = 0. \tag{D.2}$$

At the dielectric boundary, the potential and normal component of the displacement field are continuous:

$$\varphi_I(r_\Omega) = \varphi_{II}(r_\Omega) \tag{D.3}$$

$$\varepsilon_I\frac{\partial\varphi_I}{\partial n}(r_\Omega) = \varepsilon_{II}\frac{\partial\varphi_{II}}{\partial n}(r_\Omega). \tag{D.4}$$

D.2.2 The Boundary-Element Method

Consider the problem of computing the capacitance of a conducting sphere, whose surface is S, suspended in free space. By setting the potential on the sphere to unity and solving

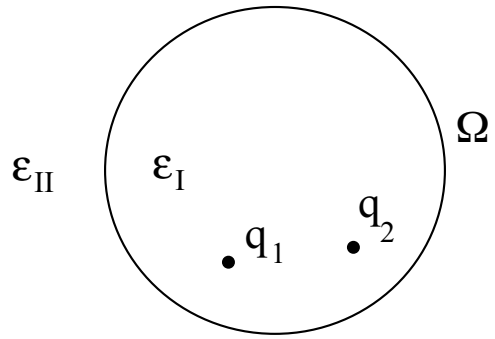


Figure D-1: Mixed discrete-continuum electrostatics model.

the first kind integral equation

$$\int_S \frac{\sigma(r')dA'}{4\pi\varepsilon_0||r-r'||} = \Psi(r), \tag{D.5}$$

we can integrate $\sigma(r)$ over S to find the capacitance. To solve the problem numerically, we discretize the boundary surface into a set of n_p panels and represent the solution $\sigma(r)$ on the discretized surface as a weighted combination of compactly supported basis functions:

$$\sigma(r) = \sum_{i=1}^{n_p} y_i \chi_i(r). \tag{D.6}$$

Here, $\chi_i(r)$ denotes the i^{th} basis function and y_i the associated weight. In this paper, we use piecewise-constant basis functions such that each function takes value unity on a single panel and is zero everywhere else:

$$\chi_i(r) = \begin{cases} 1 & \text{if } r \text{ is on panel } i \\ 0 & \text{otherwise.} \end{cases} \tag{D.7}$$

In general, the span of the basis functions will not permit exact solution of the original integral equation. Instead, consider computing the basis function weights so as to reduce the residual $R(r)$, which is the difference between the known potential $\Psi(r)$ and the result

of applying the integral operator to the approximate solution:

$$R(r) = \Psi(r) - \int_S G(r;r') \left(\sum_{i=1}^{n_p} y_i \chi_i(r') \right) dA'. \tag{D.8}$$

In the commonly used centroid collocation scheme, $R(r)$ is forced to be zero at the basis function centroids [196]. The resulting linear system is of the form $Ay = b$ with

$$A_{ij} = \int_S G(r_{c_i};r')\chi_j(r')dA' \tag{D.9}$$

$$b_i = \Psi(r_{c_i}), \tag{D.10}$$

where r_{c_i} is the centroid of panel i. Alternatively, Galerkin methods force the residual to be orthogonal to the basis functions $\{\chi_1, \chi_2, \ldots, \chi_{n_p}\}$. Galerkin methods produce linear systems of equations of the same $Ay = b$ form, though now the entries are

$$A_{ij} = \int_S \int_S \chi_i(r')G(r';r'')\chi_j(r'')dA'dA'' \tag{D.11}$$

$$b_i = \int_S \chi_i(r')\Psi(r')dA'. \tag{D.12}$$

For both the collocation and Galerkin methods, the linear systems can be solved using sparsification-accelerated iterative methods [82, 96, 197].

D.3 THE ECF–QUALOCATION METHOD

D.3.1 Integral Formulation

The essential idea of the equivalent charge formulation is to replace the original problem, which has two dielectric regions, with a simpler problem, shown in Figure D-2, which is a Poisson problem with the same dielectric constant everywhere in space. In Figure D-2, we have replaced the solvent dielectric ε_{II} with ε_I from the interior and introduced a fictitious layer of charge $\sigma_p(r)$ on the surface. The variables $\hat{\varphi}_I$ and $\hat{\varphi}_{II}$ denote the potential in the modified problem. Finding a surface charge layer $\sigma_p(r)$ such that the original boundary conditions (Equations D.3 and D.4) hold ensures that the solution of the homogeneous

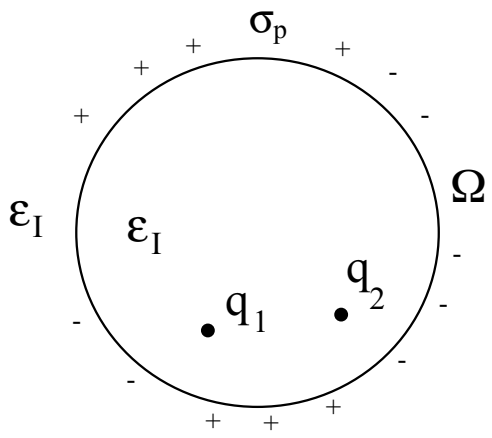


Figure D-2: Physical model of the equivalent charge formulation.

dielectric problem is equivalent to that of the original multiple dielectric region problem.

Because the dielectric constant is homogeneous throughout space in the equivalent problem, we can write the potential as

$$\hat{\varphi}(r) = \sum_{i=1}^{n_c} \frac{q_i}{4\pi\varepsilon_I ||r - r_i||} + \int_\Omega \frac{\sigma_p(r')dA'}{4\pi\varepsilon_I ||r - r'||}. \tag{D.13}$$

The normal component of the electric field at a point r on the surface is therefore

$$\frac{\partial\hat{\varphi}}{\partial n}(r) = \frac{\partial}{\partial n(r)} \sum_{i=1}^{n_c} \frac{q_i}{4\pi\varepsilon_I ||r - r_i||} + \frac{\partial}{\partial n(r)} \int_\Omega \frac{\sigma_p(r')dA'}{4\pi\varepsilon_I ||r - r'||}, \tag{D.14}$$

and the discontinuity in the integral term implies that a side of the surface must be specified. In the homogeneous dielectric problem, the charge density determines the discontinuity of the normal component of the electric field by the relation [51]:

$$\frac{\partial\hat{\varphi}_{II}}{\partial n}(r) - \frac{\partial\hat{\varphi}_I}{\partial n}(r) = \sigma_p(r)/\varepsilon_I. \tag{D.15}$$

Combining (D.15), (D.4), and (D.14) gives

$$\frac{\varepsilon_I+\varepsilon_{II}}{2\varepsilon_I(\varepsilon_I-\varepsilon_{II})}\sigma_p(r)+\oint_\Omega \frac{\partial}{\partial n(r)}\frac{\sigma_p(r')dA'}{4\pi\varepsilon_I||r-r'||} = -\frac{\partial}{\partial n(r)}\sum_{i=1}^{n_c}\frac{q_i}{4\pi\varepsilon_I||r-r_i||}, \tag{D.16}$$

which is known as the equivalent charge formulation [195,196]; the integral over Ω is taken to be the principal value integral.

D.3.2 Qualocation Method

We now motivate the qualocation approach as it was described by Tausch *et al.* [37], and present both collocation and qualocation as simplifications of the Galerkin method. To solve Equation D.16 numerically via the Galerkin method, we discretize the molecular surface into n_p flat triangles and represent the surface charge $\sigma_p(r)$ as a weighted combination of piecewise constant basis functions. We then define a residual $R(r)$ similar to Equation D.8 and enforce $\int R(r)\chi_i(r)dA = 0$ for each basis function $\chi_i(r)$. This produces a set of equations of the form:

$$\int_{\text{panel } i}\frac{(\varepsilon_I+\varepsilon_{II})y_i dA}{2\varepsilon_I(\varepsilon_I-\varepsilon_{II})}+\int_{\text{panel } i}\oint_{\text{panel } j}\frac{\partial}{\partial n(r)}\frac{y_j dA' dA}{4\pi\varepsilon_I||r-r'||} = -\int_{\text{panel } i}\frac{\partial}{\partial n(r)}\sum_k\frac{q_k dA}{4\pi\varepsilon_I||r-r_k||}, \tag{D.17}$$

where again y_i is the weight associated with the i^{th} basis function.

The centroid collocation method simplifies the Galerkin method by replacing each integral over panel i with a midpoint quadrature rule; the inner integral of the double integral is then evaluated analytically [45,46]. However, the integrand of the outer integral is non-smooth for nearby panels because the normal $n(r)$ on panel i has a component in the plane of panel j. As a result, midpoint quadrature and the resulting collocation scheme are inaccurate.

In contrast, the qualocation method replaces the inner integral, which is smooth, with a

midpoint quadrature rule. The resulting system has entries

$$A_{ii} = \frac{\varepsilon_I + \varepsilon_{II}}{2\varepsilon_I(\varepsilon_I - \varepsilon_{II})}\alpha_i \tag{D.18}$$

$$A_{ij} = \int_{\text{panel } i} \frac{\partial}{\partial n(r)} \frac{\alpha_j \, dA}{4\pi\varepsilon_I ||r - r_{c_j}||} \tag{D.19}$$

$$b_i = -\int_{\text{panel } i} \sum_k \frac{\partial}{\partial n(r)} \frac{q_k \, dA}{4\pi\varepsilon_I ||r - r_k||}, \tag{D.20}$$

where α_i is the area of panel i. Using qualocation, the outer, non-smooth integral can be evaluated analytically and the smooth inner integral is approximated accurately.

D.4 RESULTS

We have implemented the ECF–qualocation formulation using the FFTSVD fast BEM algorithm [83] to rapidly apply the dense discretized integral operator. The method relies on the observation that the qualocation operator is the scaled transpose of the double layer potential operator [37]. We compare the ECF–qualocation method to ECF–collocation as well as to a more complex formulation derived from Green's theorem [21, 117]. In contrast to the ECF formulation, which has one variable per panel and one integral operator, the Green's theorem formulation has two surface variables per panel and requires two integral operators.

D.4.1 Sphere

To test the accuracy of the ECF–qualocation method, we computed the electrostatic component of the solvation free energy for a sphere with a 1 Å radius and a central +1e charge. We compare the numerical results with the analytical answer as the surface discretization is refined. Figure D-3 is a plot of the results computed using collocation and qualocation methods as well as those from a Green's theorem formulation [21, 117]. The qualocation method is clearly superior in accuracy to the collocation method; surprisingly, qualocation returns a slightly more accurate answer than the Green's theorem method, which has twice as many degrees of freedom.

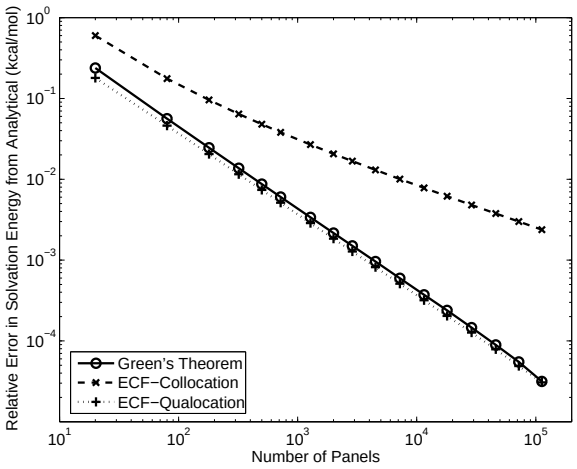


Figure D-3: Improvement in accuracy with increasing panel discretization when computing the solvation energy of a 1 Å radius sphere with a central +1e charge. Results for the Green's theorem, ECF–collocation, and ECF–qualocation formulations are shown.

D.4.2 Barnase–Barstar Protein Complex

We also computed the electrostatic component of the solvation free energy for the barnase–barstar protein complex (1BRS in the Protein Data Bank) [77]. Figure D-4 is a convergence plot that compares the ECF–qualocation result, the ECF–collocation result, and the Green's theorem result as the surface discretization is refined.

D.4.3 Iterative Method Convergence

It is well known [37] that second-kind integral operators such as the ECF formulation in Equation D.16 are well-conditioned. The discretized linear systems have tightly clustered spectra, which leads to rapid convergence when Krylov iterative methods are used instead of Gaussian elimination. The Green's theorem formulation [21, 117] is instead a mixed first-second kind equation; its poorer conditioning necessitates the development of effective preconditioners [96, 117]. To illustrate the advantageous conditioning, we have solved the barnase–barstar problem using the ECF–qualocation method using both no preconditioner and a diagonal preconditioner, and the Green's theorem formulation with no preconditioner as well as with the block diagonal preconditioner presented by Kuo *et al.* [117]. In Figure D-5 we plot the relative GMRES residuals as a function of iteration count.

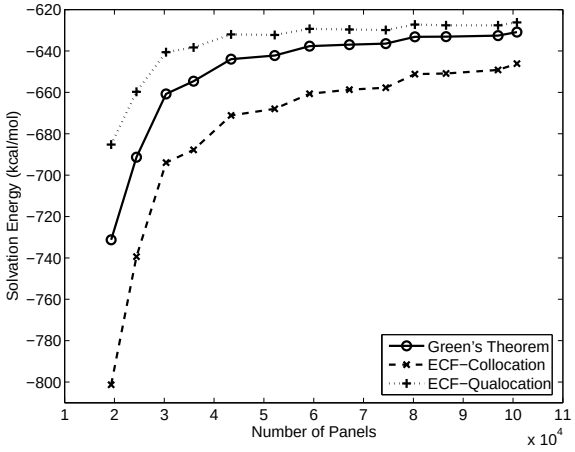


Figure D-4: Computed electrostatic components of the solvation free energy of the barnase–barstar protein complex with increasing panel discretization for the Green's theorem, ECF–collocation, and ECF–qualocation formulations.

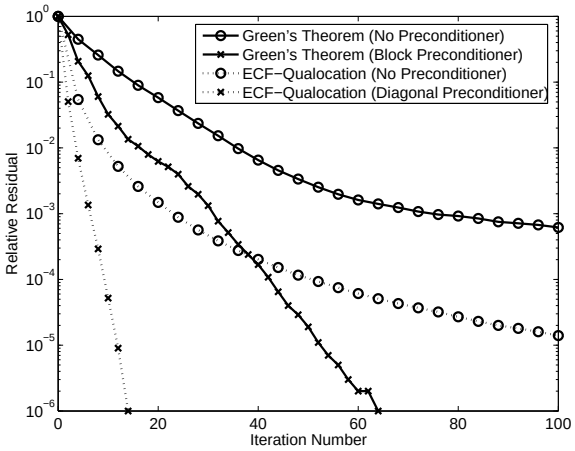


Figure D-5: Reduction in relative residual with iteration count for a 74,466 panel discretization of the barnase–barstar complex. Results are shown for the Green's theorem formulation, with and without block diagonal preconditioning, as well as for the ECF–qualocation formulation, with and without diagonal preconditioning.

D.5 SUMMARY

We have presented a numerical technique for calculating the electrostatic component of the solvation free energy of biomolecules for solutions with zero ionic strength. The technique is based on the equivalent charge formulation [16, 196] of the electrostatics problem. Our technique differs from earlier presentations because we form a linear system of equations using qualocation [37] rather than centroid collocation or Galerkin methods. We have demonstrated that the qualocation approach exhibits superior accuracy, and that Krylov iterative methods converge rapidly for ECF–qualocation problems because the second-kind integral formulation is extremely well-conditioned. It is non-trivial to extend the ECF formulation to treat problems in which the solvent ionic strength is non-zero.

Bibliography

[1] R. A. Friesner and B. D. Dunietz. Large-scale ab initio quantum chemical calculations on biological systems. *Accounts of Chemical Research*, 34:351–358, 2001.

[2] A. G. Street and S. L. Mayo. Computational protein design. *Structure with Folding and Design*, 7:R105–R109, 1999.

[3] B. Honig, K. Sharp, and A. S. Yang. Macroscopic models of aqueous solutions: Biological and chemical applications. *Journal of Physical Chemistry*, 97(6):1101–1109, 1993.

[4] A. Leach. *Molecular Modelling: Principles and Applications*. Prentice Hall, second edition, 2001.

[5] M. K. Gilson, J. A. Given, B. L. Bush, and J. A. McCammon. The statistical-thermodynamic basis for computation of binding affinities: A critical review. *Biophysical Journal*, 72:1047–1069, 1997.

[6] B. Roux and T. Simonson. Implicit solvent models. *Biophysical Chemistry*, 78:1–20, 1999.

[7] A. Jean-Charles, A. Nicholls, K. Sharp, B. Honig, A. Tempczyk, T. F. Hendrickson, and W. C. Still. Electrostatic contributions to solvation energies: Comparison of free energy perturbation and continuum calculations. *Journal of the American Chemical Society*, 113:1454–1455, 1991.

[8] R. M. Levy, L. Y. Zhang, E. Gallicchio, and A. K. Felts. On the nonpolar hydration free energy of proteins: Surface area and continuum solvent models for the solute–solvent interaction energy. *Journal of the American Chemical Society*, 125:9523–9530, 2003.

[9] K. A. Sharp and B. Honig. Electrostatic interactions in macromolecules: Theory and applications. *Annual Review of Biophysics and Biophysical Chemistry*, 19:301–332, 1990.

[10] T. Simonson. Macromolecular electrostatics: Continuum models and their growing pains. *Current Opinions in Structural Biology*, 11:243–252, 2001.

[11] A. D. MacKerell Jr., D. Bashford, M. Bellott, R. L. Dunbrack Jr., J. D. Evanseck, M. J. Field, S. Fischer, J. Gao, H. Guo, S. Ha, D. Joseph–McCarthy, L. Kuchnir, K. Kuczera, F. T. K. Lau, C. Mattos, S. Michnick, T. Ngo, D. T. Nguyen, B. Prodhom, W. E. Reiher III, B. Roux, M. Schlenkrich, J. C. Smith, R. Stote, J. Straub, M. Watanabe, J. Wiorkiewicz–Kuczera, D. Yin, and M. Karplus. All-atom empirical potential for molecular modeling and dynamics studies of proteins. *Journal of Physical Chemistry B*, 102:3586–3616, 1998.

[12] D. Sitkoff, K. A. Sharp, and B. Honig. Accurate calculation of hydration free energies using macroscopic solvent models. *Journal of Physical Chemistry B*, 98:1978–1988, 1994.

[13] D. F. Green and B. Tidor. Evaluation of ab initio charge determination methods for use in continuum solvation calculations. *Journal of Physical Chemistry B*, 107(37):10261–10273, 2003.

[14] I. Massova and P. A. Kollman. Computational alanine scanning to probe protein-protein interactions: A novel approach to evaluate binding free energies. *Journal of the American Chemical Society*, 121:8133–8143, 1999.

[15] J. Warwicker and H. C. Watson. Calculation of the electric potential in the active site cleft due to alpha-helix dipoles. *Journal of Molecular Biology*, 157:671–679, 1982.

[16] P. B. Shaw. Theory of the Poisson Green's-function for discontinuous dielectric media with an application to protein biophysics. *Physical Review A*, 32(4):2476–2487, 1985.

[17] R. J. Zauhar and R. S. Morgan. A new method for computing the macromolecular electric-potential. *Journal of Molecular Biology*, 186(4):815–820, 1985.

[18] I. Klapper, R. Hagstrom, R. Fine, K. Sharp, and B. Honig. Focusing of electric fields in the active site of Cu-Zn superoxide dismutase: Effects of ionic strength and amino-acid modification. *Proteins: Structure, Function, Genetics*, 1:47–59, 1986.

[19] R. J. Zauhar and R. S. Morgan. The rigorous computation of the molecular electric potential. *Journal of Computational Chemistry*, 9(2):171–187, 1988.

[20] M. K. Gilson and B. Honig. Calculation of the total electrostatic energy of a macromolecular system: Solvation energies, binding energies, and conformational analysis. *Proteins: Structure, Function, Genetics*, 4:7–18, 1988.

[21] B. J. Yoon and A. M. Lenhoff. A boundary element method for molecular electrostatics with electrolyte effects. *Journal of Computational Chemistry*, 11(9):1080–1086, 1990.

[22] A. Nicholls and B. Honig. A rapid finite difference algorithm, utilizing successive over-relaxation to solve the Poisson–Boltzmann equation. *Journal of Computational Chemistry*, 12:435–445, 1991.

[23] A. H. Juffer, E. F. F. Botta, B. A. M. van Keulen, A. van der Ploeg, and H. J. C. Berendsen. The electric potential of a macromolecule in a solvent: A fundamental approach. *Journal of Computational Physics*, 97(1):144–171, 1991.

[24] M. J. Holst. *Multilevel Methods for the Poisson-Boltzmann Equation*. PhD thesis, Univ. of Ill. at Urbana-Champaign, 1993.

[25] T. J. You and S. C. Harvey. Finite-element approach to the electrostatics of macromolecules with arbitrary geometries. *Journal of Computational Chemistry*, 14:484–501, 1993.

[26] J. D. Madura, J. M. Briggs, R. C. Wade, M. E. Davis, B. A. Luty, A. Ilin, J. Antosiewicz, M. K. Gilson, B. Bagheri, L. Ridgway-Scott, and J. A. McCammon. Electrostatics and diffusion of molecules in solution: Simulations with the University of Houston Brownian Dynamics program. *Computer Physics Communications*, 91:57–95, 1995.

[27] E. O. Purisima and S. H. Nilar. A simple yet accurate boundary-element method for continuum dielectric calculations. *Journal of Computational Chemistry*, 16(6):681–689, 1995.

[28] R. Bharadwaj, A. Windemuth, S. Sridharan, B. Honig, and A. Nicholls. The fast multipole boundary element method for molecular electrostatics: An optimal approach for large systems. *Journal of Computational Chemistry*, 16(7):898–913, 1995.

[29] J. Liang and S. Subramaniam. Computation of molecular electrostatics with boundary element methods. *Biophysical Journal*, 73(4):1830–1841, 1997.

[30] M. Holst, N. Baker, and F. Wang. Adaptive multilevel finite element solution of the Poisson–Boltzmann equation I. Algorithms and examples. *Journal of Computational Chemistry*, 21:1319–1342, 2000.

[31] W. Rocchia, E. Alexov, and B. Honig. Extending the applicability of the nonlinear Poisson–Boltzmann equation: Multiple dielectric constants and multivalent ions. *Journal of Physical Chemistry B*, 105:6507–6514, 2001.

[32] A. H. Boschitsch, M. O. Fenley, and H.-X. Zhou. Fast boundary element method for the linear Poisson–Boltzmann equation. *Journal of Physical Chemistry B*, 106(10):2741–54, 2002.

[33] A. H. Boschitsch and M. O. Fenley. Hybrid boundary element and finite difference method for solving the nonlinear Poisson–Boltzmann equation. *Journal of Computational Chemistry*, 25(7):935–955, 2004.

[34] L.-P. Lee and B. Tidor. Optimization of electrostatic binding free energy. *Journal of Chemical Physics*, 106:8681–8690, 1997.

[35] E. Kangas and B. Tidor. Optimizing electrostatic affinity in ligand–receptor binding: Theory, computation, and ligand properties. *Journal of Chemical Physics*, 109:7522–7545, 1998.

[36] M. D. Altman, J. P. Bardhan, J. K. White, and B. Tidor. An efficient and accurate surface formulation for biomolecule electrostatics in non-ionic solution. In *Engineering in Medicine and Biology Conference (EMBC)*, 2005.

[37] J. Tausch, J. Wang, and J. White. Improved integral formulations for fast 3-D method-of-moment solvers. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 20(12):1398–1405, 2001.

[38] J. P. Bardhan, M. D. Altman, J. K. White, and B. Tidor. Numerical integration techniques for curved-panel discretizations of molecule–solvent interfaces. *(in preparation)*.

[39] R. J. Zauhar. SMART: A solvent-accessible triangulated surface generator for molecular graphics and boundary-element applications. *Journal of Computer-Aided Molecular Design*, 9(2):149–159, 1995.

[40] C. Tanford and J. G. Kirkwood. Theory of protein titration curves I. general equations for impenetrable spheres. *Journal of the American Chemical Society*, 59:5333–5339, 1957.

[41] W.C. Still, A. Tempczyk, R. C. Hawley, and T. F. Hendrickson. Semianalytical treatment of solvation for molecular mechanics and dynamics. *Journal of the American Chemical Society*, 112(16):6127–6129, 1990.

[42] A. Ghosh, C. S. Rapp, and R. A. Friesner. Generalized Born model based on a surface integral formulation. *Journal of Physical Chemistry B*, 102:10983–10990, 1998.

[43] J. P. Bardhan, M. D. Altman, S. M. Lippow, B. Tidor, and J. K. White. A curved panel integration technique for molecular surfaces. In *Modeling and Simulation of Microsystems (Nanotech)*, volume 1, pages 512–515, 2005.

[44] A. Stroud. *Approximate Calculation of Multiple Integrals*. Prentice Hall, 1971.

[45] J. L. Hess and A. M. O. Smith. Calculation of non-lifting potential flow about arbitrary three-dimensional bodies. *Journal of Ship Research*, 8(2):22–44, 1962.

[46] J. N. Newman. Distribution of sources and normal dipoles over a quadrilateral panel. *Journal of Engineering Mathematics*, 20(2):113–126, 1986.

[47] C. Pozrikidis. *A Practical Guide to Boundary-Element Methods with the Software Library BEMLIB*. Chapman & Hall/CRC Press, 2002.

[48] E. Turco and M. Aristodemo. A three-dimensional B-spline boundary element. *Computer Methods in Applied Mechanics and Engineering*, 155(1-2):119–128, 1998.

[49] F. M. Richards. Areas, volumes, packing, and protein structure. *Annual Review of Biophysics and Bioengineering*, 6:151–176, 1977.

[50] M. D. Altman, J. P. Bardhan, J. K. White, and B. Tidor. Accurate solution of multi-region continuum electrostatic problems using the linearized Poisson–Boltzmann equation and curved boundary elements. *(in preparation)*.

[51] J. D. Jackson. *Classical Electrodynamics*. Wiley, 3^{rd} edition, 1998.

[52] D. M. Chipman. Solution of the linearized Poisson–Boltzmann equation. *Journal of Chemical Physics*, 120(12):5566–5575, 2004.

[53] R. Kress. *Linear Integral Equations*. Springer–Verlag, second edition, 1999.

[54] K. E. Atkinson. *The Numerical Solution of Integral Equations of the Second Kind*. Cambridge University Press, 1997.

[55] T. Grycuk. Deficiency of the Coulomb-field approximation in the generalized Born model: An improved formula for Born radii evaluation. *Journal of Chemical Physics*, 119(9):4817–4826, 2003.

[56] M. Wojciechowski and B. Lesyng. Generalized Born model: Analysis, refinement, and applications to proteins. *Journal of Physical Chemistry B*, 108:18368–18376, 2004.

[57] B. Lee and F. M. Richards. The interpretation of protein structures: Estimation of static accessibility. *Journal of Molecular Biology*, 55(3):379–400, 1971.

[58] B. R. Brooks, R. E. Bruccoleri, B. D. Olafson, D. J. States, S. Swaminathan, and M. Karplus. CHARMM: a program for macromolecular energy, minimization, and dynamics calculations. *Journal of Computational Chemistry*, 4:187–217, 1983.

[59] M. L. Connolly. Analytical molecular surface calculation. *Journal of Applied Crystallography*, 16:548–558, 1983.

[60] C. L. Bajaj, V. Pascucci, A. Shamir, R. J. Holt, and A. N. Netravali. Dynamic maintenance and visualization of molecular surfaces. *Discrete Applied Mathematics*, 127(1):23–51, 2003.

[61] P. Laug and H. Borouchaki. Molecular surface modeling and meshing. *Engineering with Computers*, 18(3):199–210, 2002.

[62] W. S. Cai, M. S. Zhang, and B. Maigret. New approach for representation of molecular surface. *Journal of Computational Chemistry*, 19(16):1805–1815, 1998.

[63] J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam. Analytical shape computation of macromolecules: I. Molecular area and volume through alpha shape. *Proteins: Structure, Function, and Bioinformatics*, 33(1):1–17, 1998.

[64] J. Liang, H. Edelsbrunner, P. Fu, P. V. Sudhakar, and S. Subramaniam. Analytical shape computation of macromolecules: II. Inaccessible cavities in proteins. *Proteins: Structure, Function, and Bioinformatics*, 33(1):18–29, 1998.

[65] A. H. Juffer and P. J. Vogel. A flexible triangulation method to describe the solvent-accessible surface of biopolymers. *Journal of Computer-Aided Molecular Design*, 12(3):289–299, 1998.

[66] S. L. Chan and E. O. Purisima. Molecular surface generation using marching tetrahedra. *Journal of Computational Chemistry*, 19(11):1268–1277, 1998.

[67] M. Totrov and R. Abagyan. The contour-buildup algorithm to calculate the analytical molecular surface. *Journal of Structural Biology*, 116(1):138–143, 1996.

[68] M. Sanner, A. J. Olson, and J. C. Spehner. Reduced surface: An efficient way to compute molecular surfaces. *Biopolymers*, 38:305–320, 1996.

[69] R. J. Zauhar and R. S. Morgan. Computing the electric-potential of biomolecules: application of a new method of molecular-surface triangulation. *Journal of Computational Chemistry*, 11(5):603–622, 1990.

[70] A. J. Bordner and G. A. Huber. Boundary element solution of the linear Poisson–Boltzmann equation and a multipole method for the rapid calculation of forces on macromolecules in solution. *Journal of Computational Chemistry*, 24(3):353–367, 2003.

[71] X. Wang, J. White, and J. Newman. Robust algorithms for boundary-element integrals on curved surfaces. In *Modeling and Simulation of Microsystems (MSM)*, pages 473–476, 2000.

[72] X. Wang. *FastStokes: A fast 3-D fluid simulation program for Micro-electro-mechanical systems*. PhD thesis, Massachusetts Institute of Technology, 2002.

[73] D. J. Willis, J. Peraire, and J. K. White. A quadratic basis function, quadratic geometry, high order panel method. In *44th AIAA Aerospace Sciences Meeting, AIAA-2006-1253*, 2006.

[74] M. P. do Carmo and M. P. Carmo. *Differential Geometry of Curves and Surfaces*. Pearson Education, 1976.

[75] *Matlab v.6*. Mathworks, Inc.

[76] M. Connolly. Molecular surface package. http://connolly.best.vwh.net, 2000.

[77] A. M. Buckle, G. Schreiber, and A. R. Fersht. Protein-protein recognition: Crystal structural analysis of a barnase-barstar complex at 2.0-å resolution. *Biochemistry*, 33(30):8878–8889, 1994.

[78] H. M. Berman, J. Westbrook, Z. Feng, G. Gilliland, T. N. Bhat, H. Weissig, I. N. Shindyalov, and P. E. Bourne. The protein data bank. *Nucleic Acids Research*, 28:235–242, 2000.

[79] J. N. Scarsdale, C. Van Alsenoy, V. J. Klimkowski, L. Schafer, and F. A. Momany. Ab initio studies of molecular geometries. 27. optimized molecular structures and conformational analysis of N^{α}-*N*-methylalaninamide and comparison with peptide crystal data and empirical calculations. *Journal of the American Chemical Society*, 105:3438–3445, 1983.

[80] 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. *Journal of the American Chemical Society*, 118(45):11225–11236, 1996.

[81] W. L. Jorgensen and J. D. Madura. Temperature and size dependence for Monte-Carlo simulations of TIP4P water. *Molecular Physics*, 56(6):1381–1392, 1985.

[82] Y. Saad and M. Schultz. GMRES: A generalized minimal residual algorithm for solving nonsymmetric linear systems. *SIAM Journal of Scientific and Statistical Computing*, 7:856–869, 1986.

[83] M. D. Altman, J. P. Bardhan, B. Tidor, and J. K. White. FFTSVD: A fast multiscale boundary-element method solver suitable for BioMEMS and biomolecule simulation. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 25(2):274–284, 2006.

[84] T. E. Creighton. *Proteins: structures and molecular properties*. W. H. Freeman and Company, 1993.

[85] J. Voldman, M. L. Gray, and M. A. Schmidt. Microfabrication in biology and medicine. *Annual Review of Biomedical Engineering*, 1:401–425, 1999.

[86] D. S. Gray, T. L. Tan, J. Voldman, and C. S. Chen. Dielectrophoretic registration of living cells to a microelectrode array. *Biosensors and Bioelectronics*, 19(7):771–80, 2004.

[87] G.-B. Lee, S.-H. Chen, G.-R. Huang, W.-C. Sung, and Y.-H. Lin. Microfabricated plastic chips by hot embossing methods and their applications for DNA separation and detection. *Sensors and Actuators B*, 75:142–148, 2001.

[88] T. P. Burg and S. R. Manalis. Suspended microchannel resonators for biomolecule detection. *Applied Physics Letters*, 83(13):2698–2700, 2003.

[89] T. Korsmeyer. Design tools for bioMEMS. In *IEEE/ACM Design Automation Conference (DAC)*, pages 622–627, 2004.

[90] J. White. CAD challenges in BioMEMS design. In *IEEE/ACM Design Automation Conference (DAC)*, pages 629–632, 2004.

[91] S. D. Senturia, R. M. Harris, B. P. Johnson, S. Kim, K. Nabors, M. A. Shulman, and J. K. White. A computer-aided design system for microelectromechanical systems (MEMCAD). *Journal of Microelectromechanical Systems*, 1(1):3–13, 1992.

[92] C. A. Savran, S. M. Knudsen, A. D. Ellington, and S. R. Manalis. Micromechanical detection of proteins using aptamer-based receptor molecules. *Analytical Chemistry*, 76:3194–3198, 2004.

[93] R. A. Potyrailo, R. C. Conrad, A. D. Ellington, and G. M. Hieftje. Adapting selected nucleic acid ligands (aptamers) to biosensors. *Analytical Chemistry*, 70:3419–3425, 1998.

[94] B. Honig and A. Nicholls. Classical electrostatics in biology and chemistry. *Science*, 268:1144–1149, 1995.

[95] M. K. Gilson, K. A. Sharp, and B. H. Honig. Calculating the electrostatic potential of molecules in solution: Method and error assessment. *Journal of Computational Chemistry*, 9:327–335, 1988.

[96] K. Nabors, F. T. Korsmeyer, F. T. Leighton, and J. White. Preconditioned, adaptive, multipole-accelerated iterative methods for three-dimensional first-kind integral equations of potential theory. *SIAM Journal on Scientific Computing*, 15(3):713–735, 1994.

[97] L. Greengard and V. Rokhlin. A fast algorithm for particle simulations. *Journal of Computational Physics*, 73:325–348, 1987.

[98] L. Greengard. *The Rapid Evaluation of Potential Fields in Particle Systems*. MIT Press, 1988.

[99] W. Hackbusch. A sparse matrix arithmetic based on H-matrices. i. introduction to H-matrices. *Computing*, 62(2):89–108, 1999.

[100] W. Hackbusch and B. N. Khoromskij. A sparse H-matrix arithmetic. ii. application to multi-dimensional problems. *Computing*, 64(1):21–47, 2000.

[101] S. Borm, L. Grasedyck, and W. Hackbusch. Introduction to hierarchical matrices with applications. *Engineering Analysis with Boundary Elements*, 27(5):405–22, 2003.

[102] J. R. Phillips and J. K. White. A precorrected-FFT method for electrostatic analysis of complicated 3-D structures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 16(10):1059–1072, 1997.

[103] W. Shi, J. Liu, N. Kakani, and T. Yu. A fast hierarchical algorithm for 3-D capacitance extraction. In *IEEE/ACM Design Automation Conference (DAC)*, 1998.

[104] J. Tausch and J. White. A multiscale method for fast capacitance extraction. In *IEEE/ACM Design Automation Conference (DAC)*, pages 537–542, 1999.

[105] E. T. Ong, K. M. Lim, K. H. Lee, and H. P. Lee. A fast algorithm for three-dimensional potential fields calculation: fast Fourier transform on multipoles. *Journal of Computational Physics*, 192(1):244–61, 2003.

[106] E. T. Ong, H. P. Lee, and K. M. Lim. A parallel fast Fourier transform on multipoles (FFTM) algorithm for electrostatics analysis of three-dimensional structures. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 23(7):1063–1072, 2004.

[107] G. Biros, L. Ying, and D. Zorin. A fast solver for the Stokes equations with distributed forces in complex geometries. *Journal of Computational Physics*, 193(1):317–348, 2004.

[108] L. Ying, G. Biros, and D. Zorin. A kernel-independent adaptive fast multipole algorithm in two and three dimension. *Journal of Computational Physics*, 196(2):591–626, 2004.

[109] S. Kapur and D. E. Long. IES^3: A fast integral equation solver for efficient 3-dimensional extraction. In *International Conference on Computer Aided Design (ICCAD)*, pages 448–55, 1997.

[110] S. Kapur and D. E. Long. IES^3: Efficient electrostatic and electromagnetic simulation. *IEEE Computational Science and Engineering*, 5(4):60–7, 1998.

[111] L. Greengard, J. Huang, V. Rokhlin, and S. Wandzura. Accelerating fast multipole methods for the Helmholtz equation at low frequencies. *IEEE Computational Science and Engineering*, 5(3):32–38, 1998.

[112] D. Gope and V. Jandhyala. PILOT: A fast algorithm for enhanced 3D parasitic extraction efficiency. In *IEEE Electrical Performance of Electronic Packaging*, 2003.

[113] K. Nabors and J. White. FASTCAP: A multipole accelerated 3-D capacitance extraction program. *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 10(10):1447–1459, 1991.

[114] Z. Zhu, B. Song, and J. White. Algorithms in FastImp: A fast and wideband impedance extraction program for complicated 3D geometries. *IEEE/ACM Design Automation Conference (DAC)*, 2003.

[115] N. R. Aluru and J. White. A fast integral equation technique for analysis of microflow sensors based on drag force calculations. In *Modeling and Simulation of Microsystems (Nanotech)*, 1998.

[116] W. Ye, X. Wang, and J. White. A fast Stokes solver for generalized flow problems. In *Modeling and Simulation of Microsystems (Nanotech)*, 2000.

[117] S. S. Kuo, M. D. Altman, J. P. Bardhan, B. Tidor, and J. K. White. Fast methods for simulation of biomolecule electrostatics. In *International Conference on Computer Aided Design (ICCAD)*, 2002.

[118] J. Kanapka, J. Phillips, and J. White. Fast methods for extraction and sparsification of substrate coupling. In *IEEE/ACM Design Automation Conference (DAC)*, pages 738–743, 2000.

[119] J. Fliege and U. Maier. *IMA Journal of Numerical Analysis*, 19:317–334, 1999.

[120] M. Frigo and S. G. Johnson. FFTW: An adaptive software architecture for the FFT. In *Proc. 1998 IEEE Intl. Conf. Acoustics Speech and Signal Processing*, volume 3, pages 1381–1384. IEEE, 1998.

[121] B. P. Mosier, J. I. Molho, and J. G. Santiago. Photobleached-fluorescence imaging of microflows. *Experiments in Fluids*, 33(4):545–554, 2002.

[122] S. I. Cho, S.-H. Lee, D. S. Chung, and Y.-K. Kim. Bias-free pneumatic sample injection in microchip electrophoresis. *Journal of Chromatography A*, 1063(1-2):253–256, 2005.

[123] Z. Zhu. Efficient techniques for wideband impedance extraction of complex 3-dimensional geometries. Master's thesis, Massachusetts Institute of Technology, 2002.

[124] A. E. Ruehli and P. A. Brennan. Efficient capacitance calculations for three-dimensional multiconductor systems. *IEEE Transactions on Microwave Theory and Techniques*, 21:76–82, 1973.

[125] E. T. Ong, K. H. Lee, and K. M. Lim. Singular elements for electro-mechanical coupling analysis of micro-devices. *Journal of Micromechanics and Microengineering*, 13:482–490, 2003.

[126] E. T. Ong and K. M. Lim. Three-dimensional singular boundary elements for corner and edge singularities in potential problems. *Engineering Analysis with Boundary Elements*, 29:175–189, 2005.

[127] N. A. Baker. Poisson–Boltzmann methods for biomolecular electrostatics. *Methods in Enzymology*, 383:94–+, 2004.

[128] I. Stakgold. *Green's Functions and Boundary Value Problems*. John Wiley & Sons, 2nd edition, 1998.

[129] M. Holst and F. Saied. Multigrid solution of the Poisson–Boltzmann equation. *Journal of Computational Chemistry*, 14:105–113, 1993.

[130] N. A. Baker, D. Sept, M. J. Holst, and J. A. McCammon. Electrostatics of nanoysystems: Application to microtubules and the ribosome. *Proceedings of the National Academy of Sciences of the USA*, 98:10037–10041, 2001.

[131] W. Rocchia, S. Sridharan, A. Nicholls, E. Alexov, A. Chiabrera, and B. Honig. Rapid grid-based construction of the molecular surface and the use of induced surface charge to calculate reaction field energies: Applications to the molecular systems and geometric objects. *Journal of Computational Chemistry*, 23:128–137, 2002.

[132] C. M. Cortis and R. A. Friesner. Numerical simulation of the poisson-boltzmann equation using tetrahedral finite-element meshes. *Journal of Computational Chemistry*, 18(13):1591–1608, 1997.

[133] H. X. Zhou. Boundary-element solution of macromolecular electrostatics - interaction energy between 2 proteins. *Biophysical Journal*, 65:955–963, 1993.

[134] R. J. Zauhar and A. Varnek. A fast and space-efficient boundary element method for computing electrostatic and hydration effects in large molecules. *Journal of Computational Chemistry*, 17:864–877, 1996.

[135] Y. N. Vorobjev and H. A. Scheraga. A fast adaptive multigrid boundary element method for macromolecular electrostatic computations in a solvent. *Journal of Computational Chemistry*, 18(4):569–583, 1997.

[136] S. Hofinger and T. Simonson. Dielectric relaxation in proteins: A continuum electrostatics model incorporating dielectric heterogeneity of the protein and time-dependent charges. *Journal of Computational Chemistry*, 22:290–305, 2001.

[137] B. Z. Lu, D. Q. Zhang, and J. A. McCammon. Computation of electrostatic forces between solvated molecules determined by the Poisson–Boltzmann equation using a boundary element method. *Journal of Chemical Physics*, 122, 2005.

[138] M. L. Connolly. Solvent-accessible surfaces of proteins and nucleic-acids. *Science*, 221:709–713, 1983.

[139] A. H. Boschitsch, M. O. Fenley, and W. K. Olson. A fast adaptive multipole algorithm for calculating screened Coulomb (Yukawa) interactions. *Journal of Computational Physics*, 151:212–241, 1999.

[140] H. Cheng, L. Greengard, and V. Rokhlin. A fast adaptive multipole algorithm in three dimensions. *Journal of Computational Physics*, 155:468–498, 1999.

[141] J.-L. Guermond. Numerical quadratures for layer potentials over curved domains in R^3. *SIAM Journal on Numerical Analysis*, 29(5):1347–1369, 1992.

[142] M. Prabu-Jeyabalan, E. Nalivaika, and C. A. Schiffer. How does a symmetric dimer recognize an asymmetric substrate? A substrate complex of HIV-1 protease. *Journal of Molecular Biology*, 301:1207–1220, 2000.

[143] Z. S. Hendsch and B. Tidor. Electrostatic interactions in the GCN4 leucine zipper: Substantial contributions arise from intramolecular interactions enhanced on binding. *Protein Science*, 8:1381–1392, 1999.

[144] S. Spector, M. H. Wang, S. A. Carp, J. Robblee, Z. S. Hendsch, R. Fairman, B. Tidor, and D. P. Raleigh. Rational modification of protein stability by the mutation of charged surface residues. *Biochemistry*, 39:872–879, 2000.

[145] D. F. Green and B. Tidor. Design of improved protein inhibitors of HIV-1 cell entry: Optimization of electrostatic interactions at the binding interface. *Proteins: Structure, Function, and Bioinformatics*, 60:644–657, 2005.

[146] G. H. Golub and C. F. Van Loan. *Matrix Computations*. The Johns Hopkins University Press, third edition, 1996.

[147] L.-P. Lee and B. Tidor. Optimization of binding electrostatics: Charge complementarity in the barnase-barstar protein complex. *Protein Science*, 10:362–377, 2001.

[148] A. T. Brunger and M. Karplus. Polar hydrogen positions in proteins - empirical energy placement and neutron-diffraction comparison. *Proteins*, 4:148–156, 1988.

[149] J. Schöberl. NETGEN - an advancing front 2D/3D-mesh generator based on abstract rules. *Computing and Visualization in Science*, 1:42–52, 1997.

[150] J. Shen and J. Wendoloski. Electrostatic binding energy calculation using the finite difference solution to the linearized Poisson–Boltzmann equation: Assessment of its accuracy. *Journal of Computational Chemistry*, 17:350–357, 1996.

[151] G. Schreiber and A. R. Fersht. Interaction of barnase with its polypeptide inhibitor barstar studied by protein engineering. *Biochemistry*, 32:5145–5150, 1993.

[152] G. Schreiber and A. R. Fersht. Energetics of protein-protein interactions - analysis of the barnase-barstar interface by single mutations and double mutant cycles. *Journal of Molecular Biology*, 248:478–486, 1995.

[153] F. B. Sheinerman and B. Honig. On the role of electrostatic interactions in the design of protein-protein interfaces. *Journal of Molecular Biology*, 318:161–177, 2002.

[154] L. T. Chong, S. E. Dempster, Z. S. Hendsch, L.-P. Lee, and B. Tidor. Computation of electrostatic complements to proteins: A case of charge stabilized binding. *Protein Science*, 7(1):206–210, 1998.

[155] F. Dong, M. Vijayakumar, and H. X. Zhou. Comparison of calculation and experiment implicates significant electrostatic contributions to the binding stability of barnase and barstar. *Biophysical Journal*, 85:49–60, 2003.

[156] C. Frisch, G. Schreiber, C. M. Johnson, and A. R. Fersht. Thermodynamics of the interaction of barnase and barstar: Changes in free energy versus changes in enthalpy on mutation. *Journal of Molecular Biology*, 267:696–706, 1997.

[157] G. Schreiber, C. Frisch, and A. R. Fersht. The role of Glu73 of barnase in catalysis and the binding of barstar. *Journal of Molecular Biology*, 270:111–122, 1997.

[158] D. G. Covell and A. Wallqvist. Analysis of protein-protein interactions and the effects of amino acid mutations on their energetics. The importance of water molecules in the binding epitope. *Journal of Molecular Biology*, 269:281–297, 1997.

[159] T. Kortemme and D. Baker. A simple physical model for binding energy hot spots in protein-protein complexes. *Proceedings of the National Academy of Sciences of the USA*, 99:14116–14121, 2002.

[160] T. Wang, S. Tomic, R. R. Gabdoulline, and R. C. Wade. How optimal are the binding energetics of barnase and barstar? *Biophysical Journal*, 87(3):1618–1630, 2004.

[161] B. I. Dahiyat and S. L. Mayo. Protein design automation. *Protein Science*, 5:895–903, 1996.

[162] J. G. Saven. Combinatorial protein design. *Current Opinion Structural Biology*, 12:453–458, 2002.

[163] H. Gohlke and D. A. Case. Converging free energy estimates: MM-PB(GB)SA studies on the protein-protein complex Ras-Raf. *Journal of Computational Chemistry*, 25:238–250, 2004.

[164] J. P. Bardhan, M. D. Altman, J. K. White, and B. Tidor. A co-optimization approach for optimizing electrostatic interactions between biomolecules. *(in preparation)*.

[165] J. P. Bardhan, J. H. Lee, S. S. Kuo, M. D. Altman, B. Tidor, and J. K. White. Fast methods for biomolecule charge optimization. In *Modeling and Simulation of Microsystems (Nanotech)*, 2003.

[166] J. P. Bardhan, J. H. Lee, M. D. Altman, S. Benson, S. Leyffer, B. Tidor, and J. K. White. Biomolecule electrostatic optimization with an implicit Hessian. In *Modeling and Simulation of Microsystems (Nanotech)*, 2004.

[167] L.-P. Lee and B. Tidor. Barstar is electrostatically optimized for tight-binding to barnase. *Nature Structural Biology*, 8:73–76, 2001.

[168] E. Kangas and B. Tidor. Charge optimization leads to favorable electrostatic binding free energy. *Physical Review E*, 59(5):5958–5961, 1999.

[169] E. Kangas and B. Tidor. Electrostatic complementarity at ligand binding sites: Application to chorismate mutase. *Journal of Physical Chemistry*, 105:880–888, 2001.

[170] E. Kangas and B. Tidor. Electrostatic specificity in molecular ligand design. *Journal of Chemical Physics*, 112:9120–9131, 2000.

[171] A. Mandal and D. Hilvert. Charge optimization increases the potency and selectivity of a chorismate mutase inhibitor. *Journal of the American Chemical Society*, 125(19):5598–5599, 2003.

[172] T. Sulea and E. O. Purisima. Optimizing ligand charges for maximum binding affinity. A solvated interaction energy approach. *Journal of Physical Chemistry B*, 105(4):889–899, 2001.

[173] T. Sulea and E. O. Purisima. Profiling charge complementarity and selectivity for binding at the protein surface. *Biophysical Journal*, 84(5):2883–2896, 2003.

[174] P. A. Sims, C. F. Wong, and J. A. McCammon. Charge optimization of the interface between protein kinases and their ligands. *Journal of Computational Chemistry*, 25(11):1416–1429, 2004.

[175] D. F. Green and B. Tidor. Escherichia coli glutaminyl–tRNA synthetase is electrostatically optimized for binding of its cognate substrates. *Journal of Molecular Biology*, 342(2):435–452, 2004.

[176] K. A. Armstrong, B. Tidor, and A. C. Cheng. Optimal charges in lead progression: A structure-based neuraminidase case study. *Journal of Medicinal Chemistry*, 49:2470–2477, 2006.

[177] M. K. Gilson. Sensitivity analysis and charge-optimization for flexible ligands: Applicability to lead optimization. *Journal of Chemical Theory and Computation*, 2(2):259–270, 2006.

[178] E. O. Purisima. Fast summation boundary element method for calculating solvation free energies of macromolecules. *Journal of Computational Chemistry*, 19(13):1494–1504, 1998.

[179] D. P. Bertsekas. *Nonlinear Programming*. Athena Scientific, second edition, 1999.

[180] S. J. Wright. *Primal-Dual Interior Point Methods*. SIAM, 1997.

[181] E. De Sturler and J. Liesen. Block-diagonal and constraint preconditioners for nonsymmetric indefinite linear systems. Part I: Theory. *SIAM Journal of Scientific Computing*, 26(5):1598–1619, 2005.

[182] M. F. Murphy, G. H. Golub, and A. J. Wathen. A note on preconditioning for indefinite linear systems. *SIAM Journal of Scientific Computing*, 21(6):1969–1972, 2000.

[183] R. A. Horn and C. R. Johnson. *Matrix Analysis*. Cambridge University Press, 1990.

[184] S. Balay, K. Buschelman, W. D. Gropp, D. Kaushik, M. Knepley, L. C. McInnes, B. F. Smith, and H. Zhang. PETSc home page. http://www.mcs.anl.gov/petsc, 2001.

[185] R. Byrd, M. E. Hribar, and J. Nocedal. An interior point method for large scale nonlinear programming. *SIAM Journal on Optimization*, 9:877–900, 1999.

[186] R. Krishnan, I. Mochalkin, R. Arni, and A. Tulinsky. Structure of thrombin complexed with selective non-electrophilic inhibitors having cyclohexyl moieties at p1. *Acta Crystallographica, Section D*, 56:294–303, 2000.

[187] J. J. Havranek and P. B. Harbury. Tanford–Kirkwood electrostatics for protein modeling. *Proceedings of the National Academy of Sciences of the USA*, 96(20):11145–11150, 1999.

[188] E. G. Alexov and M. R. Gunner. Incorporating protein conformational flexibility into the calculation of pH-dependent protein properties. *Biophysical Journal*, 72(5):2075–2093, 1997.

[189] F. B. Sheinerman, R. Norel, and B. Honig. Electrostatic aspects of protein-protein interactions. *Current Opinions in Structural Biology*, 11:153–159, 2000.

[190] V. Lounnas, B. M. Pettitt, L. Findsen, and S. Subramaniam. A microscopic view of protein solvation. *Journal of Physical Chemistry*, 18:7157–7159, 1992.

[191] J. A. McCammon and S. C. Harvey. *Dynamics of Proteins and Nucleic Acids*. Cambridge University Press, Cambridge, 1987.

[192] C. L. Brooks, III, M. Karplus, and B. M Pettitt. Proteins: A theoretical perspective of dynamics, structure and thermodynamics. *Adv. Chem. Phys.*, 71:1–249, 1988.

[193] S. W. Rick and B. J. Berne. The aqueous solvation of water: A comparison of continuum methods with molecular dynamics. *Journal of the American Chemical Society*, 116:3949–3954, 1994.

[194] M. K. Gilson, A. Rashin, R. Fine, and B. Honig. On the calculation of electrostatic interactions in proteins. *Journal of Molecular Biology*, 183:503–516, 1985.

[195] S. M. Rao, T. K. Sarkar, and R. F. Harrington. The electrostatic field of conducting bodies in multiple dielectric media. *IEEE Transactions on Microwave Theory and Techniques*, 32(11):1441–1448, 1984.

[196] K. Nabors. *Efficient Three-Dimensional Capacitance Extraction*. PhD thesis, Massachusetts Institute of Technology, 1993.

[197] V. Rokhlin. Rapid solution of integral equation of classical potential theory. *Journal of Computational Physics*, 60:187–207, 1985.