# Biomolecule Electrostatic Optimization with an Implicit Hessian

J. P. Bardhan*, J. H. Lee*, M. D. Altman†, S. Leyffer⋆, S. Benson⋆, B. Tidor‡,*, J. K. White*

* Department of Electrical Engineering and Computer Science,† Department of Chemistry
‡ Biological Engineering Division
77 Massachusetts Avenue, Cambridge, MA 02139
⋆ Mathematics & Computer Science Division
Argonne National Laboratory, Argonne, IL 60439

## ABSTRACT

Computational rational drug design is the application of computer simulation techniques to improve screening processes for new drugs or to design them *de novo*. The goal is to identify molecules that have high affinity and specificity for a target molecule. Optimizing the electrostatic binding free energy is tractable under certain assumptions as a quadratic optimization problem, but computationally expensive simulations have traditionally been required to determine the associated Hessian. Prior work showed that significant performance gains could be achieved by coupling physical simulation directly to the optimization process and avoiding the calculation of the Hessian. The present paper details recent improvements to this method and the implementation of a practical, full-scale code. Computational results demonstrate the code's efficiency on large problems and accuracy solving a realistic biomolecule optimization problem.

## 1 INTRODUCTION

Computational rational drug design is the application of computer simulation techniques to the problem of identifying new drug molecules that bind tightly and specifically to a given target molecule. The binding affinity is related to the change in free energy due to the binding of the drug molecule, or *ligand*, to the target molecule, or *receptor* [1]–[3]. Several types of atomic and molecular interactions contribute to this free energy change, including the hydrophobic effect, van der Waals interactions, and electrostatic interactions (including hydrogen bonding). Because electrostatic interactions are long range and contribute a significant component of the total binding free energy, it is important to understand how the partial atomic charges in the ligand and receptor interact with each other and with the surrounding solvent. To design optimal ligands, drug designers need to understand what set of ligand partial atomic charges optimize the electrostatic free energy of binding.

If continuum electrostatics are used to model the biomolecular interactions, then under certain assumptions [2] this set of optimal ligand partial charges can be determined by solving a convex quadratic program. Standard optimization techniques, however, typically require the expensive calculation of an explicit Hessian matrix to efficiently solve the program. We therefore introduced in [4] an optimization method that avoids the Hessian calculation; initial results showed promise that the method might significantly accelerate solving these optimization problems. The present paper describes an improved formulation and an implementation that is capable of solving large, realistically sized optimization problems in biomolecule design.

Section 2 introduces the optimization problem, the electrostatic modeling techniques, and the Hessian-implicit primal-dual method of [4]. Section 3 describes recent improvements to the original method and important features of the full-scale implementation. Section 4 presents results that demonstrate the efficiency and accuracy of the new formulation. Section 5 summarizes the contributions of this work and suggests directions for further investigation.

## 2 BACKGROUND

### 2.1 Biomolecule Electrostatic Optimization

We present first the analysis of the electrostatic energy of the ligand alone in solution [1], [2]. A mixed discrete-continuum model is used: the ligand partial atomic charges are treated as point charges at discrete locations (the ligand atom centers) but with continuous value. The electrostatic potential in the interior satisfies the continuum Poisson equation with low dielectric constant; the potential in the solvent satisfies the linearized Poisson-Boltzmann equation and is characterized by a high dielectric constant and the inverse Debye screening length $\kappa$.

Each point charge polarizes the solvent; this polarization in turn creates a *reaction potential* field in the ligand. The reaction potential is a linear function of the vector $x$ of point charges; since the sources are discrete points, the reaction potential must be found only at those points to calculate the electrostatic free energy. The mapping between $x$ and $\varphi_{rf}$, the resulting vector of reaction potentials at the corresponding point locations, can therefore be represented by a matrix $L_u$, which is symmetric by reciprocity. The free energy due to $x$ in the unbound ligand is then $\frac{1}{2}x^T\varphi_{rf} = \frac{1}{2}x^T L_u x$.

When analyzing the ligand–receptor complex in solution, the ligand charges again produce a field linear in $x$. The receptor charges, which are assumed to be fixed, produce an additional field $c$; the total free energy of the bound system is thus $\frac{1}{2}x^T L_b x + x^T c$. Improvements in affinity correspond to minimizing the change in free energy between the bound and unbound states, so the objective function for the opti-

mization is the difference between these energies. Kangas and Tidor [2] showed that $L = L_b - L_u$ is symmetric positive semidefinite. Constraints are imposed on the charge vector $x$: first, sum of charge constraints on subsets of charges, and possibly on the entire set, are defined, which gives a set of equality constraints $Ax = b$. Physical and computational considerations lead to the imposition of box inequality constraints, $l_i \leq x_i \leq u_i \; \forall i$.

## 2.2 Biomolecule Electrostatic Modeling

We use the boundary element method for electrostatic simulation and address the free ligand; using the integral formulation from [5], the mapping $L_u$ can be represented as a function of three integral operators $M_1, M_2$, and $M_3$ [4]:

$$L_u = M_3 M_2^{-1} M_1. \tag{1}$$

The fully dense matrices $M_1, M_2$, and $M_3$ are too large for direct storage or inversion; instead, a fast method such as precorrected-FFT [6] is used to define functions that quickly perform matrix-vector multiplication by these matrices. The lack of explicit knowledge about $M_2$ necessitates the use of preconditioned Krylov iterative methods to apply the inverse $M_2^{-1}$. If there are $n_c$ ligand charges, the matrix $L_u$ can be calculated explicitly by solving, for $i = \{1 \ldots n_c\}$:

$$M_2 \varphi_i = M_1 e_i \tag{2}$$

$$L_{u,i} = M_3 \varphi_i \tag{3}$$

where $e_i$ is the $i^{th}$ unit vector and $L_{u,i}$ is the $i^{th}$ column of $L_u$.

## 2.3 Hessian-Implicit Primal-Dual Method

The biomolecule electrostatic optimization problem can easily be transformed into standard form

$$\begin{aligned} \text{minimize} \quad & \frac{1}{2} x^T L x + x^T c \\ \text{subj. to} \quad & Ax = b \\ \text{and} \quad & x \geq 0 \end{aligned} \tag{4}$$

by introducing slack variables on both sides of the box constraints.

After introducing the Lagrange multiplier vectors $\lambda$ and $s$, the Karush-Kuhn-Tucker optimality conditions for this problem can be written as a mildly nonlinear vector-valued function $F(x, \lambda, s)$ whose zeros are optimal solutions to (4) if $(x, s) \geq 0$: Primal-dual interior point methods [7] find an optimal solution to (4) by a modified Newton method: at iteration $k$, one linearizes $F$ around the current iterate $(x^k, \lambda^k, s^k)$ and biases the Newton step so that the pairwise products $x_i^{k+1} s_i^{k+1}$ are approximately equal. Define the average pairwise product $= (x^k)^T s^k / n$ where $n$ is the size of $x$ and $s$. The linearized system solved at each step is given by

$$\begin{aligned} \begin{bmatrix} L & -A^T & -I \\ A & 0 & 0 \\ S^k & 0 & X^k \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \\ \Delta s^k \end{bmatrix} &= -F(x^k, \lambda^k, s^k) + \begin{bmatrix} 0 \\ 0 \\ \sigma \; e \end{bmatrix} \\ &= z(x^k, \lambda^k, s^k) = z^k, \end{aligned} \tag{5}$$

where $X$ is the diagonal matrix with $X_{i,i} = x_i$, $S$ is similarly defined, and $e$ satisfies $e_i = 1 \; \forall i$. The second term on the right-hand side biases the Newton step as desired, where $0 \leq \sigma \leq 1$; the role of $\sigma$ is discussed below. The calculated Newton update is then scaled by $\alpha^k$, $0 < \alpha^k \leq 1$, to ensure that $(x^{k+1}, s^{k+1}) > 0$.

If $L$ has the form $L = M_3 M_2^{-1} M_1$, one may view the Jacobian in (5) as the Schur complement of the system

$$\begin{bmatrix} 0 & -A^T & -I & M_3 \\ A & 0 & 0 & 0 \\ S^k & 0 & X^k & 0 \\ -M_1 & 0 & 0 & M_2 \end{bmatrix} \begin{bmatrix} \Delta x^k \\ \Delta \lambda^k \\ \Delta s^k \\ \Delta \varphi^k \end{bmatrix} = \begin{bmatrix} z^k \\ \\ 0 \end{bmatrix} \tag{6}$$

where $\varphi^0 = M_2^{-1} M_1 x^0$. The Hessian-implicit method avoids direct calculation of $L$ by solving (6) at each iteration using a preconditioned Krylov subspace method. To form the preconditioner we copy the expanded Jacobian from (6), set the $M_1$ and $M_3$ blocks to zero, and approximate $M_2$ by a matrix $D_2$ composed of the diagonals of the blocks of $M_2$.

# 3 IMPLEMENTATION

## 3.1 Designing an Aggressive Optimization Strategy

The parameter $\sigma$ in (5) is called the centering parameter; it dictates how strongly the algorithm attempts to keep the pairwise products $x_i^k s_i^k$ equal. If $\sigma$ is set close to unity, the algorithm makes slow progress towards an optimal solution but is robust and rarely stagnates. If instead $\sigma$ is set very small, progress can be rapid but the optimization may stagnate; an iterate may approach the boundary of the feasible region $(x, s) > 0$, in which case the algorithm makes unacceptably slow progress. The original Hessian-implicit formulation set $\sigma = 0.4$ for all iterations, as suggested in [7], which balances robustness against the rate of convergence. The new formulation uses a simple rule to pick each $\sigma^k$ independently, using the step multiplier $\alpha^{k-1}$ as the primary criterion:

**Algorithm 1** *Choosing centering parameter* $\sigma^k$

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

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 good after a few iterations, and the two cases in which $\sigma^k < 0.4$ address its shortcomings.

## 3.2 Solving Equality Constrained Problems

The Hessian-implicit method was developed to solve programs with both equality and inequality constraints. The optimality conditions for an equality constrained program are linear, so only one linear system must be solved to find an optimal solution. The Hessian-implicit optimality conditions

$$\begin{bmatrix} 0 & A^T & M_3 \\ A & 0 & 0 \\ -M_1 & 0 & M_2 \end{bmatrix} \begin{bmatrix} x^* \\ \lambda^* \\ \varphi^* \end{bmatrix} = \begin{bmatrix} -c \\ b \\ 0 \end{bmatrix} \tag{7}$$

can be solved quickly by using the preconditioner

$$P_{eq} = \left[\begin{array}{cc|c} M_3 D_2^{-1} M_1 & A^T & 0 \\ A & 0 & 0 \\ \hline 0 & 0 & D_2 \end{array}\right]. \tag{8}$$

$P_{eq}$ is effective because $M_3 D_2^{-1} M_1$ is an approximation to $L$; factoring $P_{eq}$ is inexpensive because it is block diagonal as shown, and although the upper left block is dense, it is extremely small. In contrast to the preconditioner for (6), we include the $M_3 D_2^{-1} M_1$ block to prevent singularity of the preconditioner.

## 3.3 Full-scale Implementation

To implement the Hessian-implicit primal-dual method, we coupled the pFFT++ boundary element method code [6], [8] with the PETSc scientific library [9]. The pFFT++ code allows black-box multiplication by the integral operator matrices $M_1, M_2$, and $M_3$ to be done in $O(n \log n)$ time and space, where $n$ is the number of panels used to discretize the surface. The PETSc library offers a variety of iterative linear solvers and preconditioners; each modified Newton update (6) is solved using GMRES and the LU factorized preconditioner. The preconditioner nonzero structure is fixed, so ordering need be performed only once.

# 4 RESULTS

## 4.1 Performance Comparisons

The calculations of $M_2$ matrix-vector (MV) products dominate the cost of the optimization process, so to assess the performance of different algorithms, the number of calculated $M_2$ MV products is used as the cost metric. Figure 1 illustrates the performance of the new and old formulations.

To compare the Hessian-implicit method's performance to standard algorithms, KNITRO, which implements a primal-dual interior-point nonlinear optimization algorithm [10], was used as a reference. KNITRO uses CG to calculate each Newton update; each CG iteration therefore performs one calculation of $Lx$ if the system (5) is solved. Each $Lx$ product requires one iterative solve to find $M_2^{-1} M_1 x$. We therefore estimate KNITRO's cost to solve (5) by multiplying the total number of CG iterations by the average number of $M_2$ MV products required to find a column of $L$. Figure 2 shows

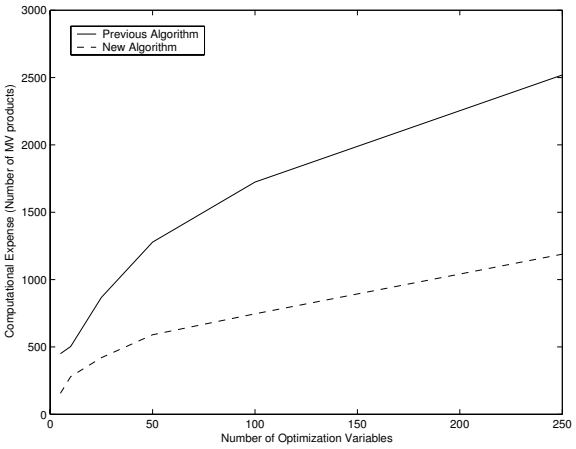


Figure 1: Performance of previous and new algorithms

that this alternative implicit scheme performs more poorly than the traditional explicit-Hessian method and much more poorly than the Hessian-implicit primal-dual method; we believe that the irregular behavior of KNITRO in Figure 2 results from particularly poor conditioning of one or more of the random test problems. When solving optimization prob-

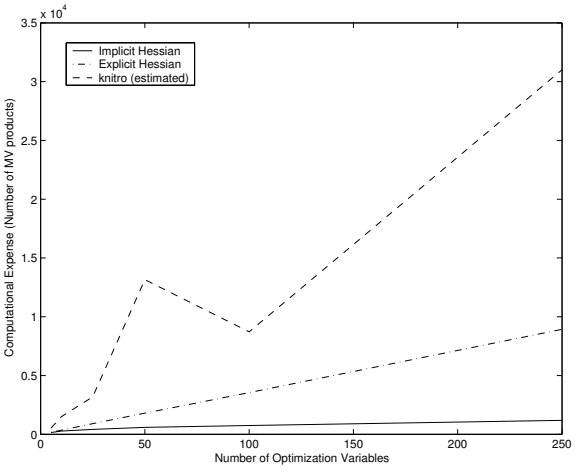


Figure 2: Performance of new and alternative methods

lems with only equality constraints, the computational advantage of using the Hessian-implicit method is even more pronounced, as illustrated in Figure 3.

## 4.2 Realistic Biomolecule Optimization

To assess the Hessian-implicit method's accuracy, we studied the ligand–receptor system of enzyme *E. coli* chorismate mutase (ECM) and an inhibitor transition-state analog (TSA) [3]. Plotted in Figure 4 are optimal charge distributions calculated by the Hessian-implicit method and an explicit Hessian method. The results agree very well. Inaccuracies are due largely to numerical Hessian asymmetry in the implicit method; when an explicit Hessian is formed, it can be symmetrized and nonphysical singular values can be removed [3],

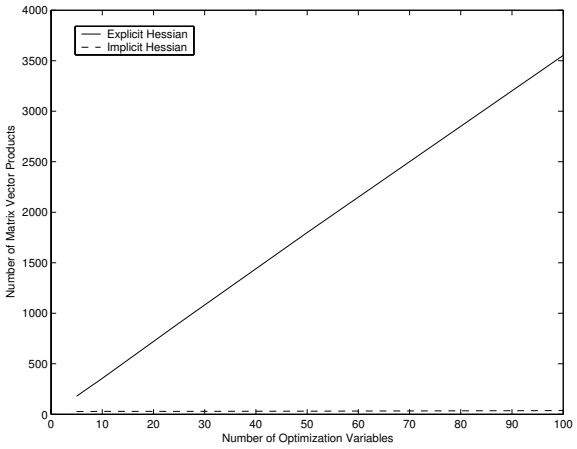


Figure 3: Performance of new algorithm on equality constrained problems

but these operations cannot be performed in the implicit optimization method. The optimization problem has 26 primary variables and the Hessian-implicit system solved at each iteration has approximately 130,000 variables.

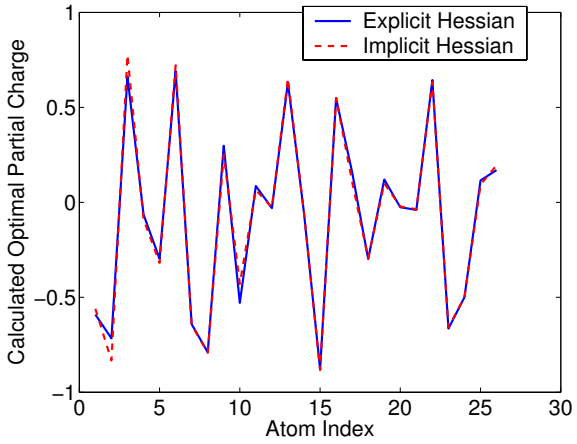


Figure 4: Accuracy of solution on realistic problem

## 5 DISCUSSION

This paper described improvements to the Hessian-implicit optimization method originally reported in [4] and a full-scale implementation of the new method. The implementation couples the pFFT++ fast boundary element method package and the PETSc scientific library. The recent improvements to the method improve performance by approximately a factor of two over the original formulation; in addition, the new full-scale code is capable of solving biologically relevant optimization problems. Simpler optimization problems with only equality constraints can be solved extremely rapidly using a newly designed preconditioner. Future work will investigate the sources of numerical Hessian asymmetry, and explore extending the formulation to allow convex constraints and the rapid update of a solution if the constraints are varied.

## 6 ACKNOWLEDGMENTS

This work was supported by the Singapore–MIT Alliance, the National Science Foundation, and the National Institutes of Health. J. Bardhan is supported by a Department of Energy Computational Science Graduate Fellowship. S. Leyffer and S. Benson are supported by the Mathematical, Information, and Computational Sciences Division subprogram of the Office of Advanced Scientific Computing Research under Contract W-31-109-ENG-38.

## REFERENCES

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

[2] 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.

[3] E. Kangas and B. Tidor. Electrostatic complementarity at ligand binding sites: Application to chorismate mutase. *Journal of Physical Chemistry*, 105:880–888, 2001.

[4] J. P. Bardhan, J. H. Lee, S. S. Kuo, M. D. Altman, B. Tidor, and J. K. White. Fast methods for biomolecule charge optimization. *Modeling and Simulation of Microsystems (MSM)*, 2003.

[5] S. S. Kuo, M. D. Altman, J. P. Bardhan, B. Tidor, and J. K. White. Fast methods for simulation of biomolecule electrostatics. *International Conference on Computer Aided Design (ICCAD)*, 2002.

[6] 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*, 16:1059–1072, 1997.

[7] S. J. Wright. *Primal-Dual Interior Point Methods*. SIAM, 1997.

[8] 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*, 2003.

[9] 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.

[10] R. Byrd, M. E. Hribar, and J. Nocedal. An interior point method for large scale nonlinear programming. *SIAM J. Optimization*, 9:877–900, 1999.