Fast direct methods for molecular electrostatics

Kenneth L. Ho

Courant Institute of Mathematical Sciences and Program in Computational Biology
New York University, 251 Mercer Street, New York, NY 10012
ho@courant.nyu.edu

Introduction

Electrostatic interactions are well-known to play an important role in determining protein structure and function, with charge interaction energies reaching the order of 10–100 times the characteristic energy kT. The electrostatic potential describing the effect of such interactions is therefore an important quantity to consider in the analysis of proteins, whether arising through the study of natural biomolecules or through the design and engineering of synthetic ones.

The calculation of the electrostatic potential of a solvated biomolecule is computationally intensive, and its cost increases very quickly with the size of the protein under consideration. Thus, fast algorithms are necessary to make the electrostatic analysis of macromolecules feasible. Current state-of-the-art methods are typically iterative, coupling fast matrix-vector multiplication routines with conjugate gradient-type schemes. Here, we consider instead *direct* methods, which enjoy several significant advantages over their iterative counterparts, and describe our recent progress in the development of fast direct solvers for molecular electrostatics.

Direct solvers of $Ax = b$

(1) Reduced sensitivity to the conditioning of A.

(2) Fast application of A^{-1}, once computed, to multiple vectors b.

(3) Efficiency at handling low-rank perturbations of A.

This is joint work with Leslie Greengard and Zydrunas Gimbutas.

Potential equations

Σ	molecular surface
Ω_1	protein interior
Ω_2	ionic solution
ν	unit normal to Σ
φ	electrostatic potential
$\varepsilon = \begin{cases} \varepsilon_1 \text{ in } \Omega_1 \\ \varepsilon_2 \text{ in } \Omega_2 \end{cases}$	dielectric permittivity
q_i	charge strengths
$\boldsymbol{r}_i$	charge locations
$\lambda = \sqrt{\frac{2I\epsilon^2}{\varepsilon_2 kT}}$	inverse Debye length
$K_L = \frac{1}{4\pi r}$	Laplace potential
$K_Y = \frac{e^{-\lambda r}}{4\pi r}$	Yukawa potential
$f * g$	convolution on Σ
σ	inner charge density
μ	outer charge density

The electrostatic potential is assumed to satisfy Poisson's equation in the protein interior, with sources given by the fixed charges of the protein; and the linearized Poisson-Boltzmann equation in the surrounding ionic solution, which describes the effect of ionic screening in the solvent. Thus, we have the interface potential problem

$$\nabla^2 \varphi(\boldsymbol{r}) = \begin{cases} -(1/\varepsilon_1) \sum_{i=1}^n q_i \delta(\boldsymbol{r} - \boldsymbol{r}_i), & \boldsymbol{r} \in \Omega_1 \\ \lambda^2 \varphi(\boldsymbol{r}), & \boldsymbol{r} \in \Omega_2 \end{cases}$$

with the jump conditions $[\varphi] = [\varepsilon(\partial\varphi/\partial\nu)] = 0$ on Σ.

We use the single-layer potential solution representation

$$\varphi(\boldsymbol{r}) = \begin{cases} (K_L * \sigma)(\boldsymbol{r}) + \varphi_0(\boldsymbol{r}), & \boldsymbol{r} \in \Omega_1 \\ (K_Y * \mu)(\boldsymbol{r}), & \boldsymbol{r} \in \Omega_2 \end{cases}$$

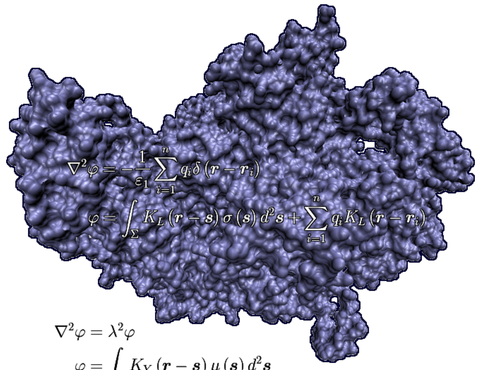

Figure 1: *Potential equations and solution represenation.*

where

$$\varphi_0(\boldsymbol{r}) = \sum_{i=1}^n q_i K_L(\boldsymbol{r} - \boldsymbol{r}_i)$$

is the potential due to the fixed charges, to obtain a *boundary integral* system in the unknown surface charge densities σ and μ. The jump conditions then become

Boundary integral systems

(1) Dimensional reduction from volume to surface

(2) High accuracy from faithful representation of geometry

$$K_L * \sigma - K_Y * \mu = \varphi_0,$$
$$\varepsilon_1 \left(\frac{\sigma}{2} + \frac{\partial K_L}{\partial \nu} * \sigma \right) - \varepsilon_2 \left(-\frac{\mu}{2} + \frac{\partial K_Y}{\partial \nu} * \mu \right) = -\varepsilon_1 \frac{\partial \varphi_0}{\partial \nu}.$$

On discretizing using collocation, with the convolution integrals evaluated by quadrature over a triangle mesh of Σ, we obtain a linear system $Ax = b$, where x is a vector of the inner and outer surface charge densities at the collocation points.

Accelerated solution

We now deal with the linear algebraic problem of solving $Ax = b$. In general, the $N \times N$ system matrix A is dense. However, smoothness of the integral operators allows the high-accuracy representation of well-separated interactions by low-rank approximations. We exploit this by *compressing* A to obtain a $K \times K$ *skeleton matrix*, where typically $K \ll N$. Since direct inversion of an $m \times m$ matrix takes $\mathcal{O}(m^3)$ operations, this dramatically reduces the work required to compute A^{-1}.

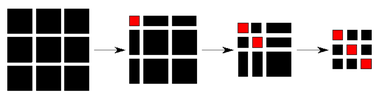

Figure 2: *One-level matrix compression.*

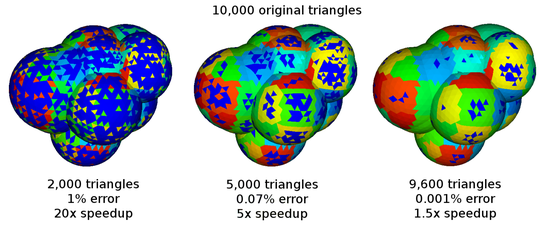

Figure 3: *Surface skeletonization.*

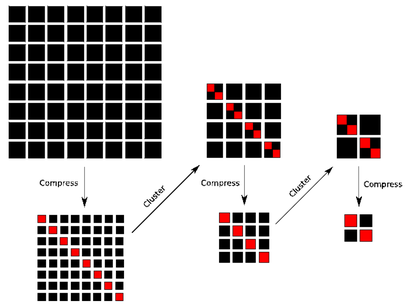

Figure 4: *Multilevel matrix compression.*

This process can be applied recursively by compressing the skeleton after a suitable reordering of the variables by clustering. This leads to a multilevel matrix inversion scheme.

Once the surface charge densities have been determined, the electrostatic potential may be evaluated at any point in space by using the single-layer potential representations. This step may be accelerated as well by using the fast multipole method (FMM) to apply the discretized integral operator.

The overall computational cost of the algorithm is expected to scale as $\mathcal{O}(N \log^2 N)$.

Table 1: *Memory requirements.*

Matrix	Dimensions	Memory use
Full	$200,000 \times 200,000$	320 Gb
Compressed	$2,000 \times 2,000$	32 Mb

Application to pK_a calculations

As an illustration of the power of direct methods, we outline the problem of computing the pK_a of a titratable residue in a protein. Briefly, the pK_a measures the strength of an acid in solution by assessing its protonation state. The pK_a of key residues, like Asp, Glu, Lys, Arg, and His, often give insight into the stability and reactivity of a protein.

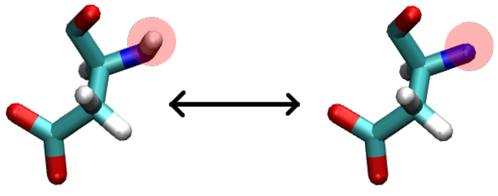

Figure 5: *Protonation of aspartate.*

$\mathrm{p}K_a$	acid dissociation constant
$\mathrm{p}K_i^{\mathrm{intr}}$	shifted pK_a of ith site
θ	protonation state of protein
$\theta_i = \begin{cases} 1 \text{ if HA} \\ 0 \text{ if A}^- \end{cases}$	protonation state of ith site
$\langle \cdot \rangle$	state average
$p(\theta)$	probability of θ
$W(\theta)$	electrostatic energy of θ
k	Boltzmann's constant
T	temeprature
Z	partition function

The pK_a of an acid can be calculated from the Henderson-Hasselbalch equation as the pH at which the acid is half-protonated on average. This involves taking a Boltzmann sum

$$\langle \theta_i \rangle = \sum_\theta \theta_i p(\theta),$$

where

$$p(\theta) = \frac{1}{Z} \exp\left[\ln 10 \sum_i \theta_i \left(\mathrm{p}K_i^{\mathrm{intr}} - \mathrm{p}H \right) - \frac{W(\theta)}{kT} \right]$$

is the probability of protonation state θ of the protein.

Since a protein typically has multiple titrating sites, the calculation of the titration curve $\langle \theta_i \rangle$ as a function of the pH, from which the pK_a may be derived, requires the sampling of many different protonation states θ. While an iterative solver must solve each case essentially independently, a direct solver can take advantage of the fact that θ only changes the vector b in the system $Ax = b$, so once A^{-1} has been computed, each solve $x = A^{-1}b$ may be obtained at minimal cost by applying A^{-1}, which requires only $\mathcal{O}(N)$ operations by using the FMM.

Protonation state sampling

(1) Assume only electrostatic interactions

(2) Boltzmann average has 2^N contributions

(3) Use Monte Carlo methods to break dimensionality barrier

Furthermore, protein conformations are dependent on the electrostatic environment. Thus, different choices of θ are expected to induce local deformations of the surface geometry. This constitutes a low-rank perturbation of A, which a direct method can handle very efficiently, given A^{-1}, by forming the Schur complement.

Table 2: *Fast application of matrix inverse.*

Tolerance	Skeletons	Factorization	Application
$\epsilon = 10^{-2}$	$K = 7,049$	20 min	0.3 s
$\epsilon = 10^{-1}$	$K = 1,596$	10 min	0.1 s

Molecular surface triangulation with $N = 20,600$. Times are for a single-processor 1.9 GHz workstation. For comparision, a FMM-based iterative algorithm requires several seconds for each solve.

Current progress

We currently have an implementation of the Poisson/linearized Poisson-Boltzmann solver in Fortran, as well as an interface to the matrix compression libraries. Multilevel matrix compression is under development, as is an interface to UMFPACK for fast matrix factorization.

Acknowledgments

This research is joint work with Leslie Greengard and Zydrunas Gimbutas. We thank NYU and NSF IGERT grant 0333389 for support.